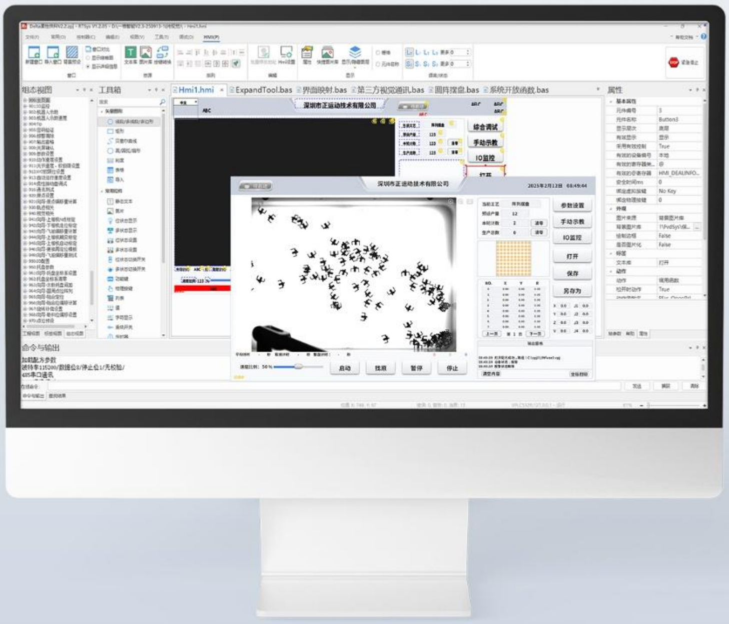
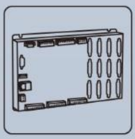
做更好用的运动控制，智造美好生活
Better Motion Control, Smarter Life

RTHmi Programming Manual

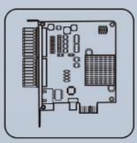
V1.4.0



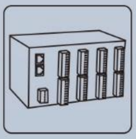
Vision Motion Controller



Motion Controller



Motion Control Card



IO Expansion Module

HMI

Content

Chapter I RTHmi Program Introduction

RTHmi is the configuration design for Zmotion motion controller. Before, ***please confirm whether the controller supports RTHmi function or not*** (valid controller models can be referred from chapter 1.1.6 RTHMI Controllers, or contact us directly).

- Notes: for low series motion controller, it also can use RTSys to do HMI programming in RTSys, but it only support RTHmi V1.0.01 functions, for those invalid elements that will be deleted automatically.

Please download RTSys (how to use it, refer to RTSys user manual) at first, then edit and debug RTHmi program in RTSys, same, RTHmi motion controllers or the simulator whose firmware version should above V5.20-20230706.

RTSys is upgraded based on ZDevelop software, which supports Basic program, PLC program, Hmi configuration. You also can draw dynamically in touch screen. Please use the latest RTSys, you can download from website (for the project that was created by higher version RTSys, don't open it by lower RTSys / ZDevelop, otherwise, it will cause abnormal project and no way to resume it).

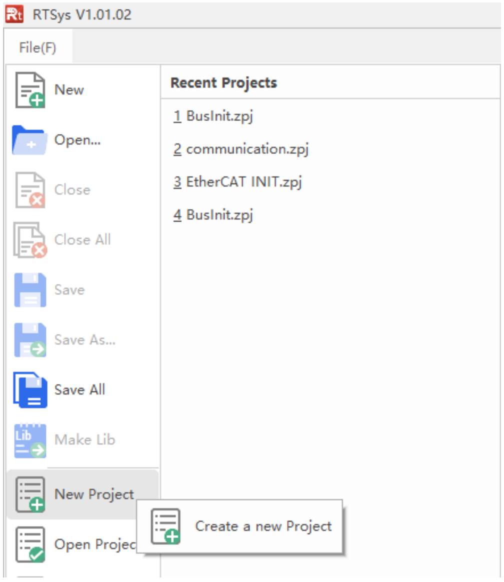
1.1. RTSys Programming Properties

1.1.1. How to Develop HMI

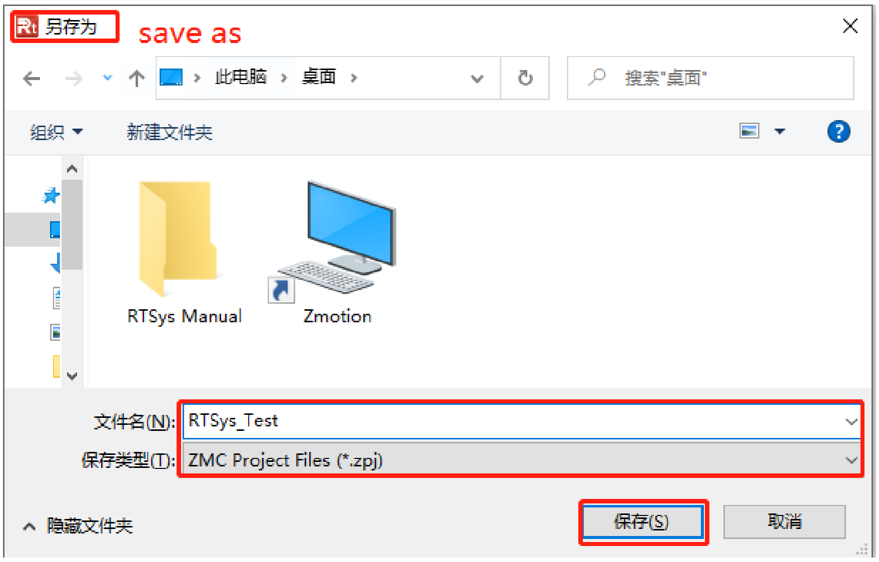
At first, build a new folder in your PC to save the project that you will create. Here take RTSys V1.01.02 as example, if you need to update, please refer to "RTSys User Manual". And there are many HMI routines in Zmotion website, you can download them to check and learm.

Basic Processes: build a new project → build a new file → select the file type → add AutoRun Task No. → edit HMI program → connect to controller → download the program into controller.

1. Build one New Project: "File" in "Menu" → "New Project".



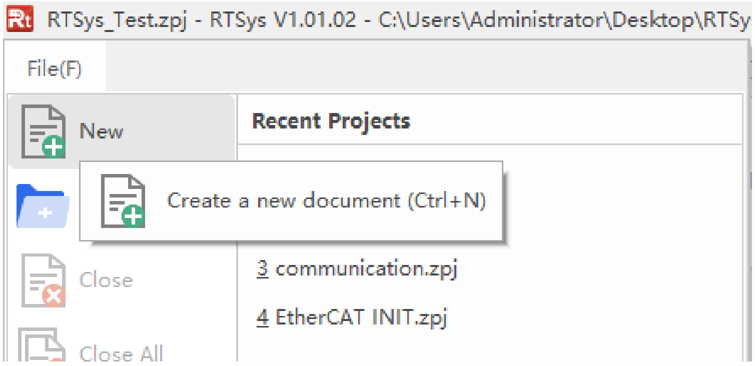
Click "New Project", then "Save as…" will be jumped, select the folder (that was created just now), and open it, then enter folder's name and save the project, pay attention to the suffix should be ".zpj".



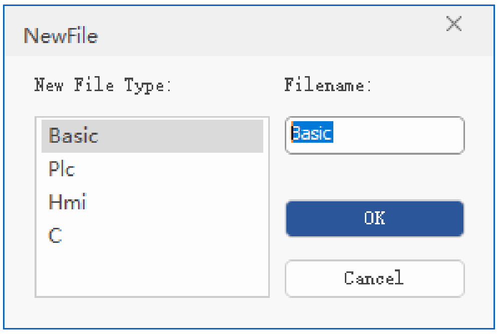
文件名(N): RTSys_Test

保存类型(T): ZMC Project Files (*.zpj)

2. Build New File: File – New


RTSys_Test.zpj - RTSys V1.01.02 - C:\Users\Administrator\Desktop\RTSy
File(F)
New
Recent Projects
Create a new document (Ctrl+N)
Close
Close All
3 communication.zpj
4 EtherCAT INIT.zpj

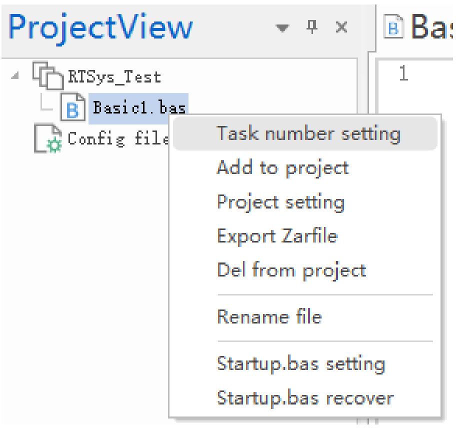
After clicking “New File”, below popping window will appear, which supports Basic/PLC/Hmi hybrid programming. Here selects the “Basic” file type and click “OK”. The name can be modified.

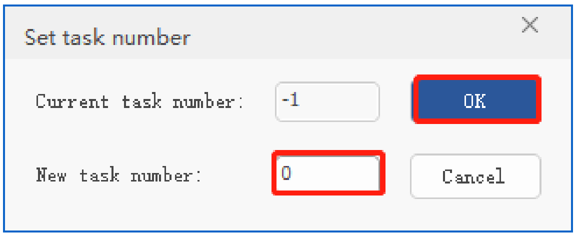


After added, it will be shown in “project view” automatically, then you can edit HMI program, and save it, after that, the HMI file has been automatically added into project .zpj folder already.

3. Set “File Automatically Run” (AutoRun Task No.)

Like below left image, right-click file, select “task number setting”, then see right image, enter the No. “0”, click OK.





After setting, in this file name right side, the task No. will be shown, such as, Basic1.bas[0]. You also can rename it. But please close it at first, then right-click it, then rename.

✧ Multi-file task No. can be set, but one task No. must be set. An it can be any values, no priority, but don’t exceed controller’s maximum task value (controller state or send “?*max” or refer to user manual).

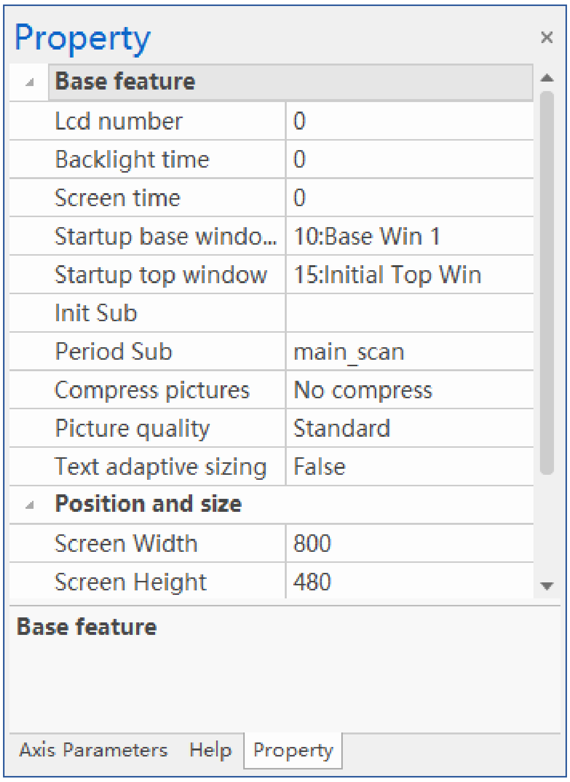
4. Edit HMI Program

Before that, open “Hmi Setting” window at first: switch to HMI programming window, then find

“HMI” in menu, then find “Hmi Setting” (or open HMI file, click empty place outside the window), according to needed teach pendant’s size, set horizontal & vertical resolution (this must be confirmed in advance!).

And see whether you need to set initial function and period function, select SUB subfunction that is global defined sub edited in BASIC.

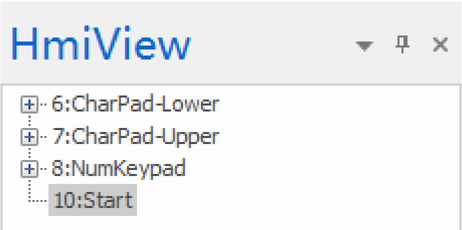
Hmi setting window (each property definition, please refer to chapter II Hmi Setting):



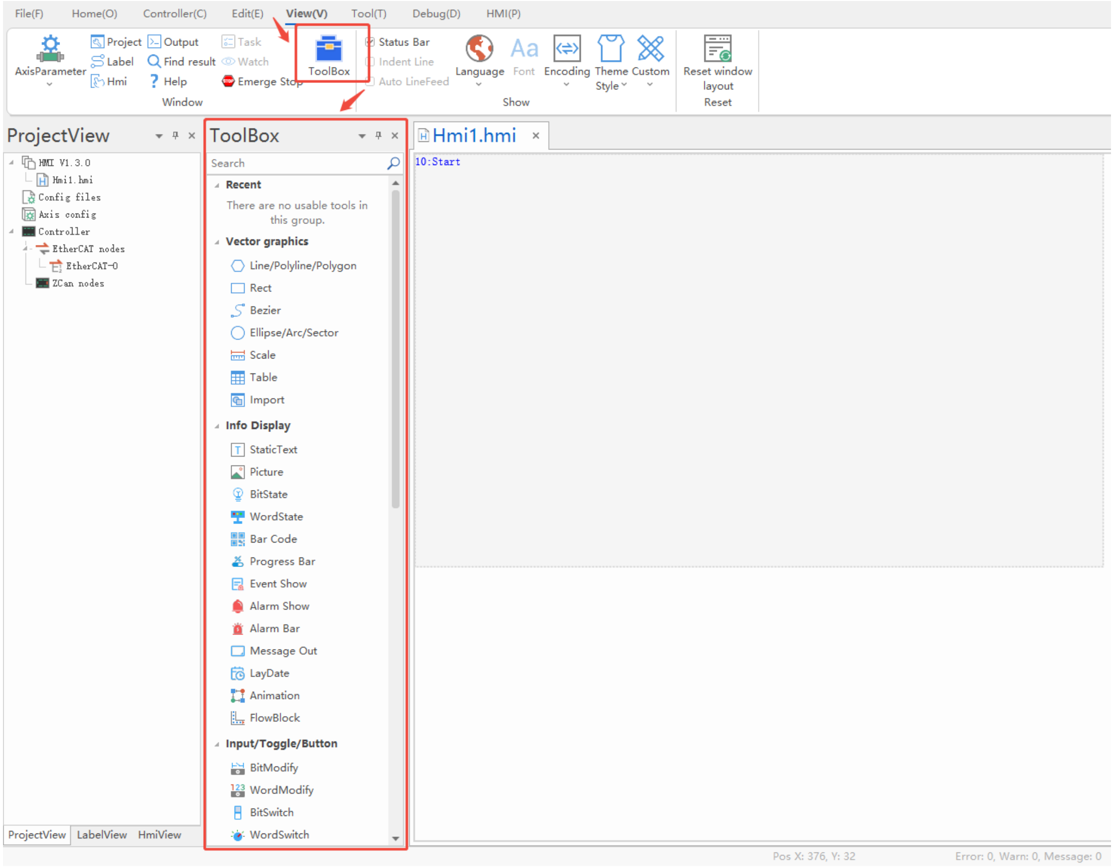
Base feature	
Lcd number	0
Backlight time	0
Screen time	0
Startup base windo...	10:Base Win 1
Startup top window	15:Initial Top Win
Init Sub	
Period Sub	main_scan
Compress pictures	No compress
Picture quality	Standard
Text adaptive sizing	False
Position and size	
Screen Width	800
Screen Height	480

Base feature

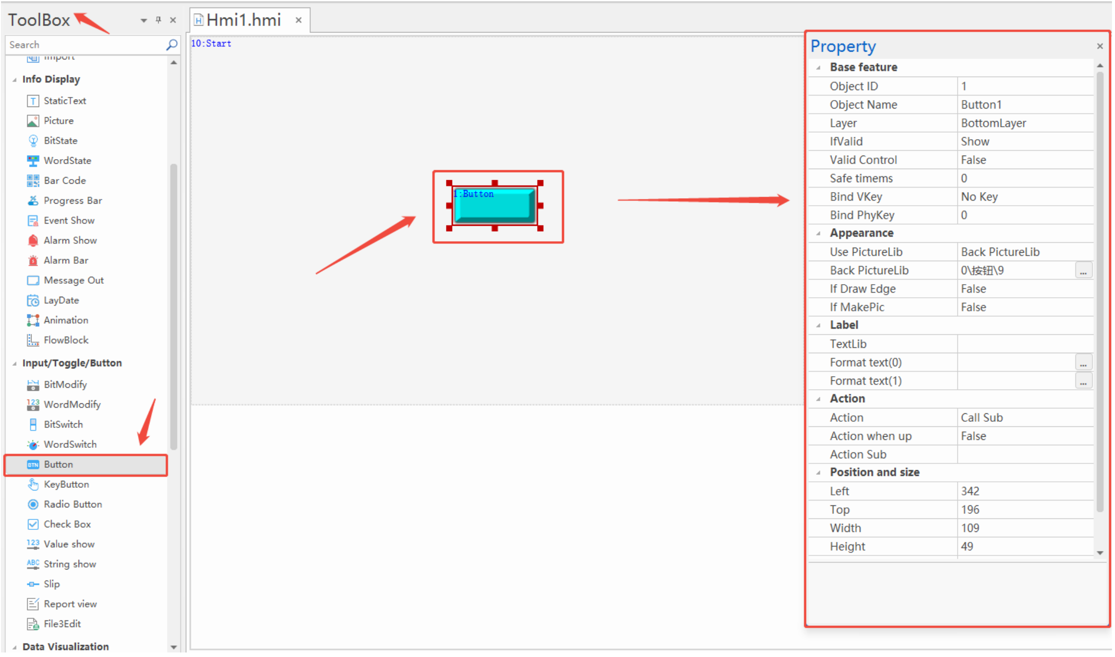
After basic setting in above window, you can add HMI components, added components will be shown in HMI window: in HMI file menu, find “Tool Boox” from View, there are many optional components (generally, after creating one HMI file, 3 soft keyboard windows and one initial window “10:Start” will be created automatically). For how to use the window and control class, please refer to “chapter III” and “chapter IV”.



- 6:CharPad-Lower
- 7:CharPad-Upper
- 8:NumKeypad
- 10:Start



In "Tool Box", you can select the object, drag one in the HMI size range directly, and each one object's property can be opened to set needed parameters.

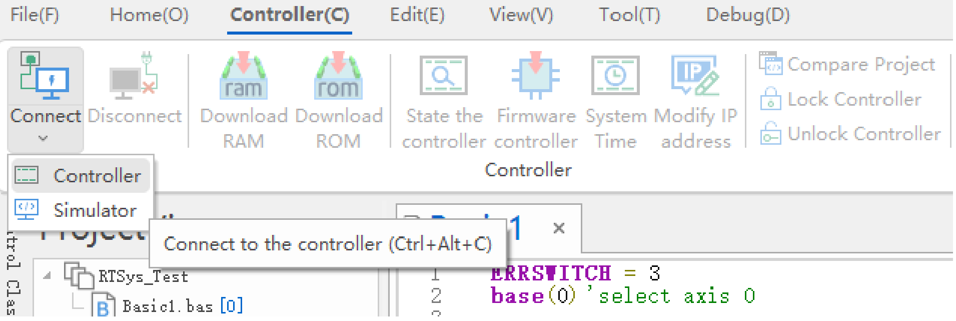


Property

Base feature	
Object ID	1
Object Name	Button1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bind VKey	No Key
Bind PhyKey	0
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	
Format text(1)	
Action	
Action	Call Sub
Action when up	False
Action Sub	
Position and size	
Left	342
Top	196
Width	109
Height	49

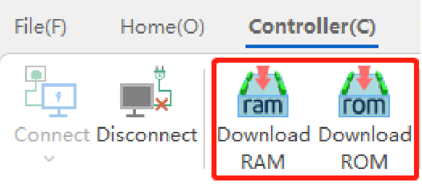
5. Connect to Controller / Simulator

After editing Basic and HMI program, then click “Controller – Connect – Controller”. If there is no controller, simulator can be used: “Controller – Connect – Simulator”, then it can be opened, please note don’t close the simulator window.

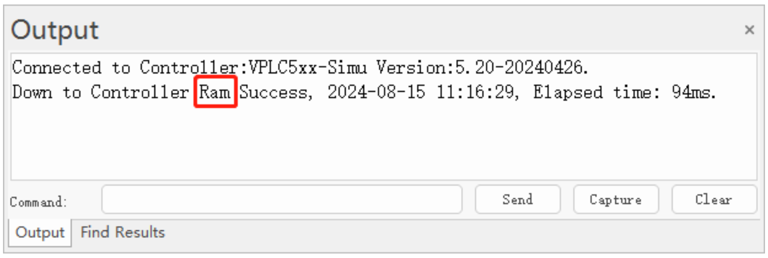


6. Download Program

In menu bar, there are two downloading methods, **Download RAM / Download ROM**. After downloaded, also, it will give you’re the feedback information in “output” window, then, program will run automatically.



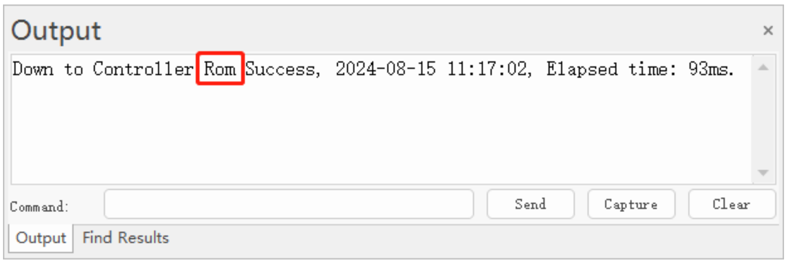
--succeed in downloading RAM--



Output

```
Connected to Controller:VPLC5xx-Simu Version:5.20-20240426.
Down to Controller Ram Success, 2024-08-15 11:16:29, Elapsed time: 94ms.
```

--succeed in downloading ROM--



Output

```
Down to Controller Rom Success, 2024-08-15 11:17:02, Elapsed time: 93ms.
```

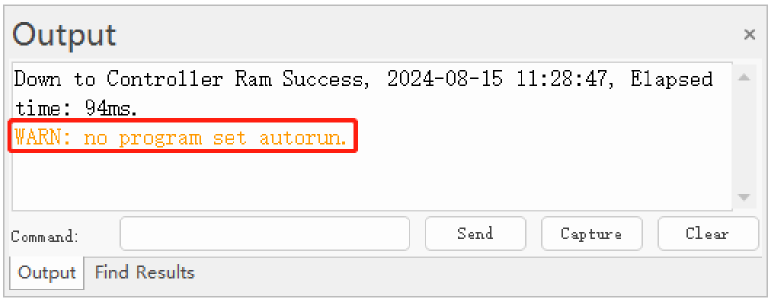
--download RAM VS download ROM--

RAM: not save program when power down

ROM: save program when power down, that is, program will run automatically according to task No., when connect to controller next time. **And please note for HMII program, it must use "Download ROM"**

NOTES:

- ZMC0XX series controllers don't support "Download RAM".
- When project is not built, that is, you only open the file, it can't be downloaded into controller.
- AutoRun No., 0 means the task No., that is, it runs as task 0. Task No. doesn't have priority.
- One No. must be set. If all files don't have task No., when downloading into controller, system will show this message: WARN: no program set autorun. Then, please set auto run task No. at first.



```
Down to Controller Ram Success, 2024-08-15 11:28:47, Elapsed
time: 94ms.
WARN: no program set autorun.
```

1.1.2. Hmi Setting

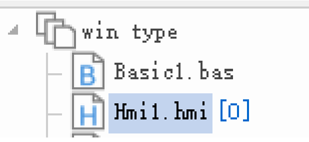
Before editing the hmi program, firstly open the "Hmi setting" window. Set the auto run No. for HMI file at first, then you can run it.

Notes:

a) Multiple HMI files can be set at the same time (for one project, HMI file must be set Auto Run No., for others, like, Basic, it is random. Same, please don't exceed its upper limit, which can be known form parameter "max_hmi" after sending the command "?*max", but one screen only can show one HMI file.

b) Usually, HMI should be programmed with Basic together: HMI component can call Basic function or register. In addition, HMI also can be programmed with PLC.

c) HMI task is not real-time, if you need high real-time, please use other independent tasks.



1.1.3. HMI Screen Display

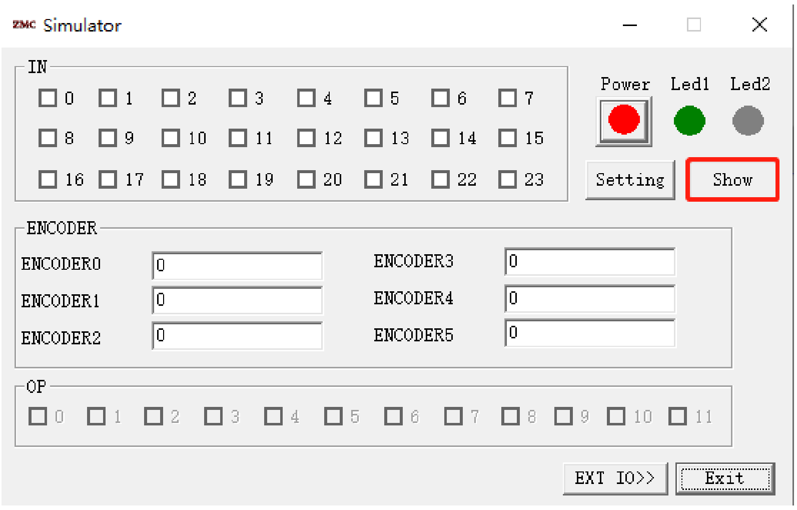
For RTHmi, you can use your computer or other touch screens as display screen through ethernet. Also, you can use the display screen, then corresponding control should be with HDMI interface.

When the controller supports multiple screens, please select LCD No. in “Hmi Setting”. If you want to know how many screens it supports, and the max resolution, you can connect to controller, then send command “?*max”, find “max_Hmi” parameter. For example: “max_Hmi: 2, x: 1024 y: 800”, which means it supports 2 remote HMI, and the max size is 1024 * 800.

1.1.4. HMI Simulation Running

After the program is downloaded to the simulator, in simulator window, you can click "show" to run the HMI, and corresponding interface will be shown in “xplc screen”.

When the controller and teaching box are not connected, it can be debugged through simulator.



ENCODER0 0
ENCODER1 0
ENCODER2 0
ENCODER3 0
ENCODER4 0
ENCODER5 0

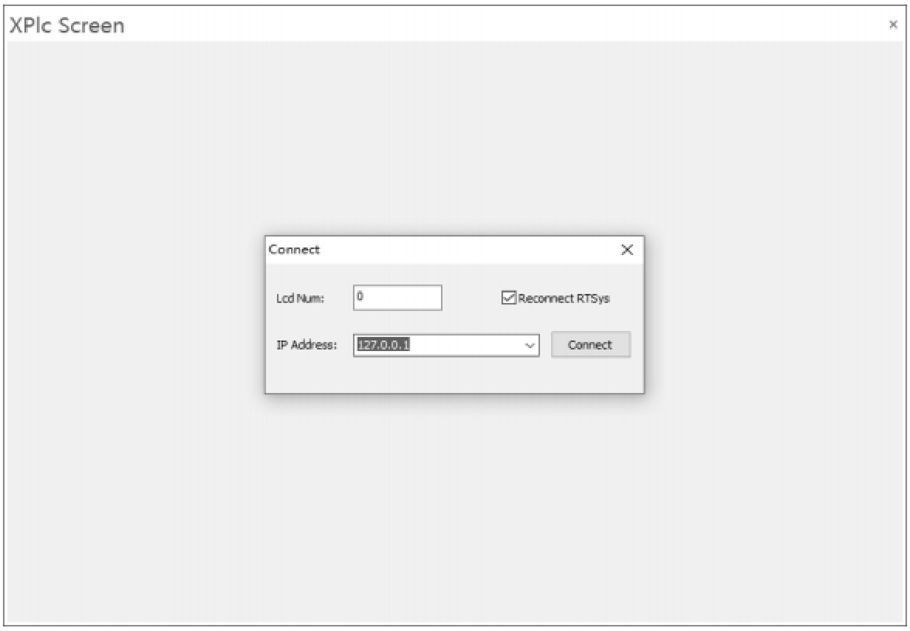
--How to Switch Simulator Language--

The simulator window is in Chinese by default, please switch it by clicking “设置 Setting” – “语言”, then select English, restart it. And this operation takes effect all the time

1.1.5. Xplcterm Running

When there is no real touch screen, use Zmotion simulation tool **“xplc screen”** to simulate the operation, or add **Xplcterm** software into “tool – plug-in”, then the PC can be used as a man-machine interface to connect to controller or simulator. In addition, Xplcterm software can be opened independently.

After the program is downloaded to the simulator or controller. Open “Xplc Screen”, click “show” in simulator, or in tool menu, find plug-in, click xplc screen, then please enter the No. and IP address, and click “Connect”. Simulator IP is shown by default.



Lcd Num: 0

☑ Reconnect RTSys

IP Address: 127.0.0.1

1.1.6. RTHMI Controllers

The controller models that support HMI programming are as follows, some controllers need to upgrade the firmware, please contact the manufacturer for details.

Controllers that support RTHmi programming: ZMC4XX and above, and corresponding firmware (that is, for lower ZMC4XX controllers, they don't support RTHmi, except the controller that is with H, for example, XPLC120H). And for lower series models that are not mentioned above, recommend use ZDevelop.

1.1.7. Valid ZHD Teach Pendant

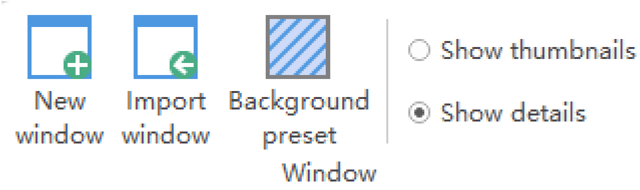
ZHD series handheld boxes with an X suffix indicate that they support the HMI configuration programming method, and the touch screen program can be downloaded to the controller (no need to download to the touch screen separately), and then connect the touch screen to the controller to communicate.

Model	ZHD300X	ZHD400X	ZHD500X
Resolution	480*272	800*480	1024*600
Buttons / Keys	47	18	16
USB	1	1	1
Emergency Stop Switch	1	1	1
Touch screen	Support	Support	Support

Chapter II HMI Menu Bar

Note: HMI menu bar is only shown while HMI is showing in the main window.

2.1. Windows



- ➢ **New window:** new build or several windows in current project, for details, please refer t.
- ➢ **Import window:** in current project, import other project or created HMI window, for details, please refer to
- ➢ **Background preset:** you can set HMI window background and object's default styles / colors. Please note it is valid when building new window or object after saving the settings, for created windows and objects, original formats keep. For details, please refer to this chapter "Picture Lib".

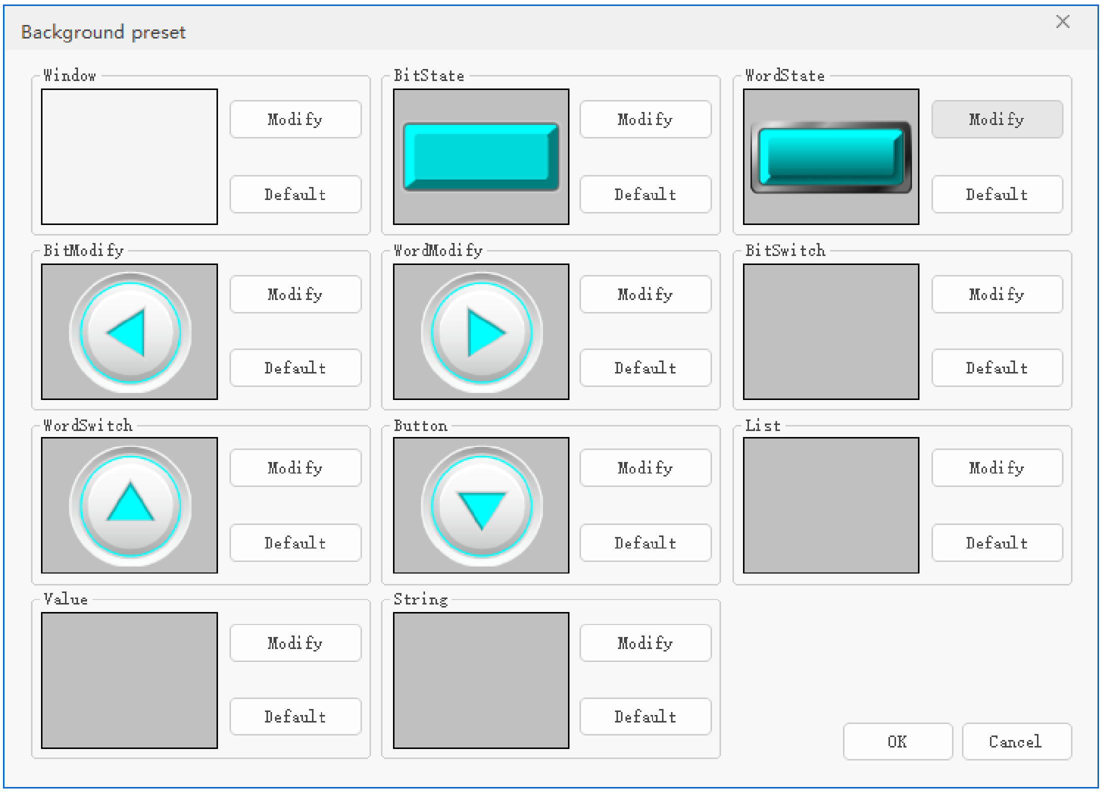
--How to Operate: Select needed object / window, then click "modify", there are two style selections.

A. Method 1: if you are ready to use picture library formats, you can import the picture from picture lib, and change the image ratio can adjust object showing size.

B. Method 2: custom colors for the state 0 and state 1.

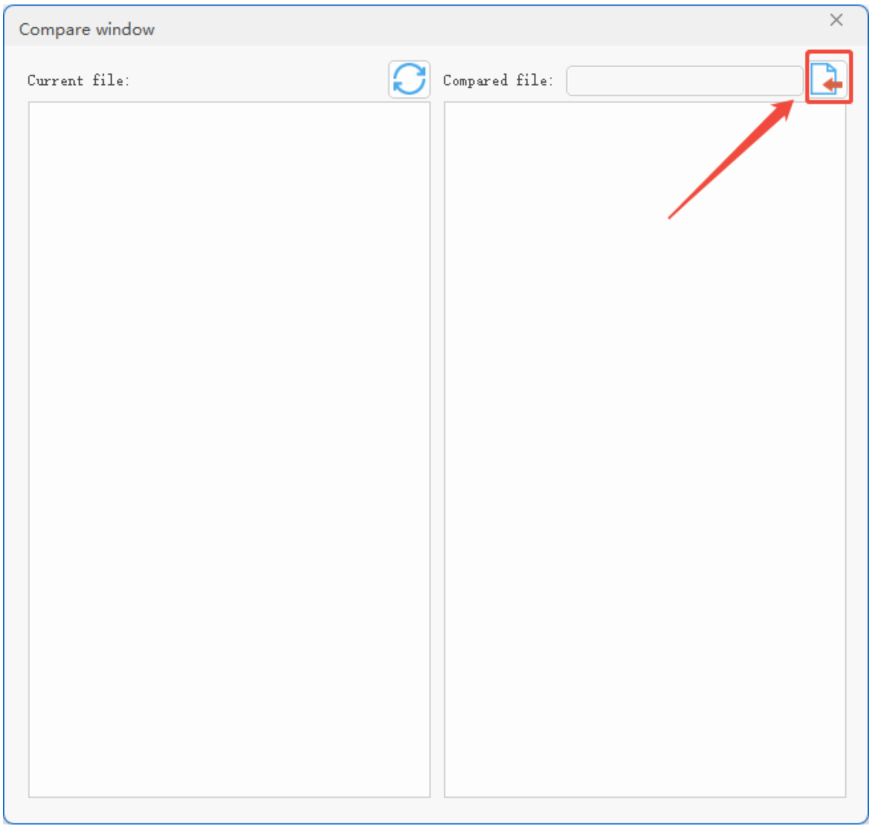
Between these two methods, only one takes effect. If "picture lib" is used, custom colors and formats will be covered.

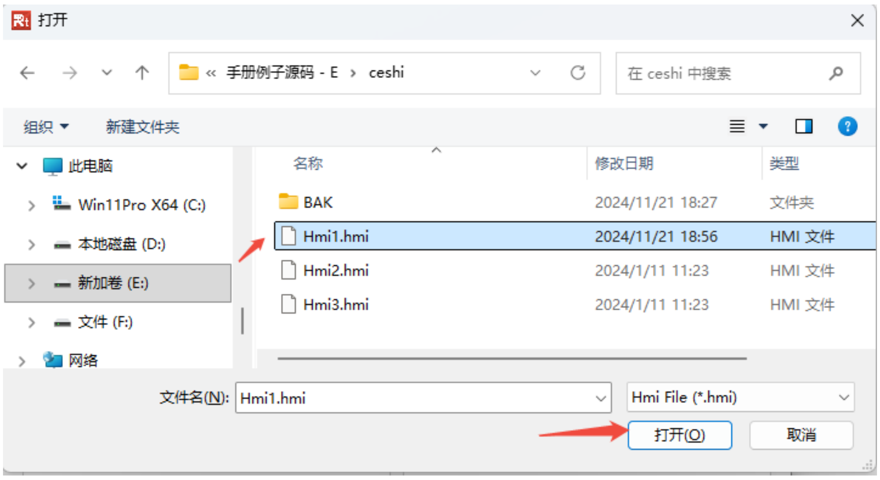
For window background preset, use method 2, and set state 0. It takes effect after building new window.



- **Compare Window:** select one HMI file, then compare its window and components with current HMI.

Step 1: click "compare window", and click right upper corner icon, then select which HMI file.





打开

« 手册例子源码 - E › ceshi

在 ceshi 中搜索

组织 ▾ 新建文件夹

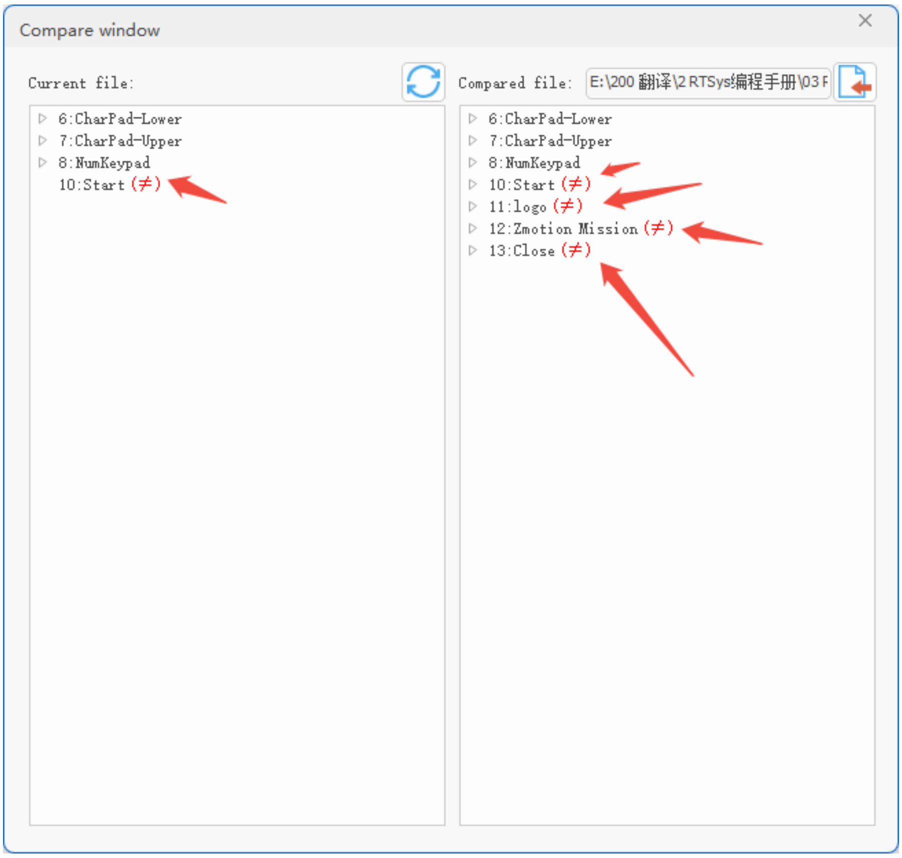
- 此电脑
- Win11Pro X64 (C:)
- 本地磁盘 (D:)
- 新加卷 (E:)
- 文件 (F:)
- 网络

名称	修改日期	类型
BAK	2024/11/21 18:27	文件夹
Hmi1.hmi	2024/11/21 18:56	HMI 文件
Hmi2.hmi	2024/1/11 11:23	HMI 文件
Hmi3.hmi	2024/1/11 11:23	HMI 文件

文件名(N): Hmi1.hmi

Hmi File (*.hmi)

Step 2: after that, this two HMI file's window information will be shown. Left shows current HMI information, right shows another one. And it will compare each corresponding window. If the window name and inner components (type, size, position, etc.) are different, the symbol "≠" will be shown behind the window.



Compare window

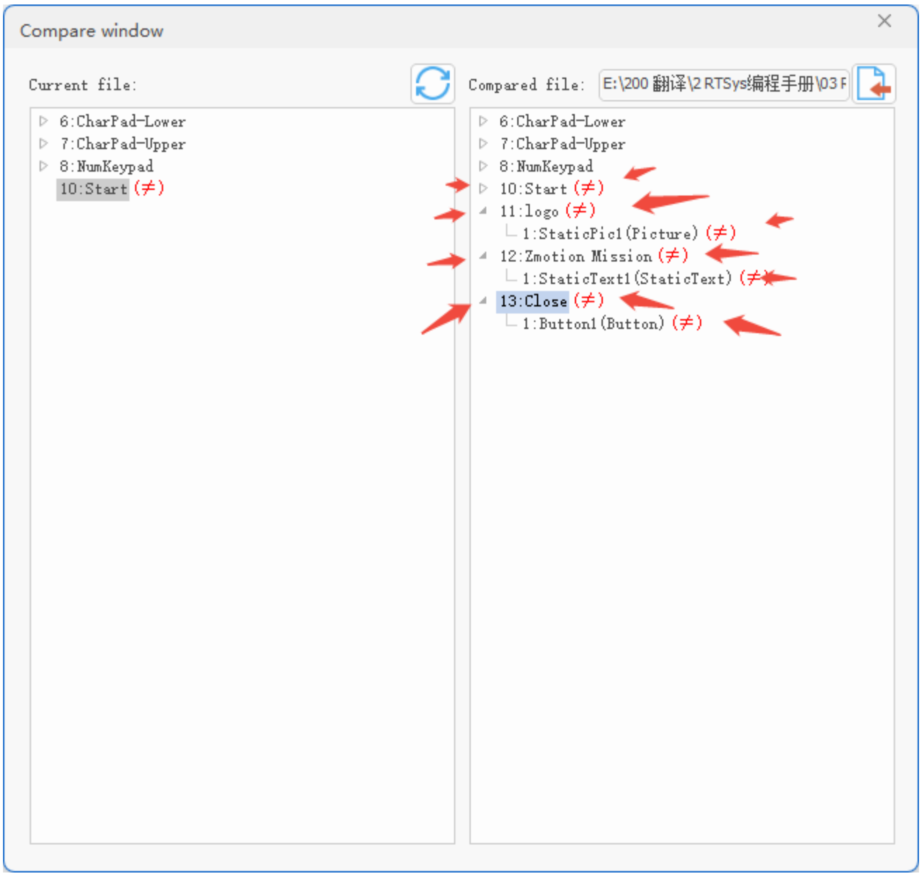
Current file:

- 6:CharPad-Lower
- 7:CharPad-Upper
- 8:NumKeypad
- 10:Start (≠)

Compared file: E:\200 翻译\2 RTSys编程手册\03 F

- 6:CharPad-Lower
- 7:CharPad-Upper
- 8:NumKeypad
- 10:Start (≠)
- 11:logo (≠)
- 12:Zmotion Mission (≠)
- 13:Close (≠)

Step 3: you could click the icon that is before the window to check specific component information, also "≠" means they are with inconsistent components.



Current file:

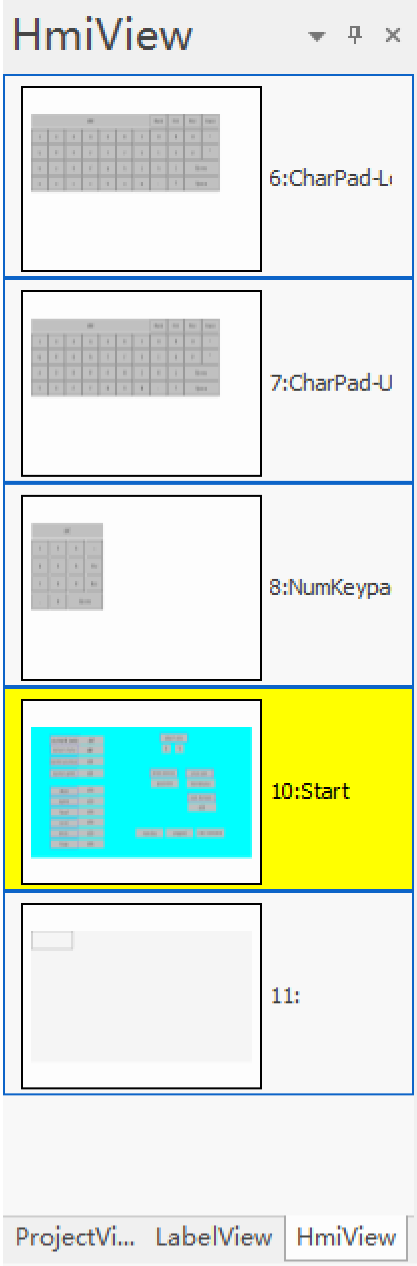
- 6:CharPad-Lower
- 7:CharPad-Upper
- 8:NumKeypad
- 10:Start (≠)

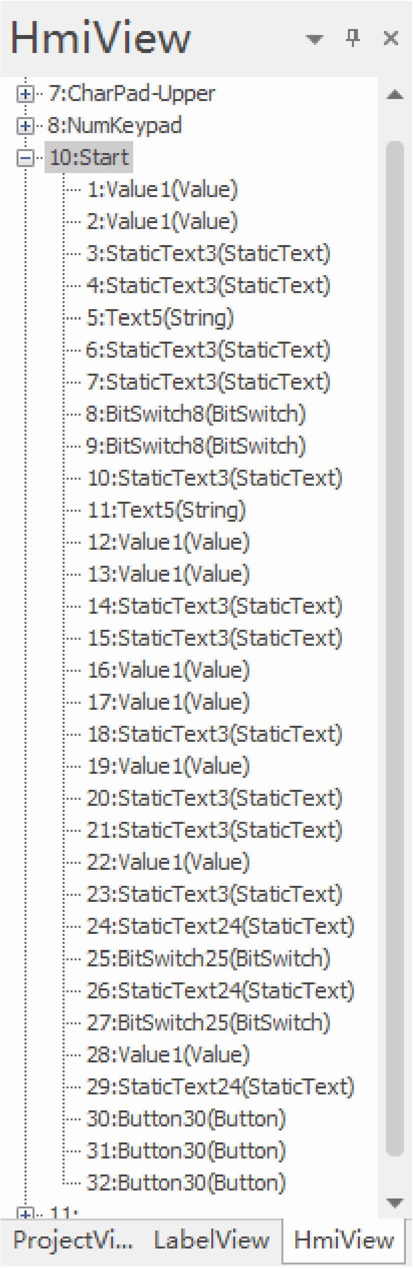
Compared file: E:\200 翻译\2 RTSys编程手册\03 F

- 6:CharPad-Lower
- 7:CharPad-Upper
- 8:NumKeypad
- 10:Start (≠)
- 11:logo (≠)
  - 1:StaticPic1(Picture) (≠)
- 12:Zmotion Mission (≠)
  - 1:StaticText1(StaticText) (≠)
- 13:Close (≠)
  - 1:Button1(Button) (≠)

- **Show thumbnails:** In HmiView, window thumbnails, window No., and window name will be shown (yellow background indicates currently opened window, you can switch through mouse clicking) [Left Image].

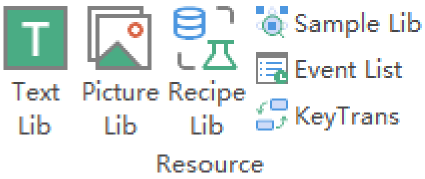
- **Show details:** In HmiView, window information (window No., window name), and object information (created object No., object name) all will be shown [Right Image].





7:CharPad-Upper
8:NumKeypad
10:Start
1:Value1(Value)
2:Value1(Value)
3:StaticText3(StaticText)
4:StaticText3(StaticText)
5:Text5(String)
6:StaticText3(StaticText)
7:StaticText3(StaticText)
8:BitSwitch8(BitSwitch)
9:BitSwitch8(BitSwitch)
10:StaticText3(StaticText)
11:Text5(String)
12:Value1(Value)
13:Value1(Value)
14:StaticText3(StaticText)
15:StaticText3(StaticText)
16:Value1(Value)
17:Value1(Value)
18:StaticText3(StaticText)
19:Value1(Value)
20:StaticText3(StaticText)
21:StaticText3(StaticText)
22:Value1(Value)
23:StaticText3(StaticText)
24:StaticText24(StaticText)
25:BitSwitch25(BitSwitch)
26:StaticText24(StaticText)
27:BitSwitch25(BitSwitch)
28:Value1(Value)
29:StaticText24(StaticText)
30:Button30(Button)
31:Button30(Button)
32:Button30(Button)
11:

2.2. Resource



2.2.1. Text Lib

(1) What is It

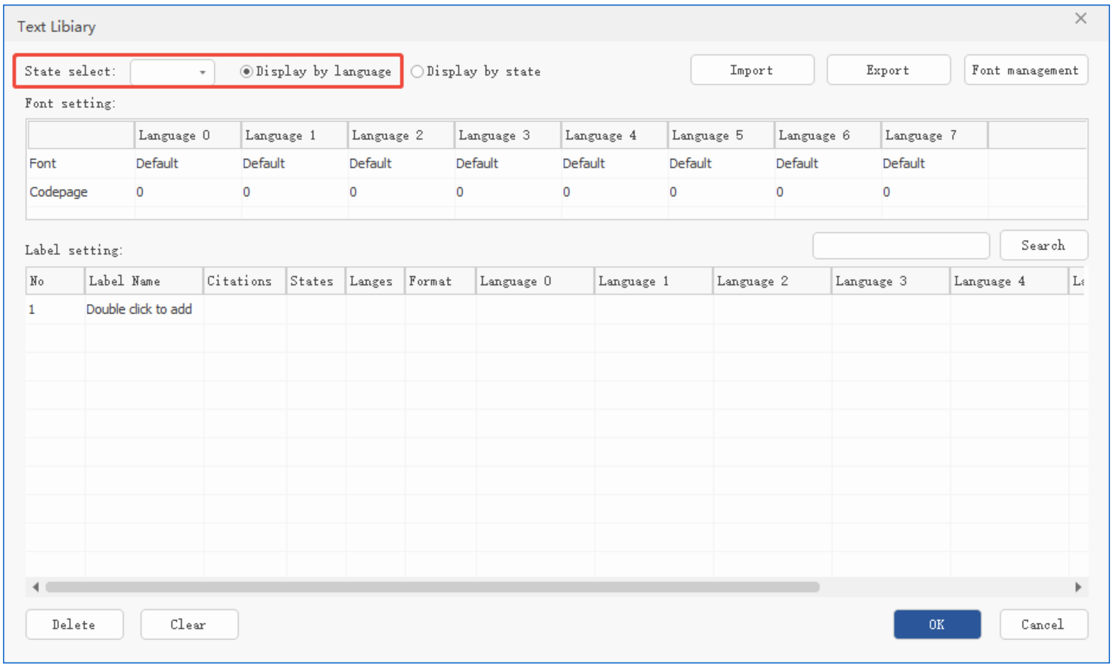
Use labels to set multiple language texts in different states and the fonts corresponding to each text at one time and call it in the corresponding component. Different text content can be filled in each language. (One label supports up to 256 states, and one state supports up to 8 language texts)

Function	Description	
Select the State	Display when “display by language” is selected, to switch the state, different text contents under different states.	
Select the Language	Display when “display by state” is selected, to switch the state, different text contents under different languages.	
Display by Language	Set “display by language”, you can set text contents under different states, 0-7 kinds of languages can be set.	
Display by State	Set “display by state”, you can set text contents under different languages, 0-255 kinds of states can be set.	
Import / Export	Import or export text library file.	
Font Management	Load or delete font file.	
Font Settings	Font	Select the corresponding font of each language.
	Codepage	Enter each language’s code (it must be filled for other languages)
Label Setting	No.	/
	Label Name	Text library name.
	Citations	How many times this text lib is cited (can’t be set).
	States	Set text library’s states.
	Langes	Set how many languages you will use.
	Format	Set language text format.
	Language 0-7	For “display by language”, write text content
	State 0-15	For “display by state”, write text content, state can be 0-15 / 16-31 / … / 240-255.

Notes:

[1] under Chinese operation window, when you need to show the text in other languages, please fill in this language's code in "codepage". If under non-Chinese operation system, when you need to show the text in Chinese, please enter Chinese code (usually it is 936).

[2] you can fill 65001 (UTF-8 code) in "codepage" to avoid garbled text.



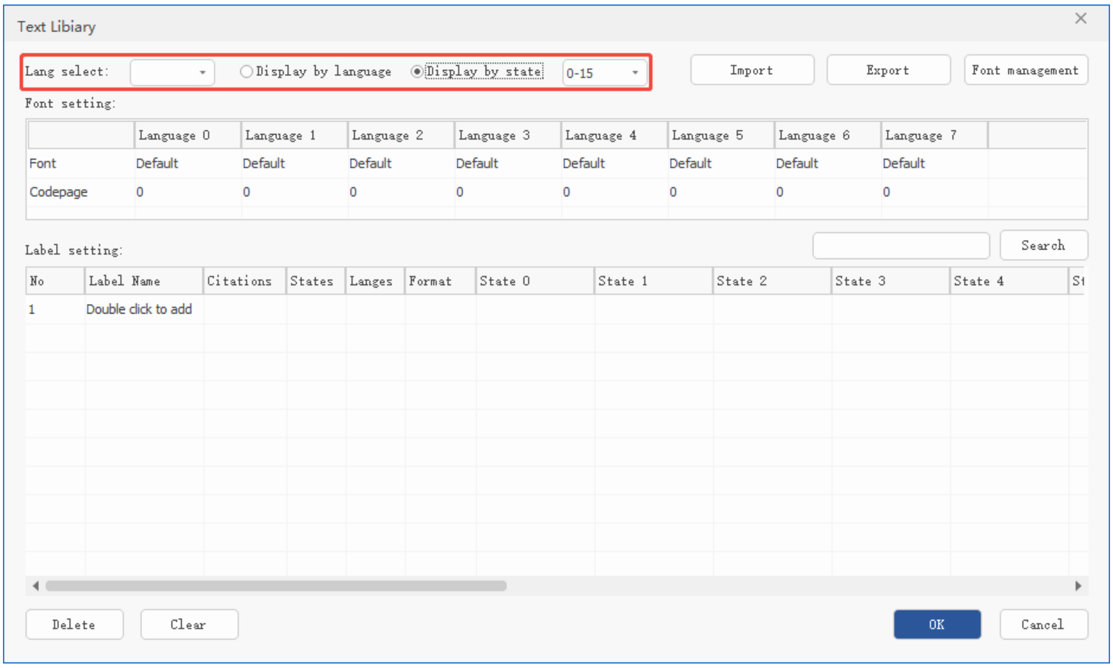
State select: ● Display by language ○ Display by state

Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	La
1	Double click to add										



Lang select: ○ Display by language ● Display by state 0-15

Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

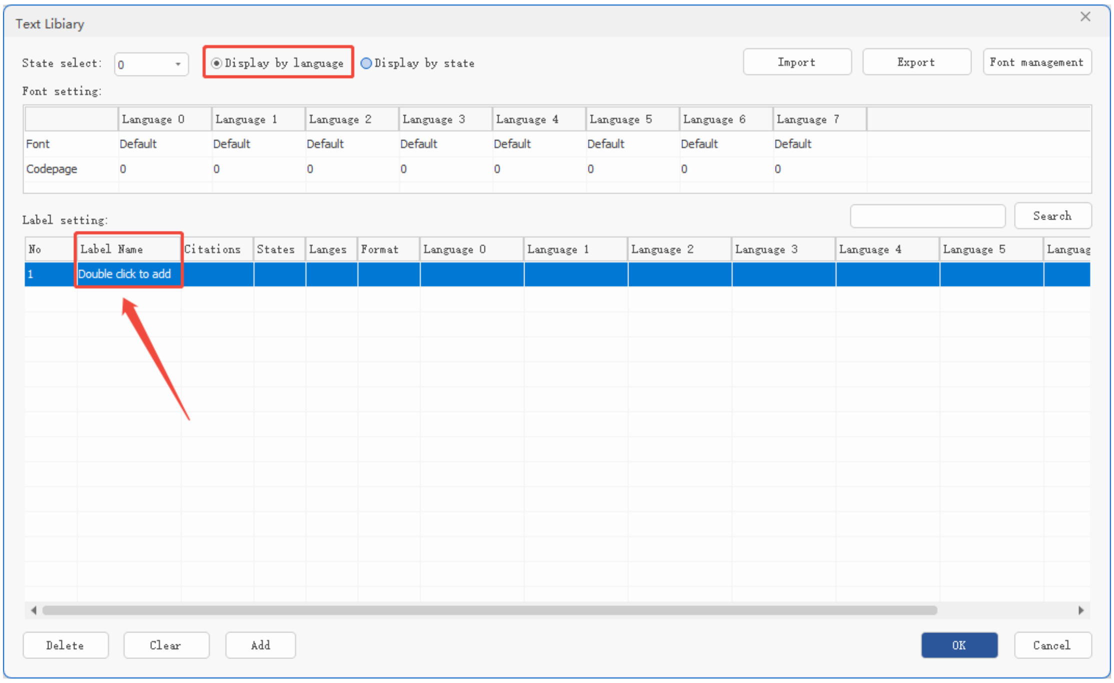
Label setting:

No	Label Name	Citations	States	Langes	Format	State 0	State 1	State 2	State 3	State 4	St
1	Double click to add										

(2) How to Operate

1) Text Lib Setting

A. Double-click “label name” to add ("Label" is used to distinguish which text library the component calls).



State select: 0

● Display by language ○ Display by state

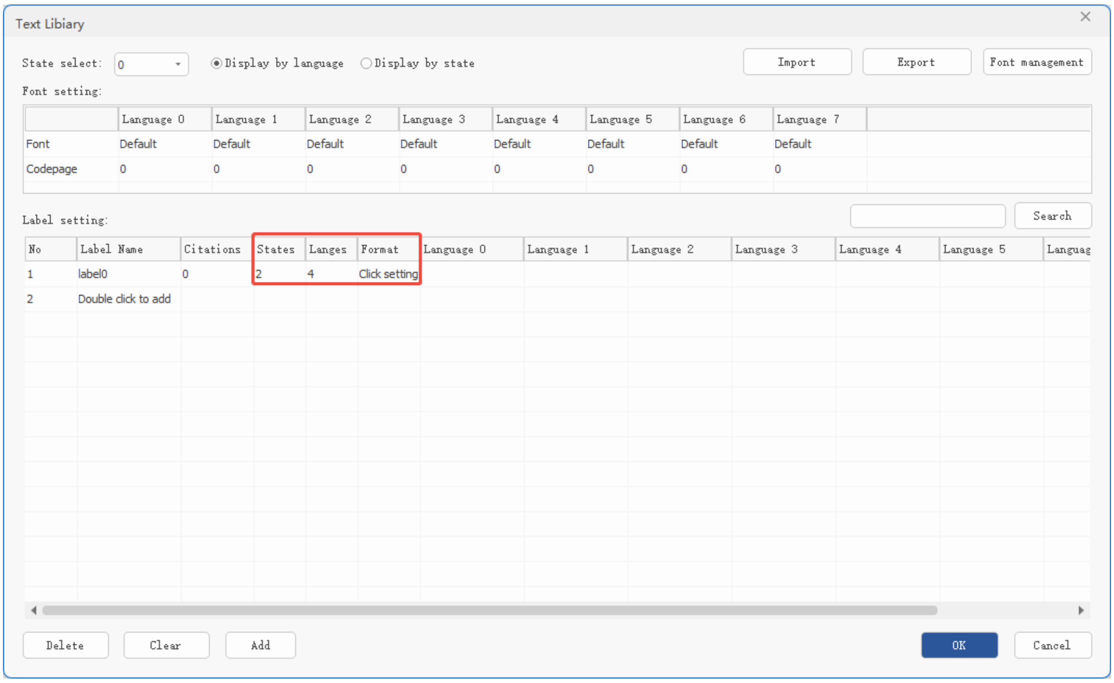
Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	Double click to add											

B. Set the required number of states and languages, and the format of the text.



State select: 0

● Display by language ○ Display by state

Font setting:

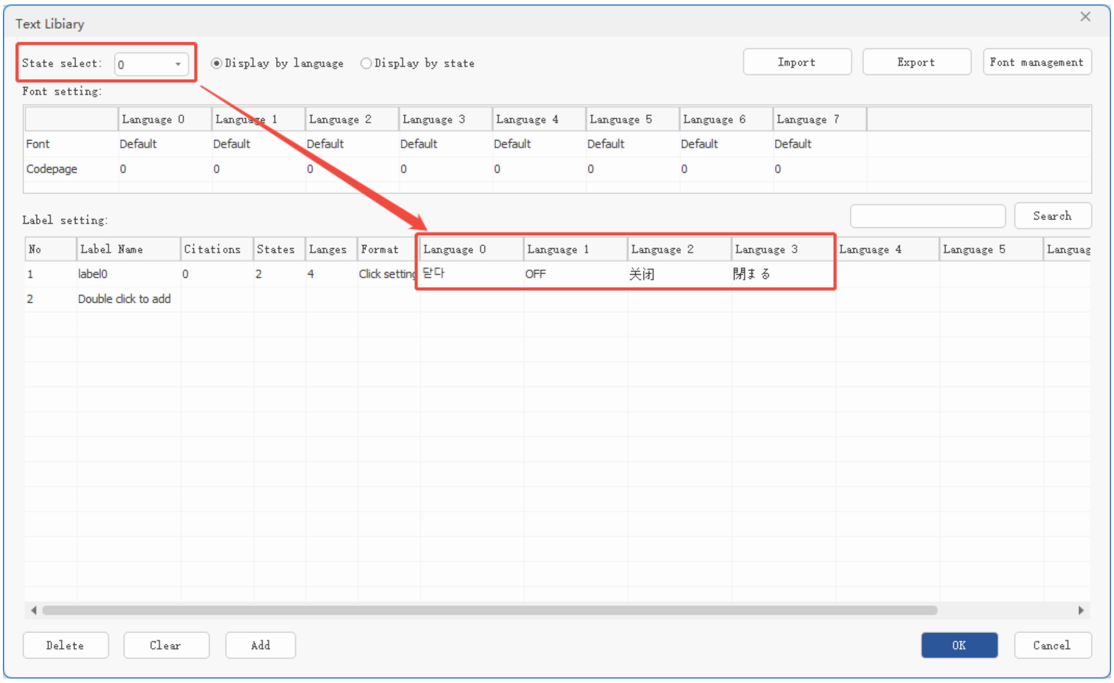
	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	label0	0	2	4	Click setting							
2	Double click to add											

C. Select the state, and add the text content to be displayed to the corresponding number of languages in the selected state.

For example: under state 0, add corresponding language text.



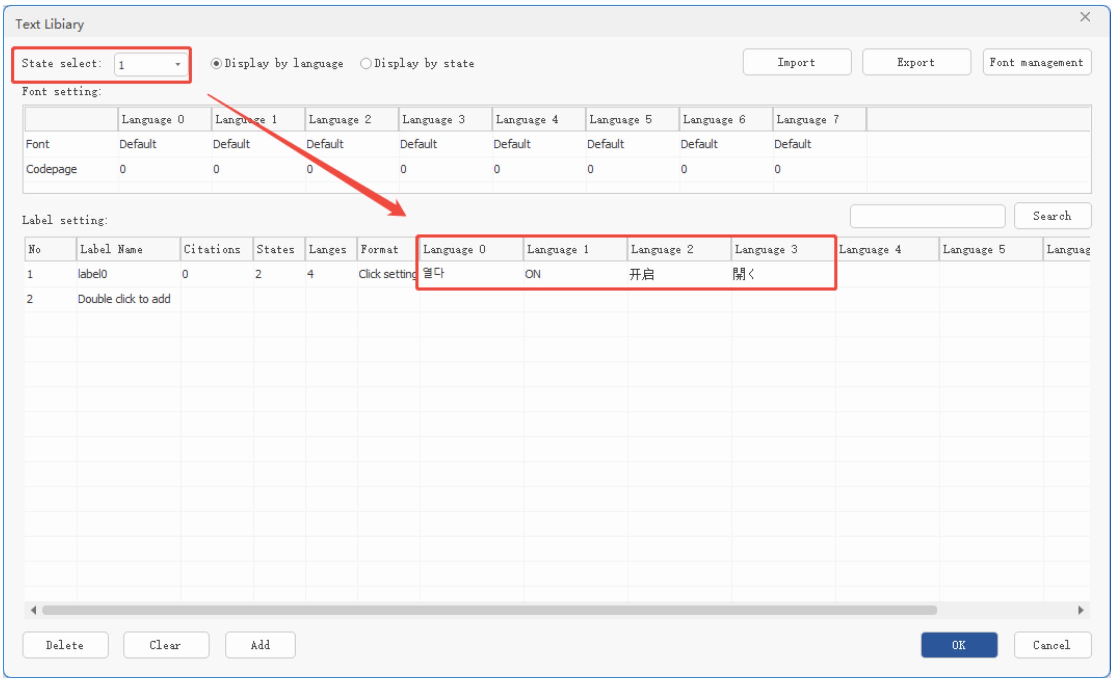
Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	label0	0	2	4	Click setting	닫다	OFF	关闭	閉まる			
2	Double click to add											

D. In “state select”, switch it into state 1, then you can add needed text content under state 1 (how many state select = how many states).



Font setting:

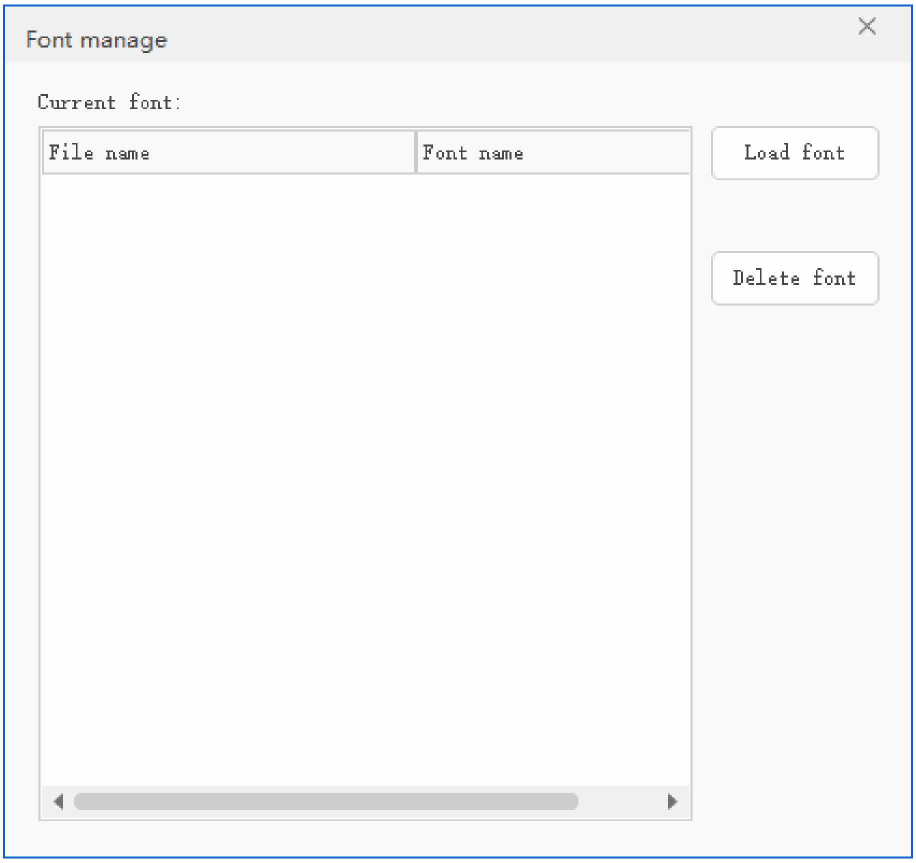
	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

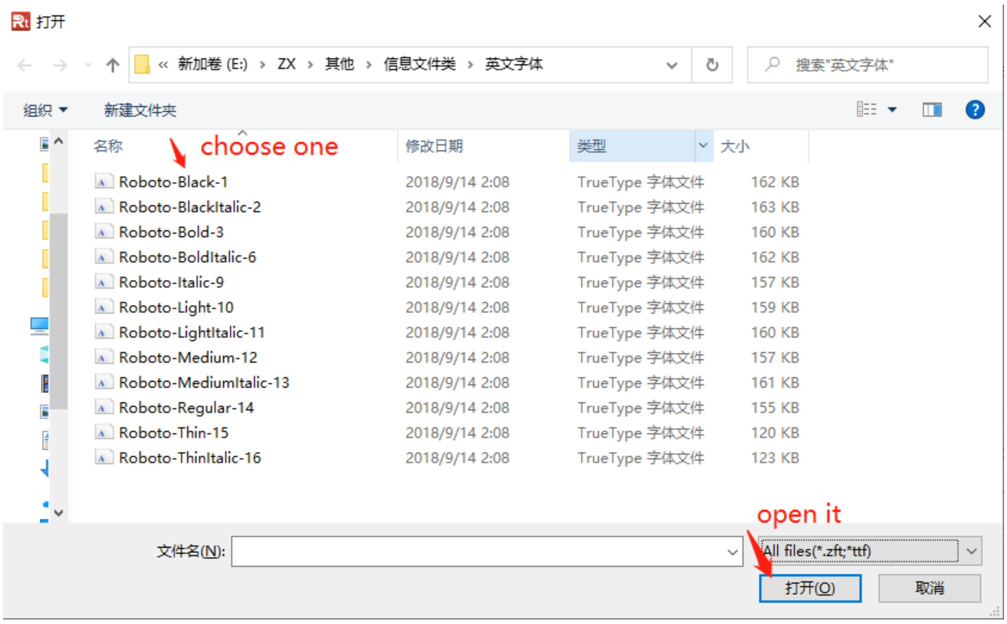
No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	label0	0	2	4	Click setting	열다	ON	开启	開く			
2	Double click to add											

E. After setting the label and language text, if no need to set font, click [OK] to save.

F. If you need to set the font, please add the font file to the project firstly. After importing the font, you can set the font for each language in the [Font Settings], then call the corresponding text library and run it to display.



G. Click "Load font", and find needed one, click "open" (.ttf & .zft are valid).

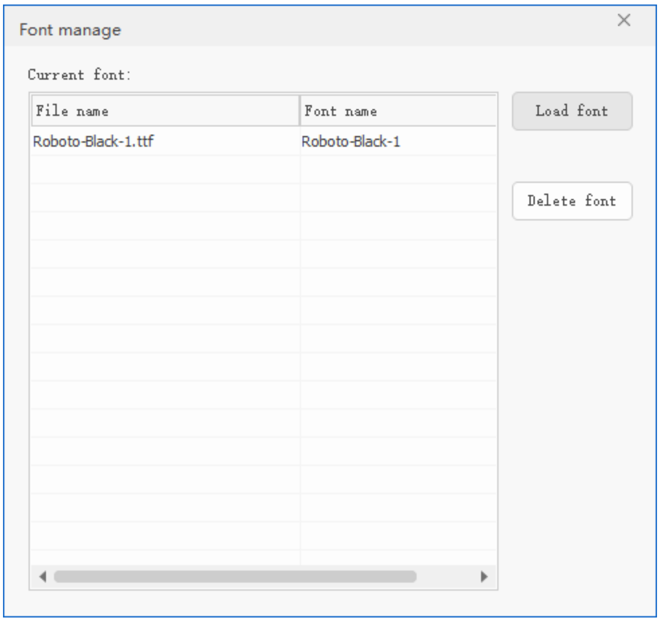


名称	修改日期	类型	大小
Roboto-Black-1	2018/9/14 2:08	TrueType 字体文件	162 KB
Roboto-BlackItalic-2	2018/9/14 2:08	TrueType 字体文件	163 KB
Roboto-Bold-3	2018/9/14 2:08	TrueType 字体文件	160 KB
Roboto-BoldItalic-6	2018/9/14 2:08	TrueType 字体文件	162 KB
Roboto-Italic-9	2018/9/14 2:08	TrueType 字体文件	157 KB
Roboto-Light-10	2018/9/14 2:08	TrueType 字体文件	159 KB
Roboto-LightItalic-11	2018/9/14 2:08	TrueType 字体文件	160 KB
Roboto-Medium-12	2018/9/14 2:08	TrueType 字体文件	157 KB
Roboto-MediumItalic-13	2018/9/14 2:08	TrueType 字体文件	161 KB
Roboto-Regular-14	2018/9/14 2:08	TrueType 字体文件	155 KB
Roboto-Thin-15	2018/9/14 2:08	TrueType 字体文件	120 KB
Roboto-ThinItalic-16	2018/9/14 2:08	TrueType 字体文件	123 KB

文件名(N):

All files(*.zft;*ttf)

H. After loading, corresponding file name and font name will be shown in the list (only for .ttf format, font name can be shown).



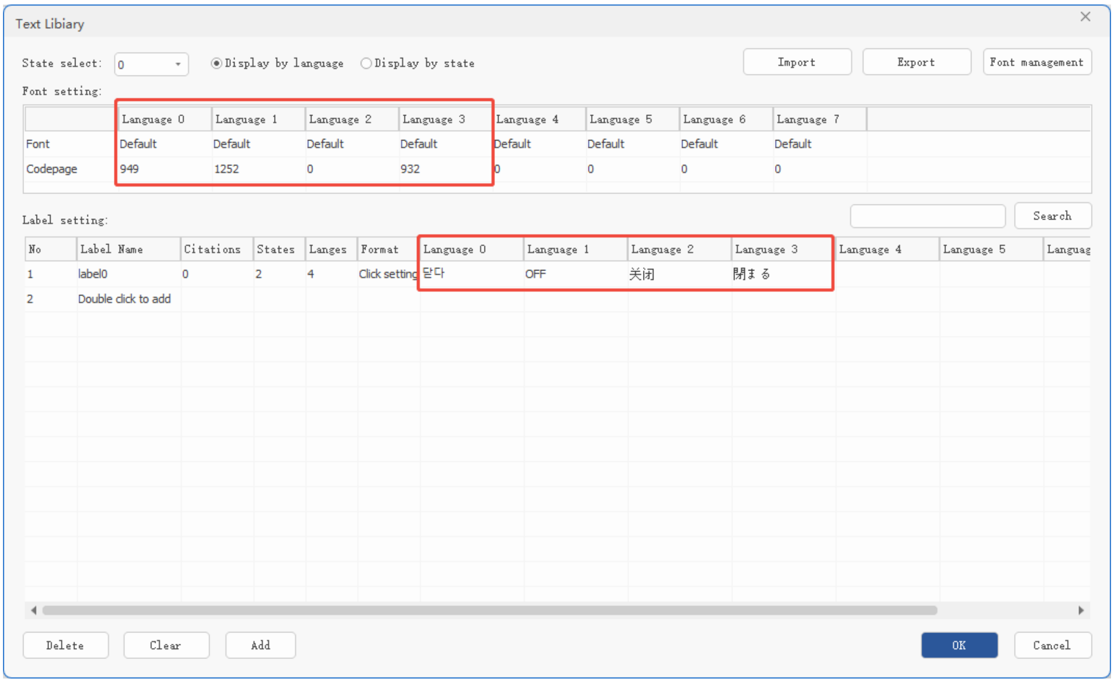
Current font:

File name	Font name
Roboto-Black-1.ttf	Roboto-Black-1

I. After that, you can set each language's font in "font setting".

Notes:

a) please enter language code in codepage, except the system language code. It can be seen language 1, language 2, and langue 4 all are not system language, corresponding codepages are 949, 1252, and 932.



State select: 0 ● Display by language ○ Display by state

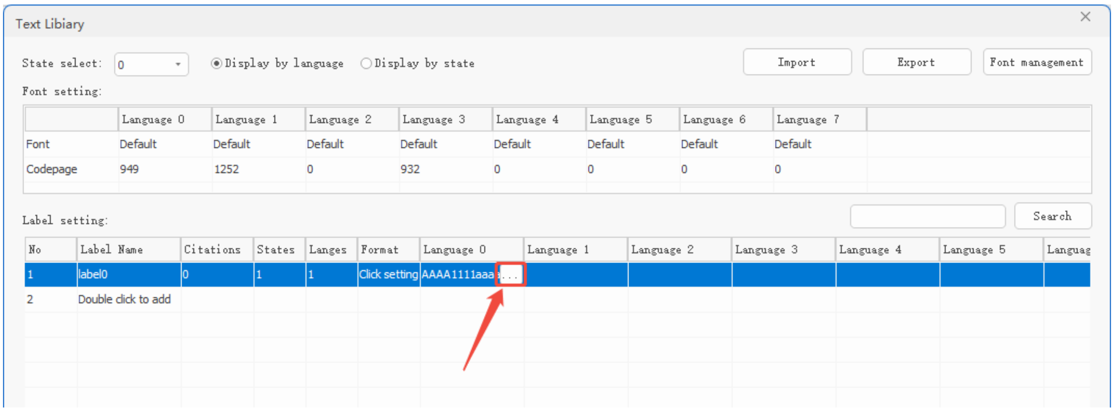
Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	949	1252	0	932	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	label0	0	2	4	Click setting	닫다	OFF	关闭	閉まる			
2	Double click to add											

b) clicking the small box in the text content allows you to wrap the text to a new line, which will then be displayed on the next line when the text content is applied.



Text Libiary

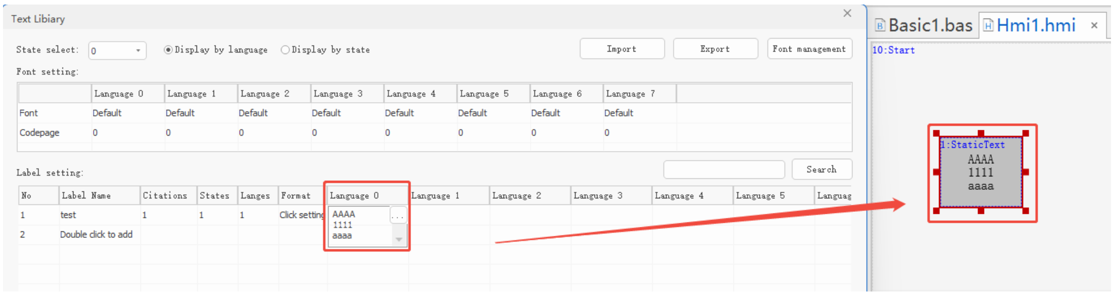
State select: 0 ● Display by language ○ Display by state

Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	949	1252	0	932	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	label0	0	1	1	Click setting	AAAA1111aaa ...						
2	Double click to add											



Text Libiary

State select: 0 ● Display by language ○ Display by state

Font setting:

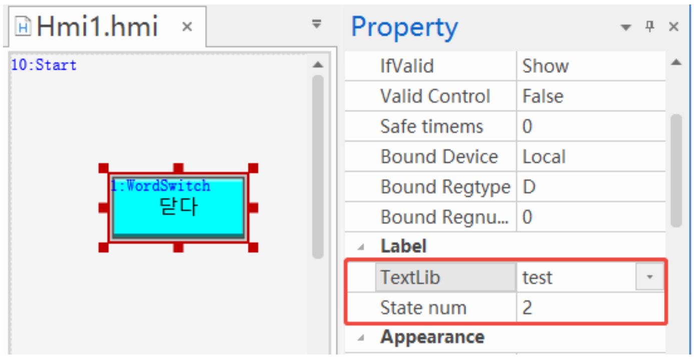
	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	test	1	1	1	Click setting	AAAA 1111 aaaa						
2	Double click to add											

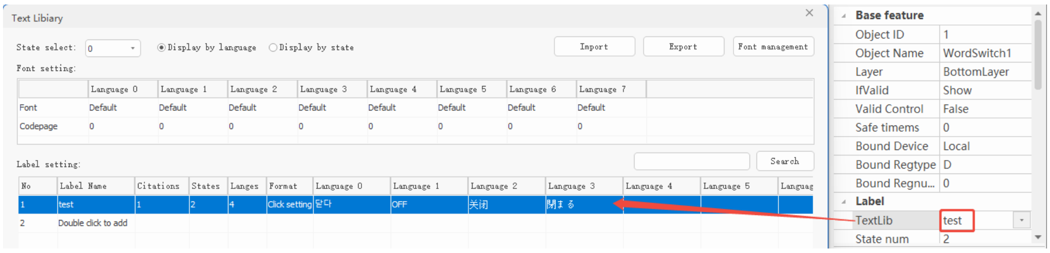
2) <u>Text Lib Calling</u>

A. According to above steps, set text library, then you can call the text lib in component “property”.



IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnu...	0
Label	
TextLib	test
State num	2
Appearance	

- Double-click text library label name to open the text library and jump to corresponding text library position.



Text Libiary

State select: 0 ● Display by language ○ Display by state

Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Languag
1	test	1	2	4	Click setting	닫다	OFF	关闭	閉まる			
2	Double click to add											

Base feature	
Object ID	1
Object Name	WordSwitch1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnu...	0
Label	
TextLib	test
State num	2

B. After downloading it, you can switch the language through the command "Hmi_LANG=ilang(Lang No.)".

And you can change the language state through modifying the value of register, for example: register value 0 shows language 0's state 0, when the register value is 1, which shows language 0's state 1.

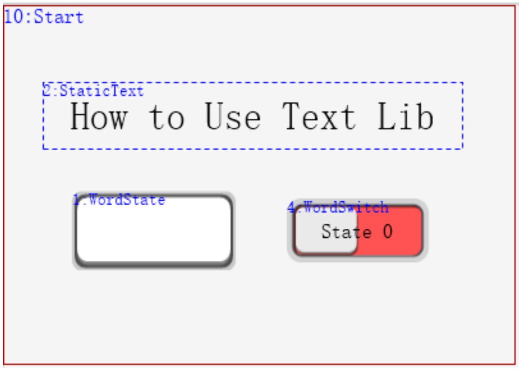
Next, let's see one routine:

Routine Design:

use "WordSwitch" and "WordState" components, and do state switching, and the "WordState" component calls the subfunction to run the command "Hmi_LANG" for language switching.

- **HMI Design:**

In HMI window, please add below components: Static Text (only used for showing the title), WordState, WordSwitch.



- **Basic Programming & Codes:**

```
GLOBAL dim runsub_sign = 0          'variable flag
GLOBAL SUB sub_switch()             'call sunfunction to switch
    IF runsub_sign = 0 THEN
        langue0
    ELSEIF runsub_sign = 1 THEN
        langue1
    ELSE
        langue2
    ENDIF
    runsub_sign=runsub_sign + 1
    IF runsub_sign > 2 THEN
        runsub_sign = 0
    ENDIF
END SUB
```

```
GLOBAL SUB langue0()   'use language 0
    HMI_LANG=0
END SUB

GLOBAL SUB langue1()   'use language 1
    HMI_LANG=1
END SUB

GLOBAL SUB langue2()   'use language 2
    HMI_LANG=2
END SUB
```

- **How to Call:**

  [WordState] Component:

  --in “property” window, please selected needed text library content in “text lib” (this example selects label 0)--

  --in “property” window, call sub function sub_switch() in “action”--

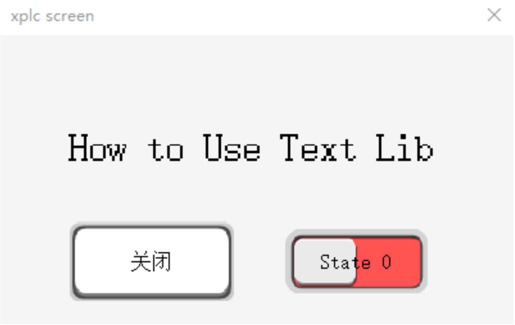
  [WordSwitch] Component:

  --in “property” window, set state num as 2, and add corresponding texts--

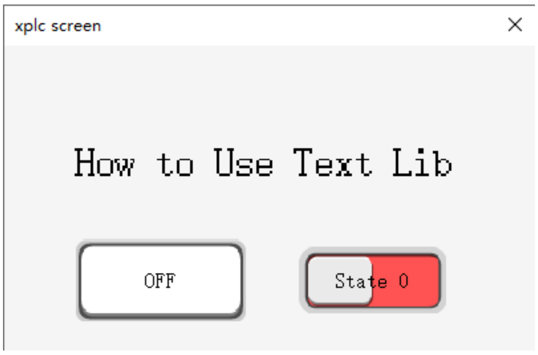
  --in “property” window, select data loop in “action”, set action data as 1--

- **How the Effect is:**

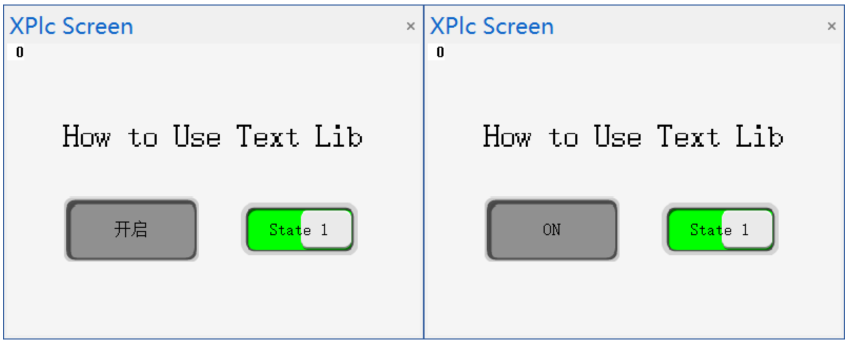
  When two components are state 0, now [WordState] will show the language 0 text under state 0, that is, it will be 关闭.



When click “关闭”, then it will switch language, switch to language 1, it will be “OFF”.



When click “State 0”, it can switch the state.



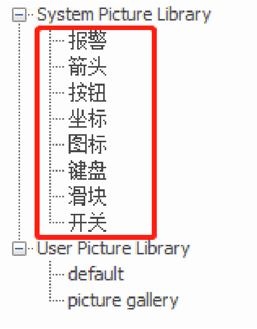
2.2.2. Picture Lib

(1) What is “Picture Lib”

A library that integrates and stores Hmi component style images or customized images. It includes system image library and user image library. This library supports modifying image style color, display content/color corresponding to different states, etc.

➢ System Picture Library

The system image library is the system default image and does not support deletion or addition. Here, rich component styles for you, you can select and use directly.

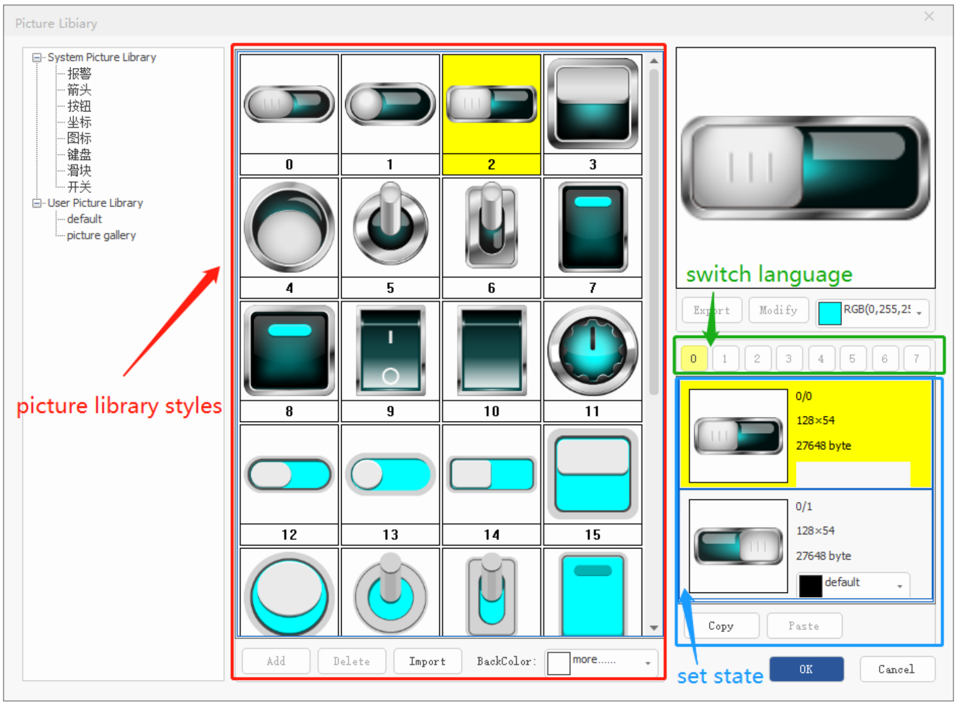


报警	Alarm
箭头	Arrow
按钮	Button
坐标	Coordinate
图标	Icon
键盘	Keyboard
滑块	Slider
开关	Switch

➢ User Picture Library

Here, you can create your own picture library, also can add external images.

Note: The maximum number of image libraries is 512!

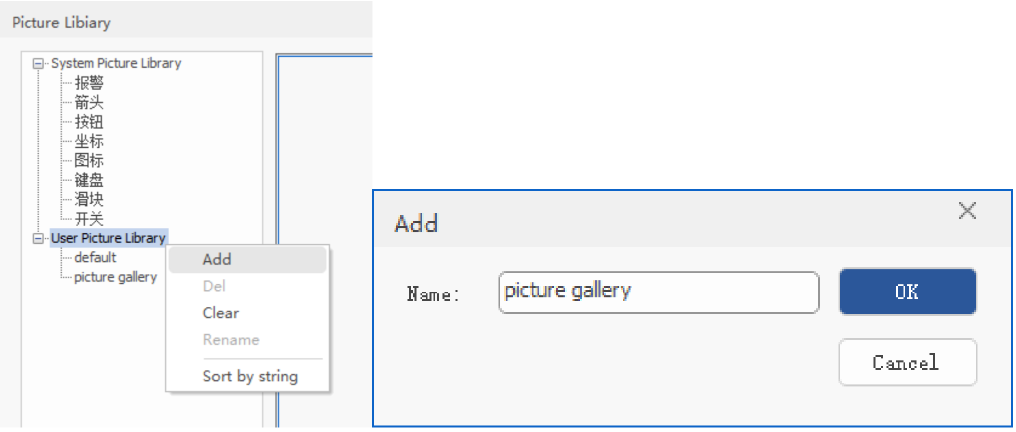


(2) How to Use (take the example of "switch on and off the led")

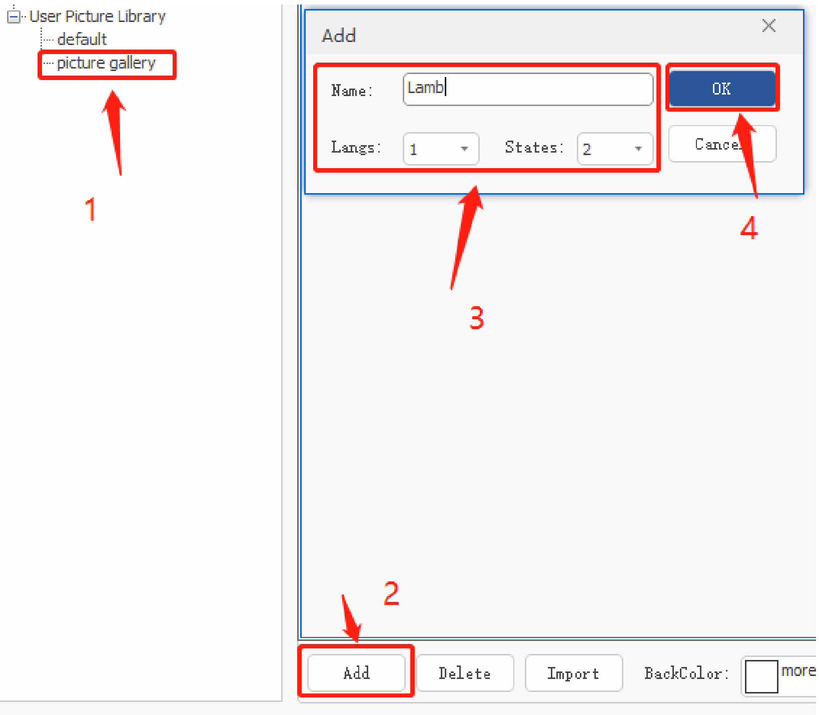
<u>Step 1: [How to Build New Picture Lib]</u>

A. Import "Lamb" Picture

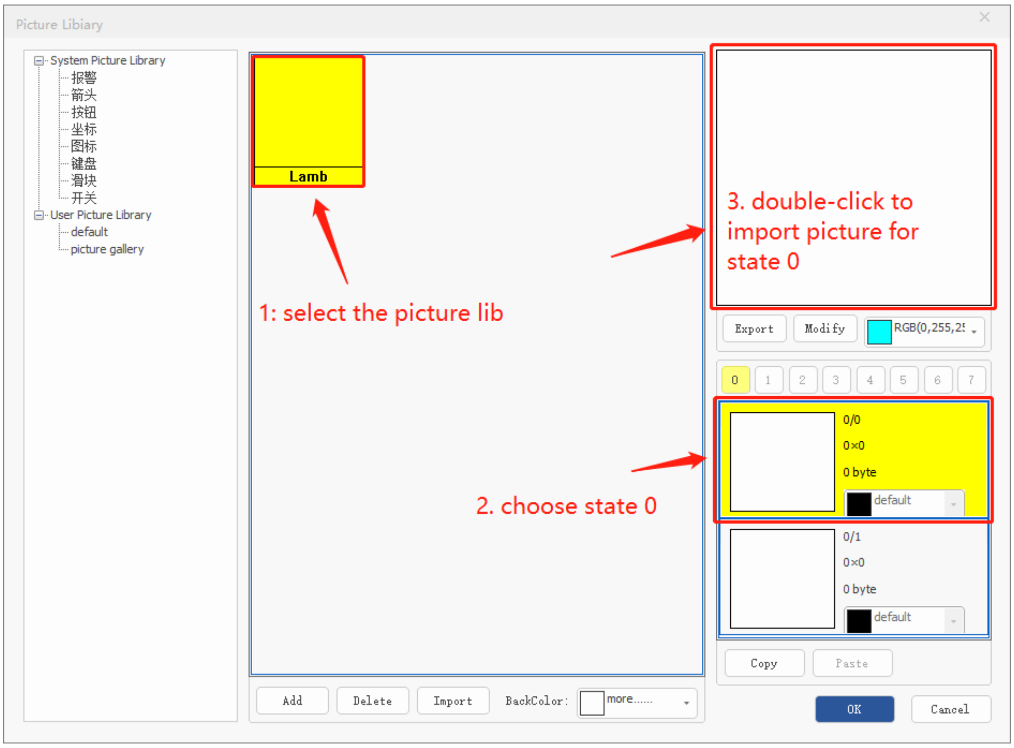
a. Right-click "user picture lib", and select "add", then enter new picture lib name in the popped window, then click OK.



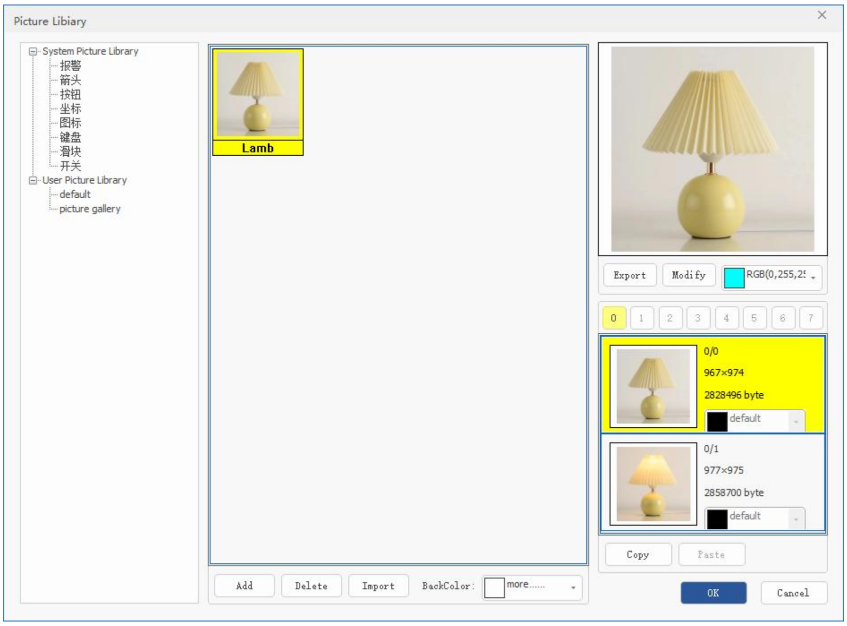
b. Double-click new created picture lib "picture gallery", then click "add", in new appeared window, enter the name, set well corresponding language numbers and state numbers.



c. After that, single click this picture library (yellow background color means it is selected), then in right side, please select corresponding states, and add pictures.

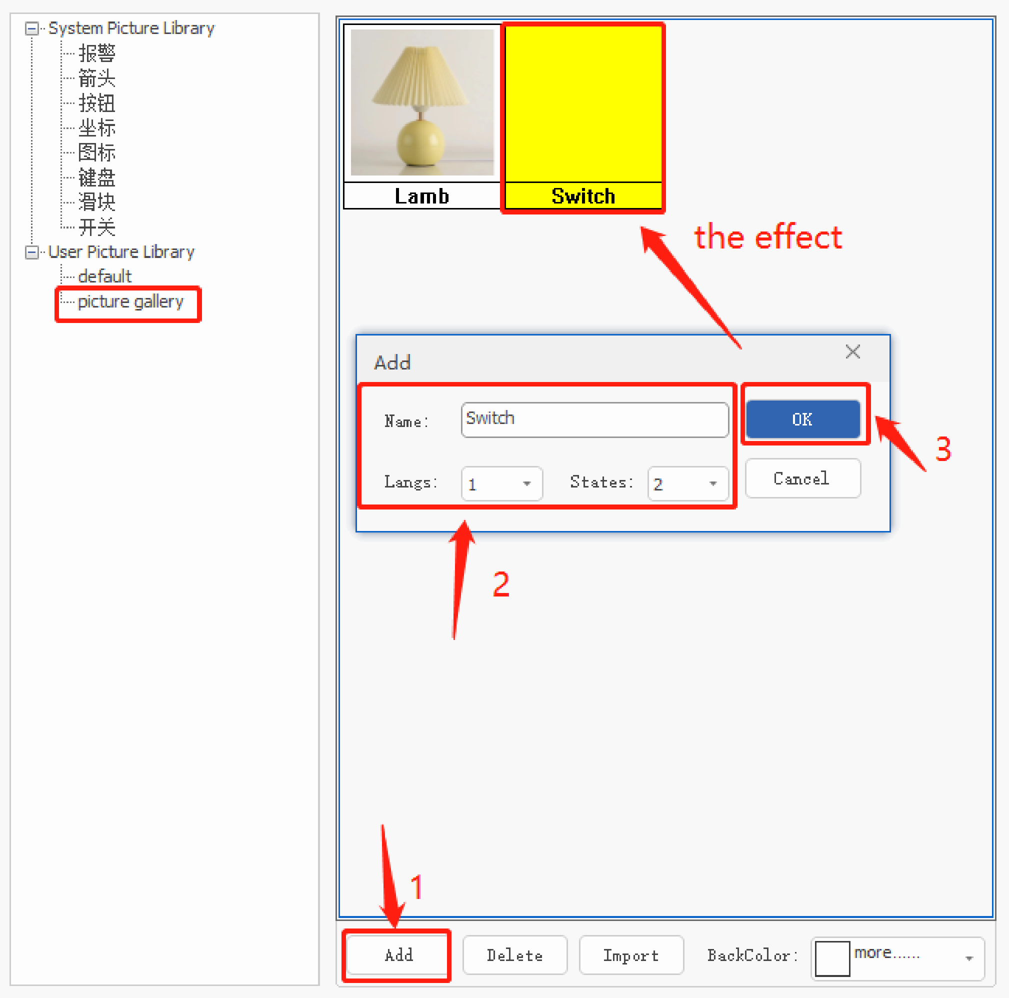


d. According to “c” to import the picture for state 0, and keep adding the picture of state 1. When both added, please click OK.

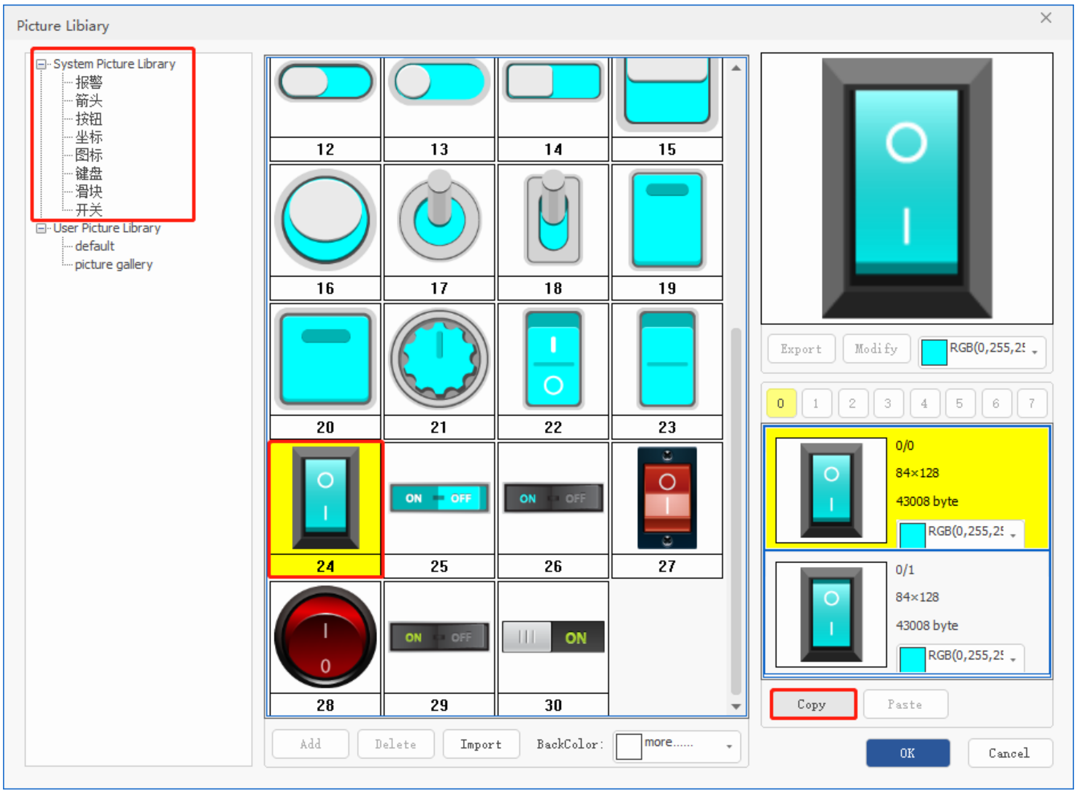


B. Copy Switch Button Styles

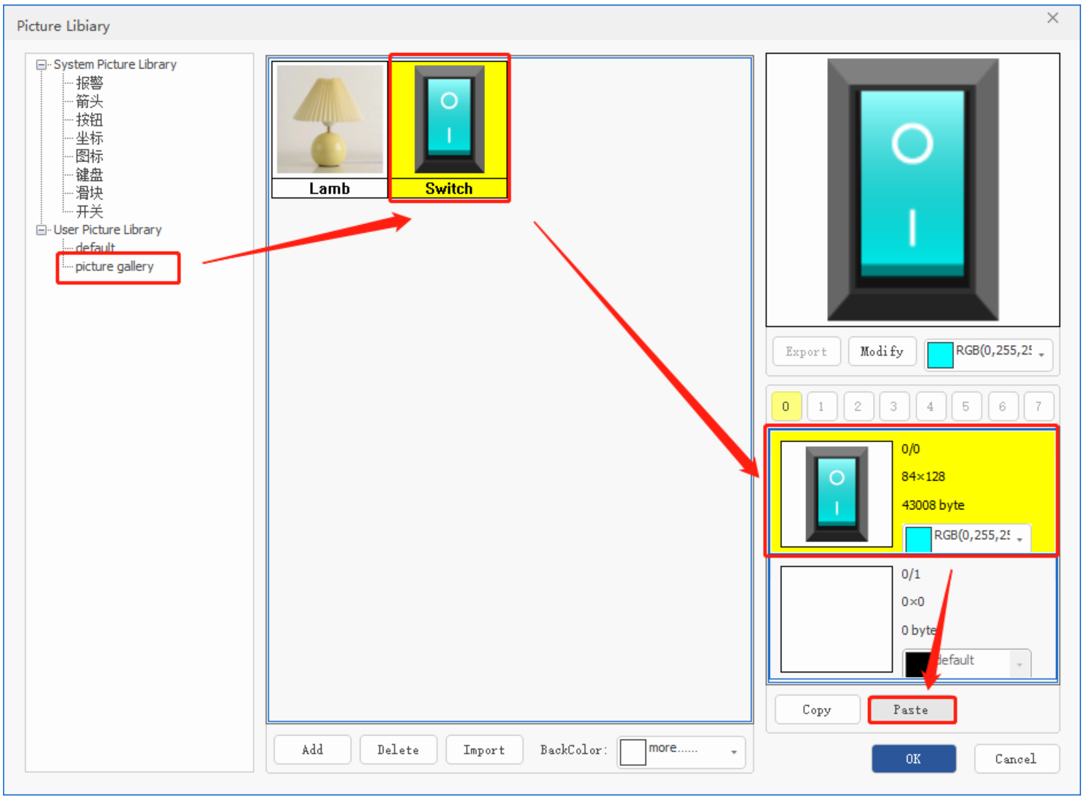
a. In new created “picture gallery”, then add one picture lib whose name is “switch”, also, set language numbers and state numbers.



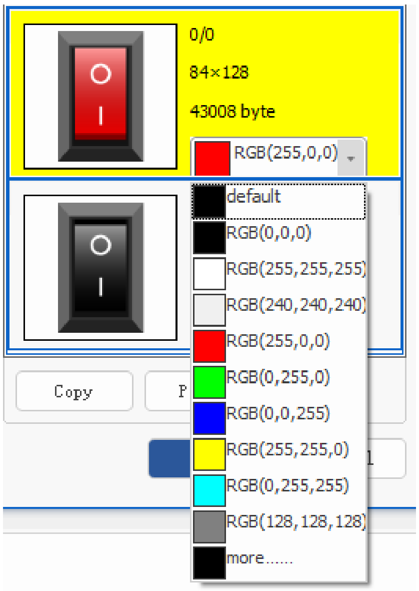
b. In “system picture library”, you can select needed format, and click “copy”. If there is no needed in system picture library, please add your own picture according to above “import”.



c. Back to new created “picture gallery”, select “switch”, choose state 0, click “paste”, new added.



d. Then, add state 1. At this time, you also can change the color, after all are OK, please click OK.

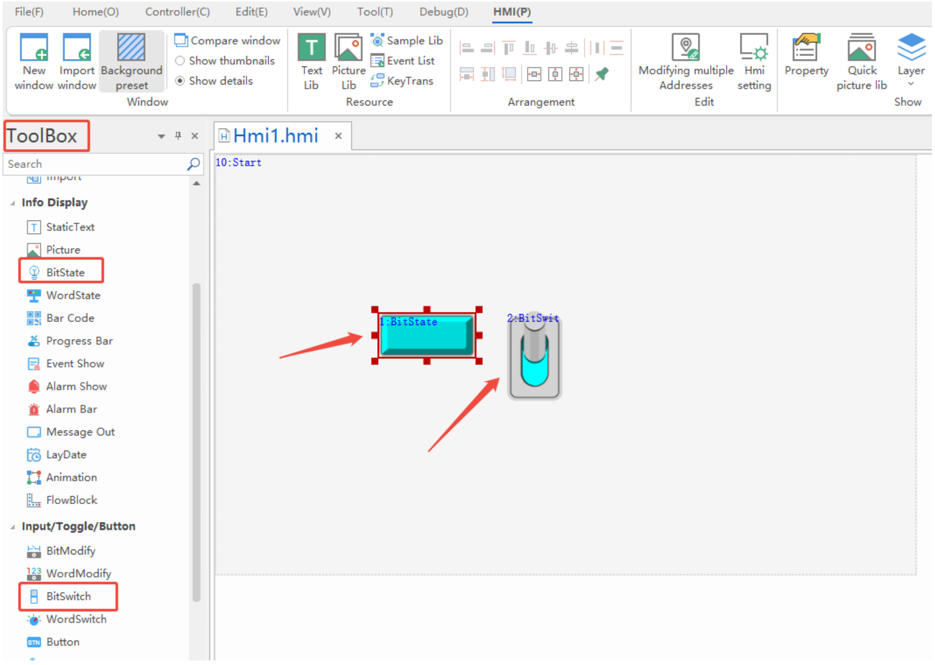


Step 2: [How to Apply Picture Lib]

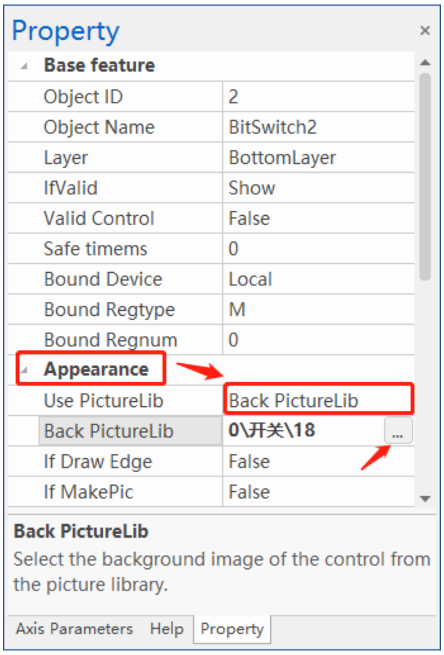
A. Apply Directly

a. After building HMI file, set auto run task No. for HMI file, then open the HMI file. Next, set HMI system property, add HMI components in HMI window.

b. In Tool Box, select and add “BitState” & “BitSwitch” into window 10.



c. Click component 2 “BitSwitch”, then in the right side, its’ property window will appear. Then in “Appearance” – “Use Picture Lib”, select “Back Picture Lib”, then in next item, click “…”, corresponding picture library will open.

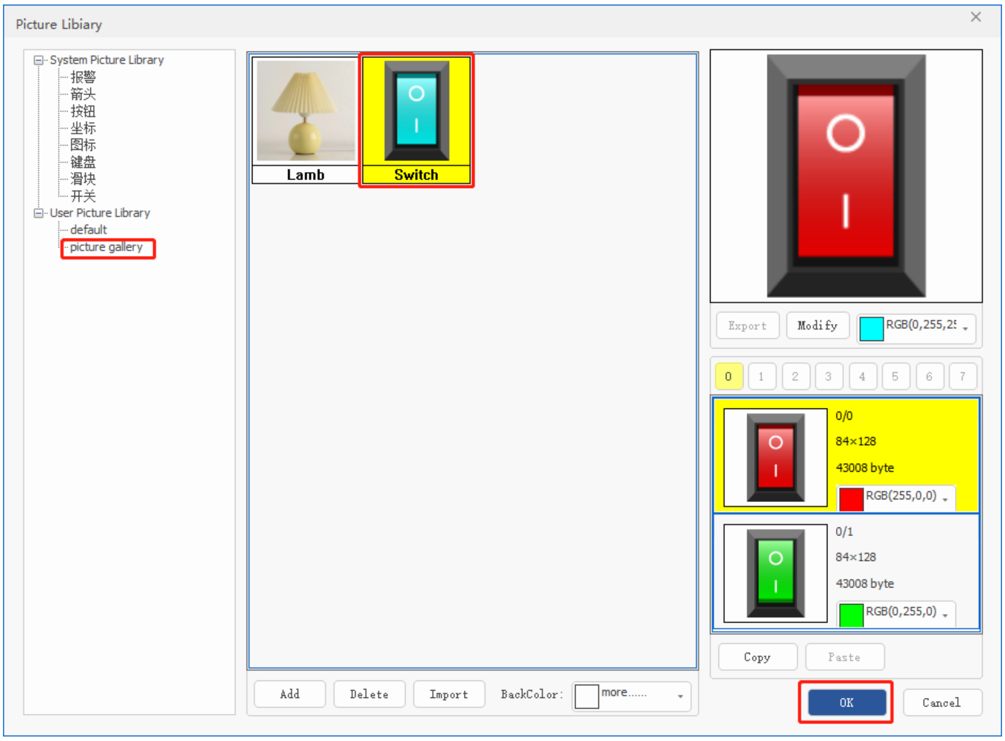


Property

Base feature	
Object ID	2
Object Name	BitSwitch2
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\开关\18
If Draw Edge	False
If MakePic	False

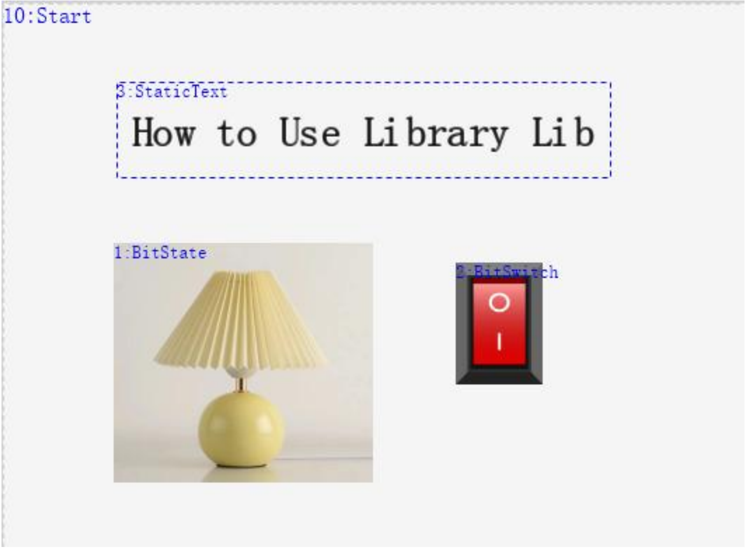
Back PictureLib
Select the background image of the control from the picture library.

d. In appeared picture library, you can find needed one, and click OK, then the picture is applied directly. And you can adjust the size of component as needed (component 1 is same as this, in “Bound RegType”, please set as M2)

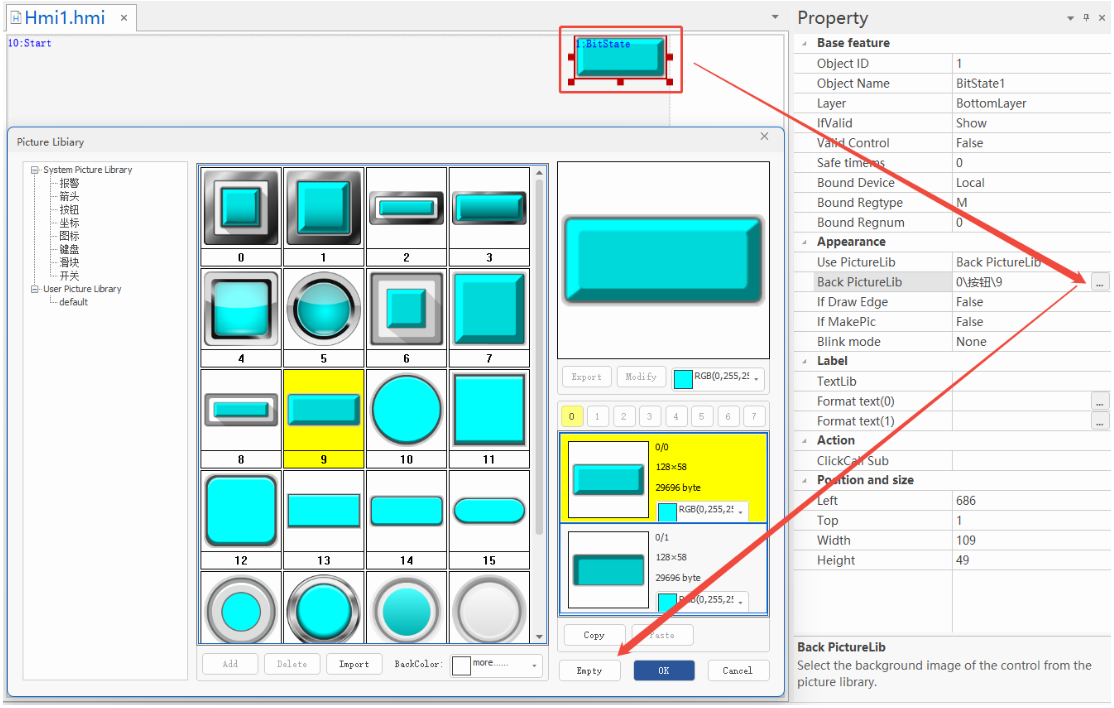


Picture Libiary

After that, apply successfully:



Tip: if it sets the button icon / background picture for the component appearance, you could click “empty” to make the background not to be shown.

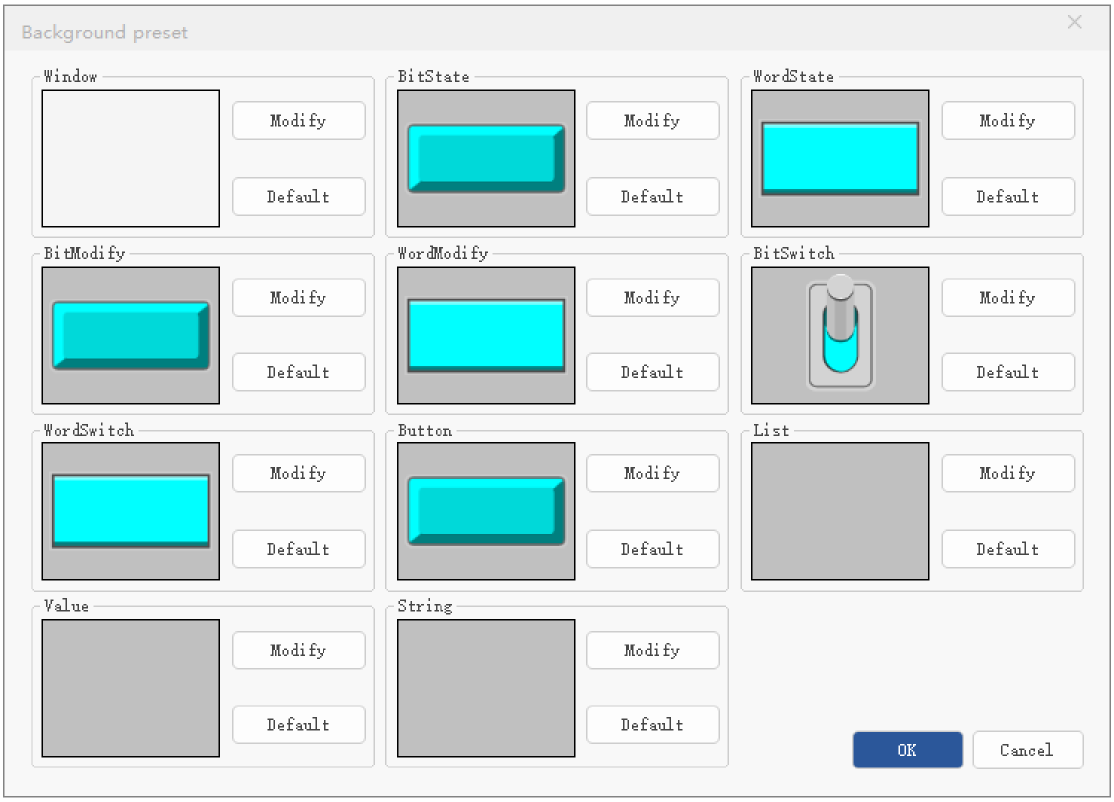


Base feature	
Object ID	1
Object Name	BitState1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Blink mode	None
Label	
TextLib	
Format text(0)	
Format text(1)	
Action	
ClickCall Sub	
Position and size	
Left	686
Top	1
Width	109
Height	49

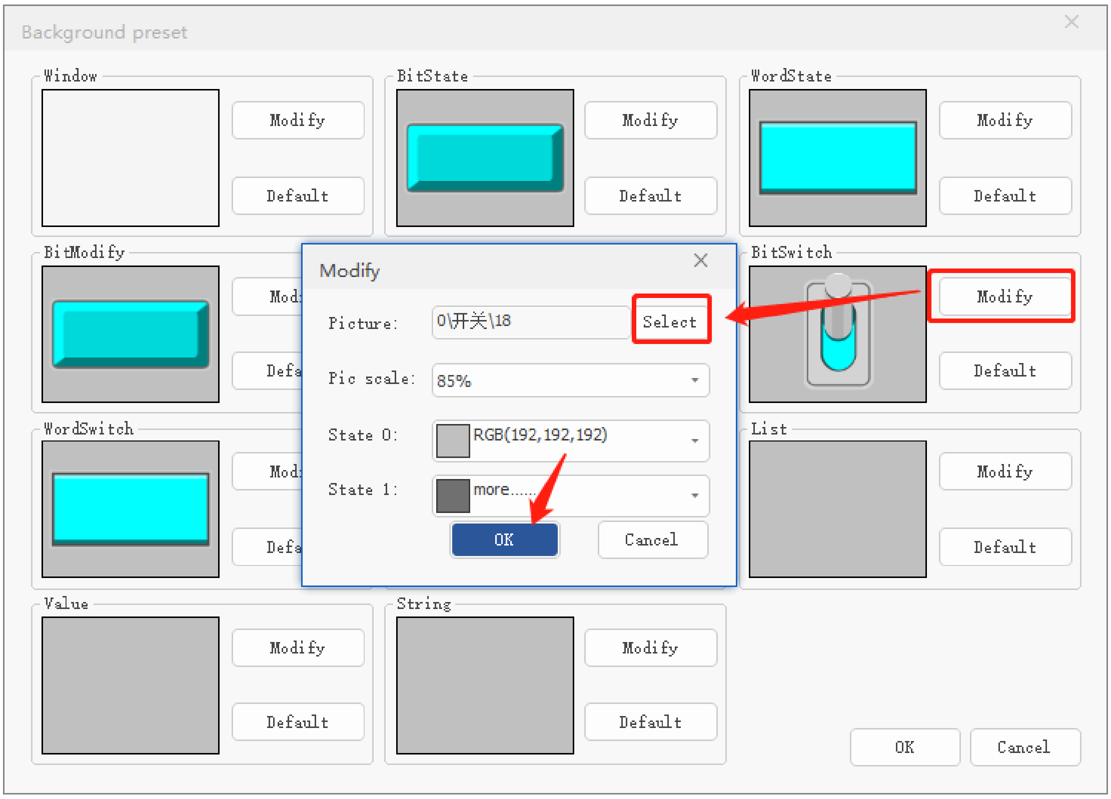
Back PictureLib
Select the background image of the control from the picture library.

B. Apply by Background Preset

a. Open HMI file, then open below window through “HMI” – “Backgroud Preset”, you can directly modify the component formats here.



b. Select needed format (here takes the example of “BitSwitch”), click “Modify”, then select in below window.



Modify

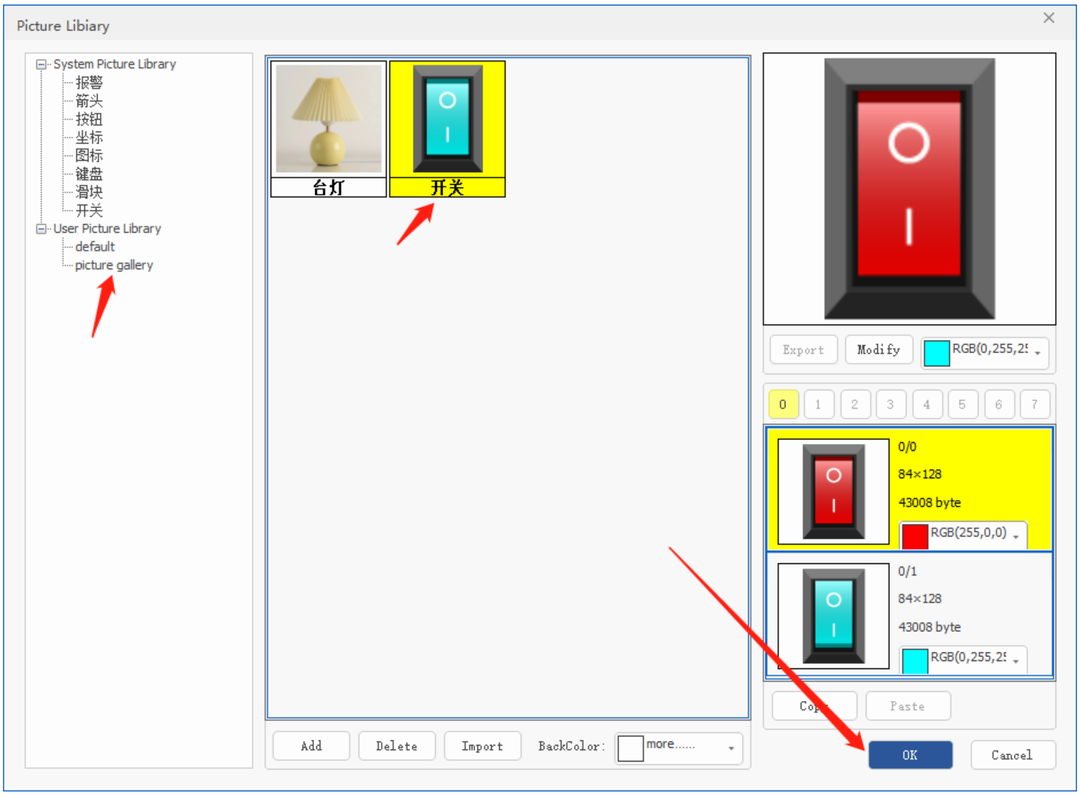
Picture: 0\开关\18

Pic scale: 85%

State 0: RGB(192,192,192)

State 1: more......

c. Click “Select”, then find which one you need, click OK.



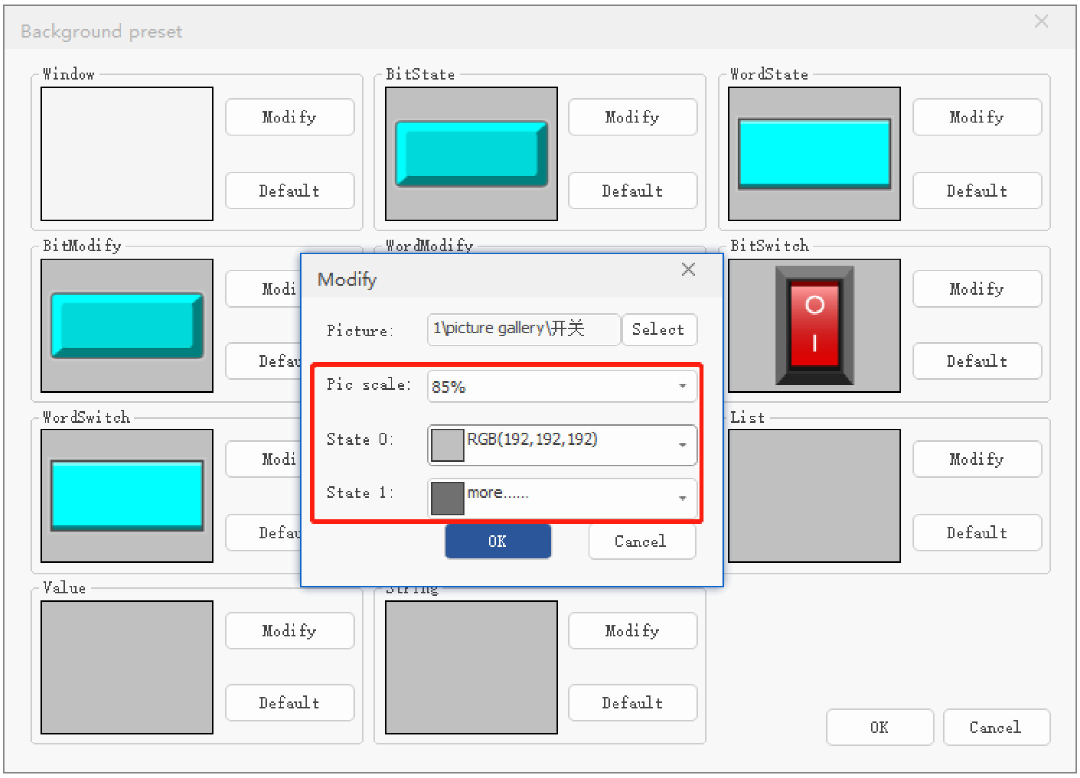
RGB(0,255,25

0/0
84×128
43008 byte
RGB(255,0,0)

0/1
84×128
43008 byte
RGB(0,255,25

BackColor: more......

d. Back to “modify” window please, you can set picture ratio, state background color here. Then, click OK (Note: state background color is only for preview, not take effect when real applied).



Modify

Picture: 1\picture gallery\开关 Select

Pic scale: 85%

State 0: RGB(192,192,192)

State 1: more......

OK Cancel

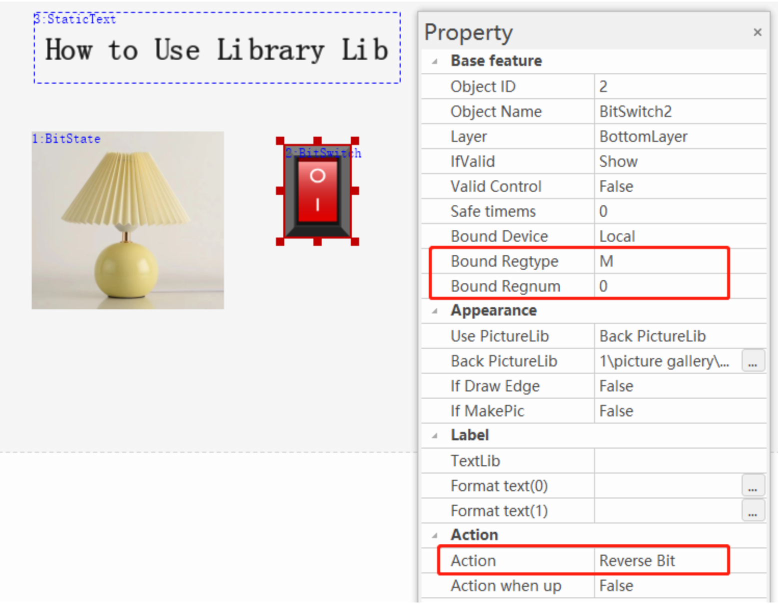
Step 3: [How to Call Sub-function]

A. “BitState” Component

Single-click the component, then in its “property” window, set “Bound Regtype” as M0.

B. “BitSwitch” Component

Single-click the component, then in its “property” window, set “Bound Regtype” as M0. And in “action”, please select “Reverse Bit”.



Property

Base feature	
Object ID	2
Object Name	BitSwitch2
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	1\picture gallery\...
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	
Format text(1)	
Action	
Action	Reverse Bit
Action when up	False

Step 4: Check the Effect

A. When the program is downloaded into controller / simulator, open xplc screen to show it.

B. When these two components are in state 0, that is, lamb and switch are OFF. When these two are state 1, that is, it is ON.

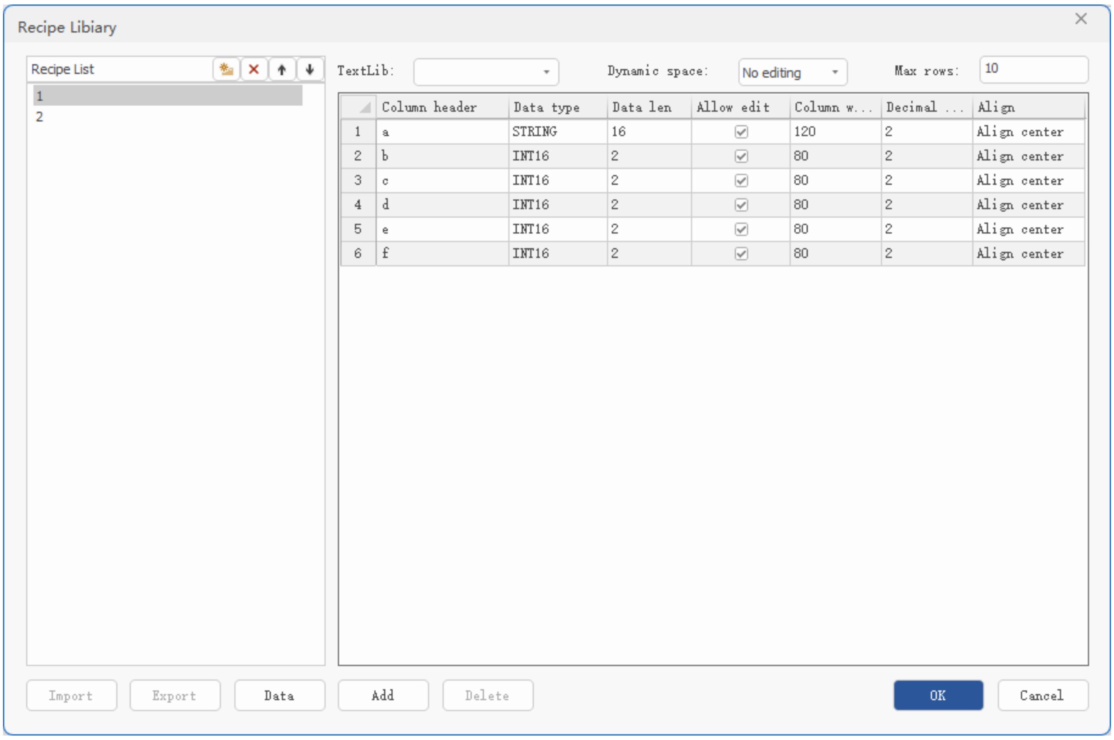
2.2.3. Recipe Lib

(1) What is “Recipe Lib”

This collects various recipe groups: the left list of the recipe library interface displays different types of recipe groups; the right table shows the raw materials and formatting settings for the selected recipe group; and the data shows recipes composed of different proportions of raw materials.

- **Properties**

Property	Descriptions
✱	Add one recipe group.
✕	Delete selected recipe group.
↑	Move the selected recipe group up one row.
↓	Move the selected recipe group down one row.
Recipe List	● Column Header: the first row of the table header when displaying recipe data, which contain a maximum of 31 characters. ● Data Type: assign the data type of the recipe list, there are UINT8, INT8, UINT16, INT16, UINT32, INT32, FLOAT, DOUBLE, STRING. When the type is FLOAT / DOUBLE, the decimal number becomes valid (can be edited), for other types, the decimal number is not valid. First row uses STRING by default, other rows use INT16 by default. ● Data Len: assign the max characters of the recipe list that can be shown in each column. Default length of first column: 16; default length of other columns: 16. ● Allow Edit: whether this data is editable. ● Column Width: each column's width. Default width of first column: 120; default width of other columns: 80. ● Decimal Places: the number of decimal places displayed when the type is floating point. ● Align: set the align mode of the recipe, there are center-align, right-align, left-align.
Text Lib	Whether to use a text library for the column headers of the recipe items. If so, pass in the text library name. Each column header occupies one state in the text library, and the column header cannot be modified at this time. When the text library is empty, the column headers take effect (become editable). For example, if a recipe group has 4 recipe ingredients, it corresponds to states 0 to 3 of the specified text library.
Dynamic Space	Byte size, providing space for recipe data editing
Max Rows	The maximum number of rows supported by the recipe data is mainly used to add rows using commands after downloading. If the entered number of rows is less than the current number of rows in the recipe data, the modification will fail. The default is 10.



Recipe List: 1, 2

TextLib:

Dynamic space: No editing

Max rows: 10

	Column header	Data type	Data len	Allow edit	Column w...	Decimal ...	Align
1	a	STRING	16	☑	120	2	Align center
2	b	INT16	2	☑	80	2	Align center
3	c	INT16	2	☑	80	2	Align center
4	d	INT16	2	☑	80	2	Align center
5	e	INT16	2	☑	80	2	Align center
6	f	INT16	2	☑	80	2	Align center

➢ Import (reserved)

It is one reserved function, which can import the recipe group.

Afte clicking “import”, select a .zbd file, and check needed recipe group name, then click “OK” to import the selected recipe groups. If the imported recipe group has the same name as an existing recipe group, the import of the recipe group with the same name will fail and a prompt will be displayed.

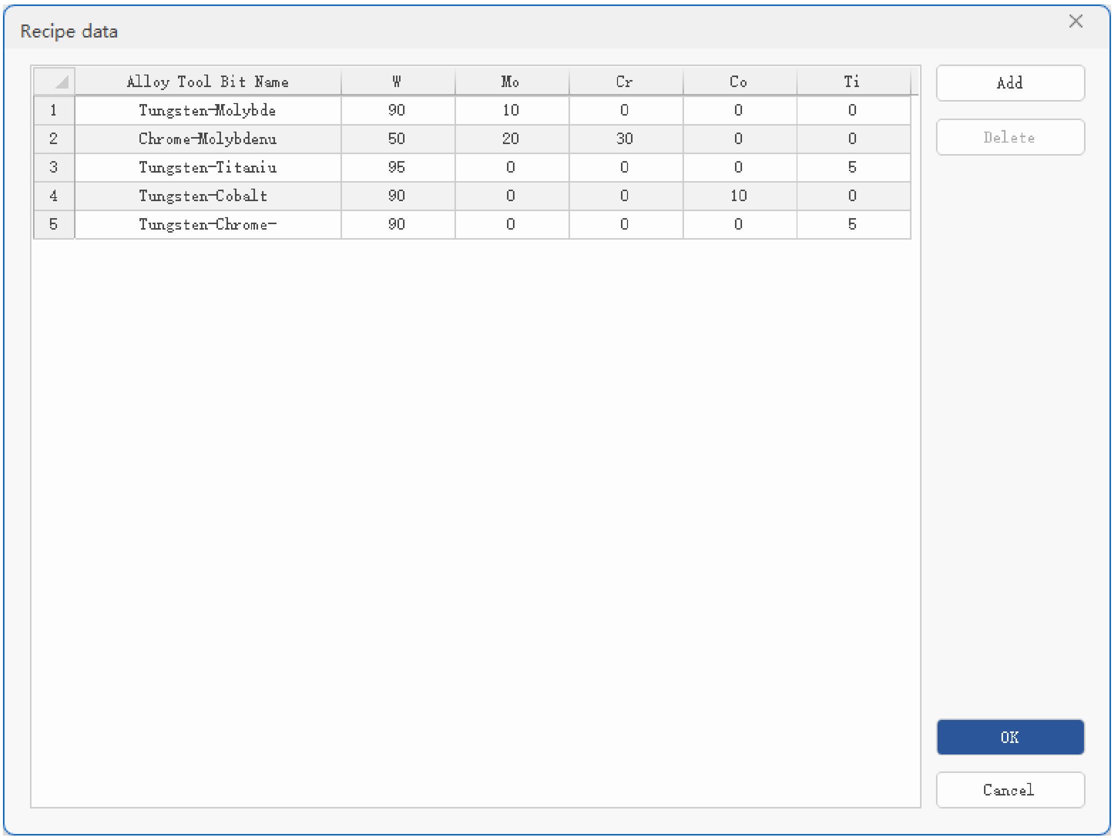
➢ Export (reserved)

It is one reserved function, which can export the recipe group.

Afte clicking “export”, check needed recipe group name, and click “OK” to export the selected recipe groups. The exported format is .zbd.

➢ Data

Select one certain recipe group, then click “Data” button, corresponding interface will appear. The interface described below displays data obtained from different proportions of the selected recipe group using the raw materials in the formula. *For example, an alloy cutter head is a recipe group composed of tungsten, molybdenum, chromium, cobalt, and titanium. The recipe data consists of various alloy cutter heads obtained from different proportions of these raw materials.*



Recipe data

	Alloy Tool Bit Name	W	Mo	Cr	Co	Ti
1	Tungsten-Molybde	90	10	0	0	0
2	Chrome-Molybdenu	50	20	30	0	0
3	Tungsten-Titaniu	95	0	0	0	5
4	Tungsten-Cobalt	90	0	0	10	0
5	Tungsten-Chrome-	90	0	0	0	5

a) Add: add a row at the end of the data. If the number of rows is equal to the maximum number of rows for this recipe, the addition of a row will fail.

b) Delete: select a row and click the delete button to remove that data.

➢ **Add**

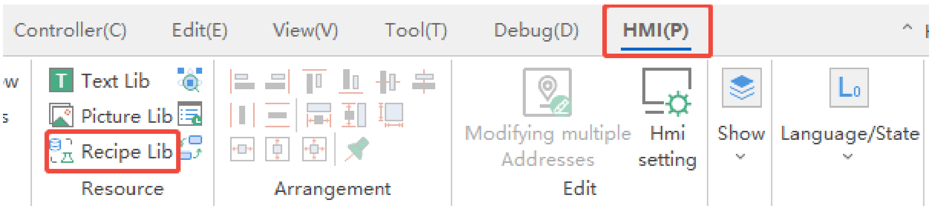
Clicking the "Add" button to add a new column for recipe ingredients. You can enter the column's title, data type, data length, editability, number of decimal places, and alignment.

➢ **Delete**

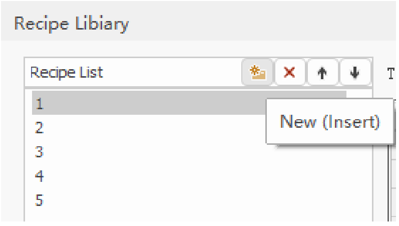
Select a row or column header and click the delete button to delete the ingredients in that row (the first row cannot be deleted).

(2) How to Use 1 (text library is not used)

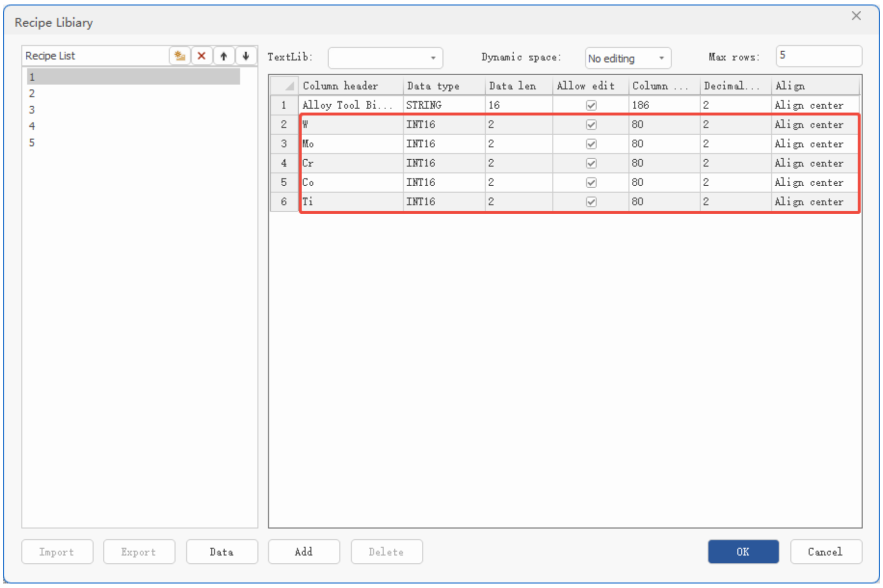
Step 1: click menu "HMI – Resource – Recipe Lib"



Step 2: click "New" to add the recipe, and give one name.



Step 3: click "add" to add recipes, and you could double-click to modify each row recipe name.



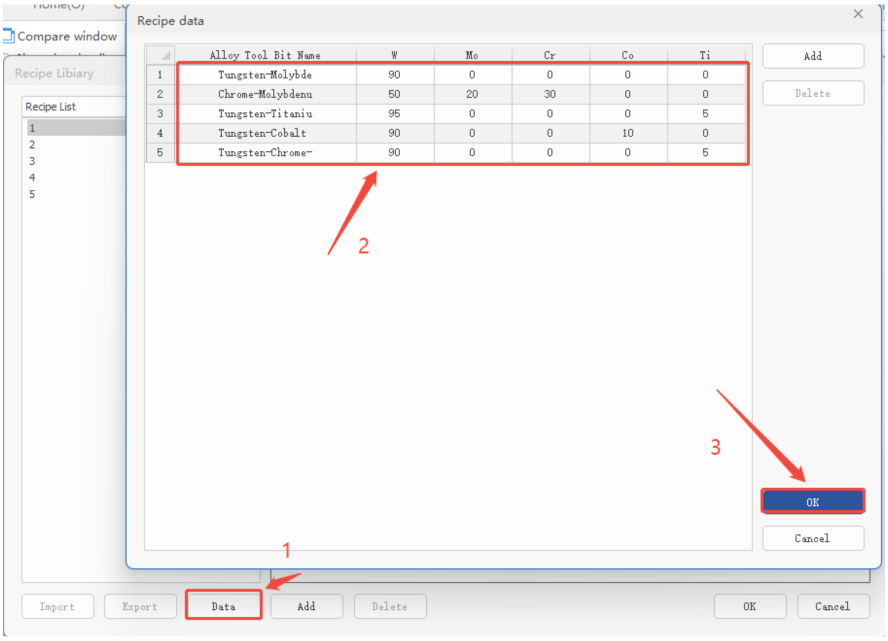
TextLib:

Dynamic space: No editing

Max rows: 5

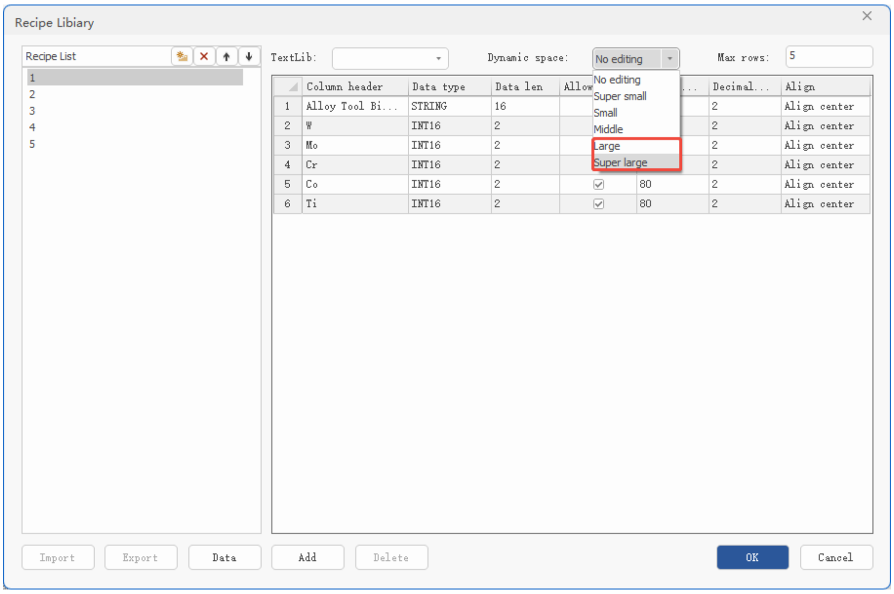
	Column header	Data type	Data len	Allow edit	Column ...	Decimal...	Align
1	Alloy Tool Bi...	STRING	16	☑	186	2	Align center
2	W	INT16	2	☑	80	2	Align center
3	Mo	INT16	2	☑	80	2	Align center
4	Cr	INT16	2	☑	80	2	Align center
5	Co	INT16	2	☑	80	2	Align center
6	Ti	INT16	2	☑	80	2	Align center

Step 4: click on "data" to bring up the recipe data window. Each row represents an alloy cutter head recipe. Name each row and set the proportion of raw materials for each recipe. Click "OK" to close the recipe data window, then click "OK" to save.



	Alloy Tool Bit Name	W	Mo	Cr	Co	Ti
1	Tungsten-Molybde	90	0	0	0	0
2	Chrome-Molybdenu	50	20	30	0	0
3	Tungsten-Titaniu	95	0	0	0	5
4	Tungsten-Cobalt	90	0	0	10	0
5	Tungsten-Chrome-	90	0	0	0	5

Note: if you need to insert or delete rows when using the recipe view, you should set the dynamic space to be larger.



Recipe Libiary

Recipe List: 1, 2, 3, 4, 5

TextLib:

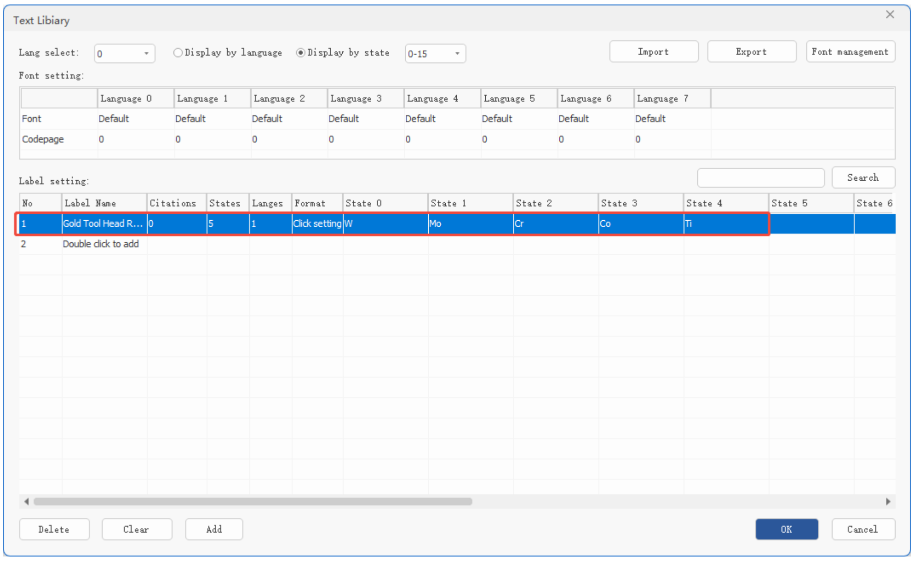
Dynamic space: No editing

Max rows: 5

	Column header	Data type	Data len	Allow	...	Decimal...	Align
1	Alloy Tool Bi...	STRING	16			2	Align center
2	W	INT16	2			2	Align center
3	Mo	INT16	2			2	Align center
4	Cr	INT16	2			2	Align center
5	Co	INT16	2	☑	80	2	Align center
6	Ti	INT16	2	☑	80	2	Align center

(3) How to Use 2 (text library is used)

Step 1: click menu "HMI – Resource – Text Lib", and set the label name, state numbers, and set each state's text, at last, click OK.



Text Libiary

Lang select: 0 ○ Display by language ● Display by state 0-15

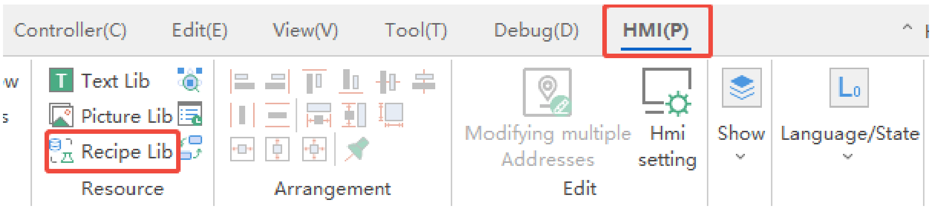
Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

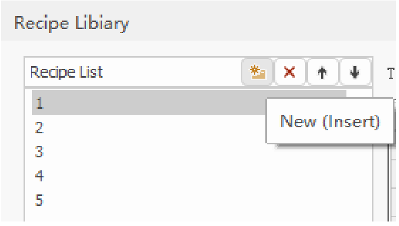
Label setting:

No	Label Name	Citations	States	Langes	Format	State 0	State 1	State 2	State 3	State 4	State 5	State 6
1	Gold Tool Head R...	0	5	1	Click setting	W	Mo	Cr	Co	Ti		
2	Double click to add											

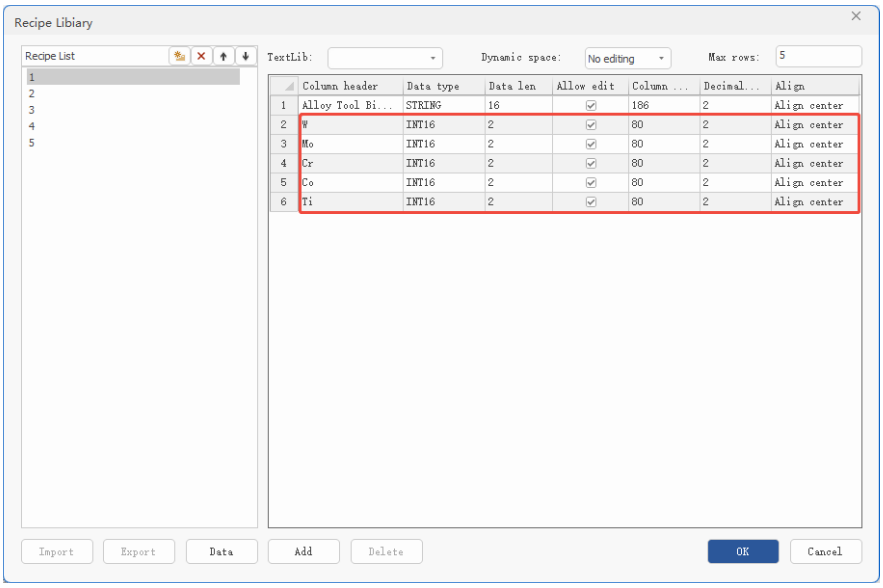
Step 2: click menu "HMI – Resource – Recipe Lib"



Step 3: click “New” to add the recipe, and give one name.



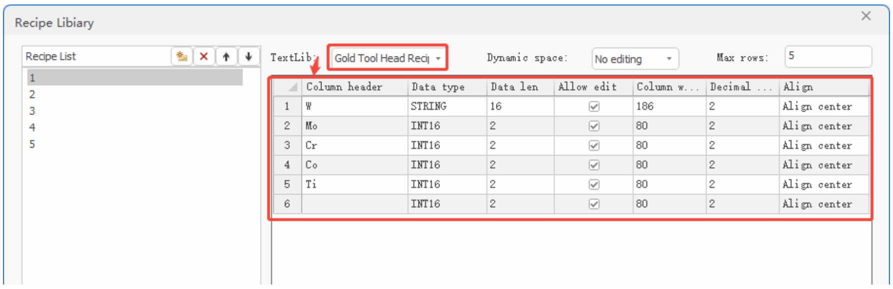
Step 4: click “add” to add recipes, and you could double-click to modify each row recipe name.



TextLib:
Dynamic space: No editing
Max rows: 5

	Column header	Data type	Data len	Allow edit	Column ...	Decimal...	Align
1	Alloy Tool Bi...	STRING	16	☑	186	2	Align center
2	W	INT16	2	☑	80	2	Align center
3	Mo	INT16	2	☑	80	2	Align center
4	Cr	INT16	2	☑	80	2	Align center
5	Co	INT16	2	☑	80	2	Align center
6	Ti	INT16	2	☑	80	2	Align center

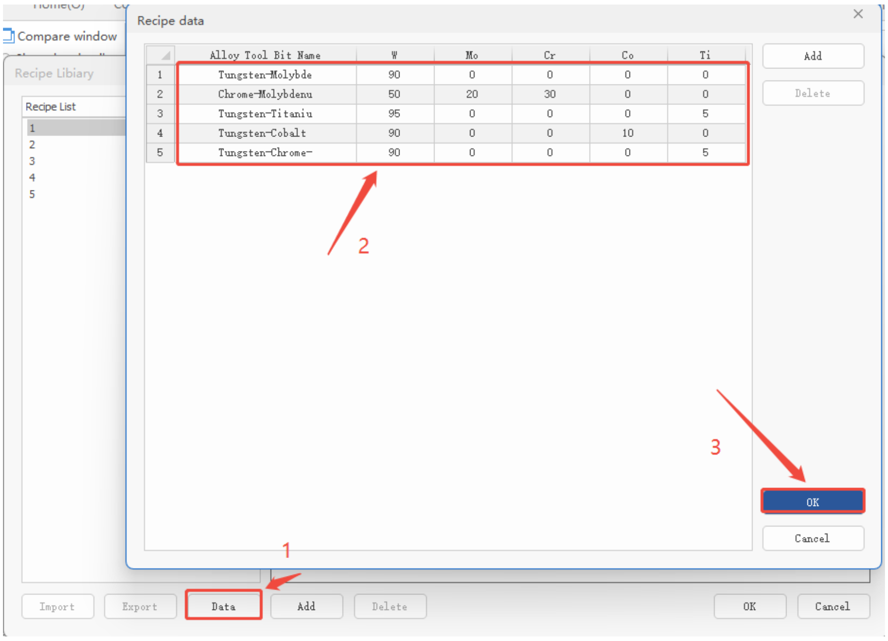
Step 5: after selecting the corresponding text library, each column title corresponds to one state of the text library.



TextLib: Gold Tool Head Reci
Dynamic space: No editing
Max rows: 5

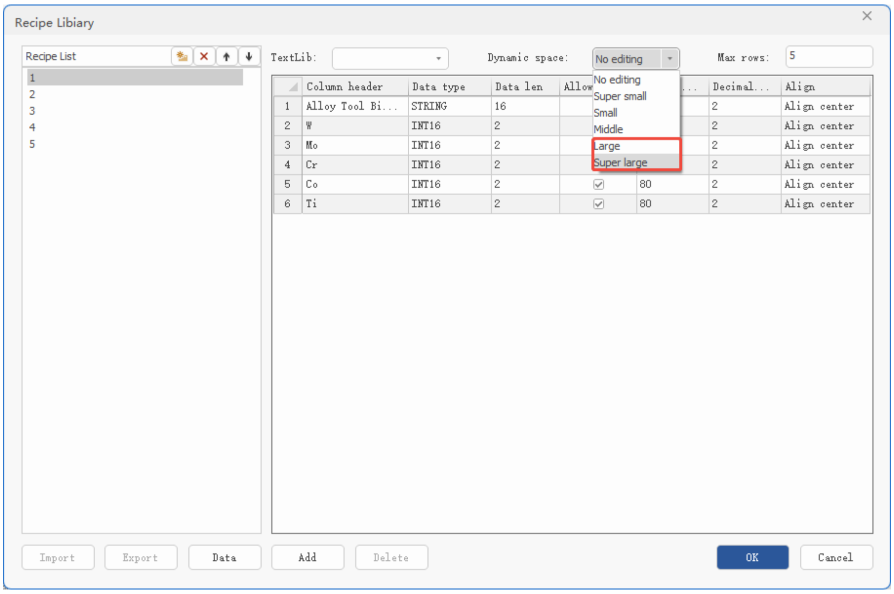
	Column header	Data type	Data len	Allow edit	Column w...	Decimal ...	Align
1	W	STRING	16	☑	186	2	Align center
2	Mo	INT16	2	☑	80	2	Align center
3	Cr	INT16	2	☑	80	2	Align center
4	Co	INT16	2	☑	80	2	Align center
5	Ti	INT16	2	☑	80	2	Align center
6		INT16	2	☑	80	2	Align center

Step 5: click on “data” to bring up the recipe data window. Each row represents an alloy cutter head recipe. Name each row and set the proportion of raw materials for each recipe. Click "OK" to close the recipe data window, then click "OK" to save.



	Alloy Tool Bit Name	W	Mo	Cr	Co	Ti
1	Tungsten-Molybde	90	0	0	0	0
2	Chrome-Molybdenu	50	20	30	0	0
3	Tungsten-Titaniu	95	0	0	0	5
4	Tungsten-Cobalt	90	0	0	10	0
5	Tungsten-Chrome-	90	0	0	0	5

Note: if you need to insert or delete rows when using the recipe view, you should set the dynamic space to be larger.



TextLib:

Dynamic space: No editing

Max rows: 5

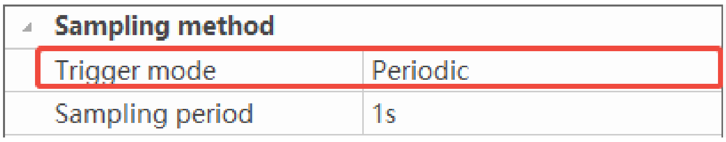
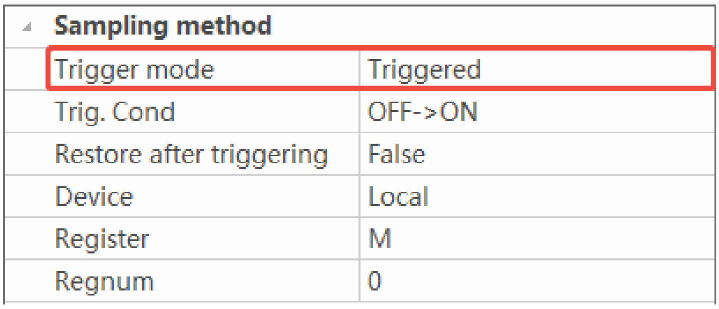
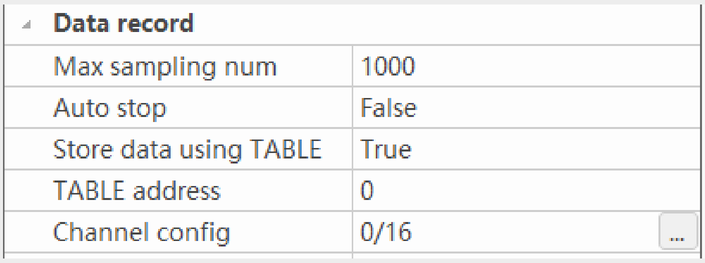
	Column header	Data type	Data len	Allow	...	Decimal...	Align
1	Alloy Tool Bi...	STRING	16			2	Align center
2	W	INT16	2			2	Align center
3	Mo	INT16	2			2	Align center
4	Cr	INT16	2			2	Align center
5	Co	INT16	2	☑	80	2	Align center
6	Ti	INT16	2	☑	80	2	Align center

“For more, please refer to [recipe view usage]”.

2.2.4. Sample Lib

It is used together with HMI “trend chart” tool, specifically, define the trend chart’s properties through resource sample library, like, sampling, and do channel configurations, then, the system will capture according to this sampling method. Sampling methods include periodic, and triggered ways. What's more, it can add, delete, copy, insert, replace, and clear resource sampling items.

- **Properties:**

<table>
<tr><th colspan="2">Property</th><th>Description</th></tr>
<tr><td colspan="3">Sampling Method</td></tr>
<tr><td rowspan="2">Trigger Mode</td><td>Periodic</td><td>Sampling once after spacing one certain time.

Sampling method
Trigger mode: Periodic
Sampling period: 1s</td></tr>
<tr><td>Triggered</td><td>Sampling when the condition is triggered.

Sampling method
Trigger mode: Triggered
Trig. Cond: OFF->ON
Restore after triggering: False
Device: Local
Register: M
Regnum: 0</td></tr>
<tr><td colspan="3"><b>Periodic</b></td></tr>
<tr><td colspan="2">Sampling Period</td><td>The spacing time</td></tr>
<tr><td colspan="3"><b>Triggered</b></td></tr>
<tr><td colspan="2">Trig. Cond</td><td>The trigger condition.</td></tr>
<tr><td colspan="2">Restore after Triggering</td><td>When the condition is triggered, it takes effect, after that, make the trigger register to be invalid (except register X).</td></tr>
<tr><td colspan="2">Device</td><td>Specify which device to be bound, the default is Local.</td></tr>
<tr><td colspan="2">Register</td><td>Select the register type, for type details, please refer to "Register" chapter.</td></tr>
<tr><td colspan="2">Regnum</td><td>The trigger register No.</td></tr>
<tr><td colspan="3">Data Record

Data record
Max sampling num: 1000
Auto stop: False
Store data using TABLE: True
TABLE address: 0
Channel config: 0/16 ...</td></tr>
</table>

Max Sampling Num	Set the max recorded sampling number, 1~40000, the default number is 1000. Note: bigger value will occupy bigger program space.
Auto Stop	Whether to make it auto-stop when sampling is full? ➢ TRUE: yes, auto-stop. ➢ FALSE: it will not stop, keep sampling (continuous sampling), and cover former data.
Store Data Using Table	Whether to save sampling points into TABLE array? ➢ TRUE: save into TABLE ➢ FALSE: save into system memory
Table Address	When you set "TRUE", please set TABLE starting address, the default is 0. *Store Format:* channel 0's 1st sampling data, channel 1's 2nd sampling data…, channel 0's N sampling data, channel 1's N sampling data.
Channel Config	Please see below 2.2.3.1 Channel Config.
And for specific usages, please refer to Chapter VIII Trend Charts Usage.	

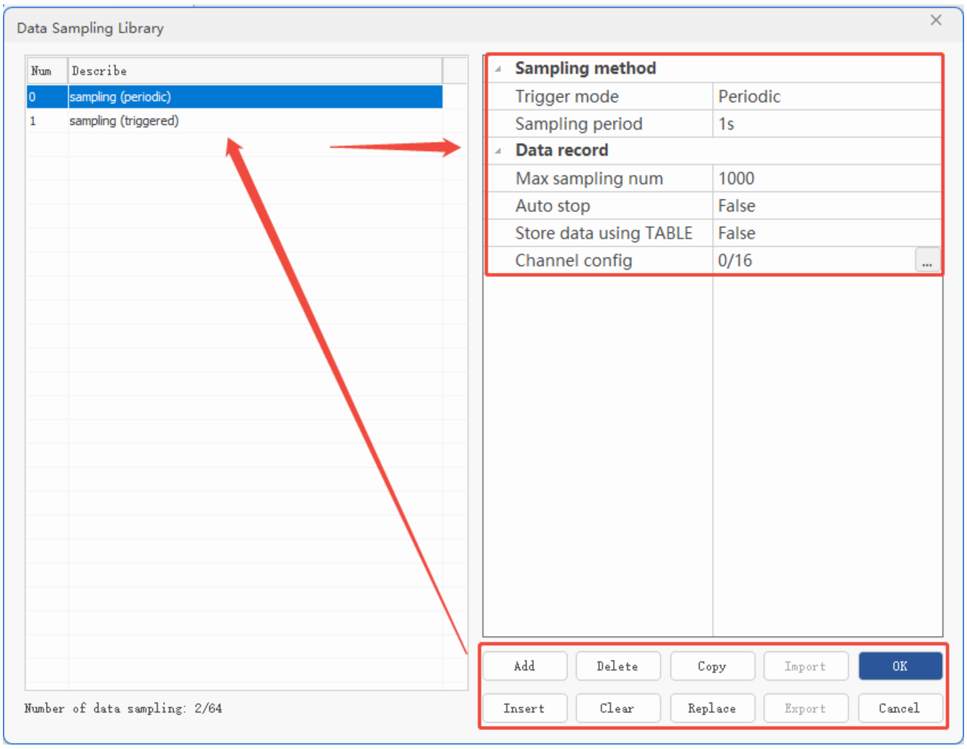
- **Basic Usage:**

1) Open "resource sampling lib" (sample lib), and click "add" to add needed sampling.

2) Double-click the content (describe) of the new added sampling, and enter information.

3) Then single-click to choose one, set its properties, like, right shown, sampling mode, sampling period, max sampling numbers, etc.

4) After that, please remember to click OK.

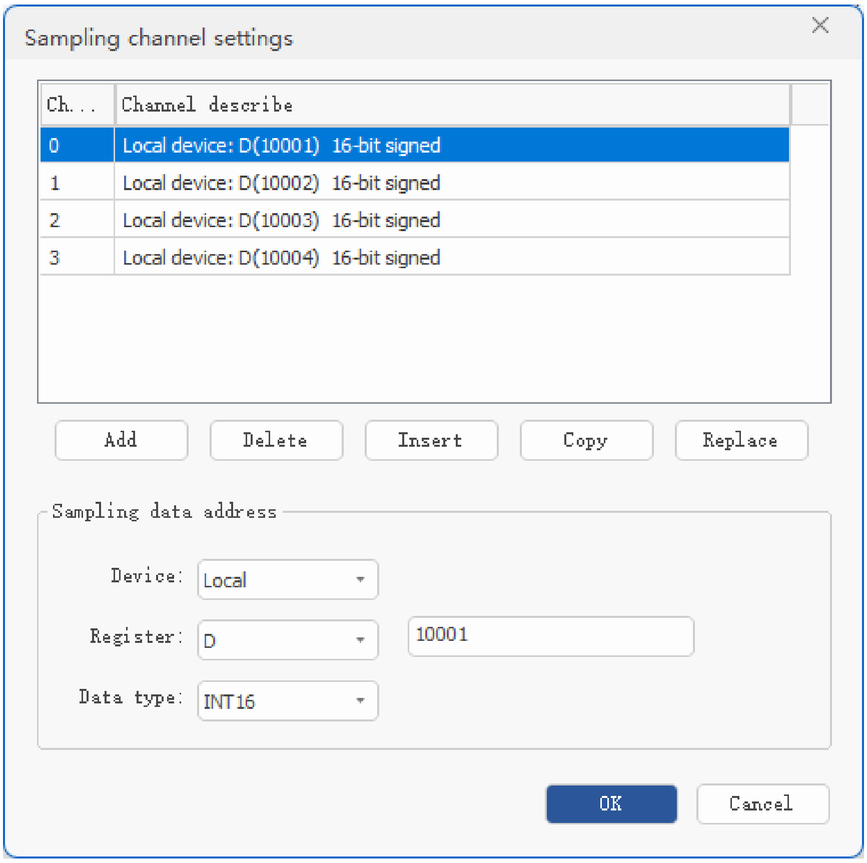

Data Sampling Library

Num	Describe
0	sampling (periodic)
1	sampling (triggered)

Sampling method	
Trigger mode	Periodic
Sampling period	1s
Data record	
Max sampling num	1000
Auto stop	False
Store data using TABLE	False
Channel config	0/16

Number of data sampling: 2/64

2.2.4.1. Channel Config


Sampling channel settings

Ch...	Channel describe
0	Local device: D(10001) 16-bit signed
1	Local device: D(10002) 16-bit signed
2	Local device: D(10003) 16-bit signed
3	Local device: D(10004) 16-bit signed

Sampling data address

Device: Local

Register: D 10001

Data type: INT16

- **Properties:**

Property	Description
Device	Specify which device to be bound, the default is Local.
Register	1) Select the register type, for type details, please refer to "Register" chapter. 2) Set the starting No.
Data Type	Select the sampling data type.

- **Basic Usage:**

1) Open "resource sampling lib" (sample lib), find "channel config", and right click to open "sampling channel settings" window.

2) Click "add" to build one new sampling channel (note: channel must be set for resource sampling library).

3) Click added channel item, then you can configure the channel property, like, register device No., register, register No., register data type. Please remember to click OK.

4) Click "delete" to delete current sampling channel.

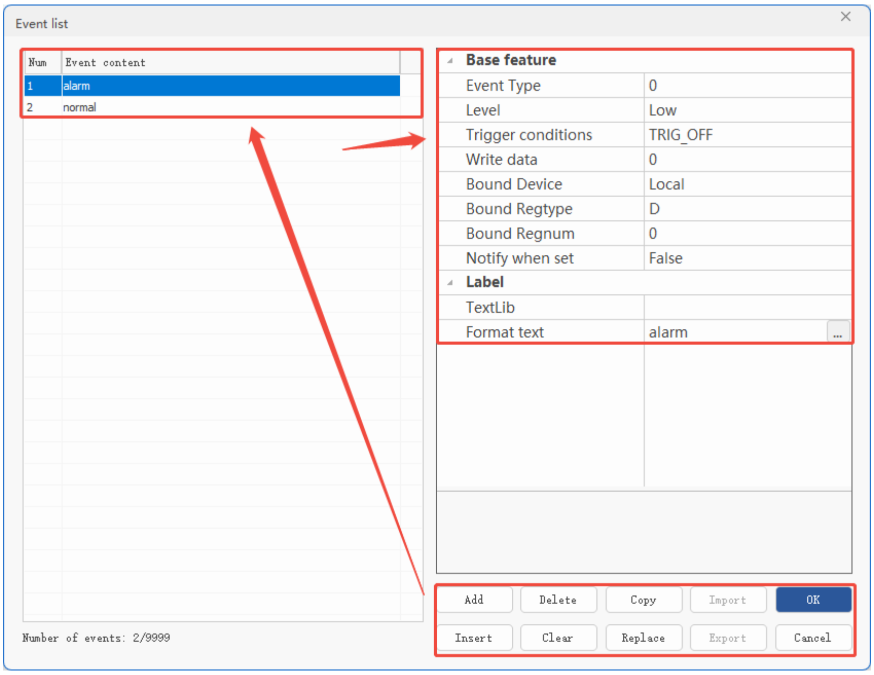
5) Click "insert" to insert one new sampling channel at selected item, the current select one will be moved behind. If no selected item, it is same as "add".

6) Click "copy" to copy current selected sampling channel.

7) Click "replace" to replace the previously copied sampling channel with the currently selected one.

2.2.5. Event List

"Event List" is used to define trigger conditions of event and the contents. Then, the system will trigger the event when conditions are met, showing contents in "info display components" of "event show", "alarm show", "alarm bar".



Num	Event content
1	alarm
2	normal

Number of events: 2/9999

Base feature	
Event Type	0
Level	Low
Trigger conditions	TRIG_OFF
Write data	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Notify when set	False
Label	
TextLib	
Format text	alarm

- **Properties:**

Property	Description
Event Type	Set the event type No.
Level	Set the event level: low, normal, high, critical.
Trigger Conditions	Select trigger conditions.
Write Data	Set the register value.
Bound Device	Specify which device to be bound, the default is Local.
Bound Regtype	Select the register type, for type details, please refer to “Register” chapter.
Bound Regnum	Select bound register No.
Notify when Set	Notify after modification, set it as ON / OFF.
Textlib	Whether to use text lib?
Format Text	Set the text format

- **Basic Usage:**

1) Open “event list”, and click “add” to add an event or events.

2) Double-click the event content, enter information.

3) Then single-click to choose one, set its properties, like, right shown, event type, level, trigger conditions, etc.

4) After that, please remember to click OK.

2.2.6. Key Transformation

Used to bind functions of physical button and virtual button, that is, when you operated the physical button, then virtual key also can work. This tool has already preset ZHD300X and ZHD400X buttons. Also, you can import and export set key values.

--How to Use--

A. Use Preset ZHD300X & ZHD400X Key Function

Step 1: open the Hmi KeyTrans in RTSys.

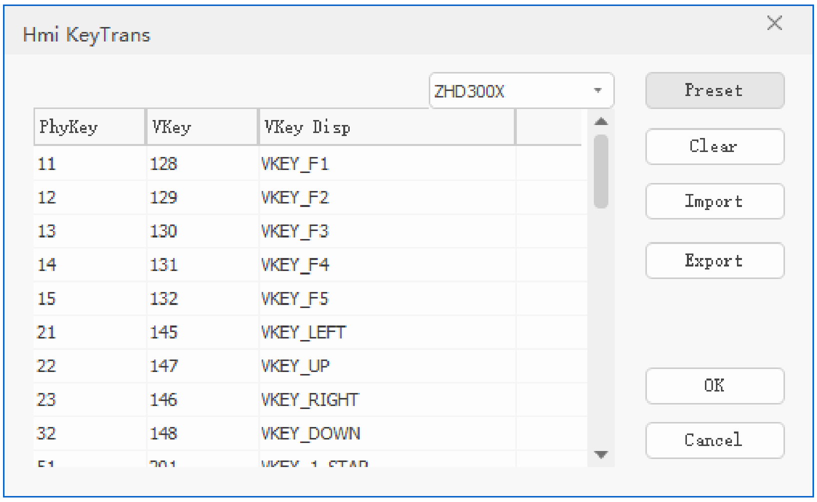
Step 2: in right upper, there is one menu you can select ZHD300X or ZHD400X.

Step 3: click preset, then corresponding key functions are shown, you can select needed, click OK. (if you want to change preset information, please click CLEAR, if you just modify someone, you can double-click it to achieve)

B. Use Customized Key Function

Step 1: open the Hmi KeyTrans in RTSys.

Step 2: double-click empty place, enter physical value and virtual value, click OK to save.



ZHD300X

PhyKey	VKey	VKey Disp
11	128	VKEY_F1
12	129	VKEY_F2
13	130	VKEY_F3
14	131	VKEY_F4
15	132	VKEY_F5
21	145	VKEY_LEFT
22	147	VKEY_UP
23	146	VKEY_RIGHT
32	148	VKEY_DOWN
51	201	VKEY_1_STAR

2.2.6.1.Physical Keys

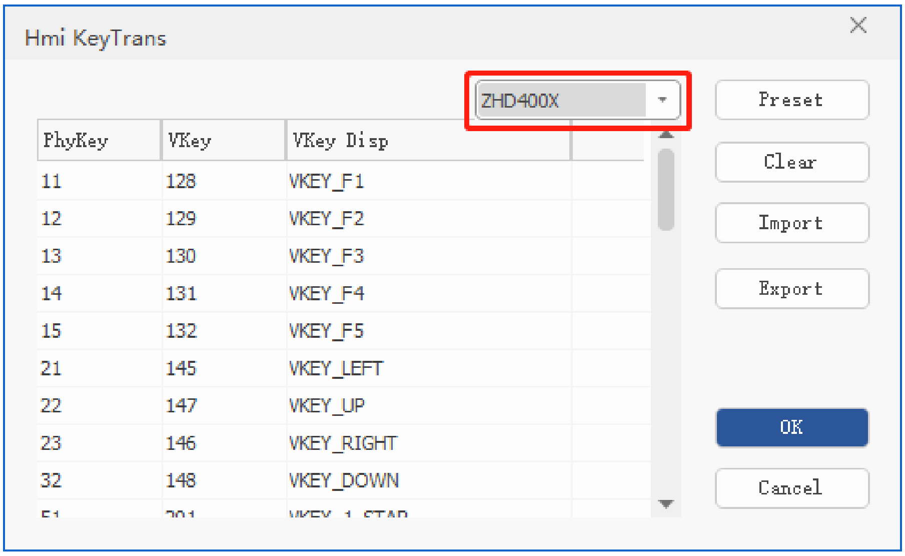
Physical keys refer to actual keys on external devices. Each key has a unique code value. When pressed, a message is sent. This message is the code value of the key.

The code value of a physical key is determined by the hardware and cannot be modified in the program. When using different external devices, the code value of the corresponding key is also different.

- **ZHD400X Standard Physical Key Codes:**

```
Global Const key_f1 = 11 'functional key 1
Global Const key_f2 = 12 'functional key 2
Global Const key_f3 = 13 'functional key 3
Global Const key_f4 = 14 'functional key 4
Global Const key_f5 = 15 'functional key 5
Global Const key_f6 = 16 'functional key 6
Global Const key_X- = 24 'axis movement key
Global Const key_X+ = 25
Global Const key_Y- = 34
Global Const key_Y+ = 35
Global Const key_Z- = 44
Global Const key_Z+ = 45
Global Const key_U- =54
Global Const key_U+ = 55
Global Const key_A- =64
Global Const key_A+ = 65
Global Const key_B- =74
Global Const key_B+ = 75
```

"in RTSys – Hmi KeyTrans"



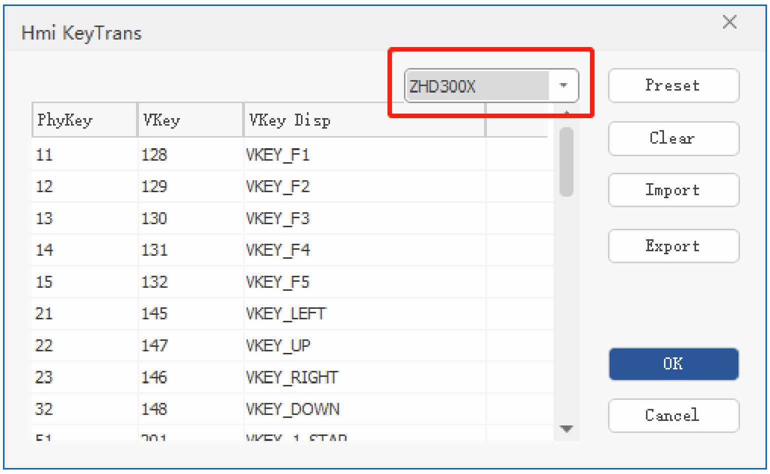
PhyKey	VKey	VKey Disp
11	128	VKEY_F1
12	129	VKEY_F2
13	130	VKEY_F3
14	131	VKEY_F4
15	132	VKEY_F5
21	145	VKEY_LEFT
22	147	VKEY_UP
23	146	VKEY_RIGHT
32	148	VKEY_DOWN
51	201	VKEY_1_STAR

The encoding of the ZHD300X physical keys is composed of rows and columns, and the key value = row No. (1-10) × 10 + column No. (1-5).

- **ZHD300X Standard Physical Key Codes:**

```
Global Const key_f1 = 11 'functional key 1
Global Const key_f2 = 12 'functional key 2
Global Const key_f3 = 13 'functional key 3
Global Const key_f4 = 14 'functional key 4
Global Const key_f5 = 15 'functional key 5
Global Const key_1 = 51 'digit key 1, at the same time, alphabet key switch
Global Const key_2 = 52
Global Const key_3 = 53
Global Const key_4 = 61
Global Const key_5 = 62
Global Const key_6 = 63
Global Const key_7 = 71
Global Const key_8 = 72
Global Const key_9 = 73
Global Const key_0 = 81 'digit key 0
Global Const key_Add= 83 '+
Global Const key_Point=82 '. (decimal part)
Global Const key_xUp=25 'JOG first axis
Global Const key_yUp=35 'second axis
Global Const key_zUp=45 'third axis
Global Const key_rUp=55 'fourth axis
Global Const key_xDown =24 'JOG first axis
Global Const key_yDown =34
Global Const key_zDown =44
Global Const key_rDown =54
Global Const key_Jog5L=64
Global Const key_Jog5R=65
Global Const key_Jog6L=74
Global Const key_Jog7R=75
Global Const key_Left=21 'move left
Global Const key_Up=22
Global Const key_Right=23
Global Const key_Down=32
Global Const key_SpeedUp=41
Global Const key_SpeedDown=43
Global Const key_Step=84
Global Const key_Manual=85
Global Const key_Reset =91 'reset
Global Const key_Del =92    'delete
Global Const key_Inset =93 'insert
Global Const key_Switch=94 'SHIFT switch
Global Const key_Save =95 'save
Global Const key_Esc =101 'cancel
Global Const key_Edit =102 'edit & watch
Global Const key_File =103 'file management
Global Const key_Set =104 'set parameter
Global Const key_Ent =105 'confirm IN
```

"in RTSys – Hmi KeyTrans"



PhyKey	VKey	VKey Disp
11	128	VKEY_F1
12	129	VKEY_F2
13	130	VKEY_F3
14	131	VKEY_F4
15	132	VKEY_F5
21	145	VKEY_LEFT
22	147	VKEY_UP
23	146	VKEY_RIGHT
32	148	VKEY_DOWN
51	201	VKEY_1_STAR

2.2.6.2.Virtual Keys

In actual programming, if the program is written using physical key coding, the portability of the program is very low. Therefore, when writing a program, except there is a code that can be used on all peripherals, so virtual coding appears. As long as the physical key code of the peripheral corresponds to the virtual code one by one, the program can be used on different peripherals.

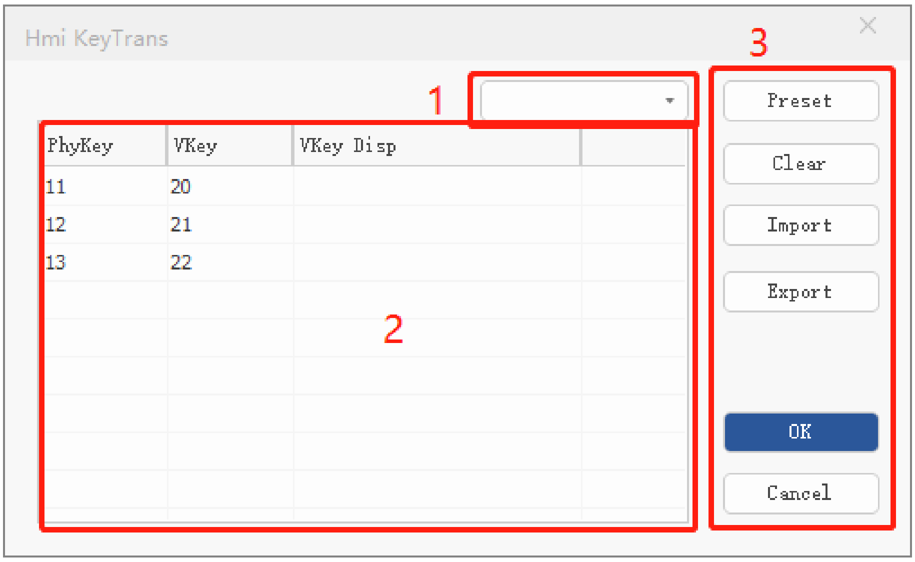
Since the operation method of virtual coding is similar to that of physical key coding, it is called **virtual key**.

In Hmi, the virtual key code is encapsulated by the bottom layer and cannot be modified in the program.

Virtual key code values 0-127 all correspond to the ASCII code table, and 128 and later support custom functions. Some virtual key values have defined functions. For details, please refer to **Appendix-Virtual Key Value Form.**

2.2.6.3.Edit “HMI KeyTrans”

You can open it through “HMI” menu – “Hmi KeyTrans”. It has 3 main parts: List & Functions & Menu Selection.



PhyKey	VKey	VKey Disp
11	20	
12	21	
13	22	

➢ **1. Menu Selection**

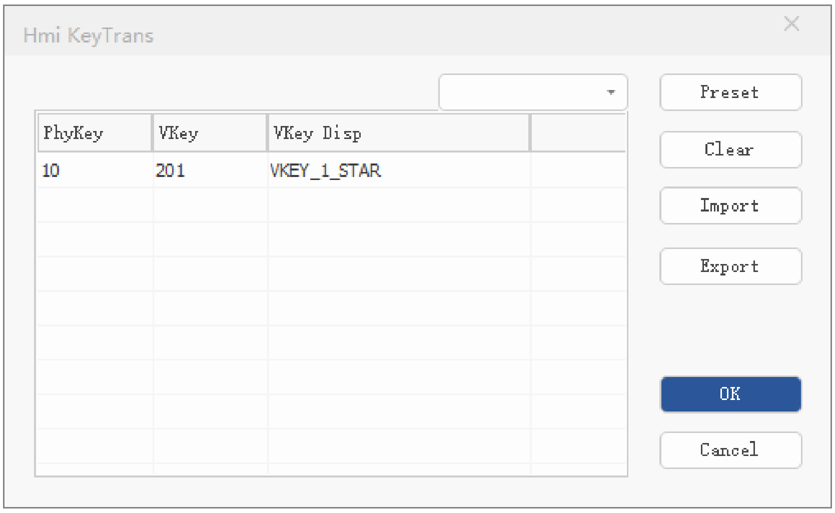
This tool has already set ZHD300X and ZHD400X conversion form, you can pull down the list to select.

➢ **2. List**

Used to edit and shown each physical key corresponding virtual axis value and function. Double-click the blank space to enter manually.

- *PhyKey:* set value is the encode value of external devices.
- *VKey:* set value is the virtual code value that binds with external physical key. 0-127 corresponds to ASCII code form, after 128 are customized functions, for details, please refer to **Appendix—Virtual Key Value Form.**
- *VKey Disp:* description of current virtual key function.

For Example: bind physical key 10 with virtual key 201:



PhyKey	VKey	VKey Disp
10	201	VKEY_1_STAR

- **3. Functions**

- *Preset:* used to take effect the information that you selected from "1 menu selection". Now, it only supports ZHD300X & ZHD400X.
- *Clear:* clear all current settings.
- *Import:* call edited conversion form from external file, usually the one that was exported and saved by "export" in this interface also, the file format is .ini.
- *Export:* export current edited conversion form as .ini file.
- *OK:* when edited well, click OK to apply it, otherwise, it will be empty when you open next time.
- *Cancel:* cancel the operation and exit.

2.2.6.4.Key Transformation Commands

Virtual key values and virtual key functions support custom binding, which can be implemented through corresponding instructions and program writing.

The basic instructions related to key conversion are mainly the following 6.

- KEY_STATE: physical key state
- KEY_EVENT: physical key state scan
- KEY_SCAN: read physical key code
- VKEY_STATE: virtual key state

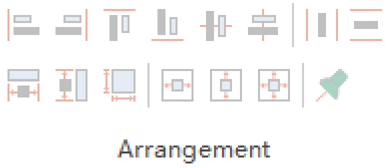
- VKEY_EVENT: virtual key state scan
- VKEY_SCAN: read virtual key code

In the program, you can use VKEY_SCAN to capture which virtual key is pressed. According to the key conversion table, you can know which physical key corresponds to it, you can also use KEY_SCAN directly to capture which physical key is pressed. However, it is generally not recommended to use KEY_SCAN and physical key related instructions, because the physical key codes of different peripherals are different, and the portability of the program is low. It is recommended to use VKEY_SCAN and other virtual key related instructions.

Since these commands can only be used in the refresh function of custom components (Hmi's "Init Sub" is also OK, but it is not recommended to do so), at least one custom component must exist. After scanning the key press, assign the return value to a custom variable. In the "Draw Sub" of the custom component, different functions are called according to the different return values to achieve different functions.

For reference examples, see "Conversion between Physical keys & Virtual Keys".

2.3. Arrangement

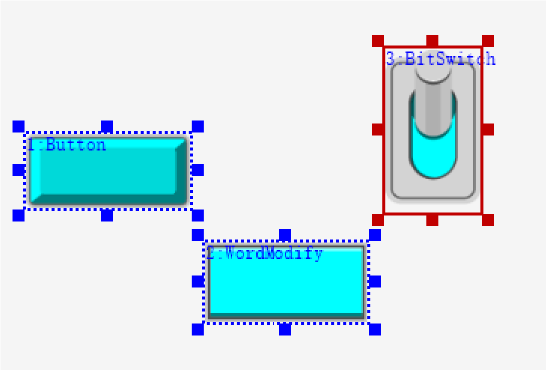


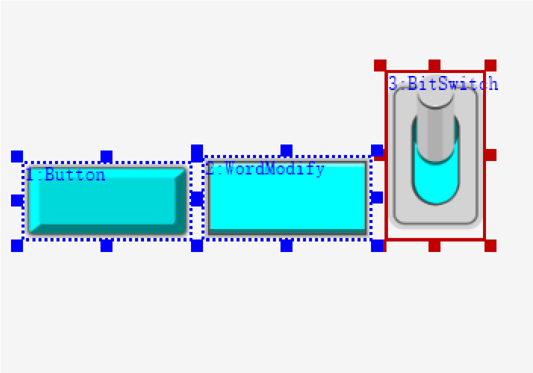
There are several alignment methods. You can select as you needed.

X: horizontal direction / width   Y: vertical direction / height

	Align Left		X center alignment		Same X
	Align Right		Y center alignment		Same Y
	Top		Same space in X		Same size in x & y
	Bottom		Same space in Y		Lock the component
	X Center		Y Center		X & Y Center

For example, make 3 components align in bottom:

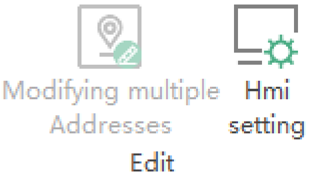




--Notes--

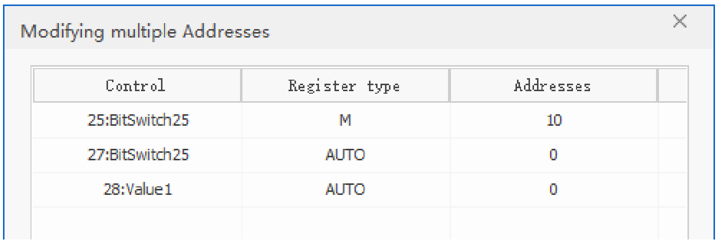
- It is used when you need to operate ≥ 2 components.
- The object that is selected will appear one red frame, and align it according to the object created sequence, not related to object No. sequence.
- If you need to custom the target object, please select the object at first, then press “Ctrl”, single click other following objects, when all selected, release the Ctrl button, then select alignment method.

2.4. Edit



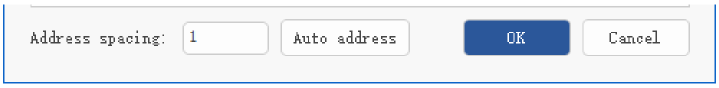
2.4.1. Modify Multiple Addresses

Through this, you can modify multiple HMI components’ register addresses at one time, for example, you can make them as the same register type, set the address space.



Modifying multiple Addresses

Control	Register type	Addresses
25:BitSwitch25	M	10
27:BitSwitch25	AUTO	0
28:Value1	AUTO	0


Address spacing: 1

- Control: when you chose several components, some objects among them support binding with register, then here will show corresponding object No. and name (only HMI object that supports binding with register).

- Register type: you can select which register to be bound, there are AUTO (automatic), X (IN), Y (OUT / OP), M (MODBUS_BIT), S (state register), D (MODBUS_REG / MODBUS_LONG / MODBUS_IEEE), D.DOT (read MODBUS_REG bit by bit), DT (TABLE), T (timer), C (counter). More details, please refer to “Register” Chapter.

- Address: set register starting address.

- Auto address: assign the address and type for the one that selects AITO type according to set starting address and address space.

--How to Operate--

1. In HMI window, select several components that set register (recommend to select components that can use same register type).

2. Open above window, then you can see related component name and register type, address.

3. Select “register type”, drop-down the list to choose.

4. Set starting address No. (AUTO: automatically follow with the last one register type).

5. If you need set address space, set by “address spacing”. Default is 1. After that, click “auto address”. Then, AUTO type will become needed one, and show according to the sequence of address. At last, click OK.

--Notes--

- The first one component can not be AUTO type.

- For the address, addresses can’t be same.

- Showing sequence is determined by adding sequence. Therefore, select them who use same register type.

- After “auto-address” each time, if you modify the type and “auto-address” again, please manually modify and set as AUTO type.

2.4.2. Hmi Setting

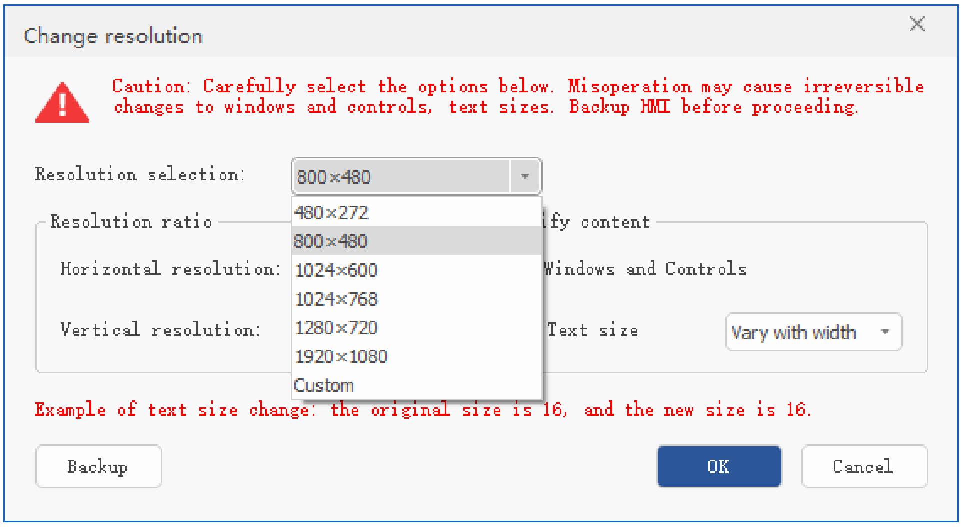
Perform initial property settings for the HMI system, and modify properties such as the HMI window resolution and the initial basic window. For details, see the table below. You can open the property window of the HMI system settings through the menu bar "HMI" → "Hmi Setting", or you also can click blank space outside the HMI screen to open it.

Note: when using HMI, usually set parameter properties in HMI Setting in advance.

Property	Function	Description
Lcd number	Set the LCD screen No.	While connecting to HMI, select which HMI file content is shown by this No.
Backlight time	The actual backlight time of the teaching box	/
Screen time	Set screen saver time	/
Startup base window	Set the HMI initial base type window	Windwo 10 is shown by default.
Starup top window	Set the HMI initial top window	/
Init Sub	Add HMI initial sub function	The sub fuinction is only called once after powered on, and it must be GLOBAL sub.
Period Sub	Add HMI period sub function	The sub fuinction is called in cyle after powered on, and it must be GLOBAL sub.
Window display protection	When it is TRUE, make the window’s component not display out of window range.	Default: False.
Window mouse protection	When it is TRUE, the mouse event is only valid in the top window at the mouse current position. And in each period, only one window can process the mouse event.	Default: False.
Compress pictures	Compress pictuers or not	Old Compress: low compress quality and can’t be resumed (that is, ZDevelop

		compress method). New Compress: compress and keep same quality No Compress: don’t compress
Picture quality	Select picture quality (when not compressed, the quality depends on if it supports “anti-aliasing zoom”	Standard: low image showing quality, but HMO performance is high. High: high image showing quality, but HMO performance is low.
Text adaptive sizing	Text adapts to component’s size.	When the text exceeds component range, font will be zoomed out automatically, the smallest font will not be lower 50% than you set one.
Format text without lib	True: use component format text False: use text lib format text	/
Format text	Set the text format of HMI notes.	
Screen Width	Window display r/esolution	Editable
Scren Height	Window display resolution	Editable

You can preset resolution / custom resolution when you editing it. Then, in “change resolution’, you can select whether “window and controls” and “text size” synchronize with new resolution.



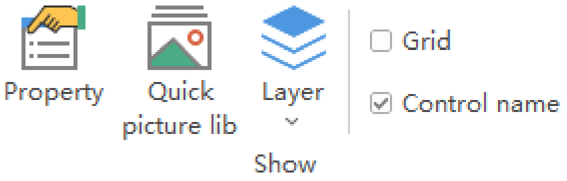
Change resolution

Caution: Carefully select the options below. Misoperation may cause irreversible changes to windows and controls, text sizes. Backup HMI before proceeding.

Resolution selection: 800×480

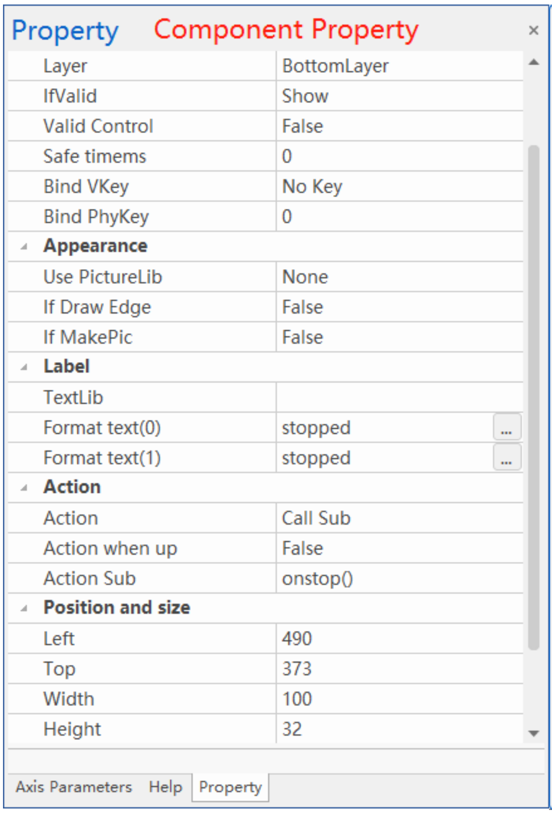
Example of text size change: the original size is 16, and the new size is 16.

2.5. Showing Settings

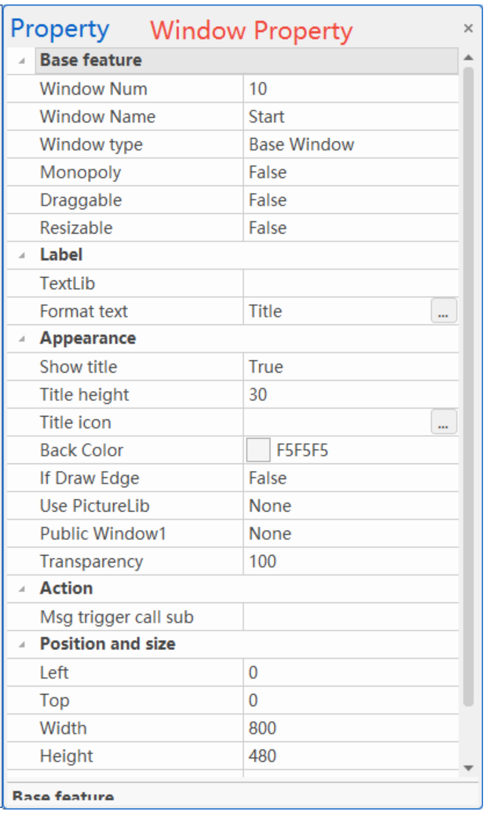


2.5.1. Property

Used to show and set properties of window / component in HMI file. Before, new build / open one HMI file, and then click “HMI” – “Property”.



Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bind VKey	No Key
Bind PhyKey	0
Appearance	
Use PictureLib	None
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	stopped
Format text(1)	stopped
Action	
Action	Call Sub
Action when up	False
Action Sub	onstop()
Position and size	
Left	490
Top	373
Width	100
Height	32



Base feature	
Window Num	10
Window Name	Start
Window type	Base Window
Monopoly	False
Draggable	False
Resizable	False
Label	
TextLib	
Format text	Title
Appearance	
Show title	True
Title height	30
Title icon	
Back Color	F5F5F5
If Draw Edge	False
Use PictureLib	None
Public Window1	None
Transparency	100
Action	
Msg trigger call sub	
Position and size	
Left	0
Top	0
Width	800
Height	480

Base feature

- **Component Property Introduction**

Property	Function	Description
Basic Feature		
Object ID	It can modify the No.	/
Object Name	It can modify the name.	/
Layer	Select component display layer	● TopLayer: the surface, it shows the most external layer, and covers

		below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Object is shown or not	● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register	Default is False. If TURE, below 3 parameters will be shown.
Valid Device	Device No.	Default is local
Valid regtype	Select register type	Select from the list
Valid regnum	Select register No.	It does not show when register value is 0, it is used when not 0.
Safe timems	The leaset button time	Unit is ms.
Bound Virtual Keys	Select virtual codes to be bound	Not used by default.
Bound Physical Keys	Select pysical codes of HMI (teach pendant) to be bound.	Button codes, please refer to "vitrual buttons".
Appearance		
Use Picture Lib	BackPicture / Back Picture	Select from picture library or backgroud
Back Picture Lib	Select background picture	After select background picture for picture source, then add
If Draw edge	Draw the edge?	/
If MakePic	Change object to image?	Default is False
Label		
Text Lib	The name of text library	If empty, indicates the use of a text label
Format text 0 / 1	Open the Format Text Settings window to set the text to be displayed by the	Default is text 0, press it, it will show information set in text 1.

	component	
Action		
Action	Motion to be executed when button	Please refer to “action”.
Action when up	Select execution action when press or release.	Default False: the action when pressed, True: the action when released
Action sub name	SUB function to be called after pressing	Select global SUB function of Basic
Position and size		
Top	Vertical starting position	Can’t exceed vertical resolution
Left	Horizontal starting position	Can’t exceed Horizonical resolution
Width	The width of element	/
Height	The height of element	/

- **Window Property Introduction**

Property	Function	Description
Basic Feature		
Window Num	Current window No.	Under one project, window No. can’t be same.
Window Name	Current window name	/
Window type	There are 5 window types	Refer to “window type”.
Monopoly	Monopoly or not?	After the monopoly, the components below the window cannot be operated
Draggable	Whether to allow the mouse to drag the title bar while holding it down?	This is valid when “Show Title is True”.
Resizable	Whether to allow mouse to be pressed and stretched at the four sides and four corners of the window?	This is valid when “Show Title is True”.
Appearance		
Show Title	whether to show the window title bar.	The title bar is used to show the window name and give one button that is to close this window. And the display area also belongs to

		the window range. If the window itself is 400*300, the title height is 35, then remain window display range is (400-35)*300.
Title Height	Set the window title bar's height	This is valid when "Show Title is True".
Title Icon	Set the title bar's icon	This is valid when "Show Title is True".
Back Color	Select window background color	/
If Draw edge	Draw the edge?	After TRUE, edge color can be selected.
Use PictureLib	BackPicture / Back Picture	You must add a picture before you can select it. The picture name cannot exceed 26 characters.
Public Window1	Set current window's public window 1	The current window can display the components of the public window. Up to 3 public windows can be set.
Transparency	Set window's transparency, but all components below the window will not become transparent along with the window.	0 – fully transparent. 100 – fully displayed.
Action		
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured triggering mode.	It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position and size		
Top	The X coordinate of the upper left corner of the window display	Don't exceed vertical resolution
Left	The Y coordinate of the upper left corner of the window display	Don't exceed Horizonical resolution
Width	Current window's showing width & height.	/
Height		/

2.5.2. Quick Picture Lib

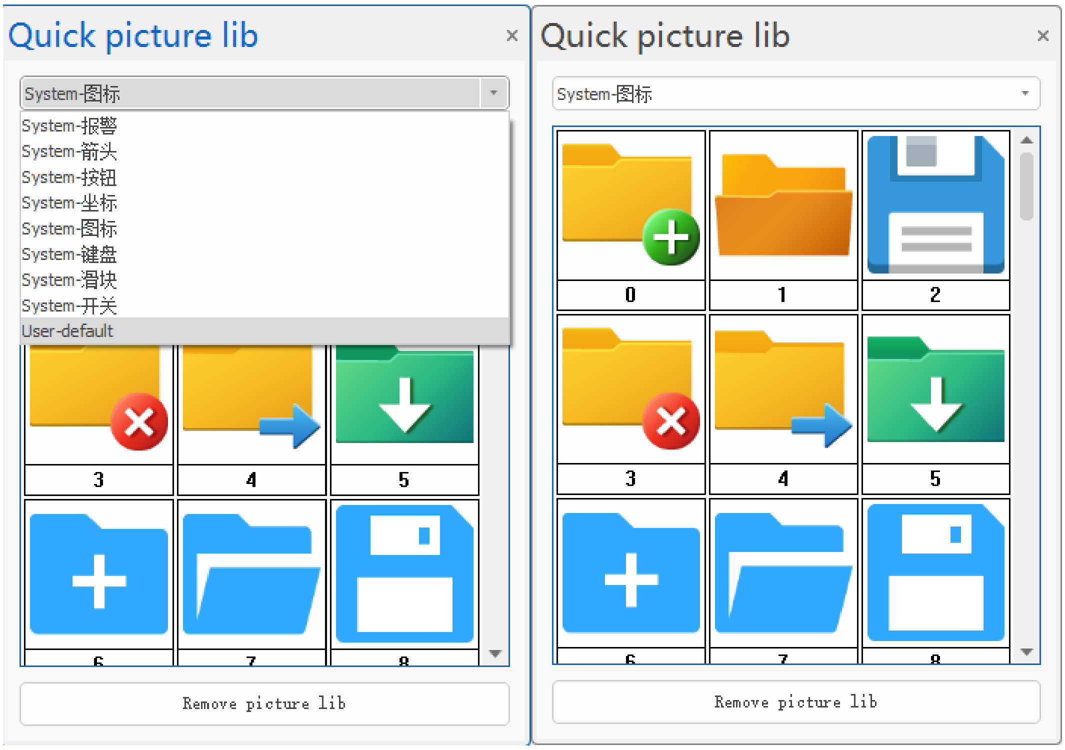
Used to quickly apply or remove styles from the picture library to HMI components. You can open this window through the menu bar "HMI" → "Quick Picture Library".

- **Add Picture Library**

Open HMI file, select one single component in HMI window, and open “quick picture lib”, find needed pictures, double-click it to use it (there are many classifications in “picture lib”, select from “drop-down” list).

- **Remove Picture Library**

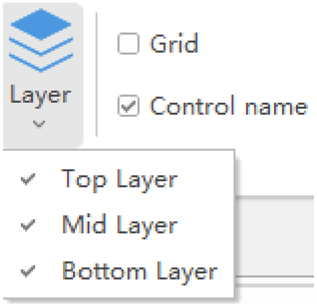
Choose the component, then remove it.



2.5.3. Show / Hide Layer

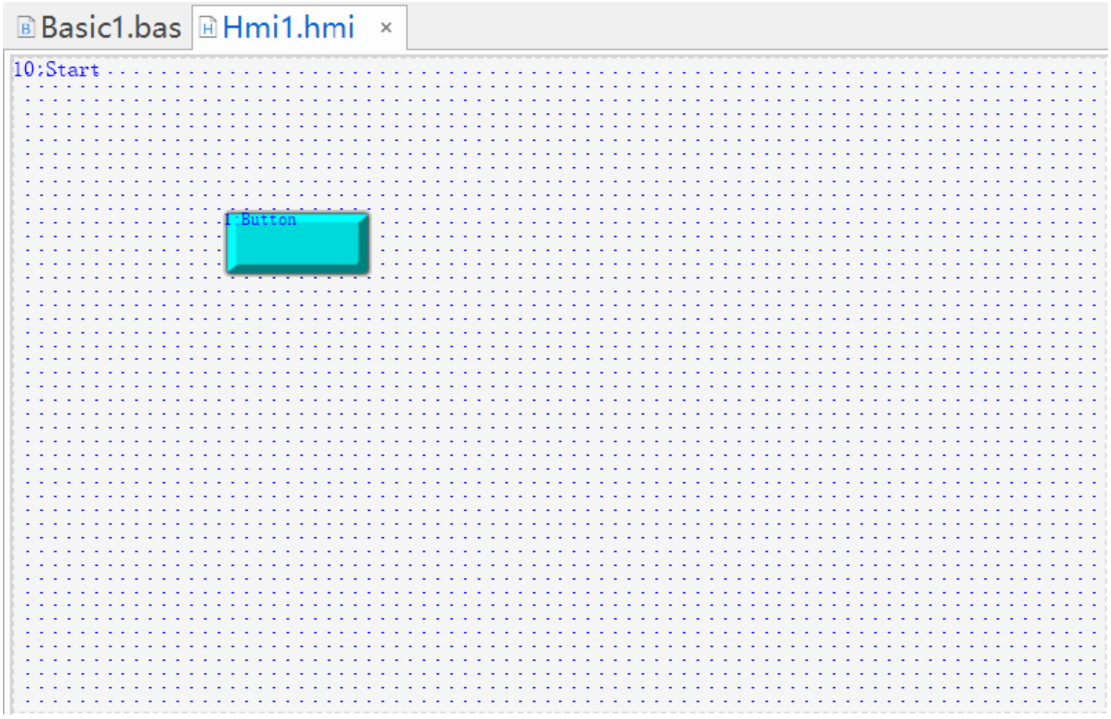
Show or hide components of different layers (HMI – Layer)

There are 3 layers operations, top, middle, bottom.

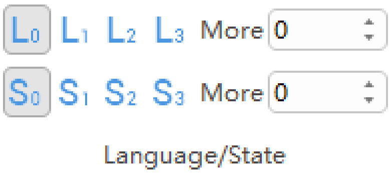


2.5.4. Grid & Control Name

Show or not (grid, component name, window name). Both are checked, as like:



2.6. Language / State



- **Switch Language**

Switch the language of the component that has called the text library. You need to set the content corresponding to each language in the current state in the text library first, call the text library in the component, then select the language to switch the text content, L0 corresponds to language 0, L1 corresponds to language 1, and so on. If there are more than that, you can enter the language number in More. A maximum of 8 languages can be set, that is, L7.

Note: you need to create a multilingual text library or image library in advance to see the effect.

- **Switch State**

To switch the state of a function key or bit state/multi-state component, select state S to switch to a different state. S0 corresponds to state 0, S1 corresponds to state 1, and so on. A maximum of 256 states can be set, that is, S_{255}. And when the state exceeds 256, it only shows max state 255 value.

Chapter III HMI Window

3.1. Window Introduction

3.1.1. Window Function

The window is the basic element of the touch screen interface and is also an important element. And with the window, the various components, graphics, text and other information on the screen can be displayed on the touch screen. In general project files, there are multiple windows, so one function needs to create multiple windows.

Since the size (resolution) of the basic window must be the same as the size of the touch screen display screen, its resolution setting must also be consistent with the resolution of the touch screen used. (Note: The default resolution of the new window of RTSys is 800*480.)

Even if the component HMI exceeds the window size, the component can also be triggered normally.

3.2. Window Operations

3.2.1. Window Property

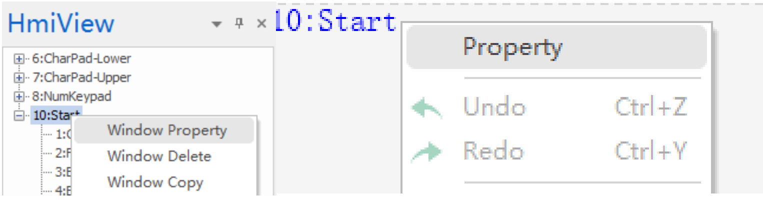
Window properties include the type, appearance, size, etc. of the window. You can set and modify the selected window through the "Properties" window;

➢ **How to open the properties window:**

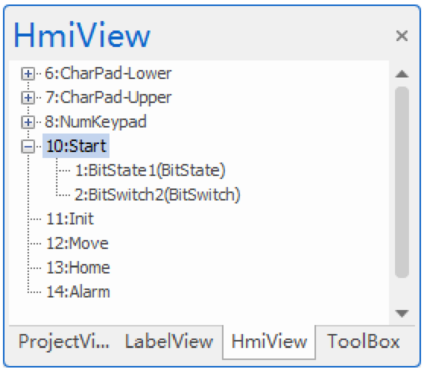
1. Open the window you want to view, click the window area with the left mouse button, then property window will appear in the right side.

2. Select the window to open, right-click, and select "Window Properties", as shown in the left figure.

3. In HMI editing window, right-click to open "Properties", as shown in the right figure.



In HmiView, you can see all windows, each window can be set in corresponding window property.



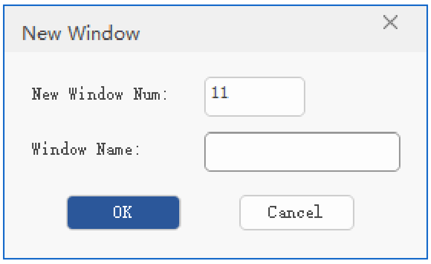
“For window property settings, please refer to 2.5.1 Window Property”

3.2.2. Create the Window (New Window)

HMI showing must use a basic window as the bottom window, which is used as the background screen of other windows, because components are shown through the window, and multiple windows of different types can be created under one HMI file. After the Hmi file is created, it comes with 3 soft keyboard windows and 1 base window (window 10). You can edit in existed window 10, also can create one new window.

- **Create New Window**

In menu HMI – New Window, you can open below window, then you can enter window No. and window name. Please remember to click OK and the Window Num can’t be the same.



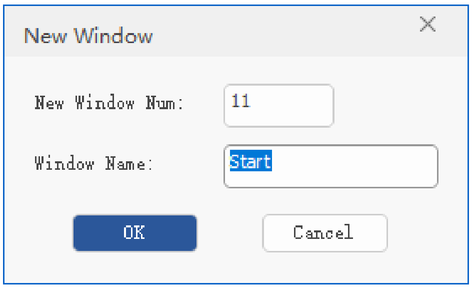
New Window Num: 11

Window Name:

- **Window Copy**

Copy already created window and corresponding information, and create one new window. That is, you can copy designed HMI interface directly and generate one another new window.

In HmiView, you select needed window, then right-click it, then will show “Window Copy”, click it, and modify the window No. and window name.



New Window Num: 11

Window Name: Start

After that, you can set related properties. You can open according to below two methods.

- **Method 1**

In HmiView, all windows and related components are shown. You select one window, then right-click and select “Window Property”. If you want to open one element property, still in HmiView, select one, then double click it.

- **Method 2**

Directly single click the window interface to open the window property. And directly click the object to open this component’s property.

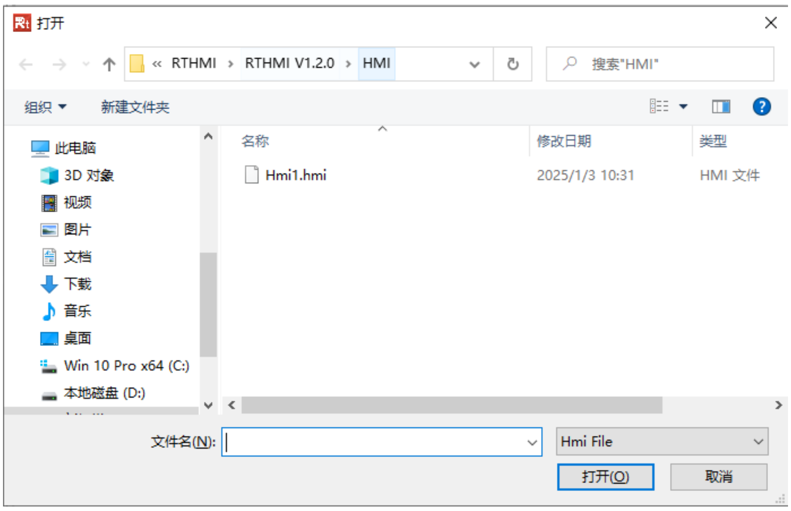
3.2.3. Import Window

Here, you can import one HMI window / several windows that was created by other projects.

➢ **How to Use**

(1) In menu HMI, click “Import Window”.

(2) In new popped window, select which one window (.Hmi file).



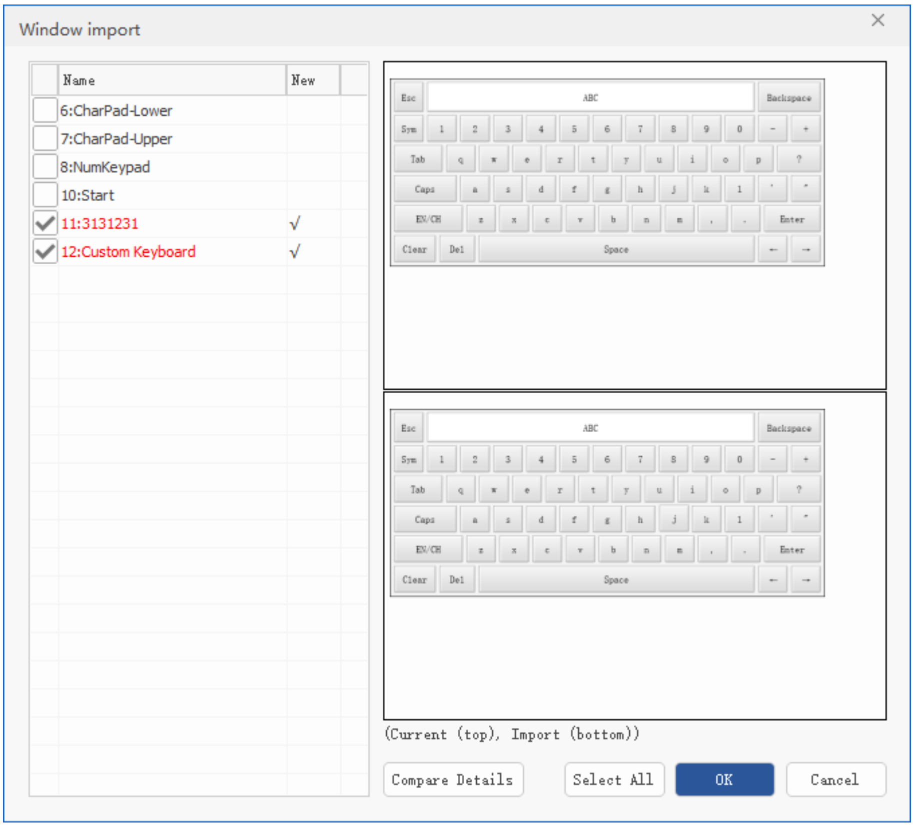
名称	修改日期	类型
Hmi1.hmi	2025/1/3 10:31	HMI 文件

文件名(N):

Hmi File

(3) Open needed HMI file, then “window import” window appears. In left side, it shows window list. In right side, it shows window comparison graphics.

Note: if the window ID and window name in the imported Hmi project are inconsistent, the window name will be displayed in red, and the window without the same window ID in the current project will be checked by default. You can choose whether to import / replace the original window. (Select “Yes” to replace, select “No” to import only the non-conflicting window.

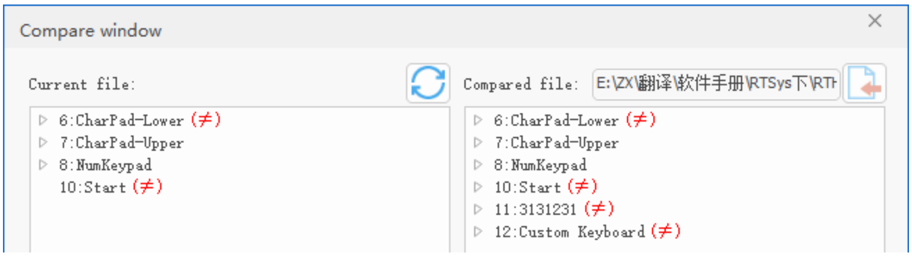


Window import

	Name	New
□	6:CharPad-Lower	
□	7:CharPad-Upper	
□	8:NumKeypad	
□	10:Start	
☑	11:3131231	√
☑	12:Custom Keyboard	√

(Current (top), Import (bottom))

After clicked “compare details”, new window of “compare window” will appear.



Compare window

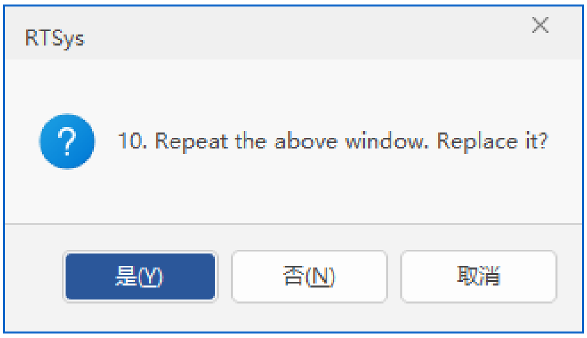
Current file:

▷ 6:CharPad-Lower (≠)
▷ 7:CharPad-Upper
▷ 8:NumKeypad
10:Start (≠)

Compared file: E:\ZX\翻译\软件手册\RTSys下\RTH

▷ 6:CharPad-Lower (≠)
▷ 7:CharPad-Upper
▷ 8:NumKeypad
▷ 10:Start (≠)
▷ 11:3131231 (≠)
▷ 12:Custom Keyboard (≠)

When all needed windows are checked, please click OK. If selected window No. is conflict with original file window No., below window will pop up, then YES to replace, NO to import non-conflict window.



10. Repeat the above window. Replace it?

3.2.4. Call Window

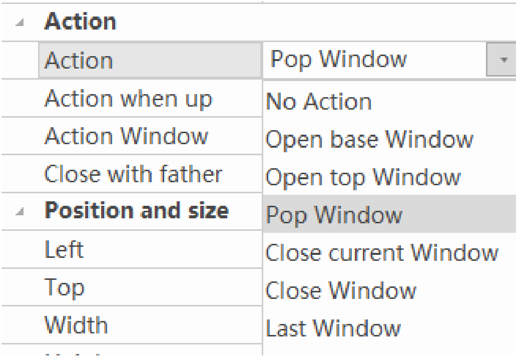
There are two methods to call the window.

■ **Method 1: Call Functional Key**

(1) Select “Control Class” – “Button”.

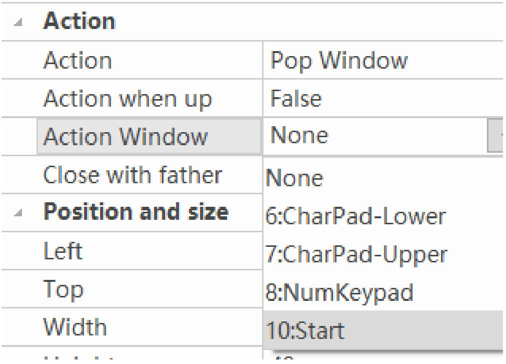
(2) Select the button, and single-click to open its property, then find “Action”, there are 3 window types. Here, for menu window, it belongs to “Pop Window”. And select it. -- **Left**

(3) After that, find “Action Window”, select needed window No. -- **Right**



Action

Action	Pop Window
Action when up	No Action
Action Window	Open base Window
Close with father	Open top Window
Position and size	Pop Window
Left	Close current Window
Top	Close Window
Width	Last Window



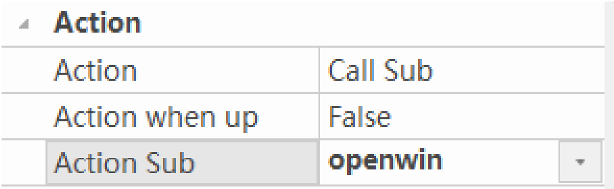
Action

Action	Pop Window
Action when up	False
Action Window	None
Close with father	None
Position and size	6:CharPad-Lower
Left	7:CharPad-Upper
Top	8:NumKeypad
Width	10:Start

Note: if the window type to be opened is inconsistent with the selected window type, the window type will be forcibly converted.

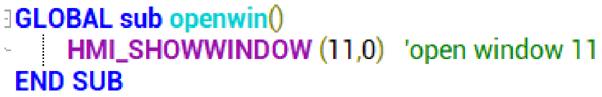
■ **Method 2: Call by Program Command**

In object property “Action”, select “Call Sub”. Then, called function can be achieved in BASIC programming, main commands are HMI_SHOWWINDOW and HMI_BASEWINDOW.



Action	
Action	Call Sub
Action when up	False
Action Sub	openwin

--Corresponding Basic Codes--



```
GLOBAL sub openwin()
    HMI_SHOWWINDOW (11,0)  'open window 11
END SUB
```

3.2.5. Close Window

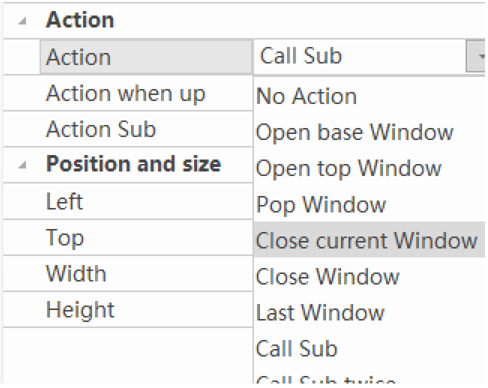
There are two methods to close the window, you can refer to below information.

Note: base window can’t be closed!

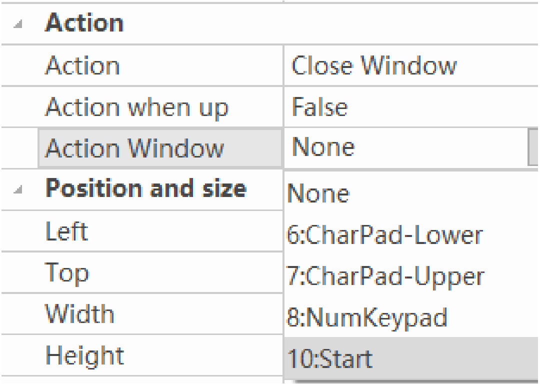
- **Method 1: Close Functional Key**

Same as calling the window, you need to create one functional key. Usually, when you switch the window, puy the “button” inside the “base window”. If you want to close the “pop window” and “top window”, generally, put the “button” inside the window that is to be closed.

There are main 2 action types of “Window OFF”: close current window & close window (it is to close assigned window, you must set window No. action window).



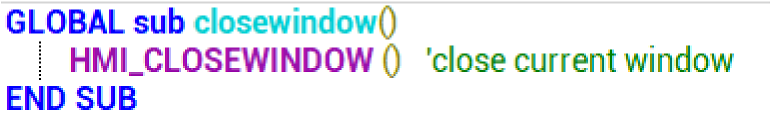
Action	
Action	Call Sub
Action when up	
Action Sub	
Position and size	
Left	
Top	
Width	
Height	



Action	
Action	Close Window
Action when up	False
Action Window	None
Position and size	
Left	
Top	
Width	
Height	

- **Method 2: Close by Program Command**

Window also can be closed by program command in BASIC programing, main command is HMI_CLOSEWINDOW. For more details, please refer to Chapter V.



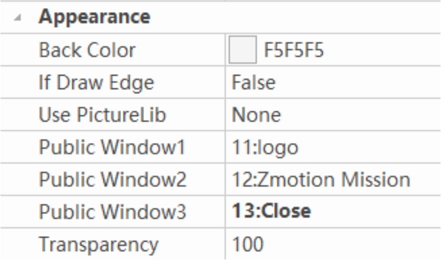
```
GLOBAL sub closewindow()
    HMI_CLOSEWINDOW ()  'close current window
END SUB
```

3.2.6. Public Window

This is used to assign one same window for several windows, in this way, public information can be shown in current window.

You can select current as public window in "Property". One window can set 3 public windows at most.

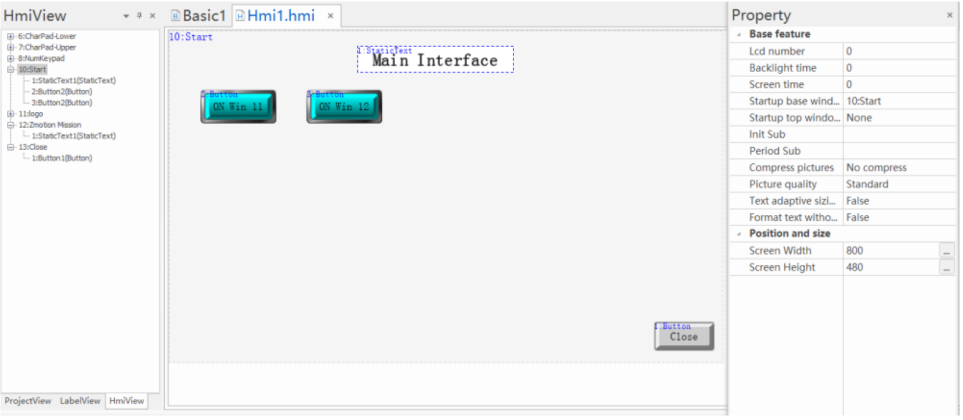
Note: after adding a public window, the components in the public window are only displayed in the current window (the window background is not displayed), and the component properties cannot be modified. To modify the component properties, you must open the window where the component is located. After downloading and running the set component actions, the actions can be executed and take effect.



Appearance	
Back Color	F5F5F5
If Draw Edge	False
Use PictureLib	None
Public Window1	11:logo
Public Window2	12:Zmotion Mission
Public Window3	**13:Close**
Transparency	100

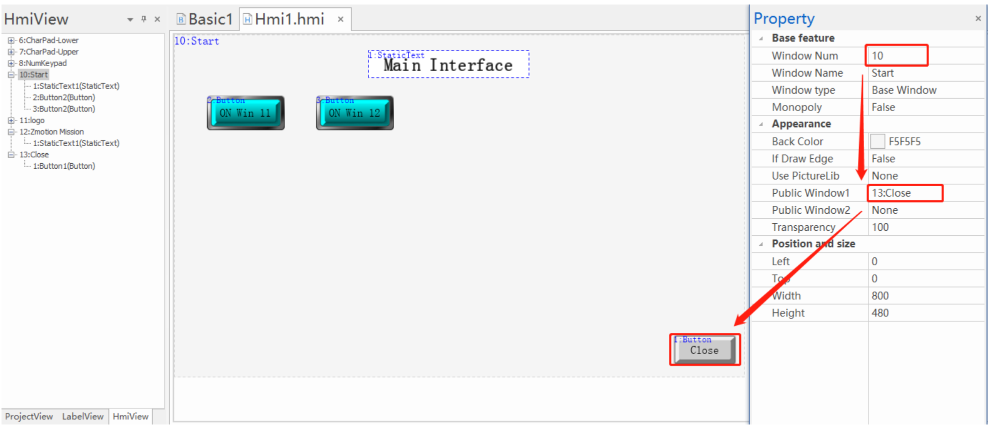
For Example:

- See below, if no public window is added, each window only shows current window's component.



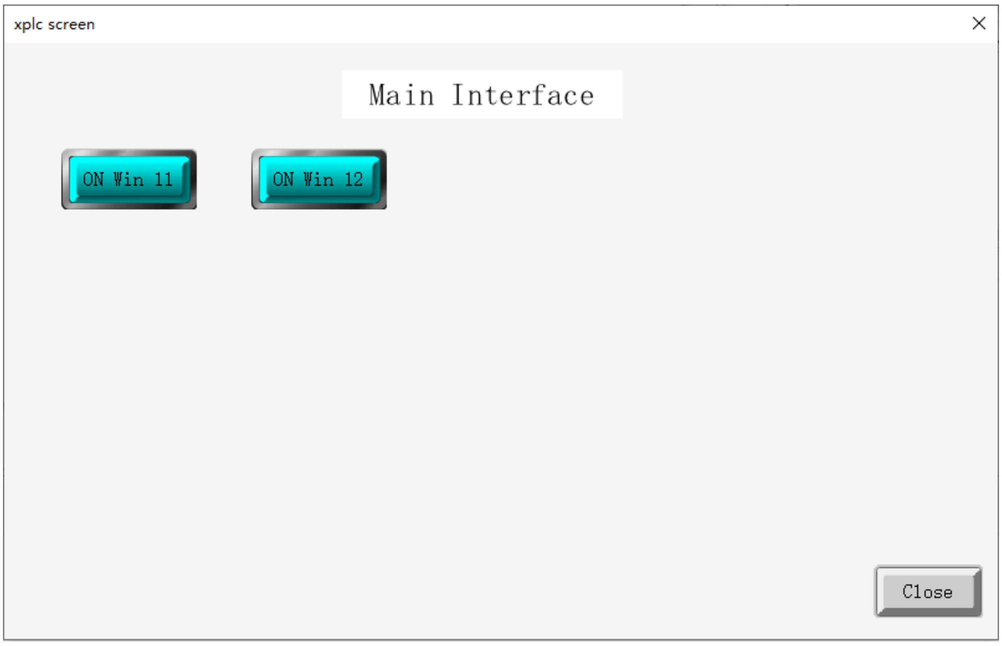
Property	
Base feature	
Lcd number	0
Backlight time	0
Screen time	0
Startup base wind...	10:Start
Startup top windo...	None
Init Sub	
Period Sub	
Compress pictures	No compress
Picture quality	Standard
Text adaptive sizi...	False
Format text witho...	False
Position and size	
Screen Width	800
Screen Height	480

- Add one public window 13 for window 10 (max 3), then this window component will be shown in window 10, but you can’t modify these components in window 10.



Base feature	
Window Num	10
Window Name	Start
Window type	Base Window
Monopoly	False
Appearance	
Back Color	F5F5F5
If Draw Edge	False
Use PictureLib	None
Public Window1	13:Close
Public Window2	None
Transparency	100
Position and size	
Left	0
Top	0
Width	800
Height	480

- After downloading & running, the current window only shows public window’s objects, but doesn’t show public window’s background.



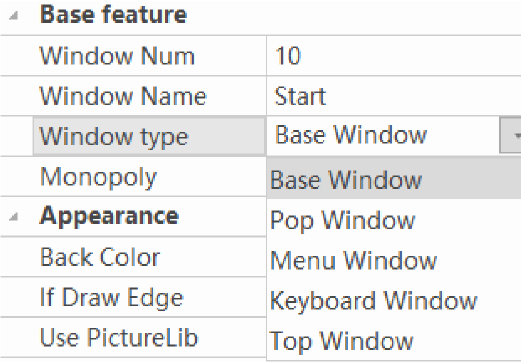
You can open corresponding window through operating main window (window 10), and the public window objects actions in window 10 can take effect.

When you open window 11 and window 12, main window shows window 11 and 12, public window 13 components can close window 11 and window 12, but can’t operate window 10 and window 13.

3.3. Window Type

According to different functions and usages. HMI window types have 5: Base Window, Keyboard Window, Pop Window, Menu Window, Top Window.

And new created window is base window by default. You also can modify it in Property window.



Base feature	
Window Num	10
Window Name	Start
Window type	Base Window
Monopoly	
Appearance	
Back Color	
If Draw Edge	
Use PictureLib	

3.3.1. Base Window

HMI showing must be with one Base Window as bottom window to be as other windows' background interface. And new created window is base window type, and it can be switched by program or component.

Notes:

- Base Window can't be closed.
- The touch screen only can show one base window at the same time.
- When connecting with touch screen, you only need to make sure HMI resolution is consistent with touch screen resolution. For each window's size, you can set as any size, but don't exceed the screen size.

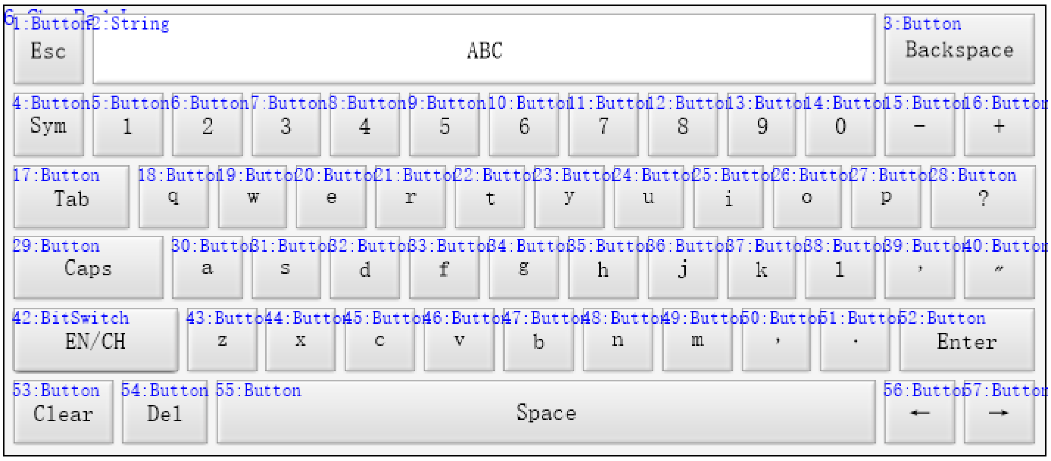
3.3.2. Keyboard Window

Used to show "value", "character" and other components that need to input data by yourself.

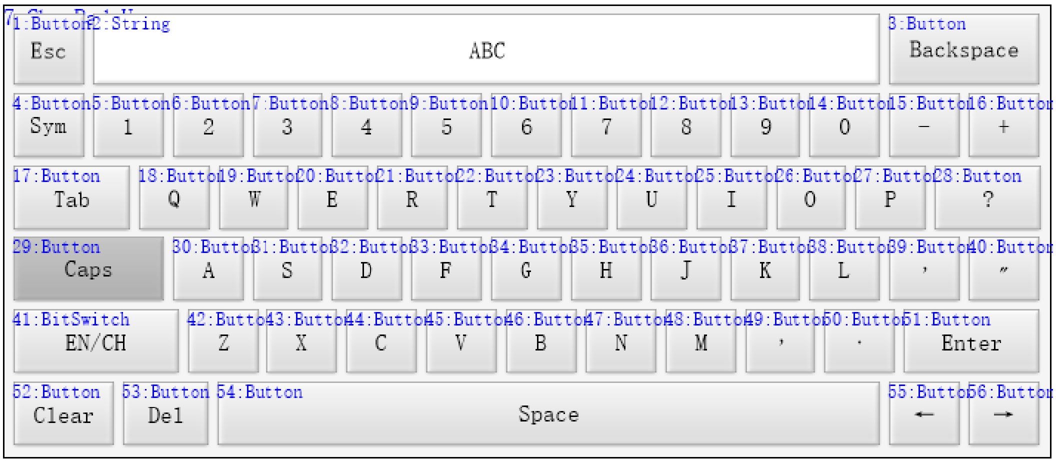
And there are 3 soft keyboard windows already inside the HMI file, you don't need to new build the HMI file. They are window 6, window 7, window 8. When you open one "value" property window, you can select "True" for "Allow Edit", then select one soft keyboard window No. in "Key Window".

Note: when calling the soft keyboard window, it only can be used for the component that supports changing the data, such as, value, character, etc., Function keys are invalid. And for each type of component, please refer to Chapter VI.

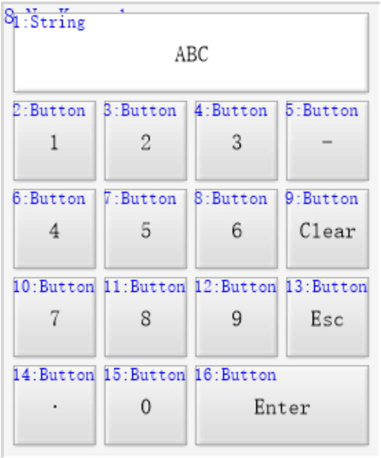
- Window 6: CharPad-Lower



- Window 7: CharPad-Upper



- Window 8: NumkeyPad



3.3.3. Pop Window

It is a kind of dynamic pop window. And it is opened by the program or component operation, for OFF, same as ON. When there are several pop windows, they will be shown according to calling sequence.

Notes:

- When multiple pop windows superpose, functional key in the pop window also can be triggered normally.
- After calling the pop window, it still supports operation for other types of windows, like, base window.

3.3.4. Menu Window

Menu window belongs to pop window, it also pops up through program or component calling.

Notes:

- After the menu window popped up, **you only can operate the menu window**. When you click non-menu window area, menu window will close automatically.
- The position of the menu window after it is opened **is displayed according to the position of the component corresponding to the operation**, and has nothing to do with the position set in the menu window properties. Therefore, when setting the pop-up menu window, pay attention to adjusting the position of the component.

3.3.5. Top Window

It is the window that always shows in the most front. It also needs program or component calling. Usually, one small window is recommended for tool bar. When there are several top windows, they are shown according to calling sequence, that is, behind called window will be above at former called window.

And there are 2 display methods: initialization showing & manual showing.

Display Method	Description
Show in Initialization	HMI – HMI Setting – Property – Startup top window to select one certain window, and set whether it shows when powered on (after setting, top window will cover initial base window).
Show after Manually Call	Show by program or component. ON / OFF by functional key's actions. One window only can select one window type to open it.

Note:

When the window is set as top window, this window will be at top state all the time after calling it. However, you also can operate other windows' components, the top window will not be closed. If you want to close it, also please by program.

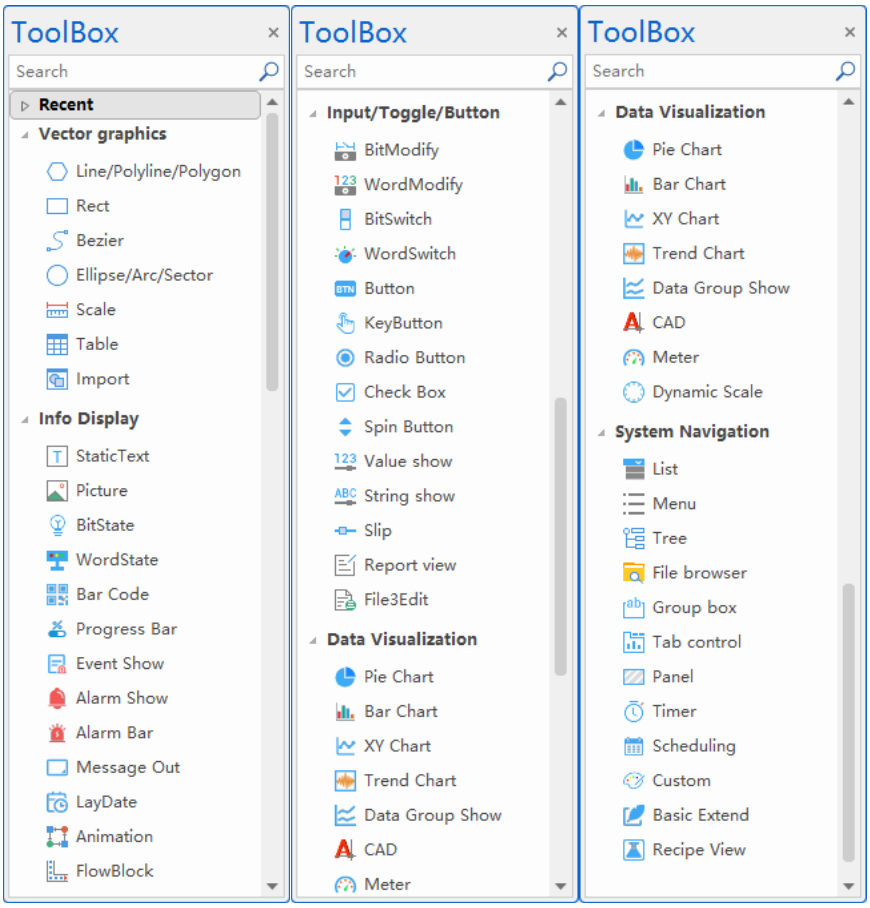
Chapter IV HMI Component

4.1. HMI Quick Tools

4.1.1. Component Menu

HMI development mainly uses windows and configuration components. First, different display windows are created, and then different HMI components are added to each window as needed, and the corresponding functions are realized in combination with program development.

Configuration components are selected and added to the window in the "Control Class" on the far left of the platform interface, or opened from "View"-"Control Class" in the menu bar.



- **Control Class Component Menu:**

Name	Image Mark	Description
Vector Graphics		

Line / Polyline / Polygon	⬡	Draw corresponding lines / polylines according to drawn points.
Rect	▭	Draw one rectangle
Bezier	S	Draw a third-order Bezier curve with four points
Ellipse / Arc / Sector	○	Drag it to draw one full circle/ellipse/arc/sector.
Scale	📏	Draw scale with equal space
Table	▦	Draw 3*3 form, you can customize table format
Import	🖼	Import vector graphics
Info Display		
Static Text	T	Add static text in the window, you can customize corresponding properties.
Picture	🖼	Insert one picture from system / background picture library
BitState	💡	Show the state of bound bit register address value
WordState	🖥	Show the state of bound word register address value
Bar Code	▣	Extract the data from specified register address, then make one QR code / bar code. When you scan it, corresponding content will be shown.
Progress Bar	%	Show the task / the process’s completion progress. You also can assign one register value as the percentage progress value.
Event Show	📄	Show the event that has been defined in [event list] and has met triggering conditions. And they are shown in sequence of triggered time.
Alarm Show	🔔	Show the event that has been defined in [event list], and the event that can meet triggering conditions under now system state, these events are called “Alarm”. [Alarm Show] component will show them in the sequence of triggered time.

Alarm Bar	🚨	Same as Alarm show, but this component will show them in the sequence of triggered time through single-line scrolling marquee format.
Message Out	▭	Output the controller’s log message, error, warn, and debugged information.
LayDate	📅	Select date and time, and you could select whether to show any date (year/month/day), any time (hour/minute/second), then output to specified register / variable.
Animation		After presetting the component movement trajectory, modify the register inner data to control the component state and the position of the motion trajectory.
FlowBlock		You can pre-define the component motion trajectory, and after downloading, it will be shown like “flow”.
Input / Toggle / Button		
BitModify		Set bit register address value according to component action state
WordModify	123	Set word register address value according to component action state
BitSwitch		Set bit register address value and show related state according to component action (BitState + BitModify)
WordSwitch		Set word register address value and show related state according to component action (WordState + WordModify)
Button	BTN	Achieve state / window / keyboard switching, there are only two showing states and can’t bind with the register.
KeyButton	👆	Used to bind with virtual key / real key, then it will act according to real button.
Radio Button	◉	Multiple [radio button] can bind with one same register.
Check Box	☑	Multiple [check box] can bind with one same register. Single-click the component can choose it / cancel it.
Value Show	123	Edit and show value, and change corresponding bound register value

String Show	ABC	Edit and show character string, and change word register value
Slip	-□-	Change word register value by dragging the slide
Report View	📝	Show multi-group data through form, then data can be shown clearly and managed easily.
File3Edit	📄	Develop the edit component of file 3 program in HMI
Data Visualization		
Pie Chart	◔	Show each channel ratio of one group of IN register's value through pie chart.
Bar Chart	📊	Show register data (percentage) through bar chart. One bar chart only represents one object. You also can combine several bar charts together.
XY Chart	📈	Show 2D XY data points through curve. Each data point includes X and Y. And XY data can be written through HMI_XYChartWhite command.
Trend Chart	📈	Use continuous segments to make the real-time captured data as one graphic for resource analysis, therefore, it is used together with resource sampling library.
Data Group Show	📈	Show multiple groups of data at the same time through line chart.
CAD	A	Show vector graphics
Meter	◔	Use instrument diagram to specify current register's data value.
Dynamic Scale	○	Show different scales, and you can set the scale' range for other components.
System Navigation		
List	☰	Show multiple lists, switch to corresponding item by bound register value
Menu	☰	Set the menu items, click one item to trigger corresponding SUB actions.
Tree	🌲	Show all form items through tree chart. Single-click the icon / double-click the content to show / hide sub-

		tree contents.
File Browser	📂	Show current content, and show file information through form format.
Group Box	ab	Used to provide identifiable grouping for other elements in the same window.
Tab Control	📊	Define the same area of the window as multiple pages, and display the corresponding information or elements when the user selects the appropriate option.
Panel	▨	Users can group all elements in a window using panel elements to categorize and divide the space.
Timer	⏱	Refresh regularly to do repeated actions.
Scheduling	📅	Set the schedule, when it arrives preset time, it will be triggered to execute needed actions. Suit to plan routine programs within one week.
Custom	🎨	Call basic function in the area of component.to achieve dynamic drawing
Basic Extend	📝	You can customize an independent component.
Recipe View	⚗	It needs to be used in conjunction with a recipe library to enable the display, selection, and editing functions of the recipe library.

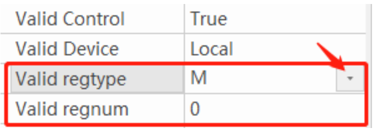
4.2. HMI Component General Property

Different types of components are with general properties, like, register calling, function calling, position, size, etc.

4.2.1. Register

Most of components include “register type” property for building data relation with register.

Register value can be controlled and shown by component. Below shows register configuration and valid register types.

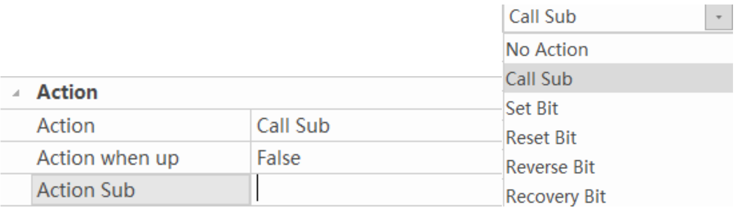


Valid Control	True
Valid Device	Local
Valid regtype	M
Valid regnum	0

Register Type	Controller Register	Description
X	IN	This register corresponds to general IN, No. 0 = MODBUS_BIT (10000)
Y	OP	This register corresponds to general OUT, No. 0 = MODBUS_BIT (20000)
M	MODBUS_BIT	Different controllers support different register numbers, for power down storage function: 2048-2175
S	State Register “S”	No. range: 0-999, No. 0 = MODBUS_BIT (30000). For power down storage function: 0-127
D	MODBUS_4X Register: According to data type: INT16: MODBUS_REG INT32: MODBUS_LONG FLOAT32: MODBUS_IEEE	Different controllers support different register numbers.
D.DOT	Read MODBUS_REG bit by bit: No. = reg No. * 16 + dot (0-15)	Please use bit state to show component
DT	TABLE	32-bit floating data
T	Timer: No.: 0-127	Register length 32-bit, when you used 16-bit command to access, it will automatically use low 16-bit.
C	Counter: No.: 0-127	
@	Variables & Arrays defined by Basic	The type must be GLOBAL.

4.2.2. Action

In the action menu, you can select different actions, and different components support different kind of “actions selections”.



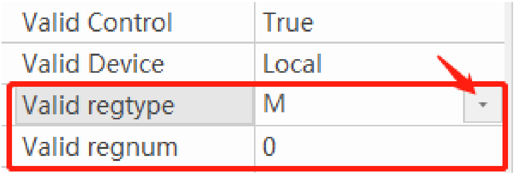
Action

Action	Call Sub
Action when up	False
Action Sub	

Action Name	Function	Description
Component General Actions		
No Action	/	/
Call Sub	Call SUB function defined by Basic	Function must be GLOBAL.
Functional Key "Button"		
Open Base Window	Open window with "Base Window"	In "Action Window", select the No., for more window type info, please refer to "3.3".
Open Top Window	Open window with "Top Window"	
Pop Window	Open window with "Pop Window"	
Close Current Window	/	Close all windows of the current key button.
Close Window	Close selected / assigned window	In "Action Window", please select the No.
Last Window	Open the last base window	Open former one base window
Call Sub	Press it, call one function, release it, call another function.	It needs to set "Action when up" & "Action Sub".
Input Physical Key	Bind with physical key	Refer to physical key form
Input Virtual Key	Set as virtual key	Select No. by "virtual key code".
Input String	Input the character string	It only can be used in soft keyboard window.
Shift Window	Switch the window	It must be used in windows that are not BASE window, and it only can switch windows that are same type.
BitSwitch (Bit State Switch) / BitModify (Bit State Configuration)		
Set Bit	When pressed, set it as 1	Switch Type
Reset Bit	When pressed, set it as 0	
Reverse Bit	Use the opposite state, when it is 1, it becomes 0, for 0, it becomes 1.	
Recovery Bit	Use normal state, when it is pressed, use state 1, when released, use 0.	
WordSwitch (Multi-State Switch) / WordModify (Multi-State Configuration)		
Data Write	Write data into register	The data value is set by "action data".
Data Plus	Add / reduce value for register	The register is selected

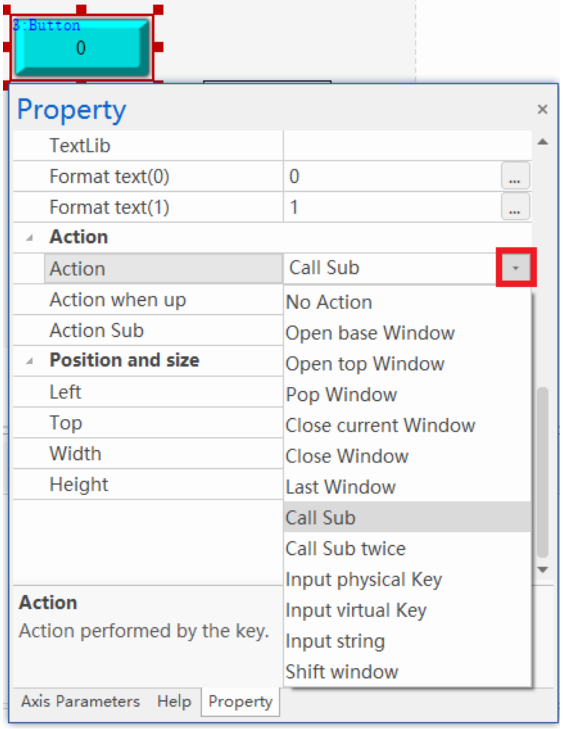
	original value	through register type (bound regtype) and No. (bound regnum)
Data Loop	Add value for register original value, then make cycle between the value and set state numbers, in this way, it can achieve periodic data switching.	The register is selected through register type and No. If the register original value is more than "state num", it will decrease to set state numbers by "original value" – "action data" at first, then enter cycle.

➢ Component's General Actions:



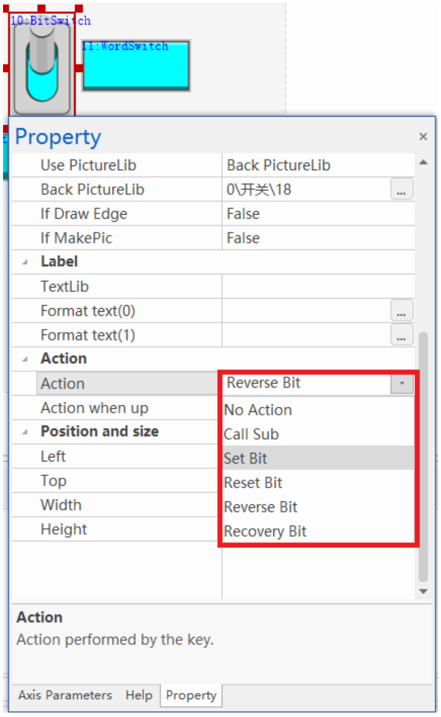
Valid Control	True
Valid Device	Local
Valid regtype	M
Valid regnum	0

➢ Functional Key "Button" + BitSwitch & BitModify:



TextLib	
Format text(0)	0
Format text(1)	1
Action	
Action	Call Sub
Action when up	
Action Sub	
Position and size	
Left	
Top	
Width	
Height	

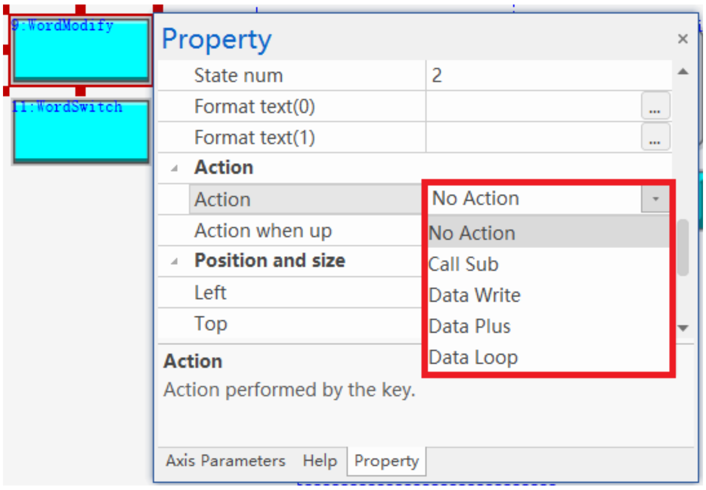
Action
Action performed by the key.



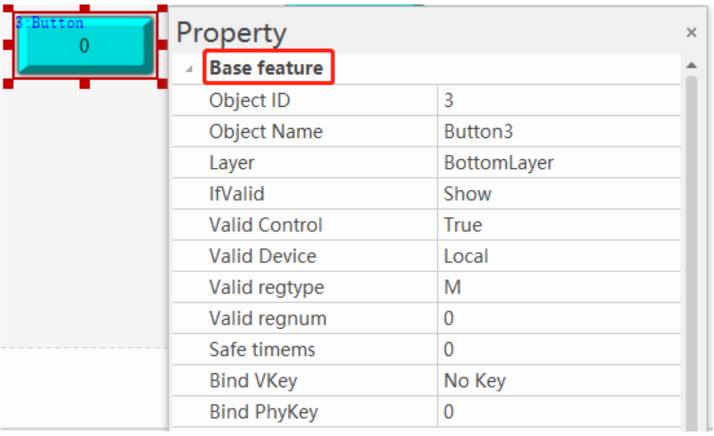
Use PictureLib	Back PictureLib
Back PictureLib	0\开关\18
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	
Format text(1)	
Action	
Action	Reverse Bit
Action when up	
Position and size	
Left	
Top	
Width	
Height	

Action
Action performed by the key.

- WordSwitch & WordModify:



4.2.3. "Base Feature" of Object Property

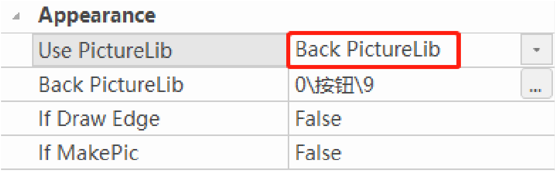


Object ID	3
Object Name	Button3
Layer	BottomLayer
IfValid	Show
Valid Control	True
Valid Device	Local
Valid regtype	M
Valid regnum	0
Safe timems	0
Bind VKey	No Key
Bind PhyKey	0

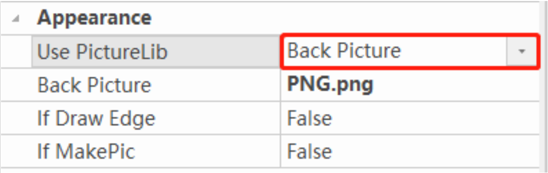
Base Feature	Description
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading.

	● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Safe timems	Unit is ms.
Bound Virtual Keys	Not used by default.
Bound Physical Keys	Button codes, please refer to "vitrual buttons".

4.2.4. "Appearance" of Object Property



Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False

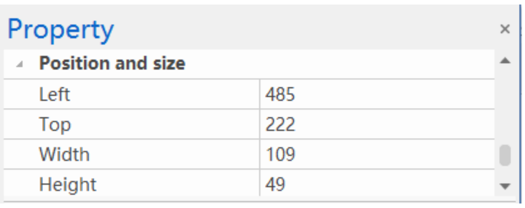


Appearance	
Use PictureLib	Back Picture
Back Picture	PNG.png
If Draw Edge	False
If MakePic	False

Appearance	Description
Use Picture Lib	Some components support displaying pictures. You can select the picture source here. After selecting the picture source, you can select the target picture to add in the next line. ● None: Do not add pictures ● Back PictureLib: Select pictures from the background picture library ● Back Picture: Select pictures from the project
Back Picture Lib	Select pictures from the background picture library, specifically, you need to add the picture in the picture library, then, select, or select already existed one.
Back Picture	Select pictures from the project, that is, add the picture into project "projectview", then select. Please note the picture name can't exceed 26 characters.
If Draw edge	Draw the edge?
Edge Color	Select the edge color.

If MakePic	Convert the component display content to images. For example, when the displayed text font is too large or contains rare characters, the font is unclear. Convert the displayed content to images. After setting it to True, the display effect can be improved.

4.2.5. “Position and Size” of Object Property



Position and size

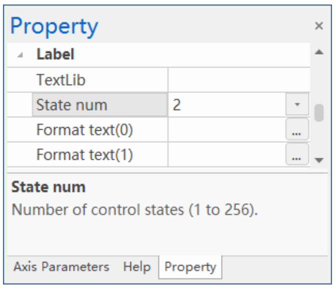
Left	485
Top	222
Width	109
Height	49

Position and Size	Description
Top	The horizontal distance of the component in the Hmi window. The upper left corner of the Hmi window is (0,0).
Left	The vertical distance of the component in the Hmi window. The upper left corner of the Hmi window is (0,0).
Width	Set component’s displaying width
Height	Set component’s displaying height
Note: please don’t make component position and size exceed the window.	

4.2.6. “Label” of Object Property

Used to edit the text display content of each state of the component. It also supports editing text directly on the component.

For components that need to set the "number of states", the "state num" supports custom modification (up to 256 states are supported). The number of "format texts=" increases or decreases as the number of states is modified.



Label

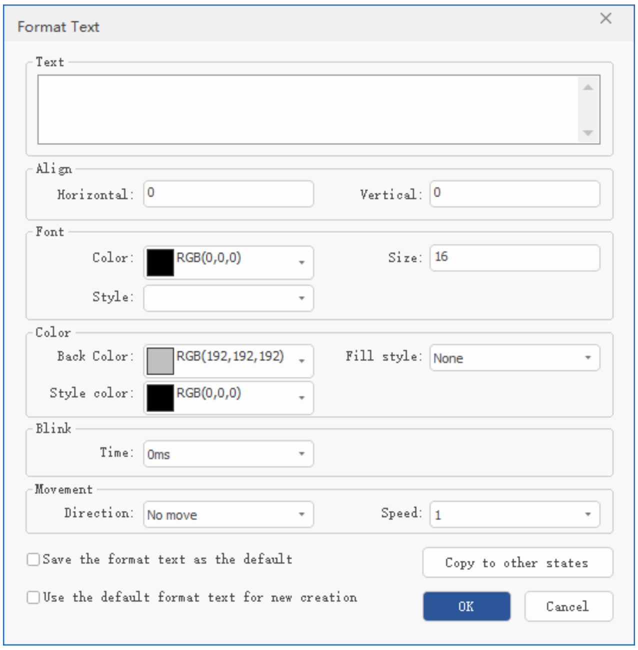
TextLib	
State num	2
Format text(0)	
Format text(1)	

State num

Number of control states (1 to 256).

When you click the "…", one window will pop up, then you can enter needed text and set related formats.

- **Text:** enter needed text, also can enter next line text. Then you can set formats: alignment, colors, etc. Note: if you want to use modified object background color and formats, please cancel already applied picture library / background picture at first.
- **Copy to other states:** copy the other settings of the format text except the "Text (content)" item to the format text of other states (that is, apply the text style of the current state to other states). Click the format text of other states to see that the parameters are consistent with the text parameters. Note: It is only applied to different states of the current component and cannot be copied to other components!
- **Save the format text as the default:** save the text style of the current component setting as the global default format, that is, apply the current text style to the component style in this project. Check to take effect. Note: It can only be applied to newly created components, and the styles of existing components will not be changed!
- **Use the default format text for new creation:** The newly created component uses the default format text that has been set. Check it to take effect.

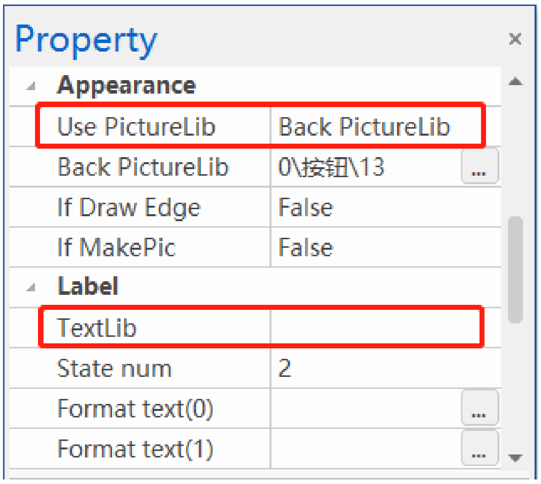


Horizontal: 0
Vertical: 0
Color: RGB(0,0,0)
Size: 16
Style:
Back Color: RGB(192,192,192)
Fill style: None
Style color: RGB(0,0,0)
Time: 0ms
Direction: No move
Speed: 1
□ Save the format text as the default
□ Use the default format text for new creation

Operation	Function	Description
Text	Enter needed text info	/
Align	Align-Horizontal	0: center alignment (default) >0: left / upper alignment, the value indicates the
	Align-Vertical	

		distance from the left / up <0: right / bottom alignment, the absolute value indicates the distance from the right / bottom
Font	Font-Color	Select font color
	Font-Size	Select font size
	Font-Style	Select font file (the suffix is .ttf/.zft) into project
Color	Back color	When you selected “None” for “use picturelib”, background color can be applied for this component
	Fill color	There are 6 formats, also used when you selected “None” for “use picturelib”.
	Style color	When you selected “None” for “use picturelib”, color format can be applied for this component
Blink Time	Select the shrink time	The format text content is displayed flashing. Then you can set the flashing interval, 0ms means no flashing.
Movement	Select the Direction	Text will move to the direction set by you.
	Set the speed	Text will move to the set direction at set speed here, if no direction set, no speed also.

4.2.7. “Picture Lib” & “Text Lib”


Property

Appearance

Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\13
If Draw Edge	False
If MakePic	False

Label

TextLib	
State num	2
Format text(0)	
Format text(1)	

HMI object supports adding pictures and text. Pictures can be added from "Back PictureLib" and "Back Picture", text can be directly called from "TextLib".

Before adding using the above methods, you need to create/add pictures or text content first. To insert a background picture, you need to add the picture to the project first, and then select the name of the picture to be loaded here.

For usage details, please refer to the [2.2.3 Picture Library] and [2.2.2 Text Library].

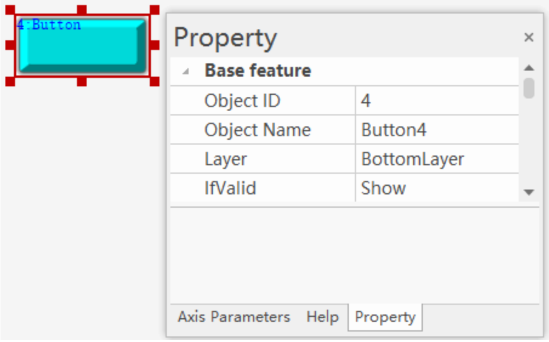
4.3. Components in Control Class Introduction & Usage

- **How to Add HMI Component?**

Open HMI file, find window 10 or create one new window, then select one object from "control class" (menu View – Control Class) and single-click any position in the window, in this way, one component is added. If you already selected one component, but no need any more, just make your mouse in the window and right click to cancel it.

- **How to Modify Component Size / Position?**

After clicking the needed object, when there is red frame, you can drag directly or change the size. What's more, you also can set in corresponding property, that is, position & size in property.



Base feature

Object ID	4
Object Name	Button4
Layer	BottomLayer
IfValid	Show

- **How to Set Component Functional Property?**

Single click the component, corresponding "property" window will be opened automatically (if not opened, click menu View – Property, then click component), then you can set through it. After setting, click "Enter" or other places, new info will be saved.

- Note: some functions in "property" only can be viewed after downloading and running.

4.3.1. Line / Polyline / Polygon

4.3.1.1. Line

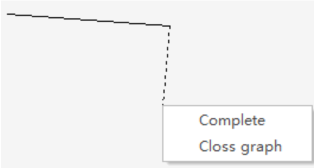
A. What is It?

“Line” is the line segment, whose length and slope by yourself. Because width and height can be set by dragging directly or in property window.

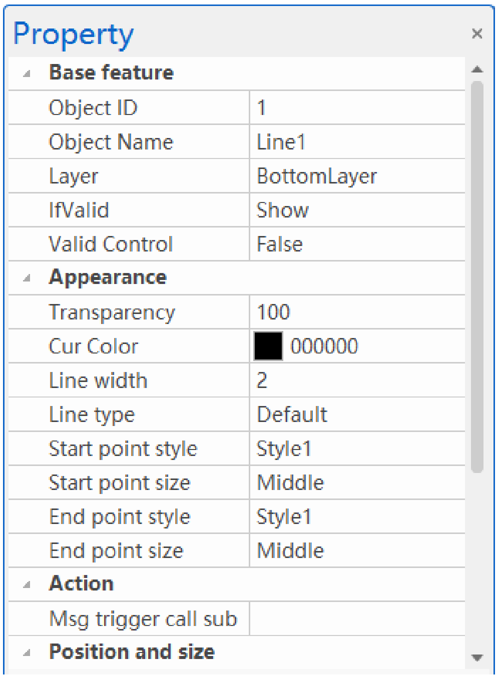
B. How to Use?

Click “View - Tool Box”, select “Line / Polyline / Polygon” from “Vector graphics”. Two points can draw one segment, when the second line drawing operation is about to start, right-click and click "Complete" in the pop-up menu to draw the line segment. The specific operation method is shown in the figure below.

Note: when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.



C. “Property” Window



Property

Base feature	
Object ID	1
Object Name	Line1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Start point style	Style1
Start point size	Middle
End point style	Style1
End point size	Middle
Action	
Msg trigger call sub	
Position and size	

D. "Property" Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Start Point Style	There are many kinds of shapes for you.
Start Point Size	You can set the starting point style size, it is valid after selecting style.
End Point Style	There are many kinds of shapes for you.
End Point Size	You can set the end point style size, it is valid after selecting style.
Action	

Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.1.2. Polyline

A. What is It?

A non-closed figure composed of multiple continuous line segments is called a polyline segment.

B. How to Use?

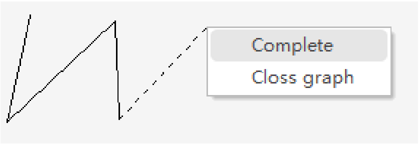
Click "View - Tool Box", select "Line / Polyline / Polygon" from "Vector graphics". Same method as "Line", but for this, you need click multi-point at different places to draw the "polyline".

Notes:

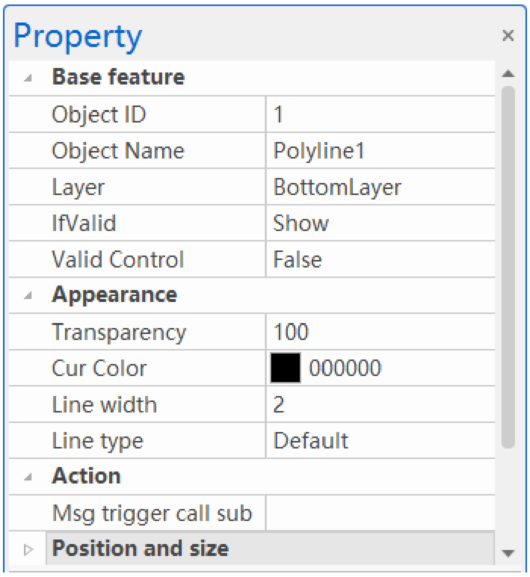
a. when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.

b. After confirmed, each point's relative position can't be modified, you only can zoom the whole shape.

c. If the starting point and end point are the same, the polyline can't be used as a closed graphic, then no way to fill the area defined by the polyline.



C. “Property” Window



Property

Base feature	
Object ID	1
Object Name	Polyline1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Action	
Msg trigger call sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.

Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.1.3. Polygon

A. What is It?

A closed figure composed of multiple continuous line segments is called a polygon, and its background color can be filled. "polygon' shape is drawn by yourself.

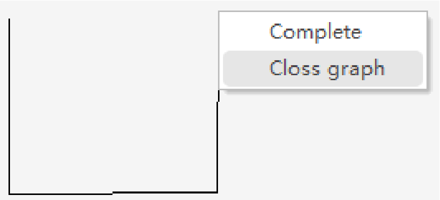
B. How to Use?

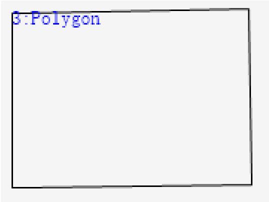
Click "View - Tool Box", select "Line / Polyline / Polygon" from "Vector graphics". Same method as "Polygon", but for this, not select "complete", please select "close graph", then, it will close automatically. That is, if you want one quadrilateral, after 4 points, no need to manually draw the last one line segment, select "close graph".

Notes:

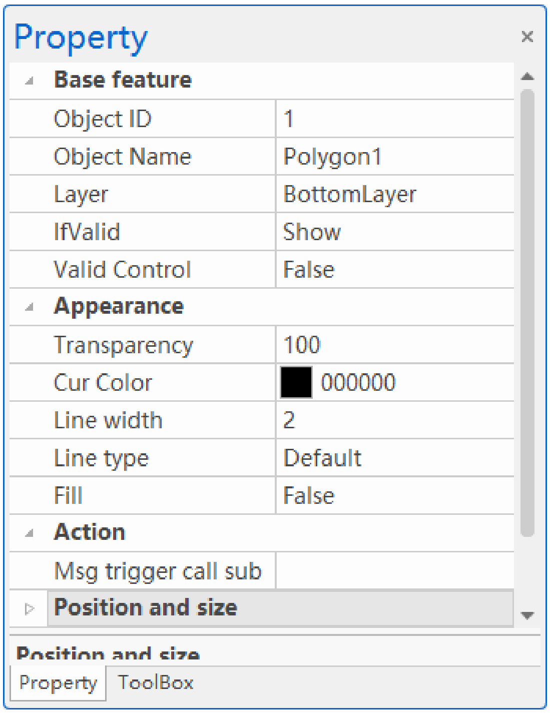
a. when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.

b. After confirmed, each point's relative position can't be modified, you only can zoom the whole shape.





C. "Property" Window



Property

Base feature	
Object ID	1
Object Name	Polygon1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Fill	False
Action	
Msg trigger call sub	
Position and size	

Position and size

Property ToolBox

D. "Property" Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading.

	● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Fill	Whetehr fill the color (for whole component), Flase / Ture
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.2. Rectangle

A. What is It?

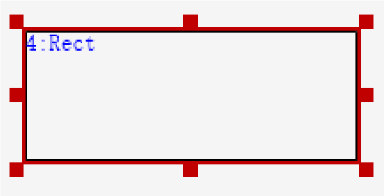
Create one rectangle, "rect" is one closed component that can fill in the background color.

B. How to Use?

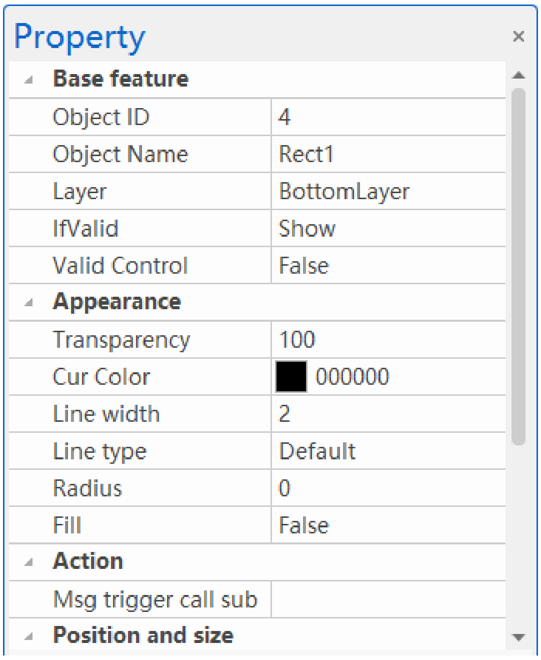
Click “View - Tool Box”, select “Rect” from “Vector graphics”. After determining the single point position, hold down the left mouse button and drag it to the appropriate length and width before releasing it.

Notes:

a. when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.



C. “Property” Window



Property

Base feature	
Object ID	4
Object Name	Rect1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Radius	0
Fill	False
Action	
Msg trigger call sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and

	covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Radius	Chamfer radius, whether the four corners need chamfering.
Fill	Whetehr fill the color (for whole component), Flase / Ture
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.3. Bezier Curve

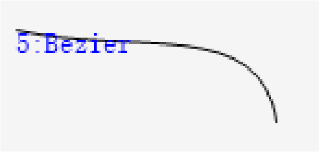
A. What is It?

The vector mathematical curve based on a two-dimensional plane.

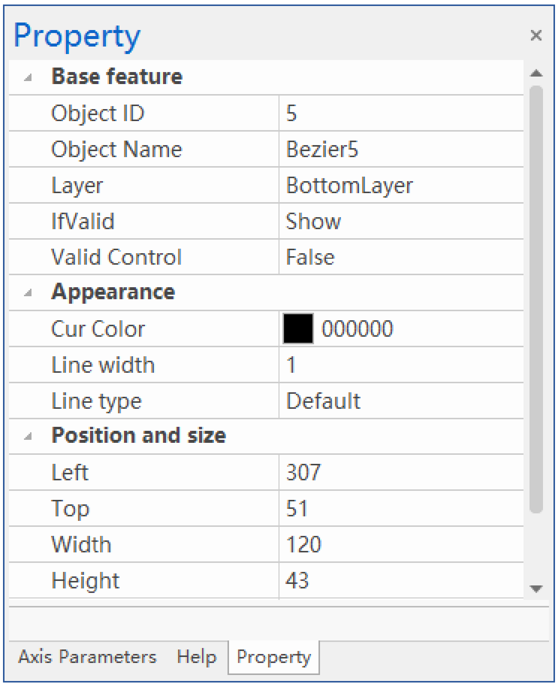
B. How to Use?

Click “View - Tool Box”, select “Bezier” from “Vector graphics”. When drawing, you only need to determine 4 points, then you can automatically create the Bezier.

Note: when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.



C. “Property” Window



Property

- Base feature

Object ID	5
Object Name	Bezier5
Layer	BottomLayer
IfValid	Show
Valid Control	False

- Appearance

Cur Color	■ 000000
Line width	1
Line type	Default

- Position and size

Left	307
Top	51
Width	120
Height	43

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.

Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Postion & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width

Height	Object height

4.3.4. Ellipse / Arc / Sector

4.3.4.1. Ellipse

A. What is It?

"Ellipse" is a closed object that can be filled with background color. It supports setting line type, color, fill color and style for circles/ellipses.

B. How to Use?

Click “View - Tool Box”, select “Ellipse / Arc / Sector” from “Vector graphics”. When you moved your mouse at the window, single click to determine the position in the window, then draw with below methods:

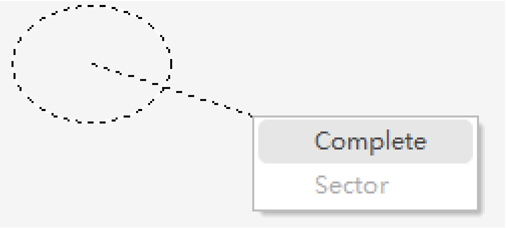
*Draw Ellipse: you only need to drag the mouse, confirm size, then right click “complete” (no need to confirm position).

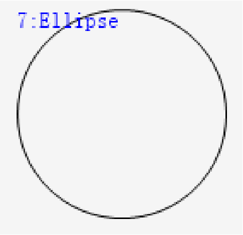
*Draw Circle: when you drag the mouse to determine the size, press Ctrl button at the same time, when confirmed, release mouse and button, then right click “complete” (no need to confirm position).

Notes:

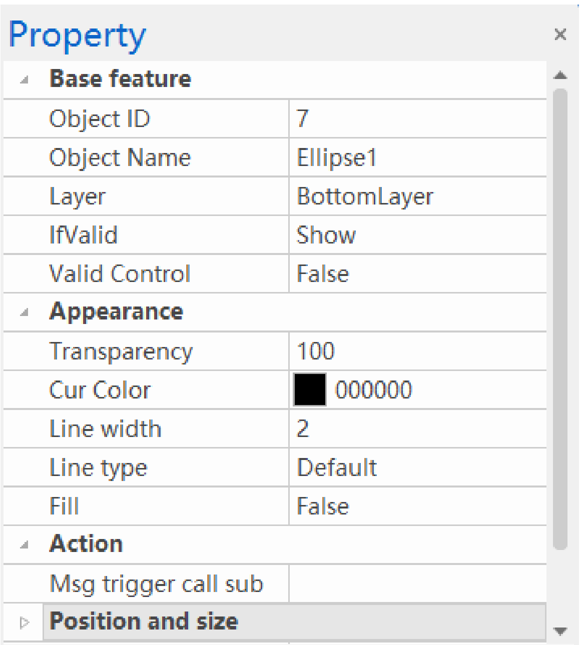
a. when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.

b. Since the arc of a circle/ellipse is distributed with countless points, adjusting the width and height can change the shape (e.g., drawing a full circle and then adjusting the size can transform it into an ellipse)





C. “Property” Window



Property

Base feature	
Object ID	7
Object Name	Ellipse1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Fill	False
Action	
Msg trigger call sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list

	● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Fill	Whetehr fill the color (for whole component), Flase / Ture
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Positon & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.4.2. Arc

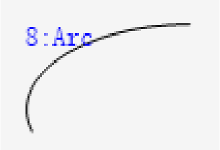
A. What is It?

The curve between any two points on a circle is an arc.

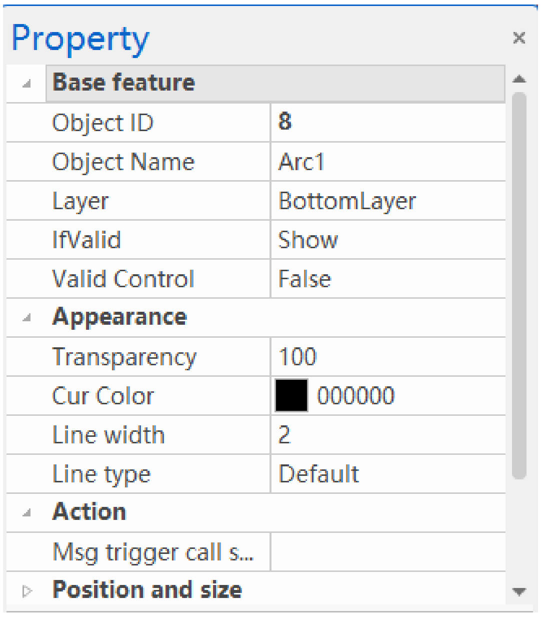
B. How to Use?

Click "View - Tool Box", select "Ellipse / Arc / Sector" from "Vector graphics". At first, draw one rectangle to confirm the size, then determine the arc starting point and end point, next, draw the arc from the starting point to end point clockwise, at last, click "complete".

Note: when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.



C. “Property” Window



Property

Base feature	
Object ID	8
Object Name	Arc1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Action	
Msg trigger call s...	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register

	is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.4.3. Sector

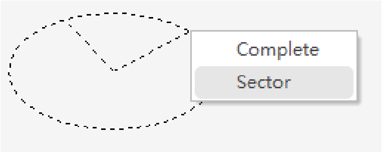
A. What is It?

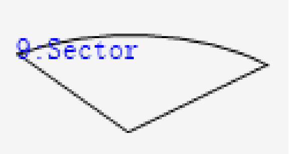
A closed object consisting of an arc and two radiuses.

B. How to Use?

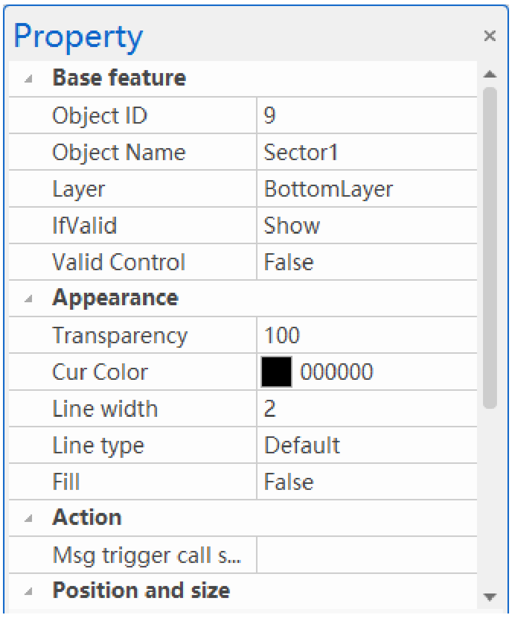
Click "View - Tool Box", select "Ellipse / Arc / Sector" from "Vector graphics". At first, draw one rectangle to confirm the size, then determine the arc starting point and end point, next, after that, right click "sector", it will generate one sector at the clockwise direction.

Note: when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.





C. “Property” Window



Property

Base feature	
Object ID	9
Object Name	Sector1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Fill	False
Action	
Msg trigger call s...	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False.

	If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Fill	Whetehr fill the color (for whole component), Flase / Ture
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.5. Scale

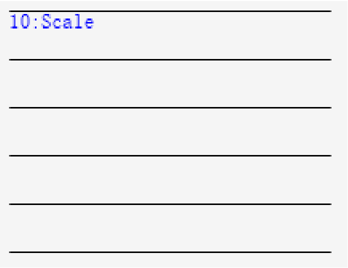
A. What is It?

Draw multiple rows of contour lines to serve as auxiliary marks for component alignment, etc.

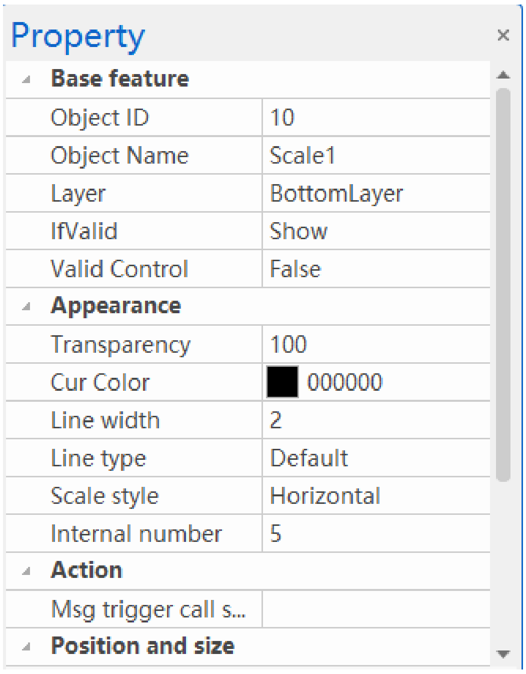
B. How to Use?

Click "View - Tool Box", select "Scale" from "Vector graphics". You only need to move the mouse to any position in the window, then click. You can adjust the size, paragraph, line direction in the property window.

Note: the row space can't be modified manually. You can zoom in or out in or change object's width and height.



C. "Property" Window



Property

Base feature

Object ID	10
Object Name	Scale1
Layer	BottomLayer
IfValid	Show
Valid Control	False

Appearance

Transparency	100
Cur Color	■ 000000
Line width	2
Line type	Default
Scale style	Horizontal
Internal number	5

Action

Msg trigger call s...	

Position and size

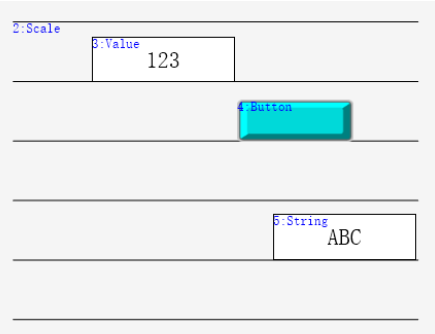
D. "Property" Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)

IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select the line segment color.
Line Width	Set the width, default is 1, max is 20.
Line Type	There are many types of line, you can choose.
Scale Style	The scale is vertical or horizontal.
Internal number	How many paragraphes, default is 5.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

E. For Example

Put the components between scale lines, which can do alignment.



4.3.6. Table

A. What is It?

Draw one net table, you can put the component in the table, which is mainly used to align objects.

B. How to Use?

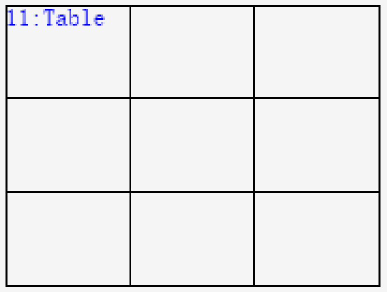
Click “View - Tool Box”, select “Table”. You only need to move the mouse to any position in the window. When you move the object to table position, it will automatically align, and there are many kinds of alignment methods.

Note:

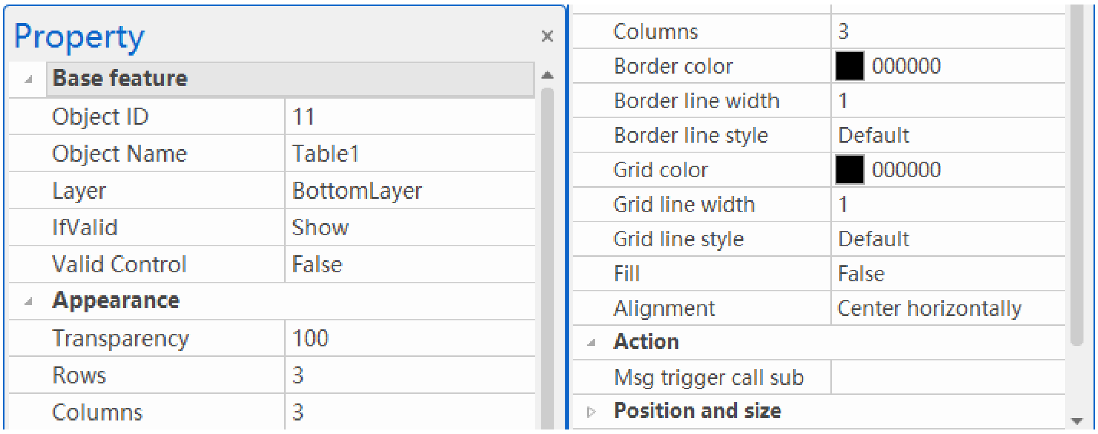
a. One table only can set one alignment method.

b. The size of the component to be placed must be < the size of a single cell in the table in order to achieve the automatic alignment adsorption.

c. While using, please attention the show layer configuration.



C. “Property” Window



Property

Base feature	
Object ID	11
Object Name	Table1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Rows	3
Columns	3
Border color	■ 000000
Border line width	1
Border line style	Default
Grid color	■ 000000
Grid line width	1
Grid line style	Default
Fill	False
Alignment	Center horizontally
Action	
Msg trigger call sub	
Position and size	

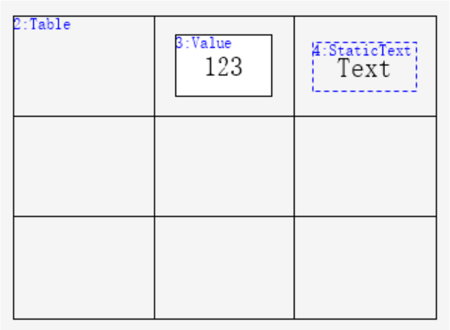
D. **"Property" Description**

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%)

	0% -- fully transparent, 100% -- fully show
Rows	Set table rows
Columns	Set table columns
Border Color	Set the border color of the table
Border Line Width	Set the border line width of the table
Border Line Style	Set the border line types of the table
Grid Color	Set grid line color of the table
Grid Line Width	Set grid line width of the table
Grid Line Style	Set grid line types of the table
Fill	Whether needs to fill in the color
Alignment	Set aligment, there are 9 alignment methods: Align top left/Align top center/Align top right/Align left center/Align center/Align right center/Align bottom left/Align bottom center/Align bottom right
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

E. For Example

After adjusting the component to the appropriate size, drag it directly into the table to automatically adsorb and align it. By setting the alignment method, the components can be placed in the table in the corresponding manner.



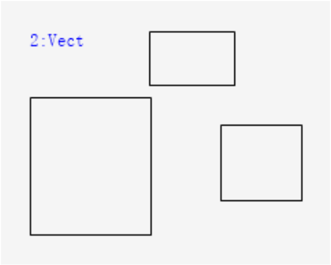
4.3.7. Import Vector Graphics

A. What is It?

Import the vector graphics.

B. How to Use?

Click "View - Tool Box", select "Import" from "Vector graphics". When creating, first determine the position of the box, pop up the file selection window, open the vector file in the system disk and fill it into the component for display. (valid graphic formats: dxf/.ai/.plt/.dst/). And you can modify the color, line width, background color, etc. of vector graphics.



C. "Property" Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown.

	● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Select segment color.
Line Width	Set line width. Default is 1.
Line Style	Set line types, there are solid or dotted line.
Fill	Whether needs to fill in the color, default is False, not to fill.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Positon & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.8. Static Text

A. What is It?

Enter and show text information. It can show single line or multiple lines.

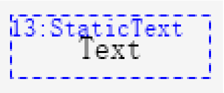
B. How to Use?

Click "View - Tool Box", select "Static Text". When you selected, put the rectangle frame at any position, then you can enter the text directly.

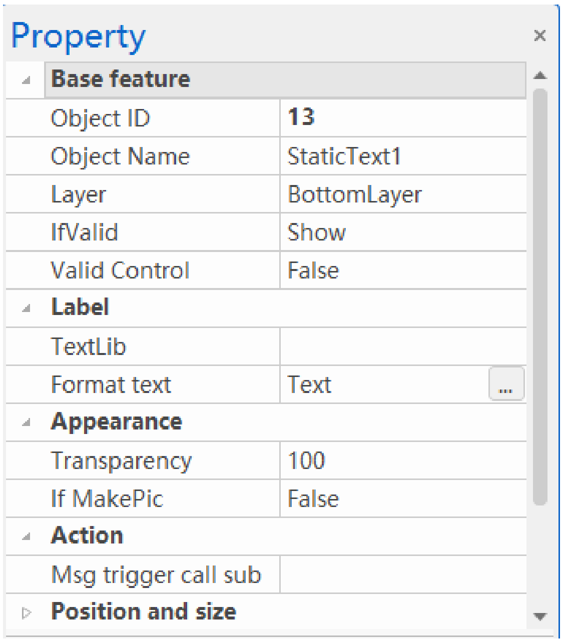
Notes:

a. It can’t directly change the line for the component that enters the text. It must be done in “property – format text”, for more details, please refer to.

b. When there is a lot of text, pay attention to adjusting the component size. If the component is too small, it may not be fully displayed.



C. “Property” Window



Property

Base feature	
Object ID	13
Object Name	StaticText1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Label	
TextLib	
Format text	Text
Appearance	
Transparency	100
If MakePic	False
Action	
Msg trigger call sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading.

	● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Label	
Textlib	The text library name, if no set, it shows "Text".
Format text	Show you edited text info, you can set what content you want to shown in the "format text" window.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If MakePic	Set whether the component changes as graphic, the default is False.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

4.3.9. Picture

A. What is It?

Import static picture or GIF picture, and show it.

B. <u>How to Use?</u>

Click “View - Tool Box”, select “Picture”. Add one “picture” component in the window, then in corresponding property, you can add the picture from Back picture lib or Back picture.

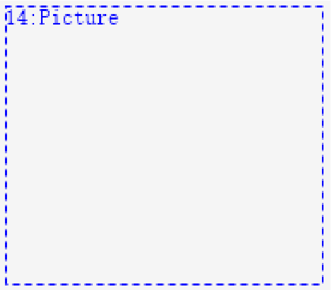
**from Back Picture Lib:* add the picture into picture library at first, then select “Back picture lib”, and find needed one.

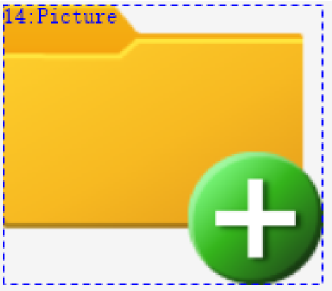
**from Back Picture:* add the picture into “Project View” at first, then select “Back picture”, and find needed one.

Notes:

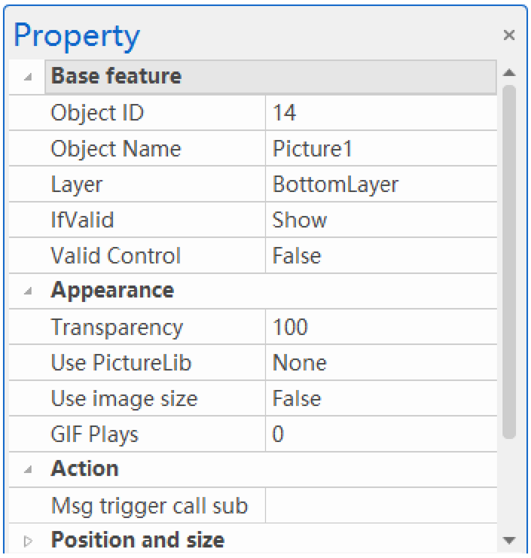
a. For GIF dynamic picture, it only can choose “from Back Picture Lib”. That is, it doesn’t support “from Back Picture”.

b. When using vision, this component can show the pictured shot by the vision, that is, select vision channel in “Back Picture”. It supports 4 vision channels, @ZV0, @ZV1, @ZV2, @ZV3, before using, save the picture into vision latch channel.





C. <u>“Property” Window</u>



Property

Base feature	
Object ID	14
Object Name	Picture1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Use PictureLib	None
Use image size	False
GIF Plays	0
Action	
Msg trigger call sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
Use Image Size	Whether to use picture original size, False: use component size, True: use picture original size.
GIF Plays	0 = periodical play, >0 means the play times, default is 0.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is

	triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

E. For Examples:

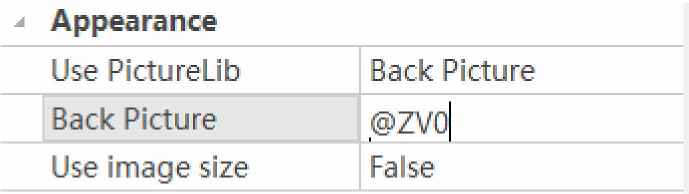
1) Example 1: read vision channel (ZV) picture

Step 1: put the picture into controller / simulator flash folder.

Step 2: read picture and save into vision channel 0 @ZV0 in Basic file.

```
GLOBAL ZVOBJECT Image
ZV_READIMAGE(Image,"test.bmp",0)     'read picture using default path
ZV_LATCH(Image,0)                    'save picture into vision latch channel 0
```

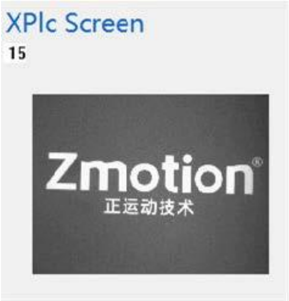
Step 3: new build the component "static picture", select back picture from the "use picture lib", then manually enter @ZV to read it.



Appearance

Use PictureLib	Back Picture
Back Picture	@ZV0
Use image size	False

Step 4: check the effect



2) Example 2:

Step 1: put the picture into the picture library.

Step 2: in picture property, select the GIF picture from Back Picture Lib.

Step 3: check the effect

4.3.10. Bit State

A. What is It?

Show format text 0 or text 1 according to register bit state 0 or 1.

B. How to Use?

Click "View - Tool Box", select "Bit State". Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed info of different states. If you want to switch the displayed state, only switching the register state.

--How to Switch Register State--

Method 1: click menu HMI – Language / State – S0/S1/S2… to switch.

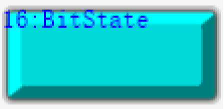
Method 2: in "output" window, enter register address and state value, and click "send", for example, enter "MODBUS_BIT(0)=1".

Method 3: in "action", set call sub, and write related register state switching program in Basic sub function, after that, press the component to call the sub to achieve switching (if no action is set, no way to achieve that directly).

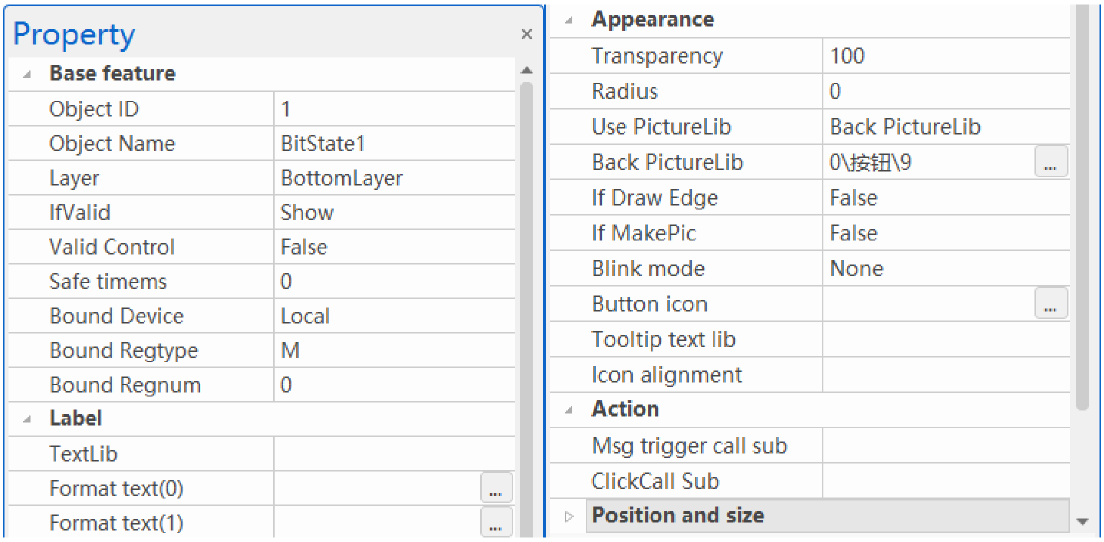
Notes:

a. When you selected bit register, register value 0 shows format text 0, register value 1 shows format text 1. When you selected register is not bit register, then register value 0 shows format text 0, register value not 1 will show format text 1.

b. If you need it to show multi-bit, please use "WordState"



C. “Property” Window

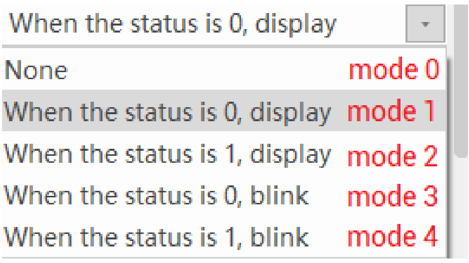


Property

Base feature	
Object ID	1
Object Name	BitState1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Label	
TextLib	
Format text(0)	
Format text(1)	
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Blink mode	None
Button icon	
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list

	● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows “Text”.
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Blink Mode	Set the component blink mode. When the status is 0, display None mode 0 When the status is 0, display mode 1 When the status is 1, display mode 2 When the status is 0, blink mode 3 When the status is 1, blink mode 4 ● Mode 0: none ● Mode 1: when the state is 0, the component will be shown bewteen state 0 format and state 1 format. ● Mode 2: when the state is 1, the component will be shown bewteen state 0 format and state 1 format. ● Mode 3: when the state is 0, the component will shrink, between

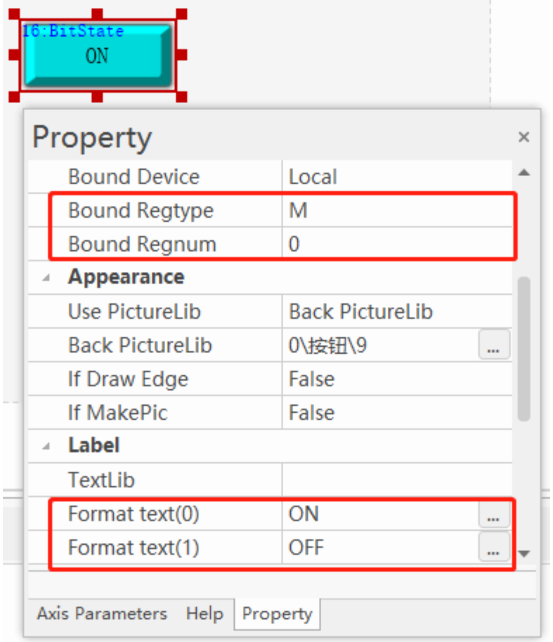
	“display” and “hide”. ● Mode 4: when the state is 1, the component will shrink, between “display” and “hide”.
Blink Time	Set the component’s shrink time, the unit is ms.
Button Icon	Select the icon from the picture library.
Icon Width	Set the icon’s width.
Icon Height	Set the icon’s height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCall Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

E. Examples

➢ Example 1: show different texts under different states of register

Step 1: select bound register type and No., M0 corresponds to MODBUS_BIT(0).

Step 2: enter corresponding texts under different states.



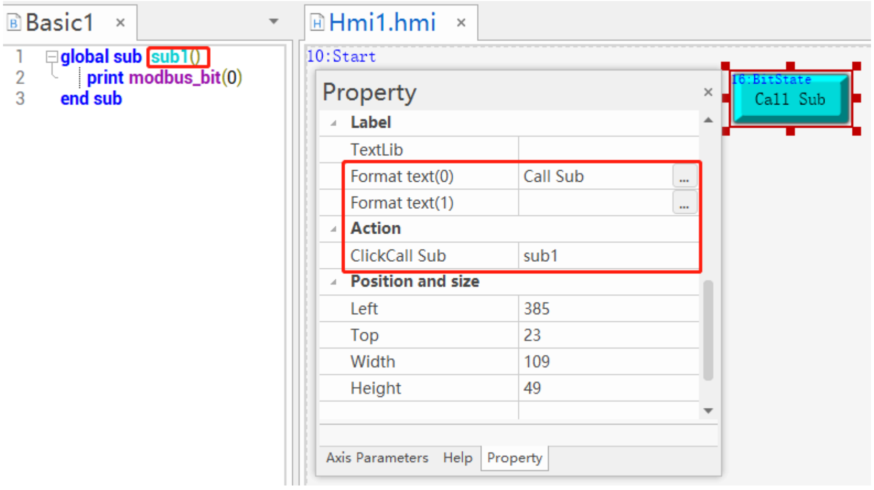
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	ON
Format text(1)	OFF

Step 3: check the effect: when MODBUS_BIT(0) = 0, it will show ON. When MODBUS_BIT(0) = 1, it will show OFF.

➢ Example 2: Call SUB Function (Click Call Sub)

Step 1: edit global SUB function that is to be called by HMI in Basic.

Step 2: select “Click CallSub” in component property, select edited SUB function name.



```
global sub sub1()
    print modbus_bit(0)
end sub
```

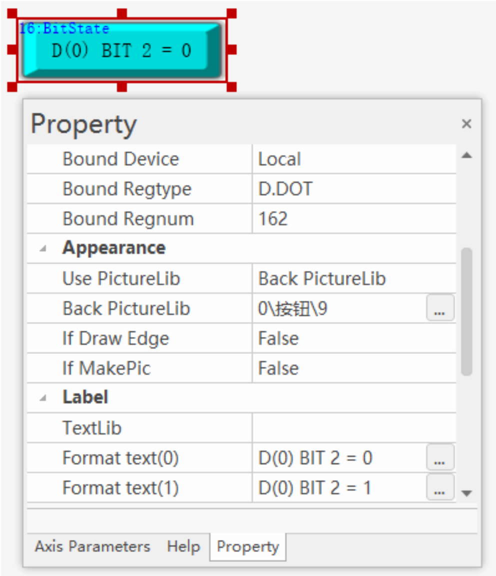
Label	
TextLib	
Format text(0)	Call Sub
Format text(1)	
Action	
ClickCall Sub	sub1
Position and size	
Left	385
Top	23
Width	109
Height	49

Step 3: check the effect: when the Call Sub function is pressed, it will call BASIC SUB function, each press, call function once.

➢ Example 3: Switch Bit

Step 1: select D.DOT for register type, register No. = reg No. * 16 + dot (0-15), 0-15 means register 0-15 bit selection (that is, you can set state value of one certain bit of MODBUS_ERG).

Step 2: enter corresponding texts under different states.



Bound Device	Local
Bound Regtype	D.DOT
Bound Regnum	162
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	D(0) BIT 2 = 0
Format text(1)	D(0) BIT 2 = 1

Step 3: check the effect: when assigned BIT 2 of D(10) is 1, the component will show "D(10) BIT 2 = 1".

4.3.11. Word State

A. What is It?

Show format text 0 or text 1 according to register bit state 0 or 1 or call SUB function.

B. How to Use?

Click "View - Tool Box", select "Word State". Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed info of different states. If you want to switch the displayed state, only switching the register state.

--How to Switch Register State--

Method 1: click menu HMI – Language / State – S0/S1/S2… to switch.

Method 2: in "output" window, enter register address and state value, and click "send", for example, enter "MODBUS_BIT(0)=1".

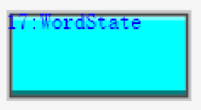
Method 3: in "action", set call sub, and write related register state switching program in Basic sub function, after that, press the component to call the sub to achieve switching (if no action is set, no way to achieve that directly).

Notes:

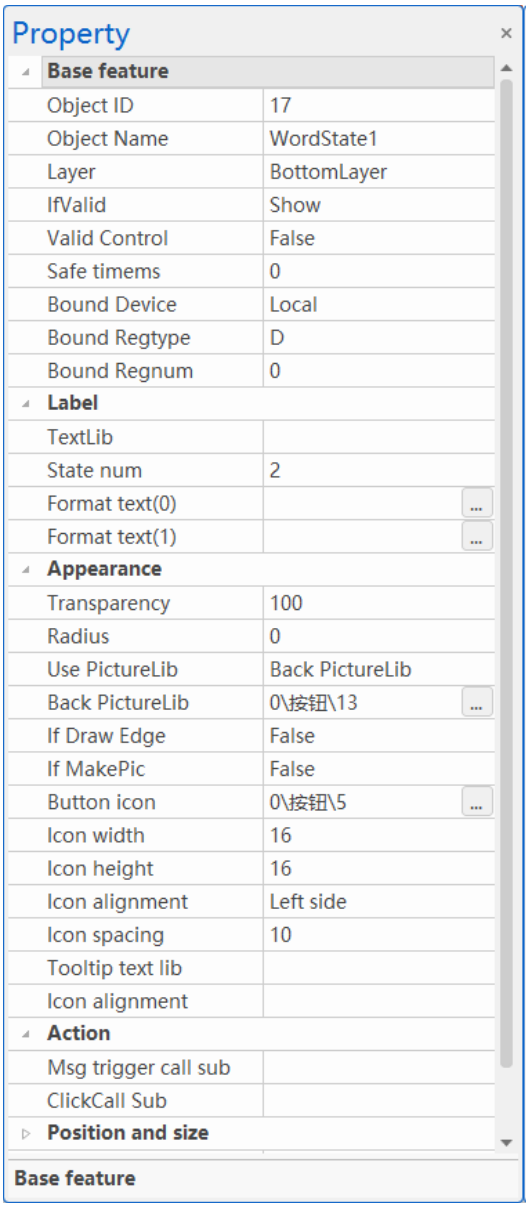
a. If no any actions are set, then it can’t switch the displayed state through pressing the component.

b. You can set how many texts by yourself in state num, the range is 1-256.

c. It is recommended to select word register, default is D0, when corresponding register value is 0, showing format text 0, when corresponding register value is 1, showing format text 1, and so on.



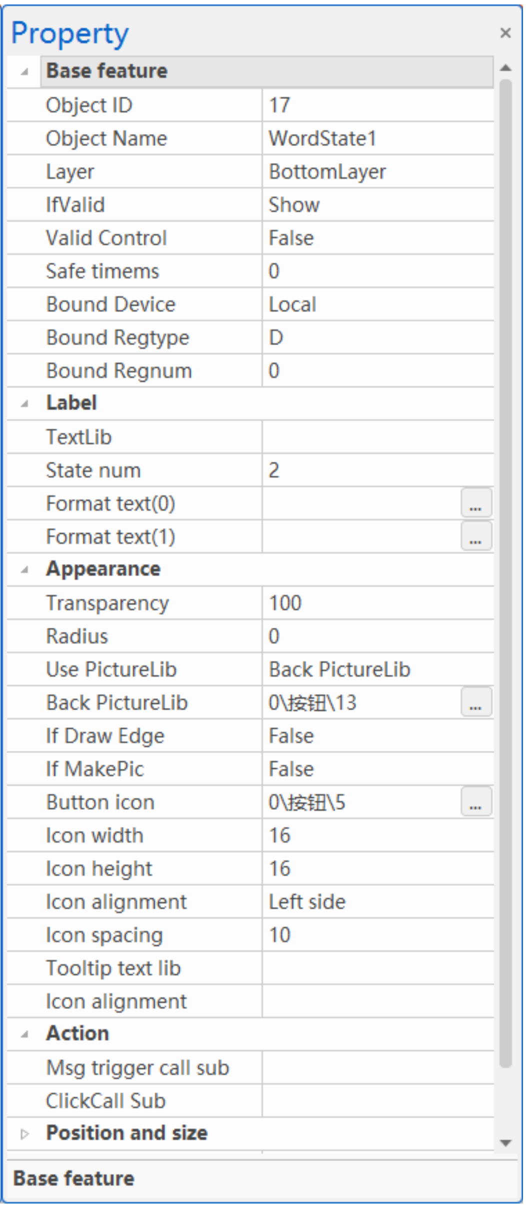
C. “Property” Window



Property ×

Base feature	
Object ID	17
Object Name	WordState1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Label	
TextLib	
State num	2
Format text(0)	...
Format text(1)	...
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\13 ...
If Draw Edge	False
If MakePic	False
Button icon	0\按钮\5 ...
Icon width	16
Icon height	16
Icon alignment	Left side
Icon spacing	10
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Base feature



Property ×

Base feature	
Object ID	17
Object Name	WordState1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Label	
TextLib	
State num	2
Format text(0)	...
Format text(1)	...
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\13 ...
If Draw Edge	False
If MakePic	False
Button icon	0\按钮\5 ...
Icon width	16
Icon height	16
Icon alignment	Left side
Icon spacing	10
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Base feature

D. **“Property” Description**

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows “Text”.
State Num	The component’s state numbers (1--256), it can show multiple states.
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc.

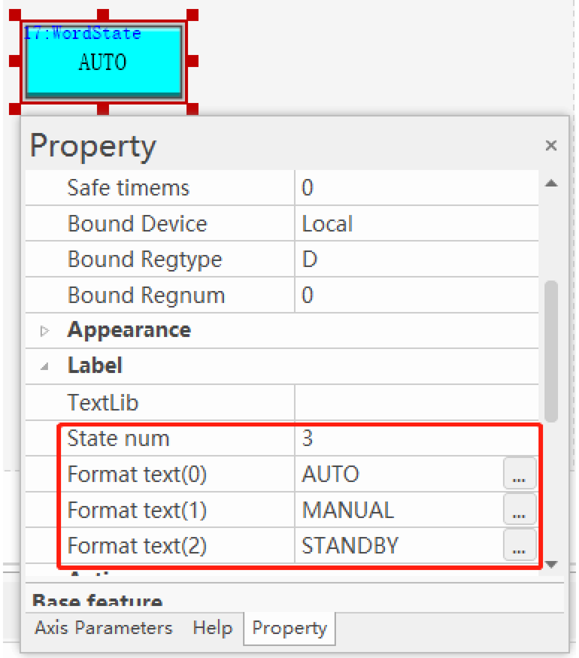
	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Button Icon	Select the icon from the picture library.
Icon Width	Set the icon’s width.
Icon Height	Set the icon’s height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCall Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

E. Examples

- **Example 1: show different texts according to different states of register**

Step 1: select bound register type and No., D0 corresponds to MODBUS_REG(0).

Step 2: enter state num and corresponding texts under different states.



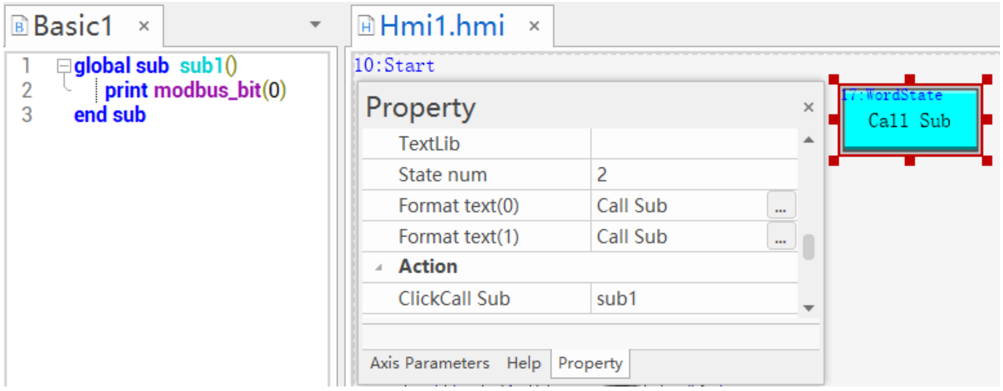
Property	Value
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
TextLib	
State num	3
Format text(0)	AUTO
Format text(1)	MANUAL
Format text(2)	STANDBY

Step 3: check the effect: when MODBUS_REG(0) = 0, it will show AUTO. When MODBUS_REG(0) = 1, it will show MANUAL, when MODBUS_REG(0) = 2, it will show STANDBY.

➢ Example 2: Call SUB Function (Click Call Sub)

Step 1: edit global SUB function that is to be called by HMI in Basic.

Step 2: select “Click CallSub” in component property, select edited SUB function name.



```
global sub sub1()
    print modbus_bit(0)
end sub
```

Property	Value
TextLib	
State num	2
Format text(0)	Call Sub
Format text(1)	Call Sub
ClickCall Sub	sub1

Step 3: check the effect: after downloading into controller / simulator, when press the object in HMI or simulation interface, it will execute SUB function.

4.3.12. Bar Code

A. What is It?

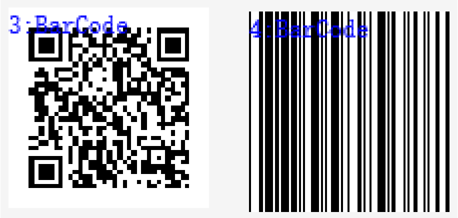
Extract the data from assigned register address, then make one corresponding QR-Code / Bar Code, then you can scan it to get data.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

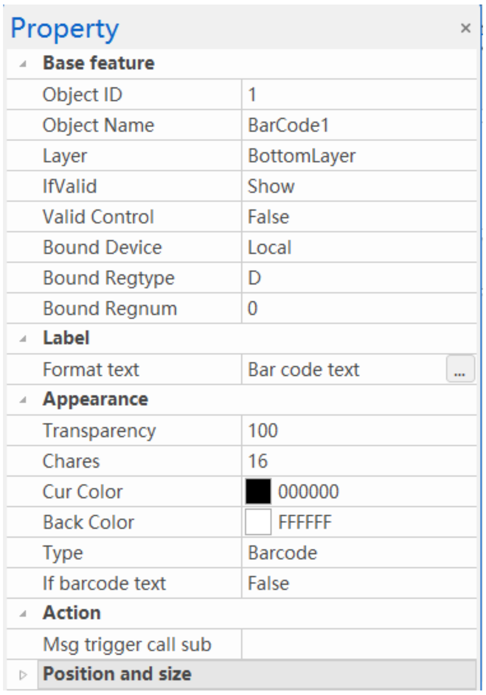
B. How to Use?

Click "View - Tool Box", select "Word State". Put the component at suitable position, then in its property window, at first, choose the type, QR-code / bar code, then set the displayed character numbers (chares), and set corresponding register contents in Basic program, bind with the register. After that, please download the program for producing the code.

Note: the character numbers should be ≥ register stored character numbers, otherwise, the information after scanned may be not complete.



C. "Property" Window



Property

Base feature	
Object ID	1
Object Name	BarCode1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Label	
Format text	Bar code text ...
Appearance	
Transparency	100
Chares	16
Cur Color	■ 000000
Back Color	□ FFFFFF
Type	Barcode
If barcode text	False
Action	
Msg trigger call sub	
Position and size	

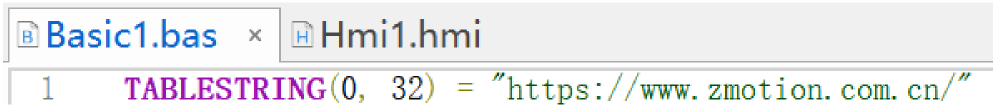
D. **“Property” Description**

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register’s variables, so it is valid when the valid regtype is @.
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Bound Reg@	select @ register’s variables, it is valid when the bound regtype is @.
Label	
Format Text	Show the text format of the bar code character string, it only can set the format, can’t modify the displayed contents.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc.

	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Chares	How many characters are shown after scanned QR code / bar code.
Cur Color	Set the QR code / bar code color, default is RGB (0,0,0)
Back Color	Set the background color, default is RGB (255, 255, 255)
Type	Select the code type, QR code / bar code When QR code is selected, you can select "correction standard". When bar code is selected, you can set the "text format"
If Barcode Text	Whether to show barcode text.
Correction Standard	Set the correction level under QR code, it uses the correction algorithm "Reed-solomon", there are 4 choices: L (7%), M (15%), Q (25%), H (30%). Higher error levels offer greater fault tolerance and resilience to data loss, but also reduce storage capacity. During encoding, data is converted to binary form and mixed with error-correcting codes to improve reading efficiency and fault tolerance. ***Note: H (30%) is default.***
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

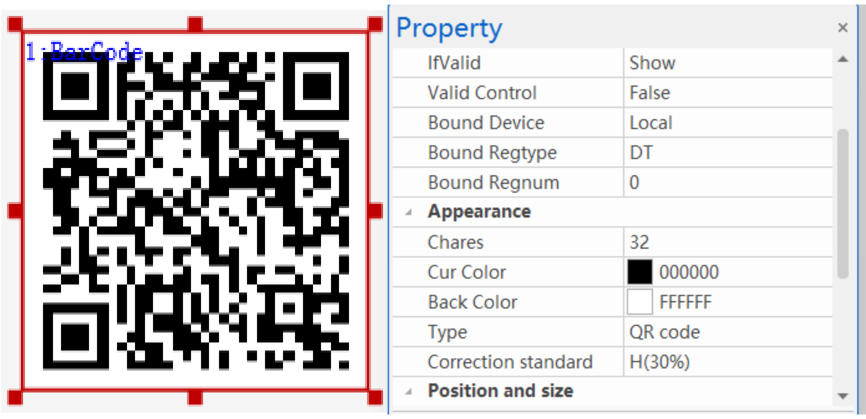
E. Examples

Step 1: in Basic program, set register stored content.



```
TABLESTRING(0, 32) = "https://www.zmotion.com.cn/"
```

Step 2: in bar code property, set the type, character numbers, and bound registers.



Property

IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	DT
Bound Regnum	0
Appearance	
Chares	32
Cur Color	■ 000000
Back Color	□ FFFFFF
Type	QR code
Correction standard	H(30%)
Position and size	

Step 3: check the effect: then you can use Wechat or others to scan it for getting register contents, because here the website is saved in the register, so it will jump to corresponding website after scanned.

4.3.13. Process Bar

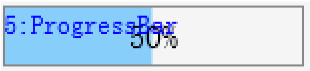
A. What is It?

Show the completion progress of the task through graphic. It can specify the register value as progress percent value (0.0~100.0), There are **bar and graphic types** of the process bar, they can be modified in “property”.

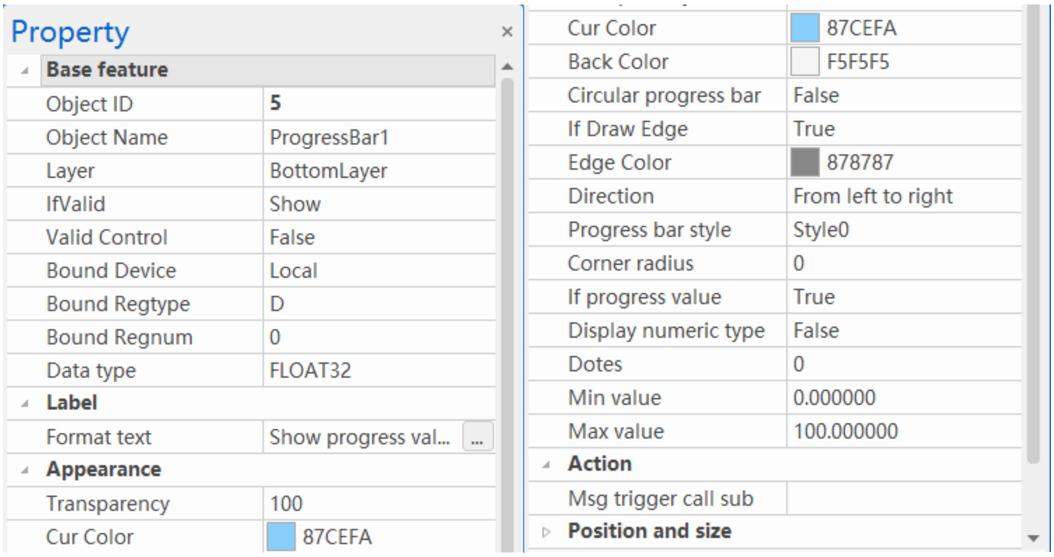
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click “View - Tool Box”, select “Process Bar”. Put the component at suitable position, then in its property window, bind with the register type, and set the register address. Also, in Basic program, write loading / decreasing function of progress bar to achieve that. Generally, use “timer” component to refresh it, please check following example.



C. “Property” Window

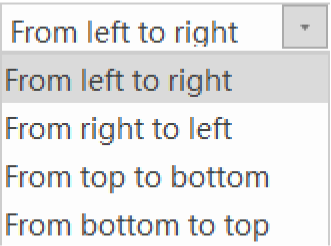


Property

Base feature	
Object ID	5
Object Name	ProgressBar1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Data type	FLOAT32
Label	
Format text	Show progress val...
Appearance	
Transparency	100
Cur Color	87CEFA
Back Color	F5F5F5
Circular progress bar	False
If Draw Edge	True
Edge Color	878787
Direction	From left to right
Progress bar style	Style0
Corner radius	0
If progress value	True
Display numeric type	False
Dotes	0
Min value	0.000000
Max value	100.000000
Action	
Msg trigger call sub	
Position and size	

D. “Property” Description

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Bound Device	Assigned device, default is local

Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Data Type	Set the progress value's data type, default is FLOAT (floating type 32-bit), generally please make it consistent with register type.
Label	
Format Text	Show the component's text.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Cur Color	Set the color for "loaded progress"
Back Color	Set the color for "not loaded progress"
Circular Progress Bar	● True / False: whether to switch current progress bar to circular progress bar. True = Yes, False = No (False). ● Direction: set the progress loading direction, clockwise or anti-clockwise. ● Outer ring width: set the progress bar's back circular width. ● Inner ring width: set the circular width of loaded progress bar. When the outer ring width is not equal to progress bar width, the smaller width will approach to inside and center.
If Draw Edge	Whether to draw the draw.
Edge Color	Set the edge color. It is valid when "If Draw Edge" = "True".
Direction	Set the loading direction, there are 4 choices.  From left to right From left to right From right to left From top to bottom From bottom to top
Progress Bar Style	Set the progress bar's loading style, style 0 and style 1.
Corner Radius	Set the progress bar's chamfer radius, the default is 0, max 33.
If Progress Value	Whether to show the progress value
Display Numeric Type	Whether to show it through numeric value.
Dotes	Set the decimal part dotes, 0~8, default is 0.

Min Value	Set the progress bar’s minimal value. It is valid when “Display Numeric Type” is True.
Max Value	Set the progress bar’s maximum value. It is valid when “Display Numeric Type” is True.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

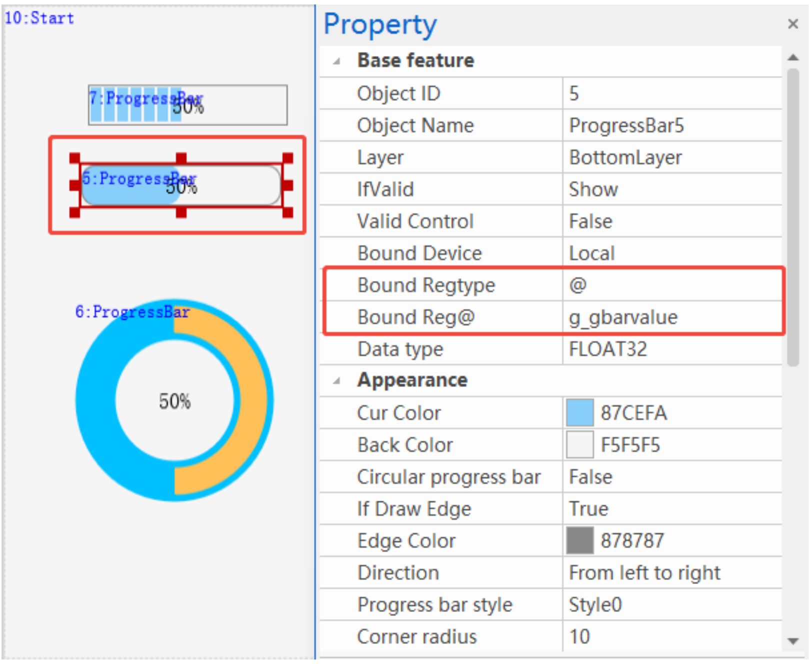
E. Example

Step 1: add “progress bar” and “timer” components, bar type or circular type.

Step 2: write Basic program.

```
GLOBAL dim g_gbarvalue      'define progress value’s variables
GLOBAL dim g_bifplus        'increasing / decreasing flag
GLOBAL dim g_bifauto        'automatic flag
g_gbarvalue = 0
g_bifplus = 1
g_bifauto = 1
end
GLOBAL sub sub_timer()
    if g_bifauto=1 then
        if g_bifplus=1 then
            g_gbarvalue = g_gbarvalue + 1
            if g_gbarvalue > 100 then 'increase to the max value, start to decrease
                g_gbarvalue = 100
                g_bifplus = 0
        endif
    else
        g_gbarvalue = g_gbarvalue - 1
        if g_gbarvalue < 0 then         'decrease to the min value, start to increase.
            g_gbarvalue = 0
            g_bifplus = 1
        endif
        endif
    endif
end sub
```

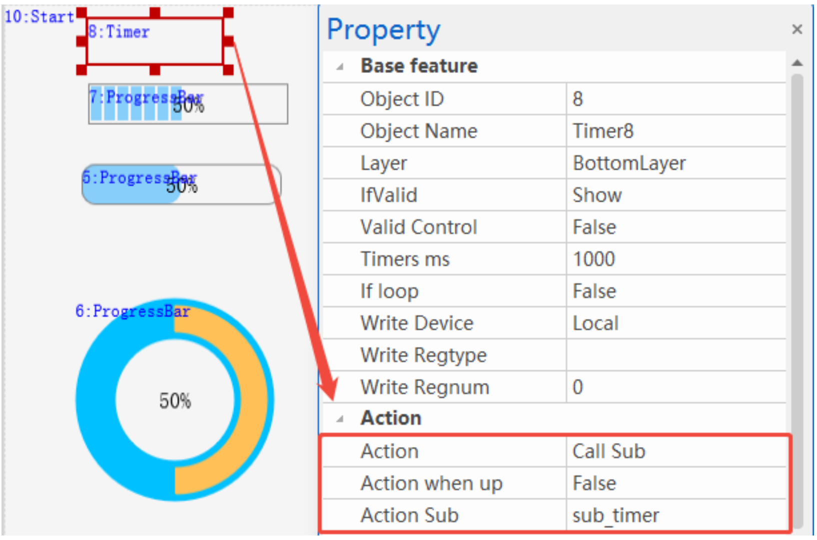
Step 3: in progress bar property, bind register / customized variables (here customized variable is “g_gbarvalue”).



Property

Base feature	
Object ID	5
Object Name	ProgressBar5
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	@
Bound Reg@	g_gbarvalue
Data type	FLOAT32
Appearance	
Cur Color	87CEFA
Back Color	F5F5F5
Circular progress bar	False
If Draw Edge	True
Edge Color	878787
Direction	From left to right
Progress bar style	Style0
Corner radius	10

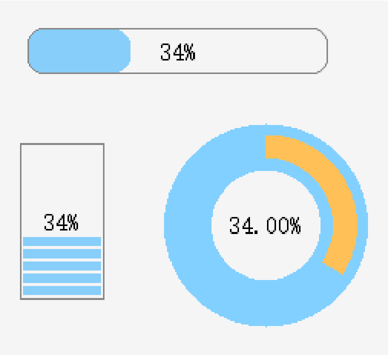
Step 4: add “timer” component, select corresponding subfunction name (here is sub_timer) for call sub.



Property

Base feature	
Object ID	8
Object Name	Timer8
Layer	BottomLayer
IfValid	Show
Valid Control	False
Timers ms	1000
If loop	False
Write Device	Local
Write Regtype	
Write Regnum	0
Action	
Action	Call Sub
Action when up	False
Action Sub	sub_timer

check the effect, after downloading into controller / simulator:



4.3.14. Event Show

A. What is It?

Used to show the event that has been defined in [event list] and has met triggering conditions. These events will be shown in order of triggering time.

It can show the event appearing date, event appearing time, event confirmation time, resume time, event information, appearing times, and the cumulative time (contents can be shown in multiple lines, which can be set in format text).

What's more, this component can be used together with HMI_EVENTHIDE and HMI_EVENTDEL commands.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

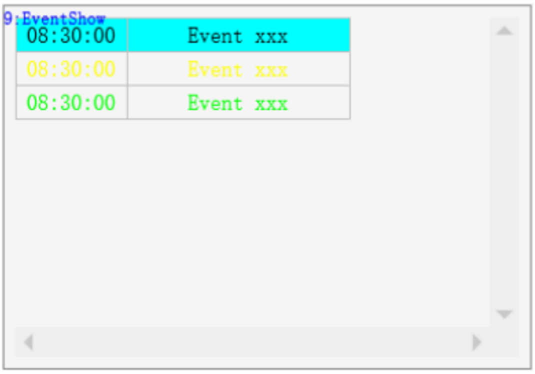
a) Add event content through clicking HMI – Event List, and configure each event content's property (you could check "Event List").

b) Add "Event Show" component by clicking "View - Tool Box – Event Box", also configure the property, like, min / max display category (used to filter event category for showing), display item settings (used to show the item sorting and character numbers), etc.

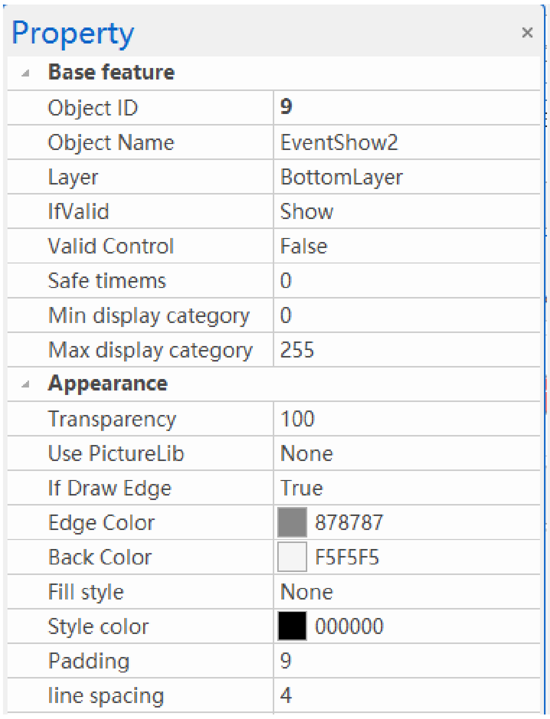
c) Set the confirmation methods of the event, there are "nonuse, double click, click" or call subfunction after the event is confirmed.

d) The event will be shown in [event show] when other components trigger the event.

C. “Property” Window

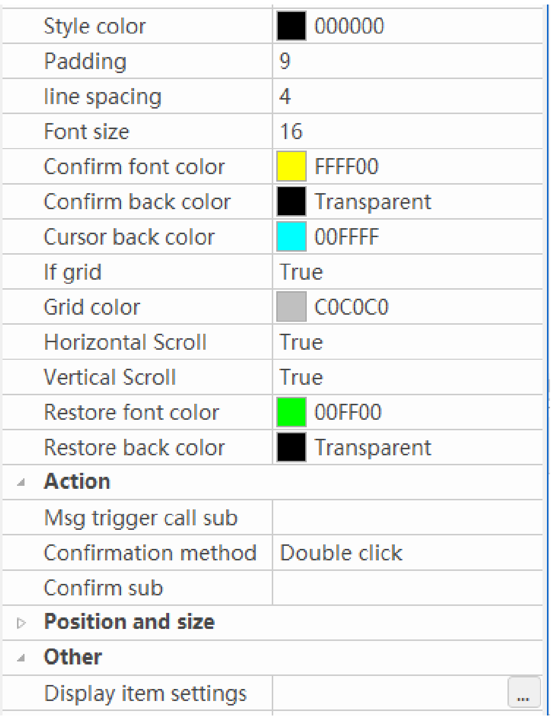


D. “Property” Description



Property

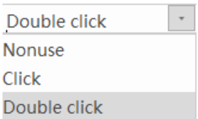
Base feature	
Object ID	9
Object Name	EventShow2
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Min display category	0
Max display category	255
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True
Edge Color	878787
Back Color	F5F5F5
Fill style	None
Style color	000000
Padding	9
line spacing	4



Style color	000000
Padding	9
line spacing	4
Font size	16
Confirm font color	FFFF00
Confirm back color	Transparent
Cursor back color	00FFFF
If grid	True
Grid color	C0C0C0
Horizontal Scroll	True
Vertical Scroll	True
Restore font color	00FF00
Restore back color	Transparent
Action	
Msg trigger call sub	
Confirmation method	Double click
Confirm sub	
Position and size	
Other	
Display item settings	...

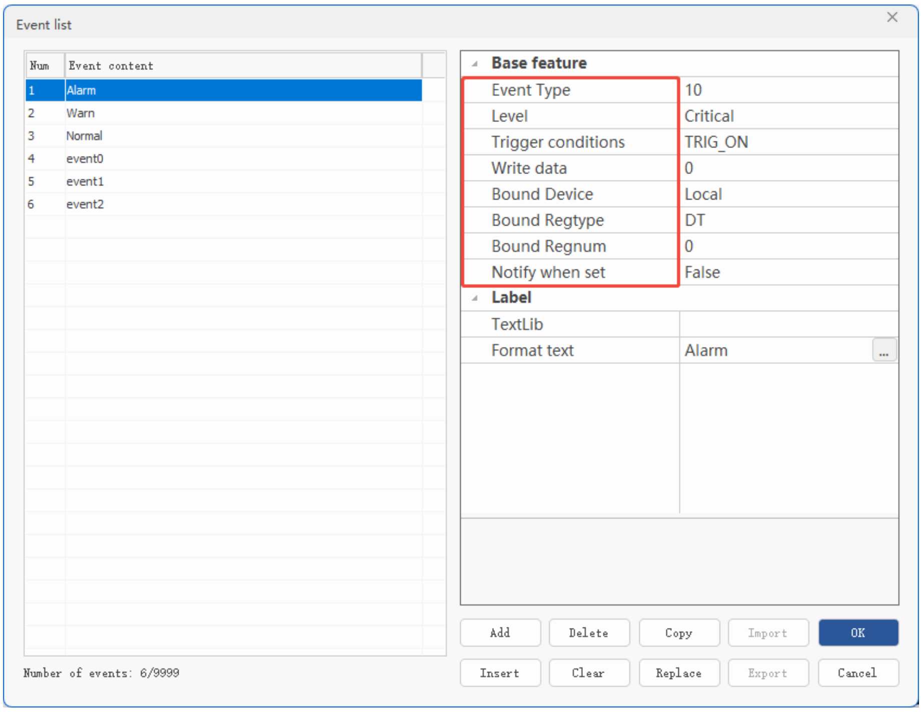
Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface.

	● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe Times	Minimal button time, the unit is ms.
Min Display Category	Set the lower limit of the range of event categories to display.
Max Display Category	Set the higher limit of the range of event categories to display.
Appearance	
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
Edge Color	Set the edge color. It is valid when "If Draw Edge" is True.
Back Color	Set the component background color.
Fill Style	Set the background filling style.
Style Color	Set the filling style's color.
Padding	Set the distance from component displayed content to the edge.
Line Spacing	Set the space for each line. 0~100.
Font Size	Set the font size.
Confirm Font Color	Set the confirmation event's font color.
Confrim Back Color	Set the confirmation event's background color.
Cursor Back Color	Set the highlight color of selected event.
If Grid	Whether to show the grid.
Grid Color	Set the grid color.
Horizonntal Scroll	Whether to show horizontal scroll bar.
Vertical Scroll	Whether to show vertical scroll bar.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is

	triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Confirmation Method	Select the event confirmation method, there are 3 methods.  Double click Nonuse Click Double click
Confirm Sub	Call sub function and transfer one parameter when the event is confirmed by the component.
Postion and Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height
Other	
Display Item Settings	Set the event sorting and displayed character numbers.

E. Example

Step 1: create the event content (can be multiple) in “event list”, and sort them according to the contents by setting the type No. and binding with register address in “event type”, and set the level, trigger conditions, etc., after that, click OK to save it. *[It can be seen here the event “alarm” binds with register TABLE (0), and the event type is 10].*



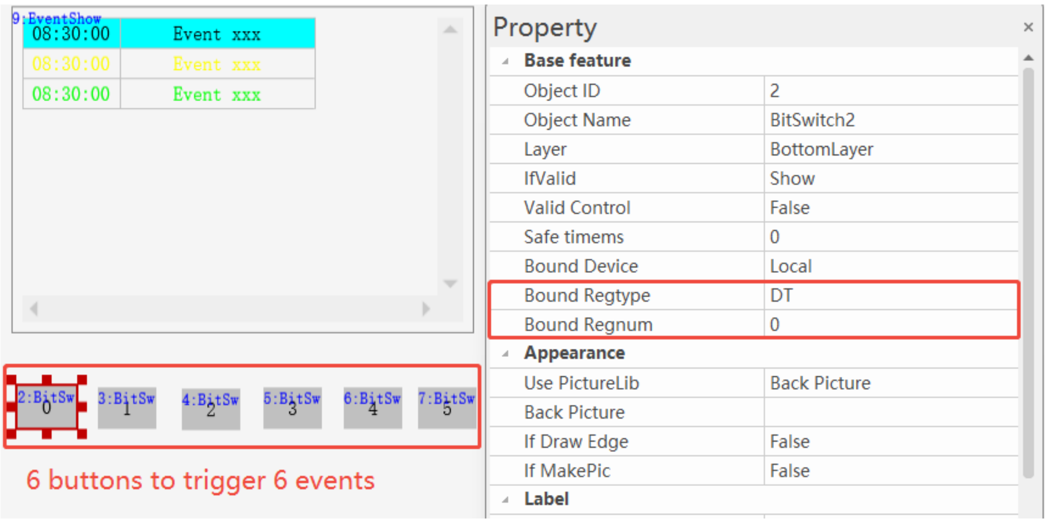
Event list

Num	Event content
1	Alarm
2	Warn
3	Normal
4	event0
5	event1
6	event2

Number of events: 6/9999

Base feature	
Event Type	10
Level	Critical
Trigger conditions	TRIG_ON
Write data	0
Bound Device	Local
Bound Regtype	DT
Bound Regnum	0
Notify when set	False
Label	
TextLib	
Format text	Alarm

Step 2: in HMI window, put the component that will trigger the event, here, take the example of "BitSwitch" (several components can be used for different events), set the action as "reverse bit" (you can set needed action), then bind with same register type and No. as the corresponding event. Like below shown, make the BitSwitch 0 bind with same register address as "Alarm" event TABLE(0), etc.



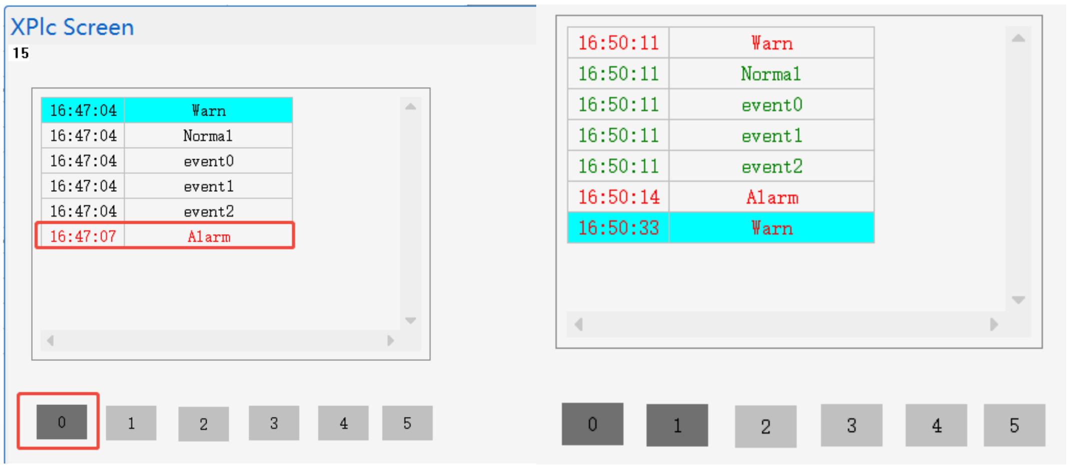
08:30:00	Event xxx
08:30:00	Event xxx
08:30:00	Event xxx

Property

Base feature	
Object ID	2
Object Name	BitSwitch2
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	DT
Bound Regnum	0
Appearance	
Use PictureLib	Back Picture
Back Picture	
If Draw Edge	False
If MakePic	False
Label	

Step 3: in HMI window, put "event show" component, and set properties, like, min / max display category, confirmation method, and so on. After that, download it into controller / simulator.

Step 4: check the effect, when clicking 0, alarm event will be triggered and shown inn "event show" component. For picture 2, all events are triggered, and among them, green font is restore event, red font is confirmed event (set by confirmation method).



16:47:04	Warn
16:47:04	Normal
16:47:04	event0
16:47:04	event1
16:47:04	event2
16:47:07	Alarm

16:50:11	Warn
16:50:11	Normal
16:50:11	event0
16:50:11	event1
16:50:11	event2
16:50:14	Alarm
16:50:33	Warn

4.3.15. Alarm Show

A. What is It?

Used to show the event that has been defined in [event list] and now the system can trigger it, then the event will be shown in order of triggering time.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

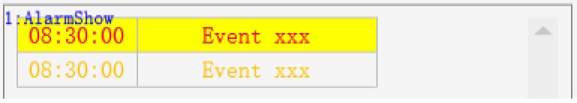
B. How to Use?

a) Add event content through clicking HMI – Event List, and configure each event content's property (you could check "Event List").

b) Add "Alarm Show" component by clicking "View - Tool Box – Alarm Show", also configure the property, like, min / max display category (used to filter event category for showing), display item settings (used to show the item sorting and character numbers), etc.

c) Set the confirmation methods of the event, there are "nonuse, double click, click" or call subfunction after the event is confirmed.

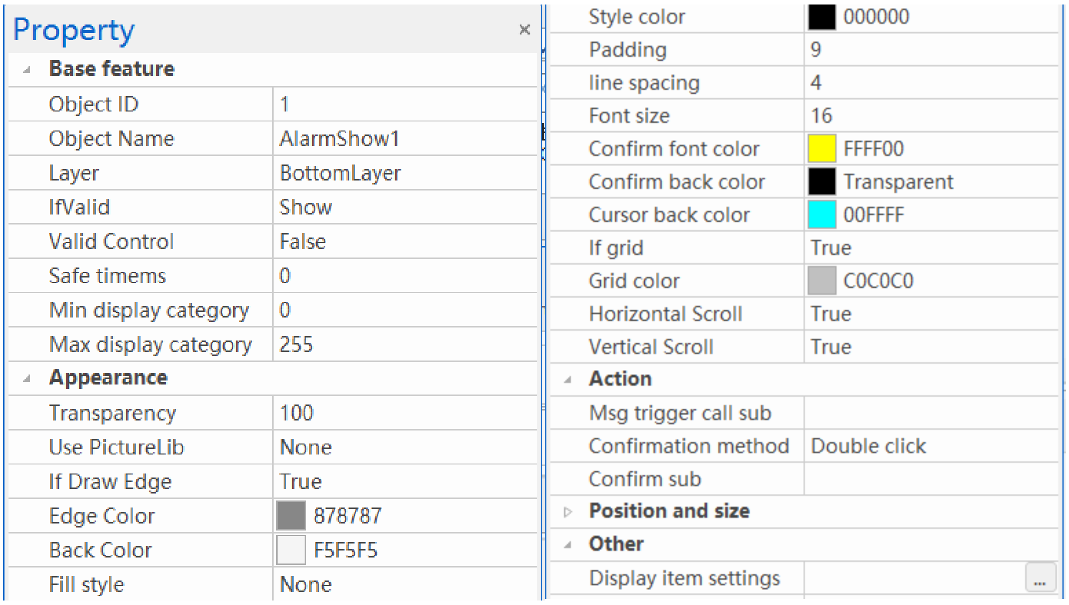
d) The event will be shown in [alarm show] when other components trigger the event.



1:AlarmShow

08:30:00	Event xxx
08:30:00	Event xxx

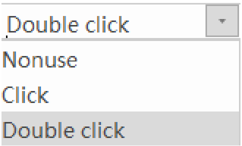
C. "Property" Window



Property

Base feature	
Object ID	1
Object Name	AlarmShow1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Min display category	0
Max display category	255
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True
Edge Color	878787
Back Color	F5F5F5
Fill style	None
Style color	000000
Padding	9
line spacing	4
Font size	16
Confirm font color	FFFF00
Confirm back color	Transparent
Cursor back color	00FFFF
If grid	True
Grid color	C0C0C0
Horizontal Scroll	True
Vertical Scroll	True
Action	
Msg trigger call sub	
Confirmation method	Double click
Confirm sub	
Position and size	
Other	
Display item settings	...

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe Times	Minimal button time, the unit is ms.
Min Display Category	Set the lower limit of the range of event categories to display.
Max Display Category	Set the higher limit of the range of event categories to display.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
Edge Color	Set the edge color. It is valid when "If Draw Edge" is True.
Back Color	Set the component background color.

Fill Style	Set the background filling style.
Style Color	Set the filling style's color.
Padding	Set the distance from component displayed content to the edge.
Line Spacing	Set the space for each line. 0~100.
Font Size	Set the font size.
Confirm Font Color	Set the confirmation event's font color.
Confrim Back Color	Set the confirmation event's background color.
Cursor Back Color	Set the highlight color of selected event.
If Grid	Whether to show the grid.
Grid Color	Set the grid color.
Horizonntal Scroll	Whether to show horizontal scroll bar.
Vertical Scroll	Whether to show vertical scroll bar.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Confirmation Method	Select the event confirmation method, there are 3 methods. 
Confirm Sub	Call sub function and transfer one parameter when the event is confirmed by the component.
Postion and Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height
Other	
Display Item Settings	Set the event sorting and displayed character numbers.

D. Example: please check "Event Show" example, same usage,

4.3.16. Alarm Bar

A. What is It?

Used to show the event that has been defined in [event list] and now the system can trigger it, the event is called “warn”, then these warn contents will be shown in component of alarm bar with the format of “single-line scrolling marquee” according to triggered time.

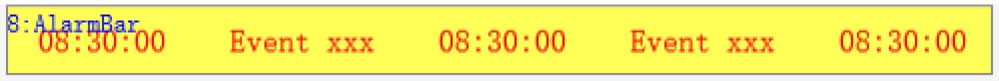
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

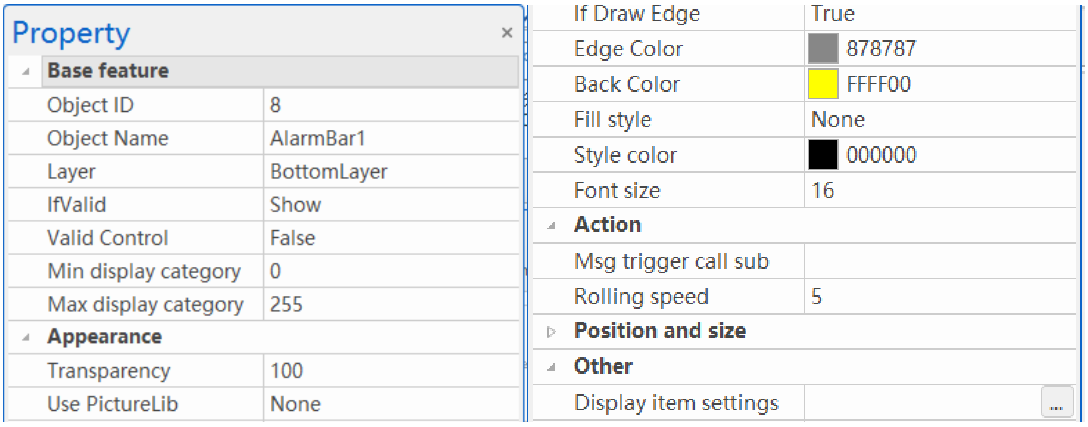
a) Add event content through clicking HMI – Event List, and configure each event content’s property (you could check “Event List”).

b) Add “Alarm Bar” component by clicking “View - Tool Box – Alarm Bar”, also configure the property, like, min / max display category (used to filter event category for showing), display item settings (used to show the item sorting and character numbers), etc.

c) The event will be shown in [alarm bar] when other components trigger the event.



C. “Property” Description


Property

Base feature	
Object ID	8
Object Name	AlarmBar1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Min display category	0
Max display category	255
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 878787
Back Color	🟨 FFFF00
Fill style	None
Style color	■ 000000
Font size	16
Action	
Msg trigger call sub	
Rolling speed	5
Position and size	
Other	
Display item settings	...

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.

Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe Times	Minimal button time, the unit is ms.
Min Display Category	Set the lower limit of the range of event categories to display.
Max Display Category	Set the higher limit of the range of event categories to display.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
Edge Color	Set the edge color. It is valid when "If Draw Edge" is True.
Back Color	Set the component background color.
Fill Style	Set the background filling style.
Style Color	Set the filling style's color.
Font Size	Set the font size.

Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Rolling Speed	Set the alarm content scrolling showing speed.
Postion and Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height
Other	
Display Item Settings	Set the event sorting and displayed character numbers.

D. Example: please check "Event Show" example, same usage,

4.3.17. Message Out

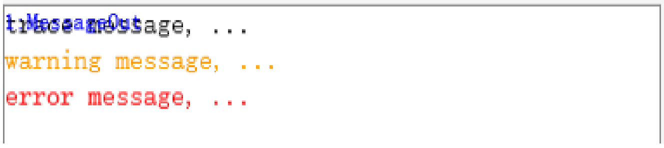
A. What is It?

Used to output controller log information, error, warn, etc., which is convenient to debug HMI. When the message exceeds the max characters, old messages will be cleared automatically.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

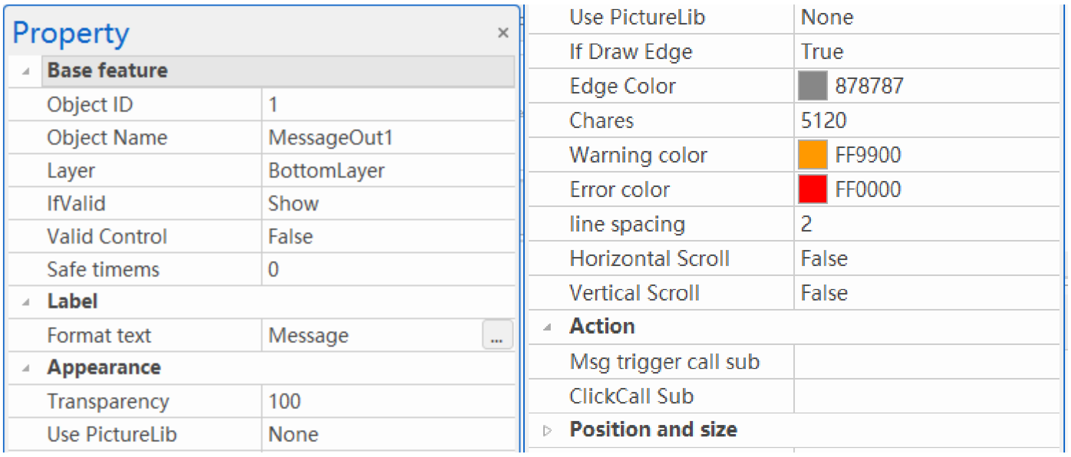
B. How to Use?

Click "View - Tool Box", select "Message Out". Put the component at suitable position, then in its property window, set needed properties. Also, in Basic program, call command "HMI_MSGOUT" to output contents.



```
trace message, ...
warning message, ...
error message, ...
```

C. “Property” Description



Property

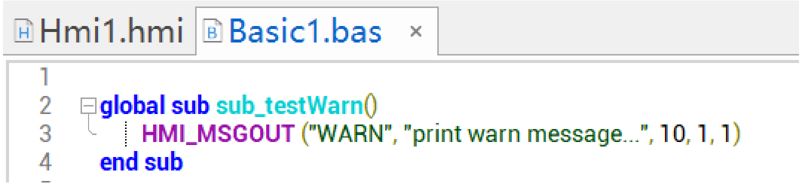
Base feature	
Object ID	1
Object Name	MessageOut1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Label	
Format text	Message
Appearance	
Transparency	100
Use PictureLib	None
Use PictureLib	None
If Draw Edge	True
Edge Color	878787
Chares	5120
Warning color	FF9900
Error color	FF0000
line spacing	2
Horizontal Scroll	False
Vertical Scroll	False
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid @reg: select @ register corresponding variables.
Safe Times	Minimal button time, the unit is ms.
Label	
Format Text	Set the message font format

Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
Edge Color	Set the edge color. It is valid when "If Draw Edge" is True.
Chares	Set the max characters. 1~65535, default is 65535.
Warning Color	Set the warning information color, default is RGB (255,153,0)
Error Color	Set the error information color, default is RGB (255,0,0)
Line Spacing	Set the space for each message.
Horizonntal Scroll	Whether to show horizontal scroll bar. Default is True.
Vertical Scroll	Whether to show vertical scroll bar. Default is True.
Label	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCallSub	Set the Basic subfunction, when clicking, it will call this sub.
Postion and Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

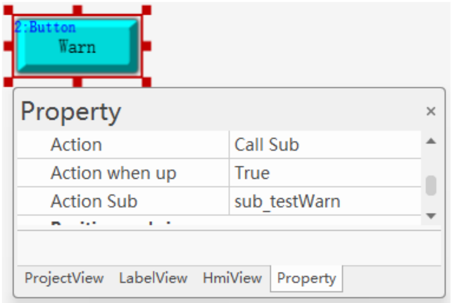
D. Example:

Step 1: write corresponding subfunction in Basic file, and use HMI_MSGOUT command to call message output type and contents.



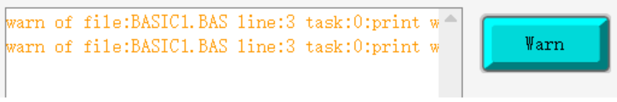
```
global sub sub_testWarn()
    HMI_MSGOUT ("WARN", "print warn message...", 10, 1, 1)
end sub
```

Step 2: add one button, and bind with the subfunction. When the button is pressed, message will be output.



Action: Call Sub
Action when up: True
Action Sub: sub_testWarn

Step 3: check the effect.



```
warn of file:BASIC1.BAS line:3 task:0:print w
warn of file:BASIC1.BAS line:3 task:0:print w
```

4.3.18. LayDate

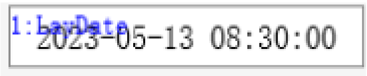
A. What is It?

Used to select the date and the time, it can be any time, then output into assigned register / variable.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

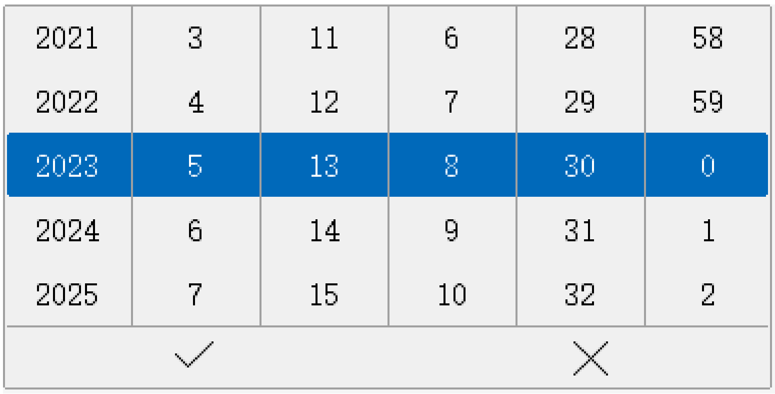
B. How to Use?

Click "View - Tool Box", select "LayDate". Put the component at suitable position, then in its property window, set needed properties, and bind with related register. After downloading, it will pop up one date and time selection window after clicking, then you could set as needed. The date and time will be shown in specified register / variable.



1:LayDate 2023-05-13 08:30:00

(display state)



2021	3	11	6	28	58
2022	4	12	7	29	59
2023	5	13	8	30	0
2024	6	14	9	31	1
2025	7	15	10	32	2

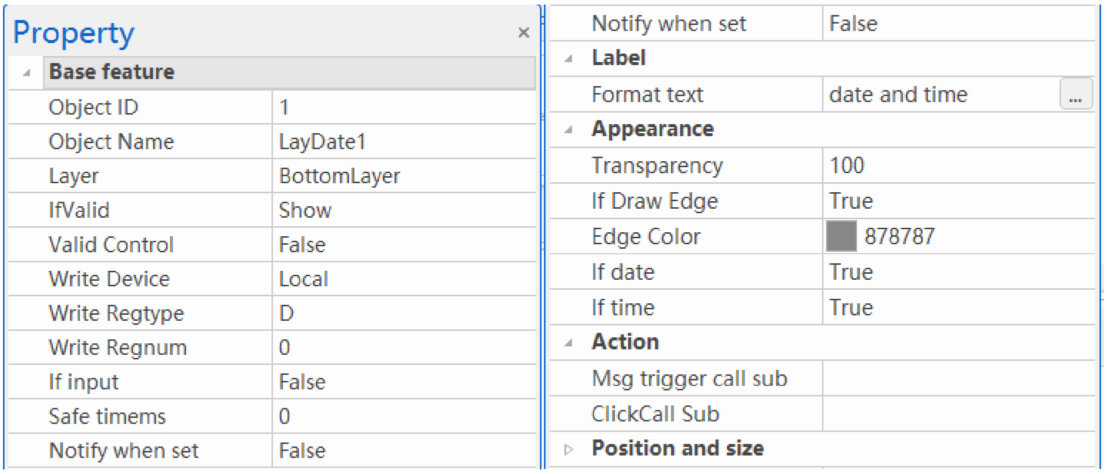
(pop-up selection state)

If you select the date and the time synchronously, then it will be output to register addr0~addr5, that is, addr0 saves the year value, addr5 saves the second value.

If you only need to show the date, then year / month / day will be output in addr0 ~ addr2 in order.

If you only need to show the time, then hour / min / second will be output in addr0 ~ addr2 in order.

C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	LayDate1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Write Device	Local
Write Regtype	D
Write Regnum	0
If input	False
Safe timems	0
Notify when set	False

Notify when set	False
Label	
Format text	date and time
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	■ 878787
If date	True
If time	True
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

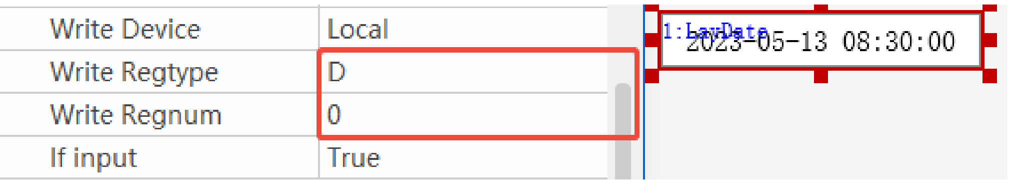
Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.

Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, not 0 = show ● Valid @reg: select @ register corresponding variables.
If Input	If it can input and show the date and the time from bound register. Default is False. True means allow to input.
Safe Times	Minimal button time, the unit is ms.
Notify when Set	Send Bit message after modified, ON / OFF. Default is True. ● Device: default is local ● regtype: select from the list ● regnum: select the register No., obtain different values from the register to control component’s different states. ● @reg: select @ register corresponding variables.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the draw
Edge Color	Set the edge color. It is valid when “If Draw Edge” is True.
If Date	Show date? Default is True.
If Time	Show time? Default is True.

Label	
Format Text	Set the time and date font format.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCallSub	Set the Basic subfunction, when clicking, it will call this sub.
Postion and Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. Example:

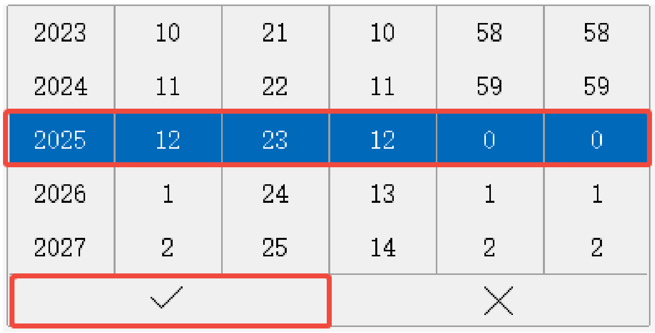
Step 1: set the register and No., then new date and time will be saved into D(0) – D(5) in order.



Write Device	Local
Write Regtype	D
Write Regnum	0
If input	True

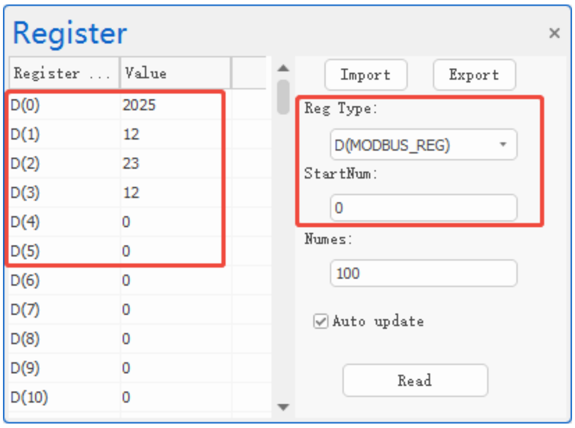
1:LayDate 2023-05-13 08:30:00

Step 2: after downloading into controller / program, select the date as 2025.12.23, select the time as 12:00:00. And click "√"



2023	10	21	10	58	58
2024	11	22	11	59	59
2025	12	23	12	0	0
2026	1	24	13	1	1
2027	2	25	14	2	2

Step 3: the new date and the time will be saved into related register. You could check from register page by clicking "Tool" – "register".



Register

Register ...	Value
D(0)	2025
D(1)	12
D(2)	23
D(3)	12
D(4)	0
D(5)	0
D(6)	0
D(7)	0
D(8)	0
D(9)	0
D(10)	0

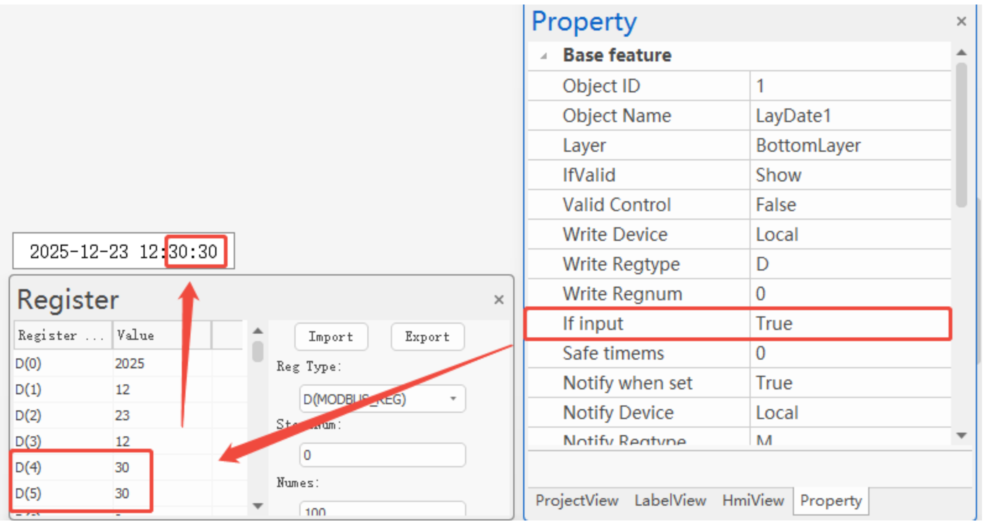
Reg Type: D(MODBUS_REG)

StartNum: 0

Numes: 100

☑ Auto update

Step 4: if the “if input” is True, you also can modify it in register to synchronize to the laydate component.



2025-12-23 12:30:30

Register

Register ...	Value
D(0)	2025
D(1)	12
D(2)	23
D(3)	12
D(4)	30
D(5)	30

Reg Type: D(MODBUS_REG)

StartNum: 0

Numes: 100

Property

Base feature

Object ID	1
Object Name	LayDate1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Write Device	Local
Write Regtype	D
Write Regnum	0
If input	True
Safe timems	0
Notify when set	True
Notify Device	Local
Notify Regtype	M

4.3.19. Animation

A. What is It?

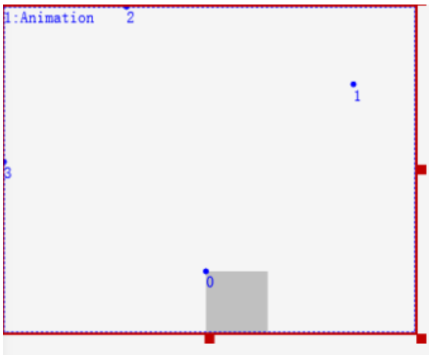
Used to preset the component movement trajectory, and control the component state and component position in trajectory through changing the register data. The “animation” component is controlled by 2 continuous registers’ data, the first one is the state, the second is the position.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

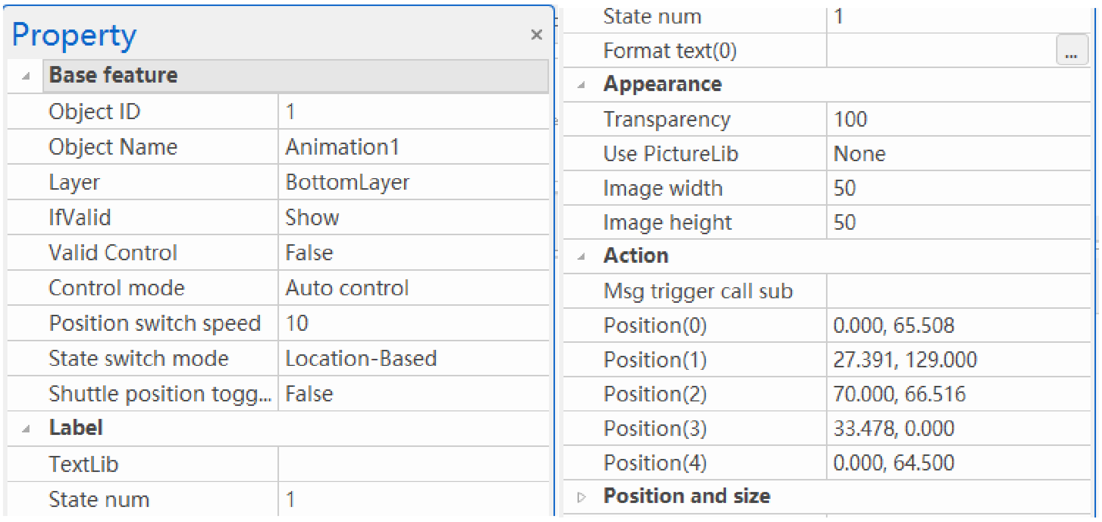
B. How to Use?

Click "View - Tool Box", select "Animation". While creating, left-click to confirm one position. And one click corresponds to one position, when clicking number is more than 2, the right-click to complete (<2 means building failed), and each animation point position will be shown in property – action.

Note: the width and height range = component range + coordinate points range, it is calculated automatically, no way to change.



C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	Animation1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Control mode	Auto control
Position switch speed	10
State switch mode	Location-Based
Shuttle position togg...	False
Label	
TextLib	
State num	1
State num	1
Format text(0)	
Appearance	
Transparency	100
Use PictureLib	None
Image width	50
Image height	50
Action	
Msg trigger call sub	
Position(0)	0.000, 65.508
Position(1)	27.391, 129.000
Position(2)	70.000, 66.516
Position(3)	33.478, 0.000
Position(4)	0.000, 64.500
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and

	covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, not 0 = show
Bound Device	Assign the device, default is local
Bound Regtype	Select the register type from the list..
Bound Regnum	select the register No.
Control Mode	Set the animation's image state and position control modes. ● Auto-Control: automatically and regularly switch the animation state and position. ● Register Control: it is only valid in "index register". Address 0: bind with the state No., to switch the animation image to the assigned state's image. Address 1: bind with the position No., to make the state graphic to move at the assigend position.
Postion Switch Speed	Set the switching speed between each position, 0~255, the unit is 100ms, and the value is smaller, the switching speed is faster.
State Switch Mode	Set the state switching method. ● Location-Based: the state is switched according to position. ● Time-Based: the state is switched according to time.
Shuttle Position Toggle	Whether to move to starting position with reverse path after the component moved at the last position.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc.

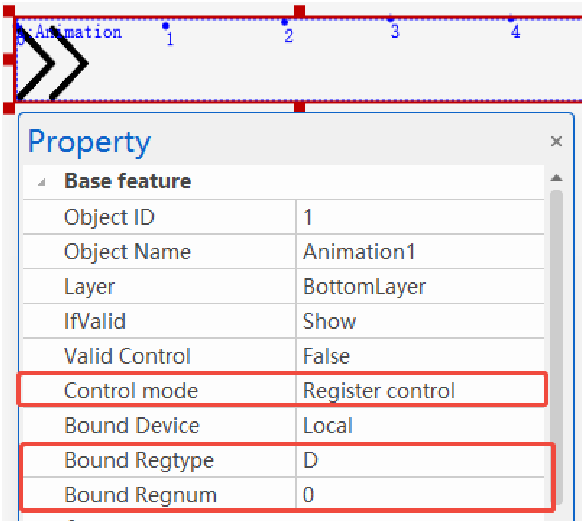
	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
Image Width	Set the animation image width.
Image Height	Set the animation image height.
Label	
TextLib	The name of text library. If not set, format text will be shown.
State Num	How many states for the component (1-256), can be multiple.
Format Text (0)	Set the text format, the numbers depends on state numbers.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position (0) / (1) / …	The position coordinates of the animation motion. While adding the component, please left-click to set the animation point, right-click to finish it.
Postion and Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. Example:

Step 1: add the needed graphic from picture library, and set states.

Step 2: in HMI file, select "animation" component from "View -- Tool Box", add it and generate one animation image (left-click to start, right-click to finish).

Step 3: set the property, select the image, set the control mode as register control. 2 registers will be used, 1st controls the state, 2nd controls the position, which means the register D(0) is for state, D(1) is for position.



Property

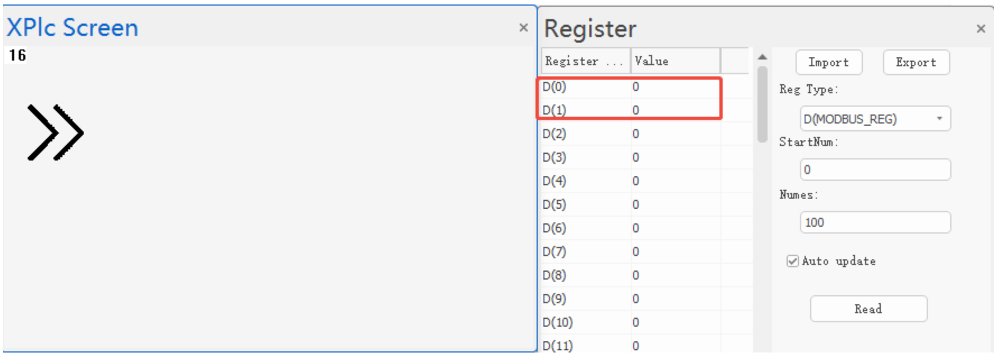
Base feature

Object ID	1
Object Name	Animation1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Control mode	Register control
Bound Device	Local
Bound Regtype	D
Bound Regnum	0

Step 4: after downloading the program, in register window, image state and position can be set by operating corresponding register.

Step 5: check the effect.

✓ Set the register D0 as 0, set register D1 as position 1.


XPlc Screen

16

Register

Register ...	Value
D(0)	0
D(1)	0
D(2)	0
D(3)	0
D(4)	0
D(5)	0
D(6)	0
D(7)	0
D(8)	0
D(9)	0
D(10)	0
D(11)	0

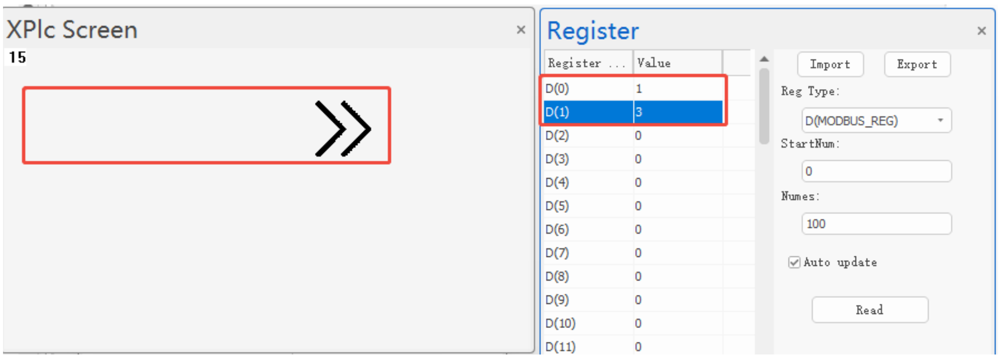
Reg Type: D(MODBUS_REG)

StartNum: 0

Numes: 100

☑ Auto update

✓ Set the register D0 as 1, set register D1 as position 4.


XPlc Screen

15

Register

Register ...	Value
D(0)	1
D(1)	3
D(2)	0
D(3)	0
D(4)	0
D(5)	0
D(6)	0
D(7)	0
D(8)	0
D(9)	0
D(10)	0
D(11)	0

Reg Type: D(MODBUS_REG)

StartNum: 0

Numes: 100

☑ Auto update

4.3.20. Flow Block

A. What is It?

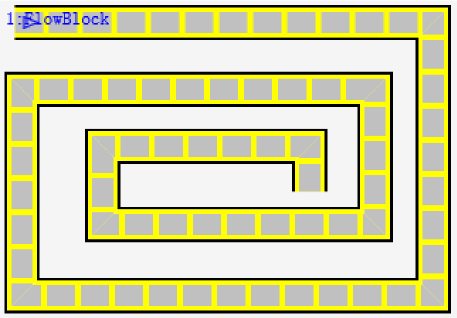
Used to preset the component movement trajectory, the corners of this component are right angles, and it exhibits a dynamic flowing effect after downloading.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

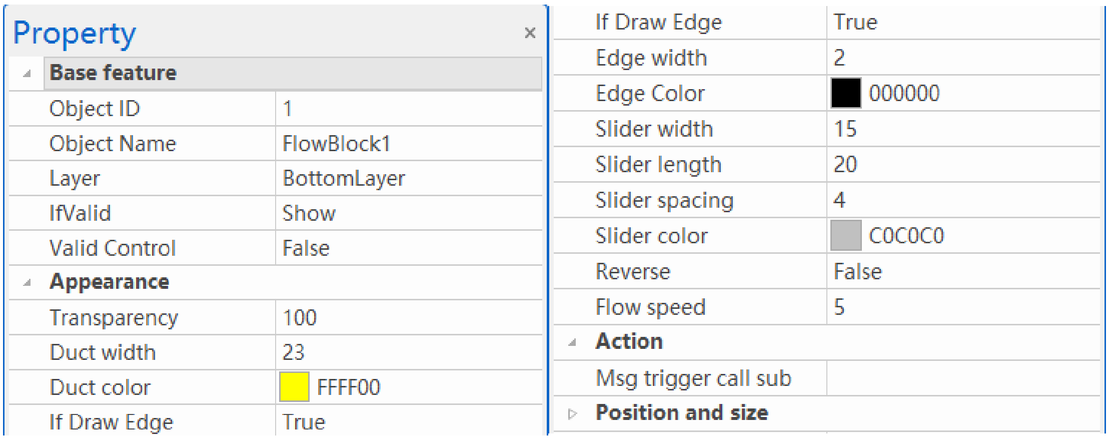
B. How to Use?

Click “View - Tool Box”, select “Flow Block” to create, at first, move the moue at needed position, now the mouse becomes “+” symbol, left-click to confirm one position. When clicking number is more than 2, the right-click to complete (<2 means building failed). The max allowable point is 256, therefore, when it reaches 256, it will automatically end.

Note: the size is confirmed while creating the flow block component, no way to change it.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	FlowBlock1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Duct width	23
Duct color	🟨 FFFF00
If Draw Edge	True
If Draw Edge	True
Edge width	2
Edge Color	■ 000000
Slider width	15
Slider length	20
Slider spacing	4
Slider color	▨ C0C0C0
Reverse	False
Flow speed	5
Action	
Msg trigger call sub	
Position and size	

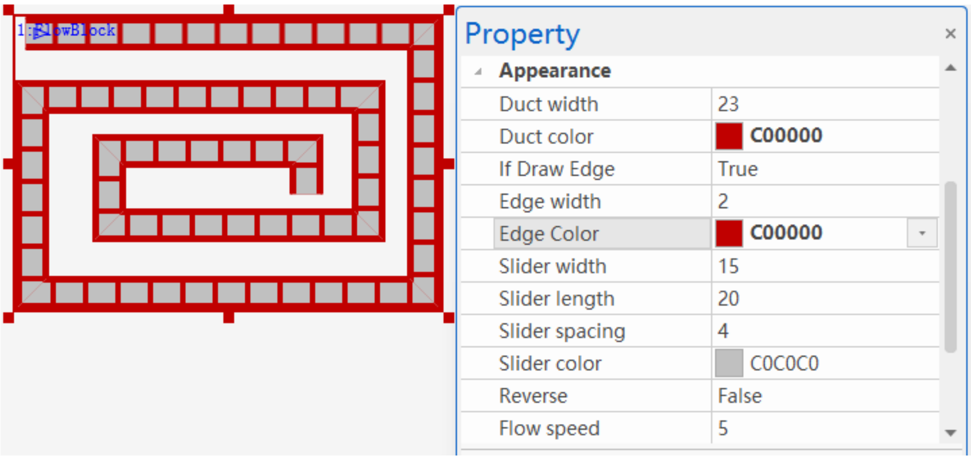
Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, not 0 = show ● Valid @reg: select @ register corresponding variables.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Duct Width	Set the duct width, 3~99 (max range depends on the flow block position), and it only can be one odd value, default is 23.
Duct Color	Set the duct color, default is RGB (255, 255, 255).
If Draw Edge	Whether to draw the edge for duct, when it is TRUE, you could set detailed properties: ● Edge Width: set the edge width, 1~20, default is 2. ● Edge Color: set the edge color, default is RGB (0, 0, 0).
Slider Width	Set the slider block width, 1~97 (max range depends on the flow block position), it only can be one odd value and smaller than the

	duct width, default is 15.
Slider Length	Set each slider block’s length, 2~255, default is 30.
Slider Spacing	Set the space between each slider block, 1~255, default is 10.
Slider Color	Set the slider color, default is RGB (192, 192, 192).
Reverse	Whther to invert the slider direction, default is False.
Flow Speed	Set the slider flowing speed, 0~25, 0 means not to flow, default is 5. When it flows each “x” pixel, flowing speed 1~25 corresponds to one slide every 1000ms to 20ms.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Postion and Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. Example:

Step 1: in HMI file, select “flow block” component from “View -- Tool Box”, add it and generate one flow block (left-click to start, right-click to finish).

Step 2: set the appearance in “property” window.



Appearance

Duct width	23
Duct color	🟥 C00000
If Draw Edge	True
Edge width	2
Edge Color	🟥 C00000
Slider width	15
Slider length	20
Slider spacing	4
Slider color	▨ C0C0C0
Reverse	False
Flow speed	5

Step 3: downloading the program to check the effect.

4.3.21. Bit Modify

A. What is It?

Set bit register address value according to component action, for bit state component, it only can show two states. It shows format text 0 firstly by default, after pressed, it will show text 1.

B. How to Use?

Click "Tool Box", select "Bit Modify" from "Input / Toggle / Button". Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed info of different states. And set "action" type. It will set bit register value according to action type. It also supports "Click CallSub" method.

--How to Check Register Value--

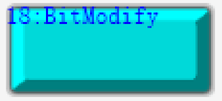
Method 1: click menu Tool – Register, select bound register type and address, then click read.

Method 2: in "output" window, enter print + bound register address and state value, and click "send", for example, enter "print MODBUS_BIT(0) / ? MODBUS_BIT(0)".

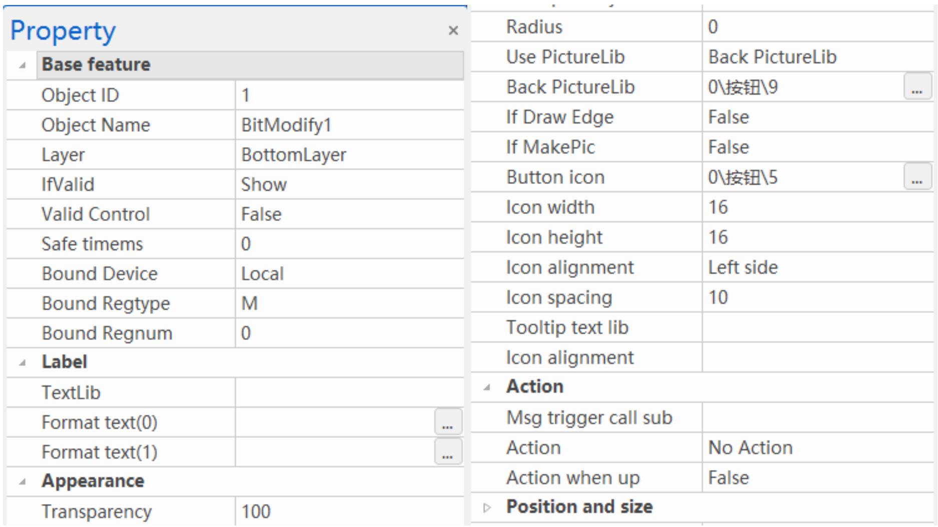
Notes:

a. It can't switch showing state through register value.

b. When the component shows one state, which may not correspond to register value state.



C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	BitModify1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Label	
TextLib	
Format text(0)	
Format text(1)	
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Button icon	0\按钮\5
Icon width	16
Icon height	16
Icon alignment	Left side
Icon spacing	10
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
Action	No Action
Action when up	False
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows "Text".
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show

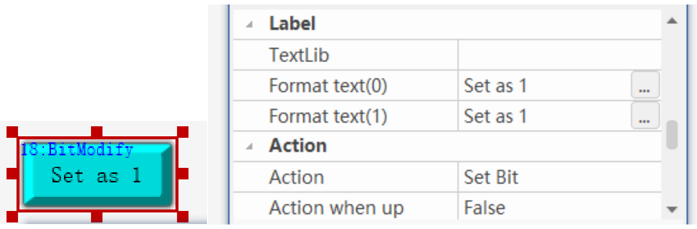
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Button Icon	Select the icon from the picture library.
Icon Width	Set the icon's width.
Icon Height	Set the icon's height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Action	The action to be executed.
Action When Up	The action when pressed or released, default is False – action when pressed, True – action when released.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. Examples

➢ Example 1: assign value 1 for register

Step 1: select register type and No.

Step 2: for "action", select "Set Bit" (set as 1).



Label	
TextLib	
Format text(0)	Set as 1
Format text(1)	Set as 1
Action	
Action	Set Bit
Action when up	False

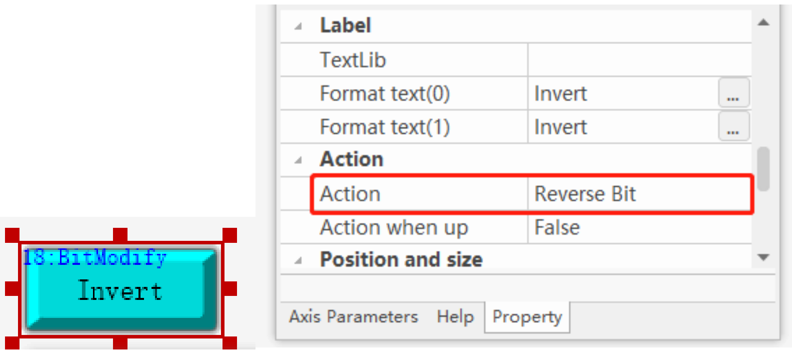
Step 3: check the effect: when set register value as 1 (that is, MODBUS_BIT(0) = 1), at this time, the object shows state 1, when released, object shows state 0, but the register value is still 1. If you selected “True” for “Action when up”, that is, after the object pressed and released, MODBUS_BIT(0)=1, MODBUS_BIT(0) value keeps value 1.

(if you select “Reset bit” for action, that is, set register as 0, which is inversed to “Set Bit”.)

- **Example 2: Invert register value**

Step 1: select register type and No.

Step 2: select “reverse bit” for “action”.



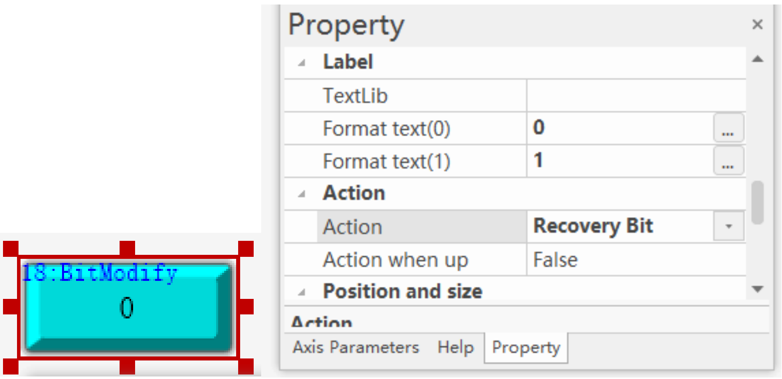
Label	
TextLib	
Format text(0)	Invert
Format text(1)	Invert
Action	
Action	Reverse Bit
Action when up	False
Position and size	

Step 3: check the effect: if initial value of MODBUS_BIT(0) is 0, when the component is pressed, the value will be inversed, that is, MODBUS_BIT(0)=1, when pressed again, MODBUS_BIT(0)=0.

- **Example 3: When pressed, Set Bit (1), when released, Reset Bit (0).**

Step 1: select register type and No.

Step 2: select “Recovery Bit” for “action”.



Property	
Label	
TextLib	
Format text(0)	0
Format text(1)	1
Action	
Action	Recovery Bit
Action when up	False
Position and size	
Action	

Step 3: check the effect: when pressed, bit register MODBUS_BIT(0)=1, when released, bit register MODBUS_BIT(0)=0.

- **Example 4: Call SUB**

Please refer to 4.3.16.

4.3.22. Word Modify

A. What is It?

Set word register address's value according to state of component's action. At the same time, it supports Call Sub. For details, please refer to below examples.

B. How to Use?

Click "Tool Box", select "Word Modify" from "Input / Toggle / Button" Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed info of different states. And set "action" type. It will set word register value according to action type.

--How to Check Register Value--

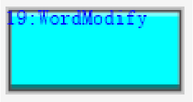
Method 1: click menu Tool – Register, select bound register type and address, then click read.

Method 2: in "output" window, enter print + bound register address and state value, and click "send", for example, enter "print MODBUS_BIT(0) / ? MODBUS_BIT(0)".

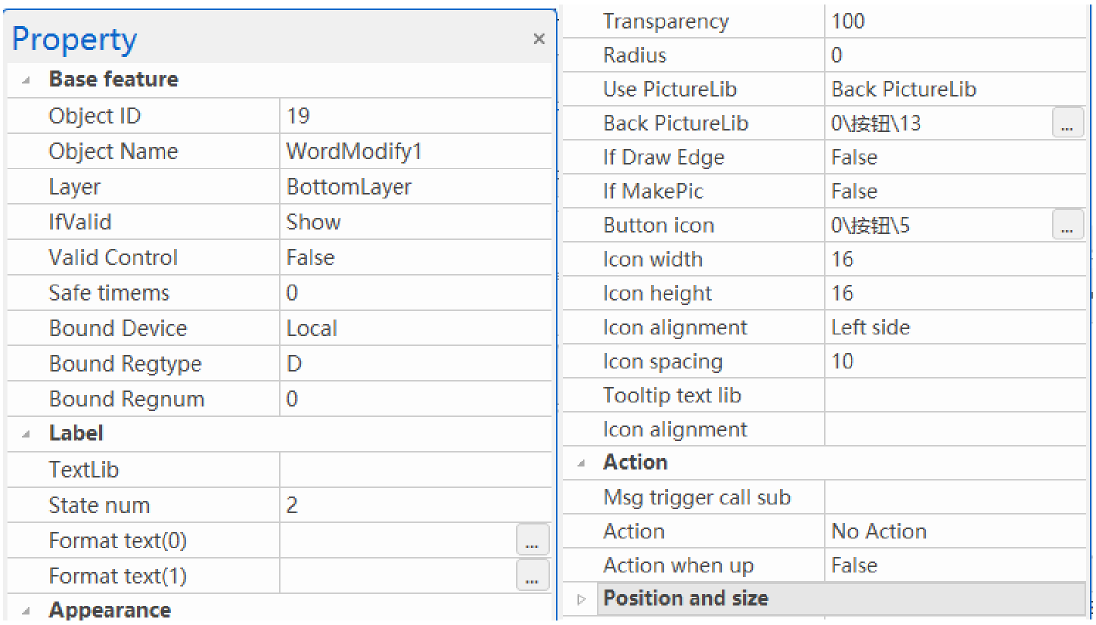
Notes:

a. It can't switch showing state through register value.

b. When the component shows one state, which may not correspond to register value state.



C. “Property” Description



Property

Base feature	
Object ID	19
Object Name	WordModify1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Label	
TextLib	
State num	2
Format text(0)	
Format text(1)	
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\13
If Draw Edge	False
If MakePic	False
Button icon	0\按钮\5
Icon width	16
Icon height	16
Icon alignment	Left side
Icon spacing	10
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
Action	No Action
Action when up	False
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list

	● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows "Text".
State Num	Set component's state numbers (1-256), it can show multiple states.
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Button Icon	Select the icon from the picture library.
Icon Width	Set the icon's width.
Icon Height	Set the icon's height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.

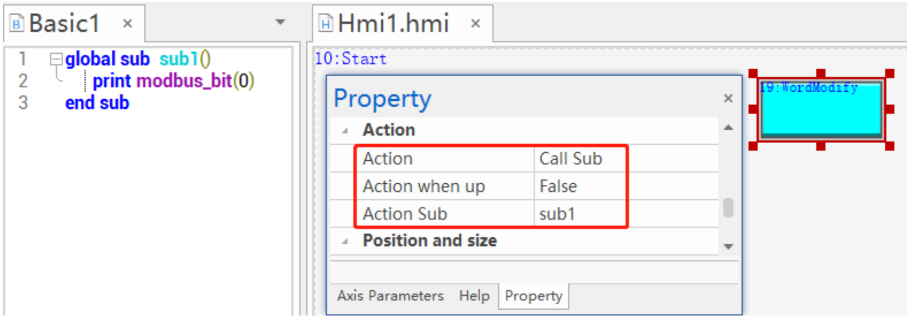
Action	The action to be executed.
Action When Up	The action when pressed or released, default is False – action when pressed, True – action when released.
Action Data	Write assigned value into the register after the button is operated.
Action Sub	The SUB function that is to be called after the button is operated. It is selected from the edited Basic SUB function list.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. Examples

➢ Example 1: Call Sub

Step 1: edit one global SUB function in Basic file.

Step 2: for "action", select "Call Sub", for Action Sub, select corresponding SUB function name.



```
global sub sub1()
    print modbus_bit(0)
end sub
```

Action
- Action: Call Sub
- Action when up: False
- Action Sub: sub1

Position and size

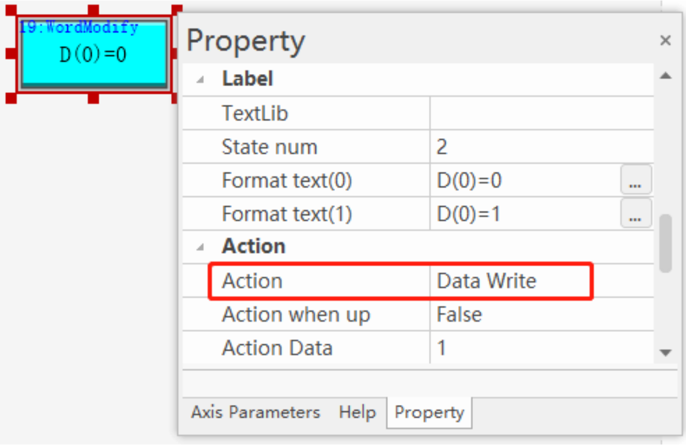
Step 3: check the effect: when the component is pressed, execute SUB function of Basic.

➢ Example 2: Write data into register

Step 1: select register type and No.

Step 2: select "Data Write" for "action", and set action data for register value, for example, set it as 1.

Two states, when pressed, it shows text 1, when released it shows text 0.



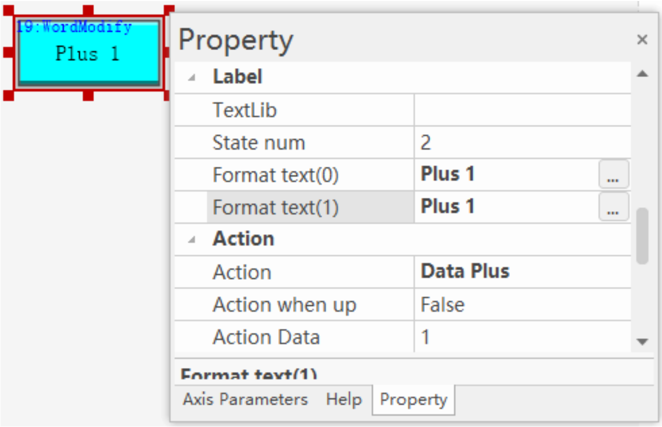
Label	
TextLib	
State num	2
Format text(0)	D(0)=0
Format text(1)	D(0)=1
Action	
Action	Data Write
Action when up	False
Action Data	1

Step 3: check the effect: when pressed, write data 1 to register MODBUS_REG(0). If set "action data" as other values, then write corresponding values into register.

- **Example 3: register original value + action data value**

Step 1: select register type and No.

Step 2: select "Data Plus" for "action", and "Action Data" fills in the data to be increased/decreased each time the register is filled.



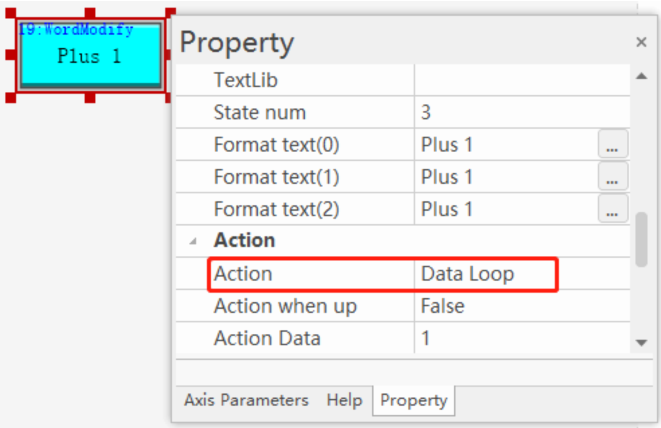
Label	
TextLib	
State num	2
Format text(0)	Plus 1
Format text(1)	Plus 1
Action	
Action	Data Plus
Action when up	False
Action Data	1

Step 3: check the effect: when the component is pressed once, MODBUS_REG(0)=original value + 1.

- **Example 4: register original value + action data value, switch between set states in loop**

Step 1: select register type and No.

Step 2: select "Data Loop" for "action", and "Action Data" fills in the data to be increased.



TextLib	
State num	3
Format text(0)	Plus 1
Format text(1)	Plus 1
Format text(2)	Plus 1
Action	
Action	Data Loop
Action when up	False
Action Data	1

Step 3: check the effect:

After pressing the component, the value of MODBUS_REG(0) switches between 0, 1, and 2.

If the initial value of MODBUS_REG(0) is > 2, pressing it once will automatically calculate and decrement to the set state number range, and then start switching between 0 and 2.

The register value cycles according to the number of states, for example:

- When the state number is 3 and the action data is 1, the corresponding register value switches between 0, 1, and 2.
- When the state number is 5 and the action data is 2, the corresponding register value switches between 0, 2, 4, 1, and 3.

4.3.23. Bit Switch

A. What is It?

Set bit register address’s value and show corresponding state according to component action. That is, when register value is set according to action, it will make state change correspondingly. When register value 0 / 1, showing text 0 / 1.

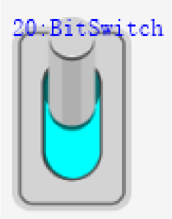
B. How to Use?

Click “Tool Box”, select “Bit Switch” from “Input / Toggle / Button”. Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed info of different states. And set “action” type. It will set bit register value according to action type.

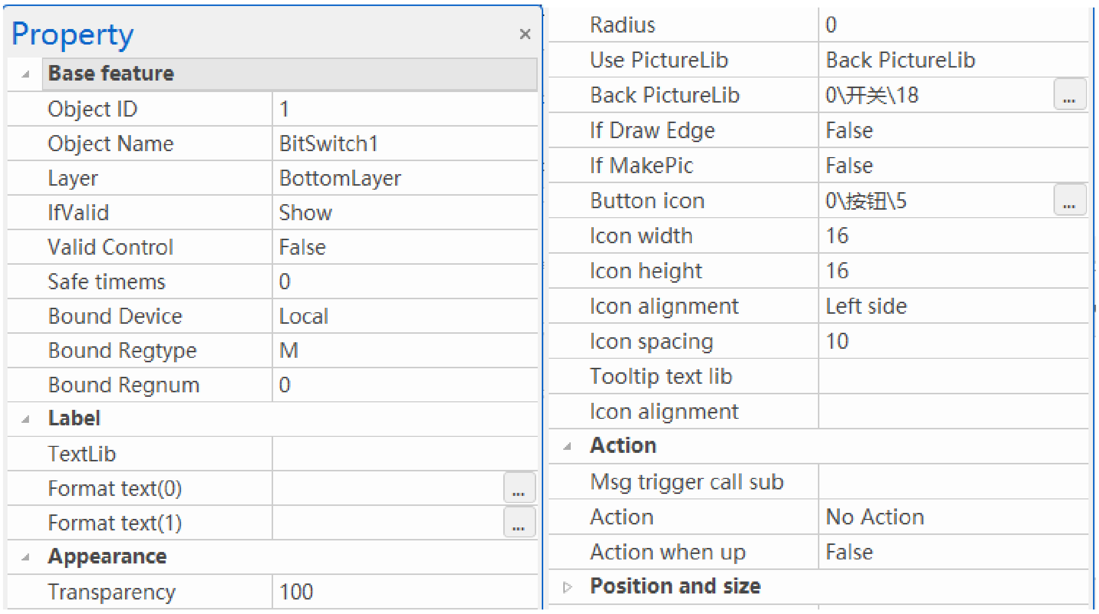
Notes:

a. It also supports Call Sub to achieve state switching, but when no action is set, no way to do that through directly pressed, you need to modify register value and switch manually.

b. When a bit register is selected, format text 0 is displayed when the register value is 0, format text 1 is displayed when the register value is 1. If the selected register is not a bit register, format text 0 is displayed when the register value is 0, format text 1 is displayed when the register value is not 1. “Word State” are used to display multiple bits.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	BitSwitch1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Label	
TextLib	
Format text(0)	
Format text(1)	
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\开关\18
If Draw Edge	False
If MakePic	False
Button icon	0\按钮\5
Icon width	16
Icon height	16
Icon alignment	Left side
Icon spacing	10
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
Action	No Action
Action when up	False
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components.

	● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows “Text”.
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Button Icon	Select the icon from the picture library.

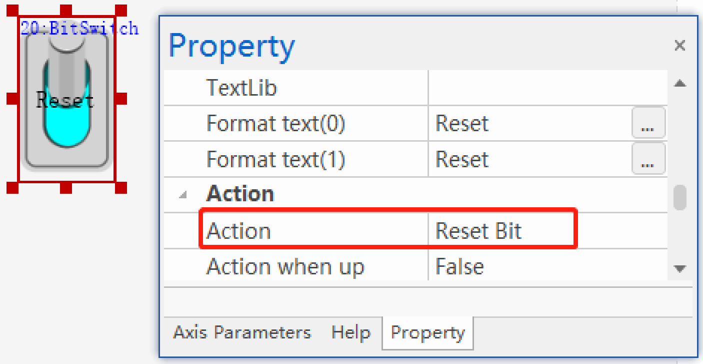
Icon Width	Set the icon’s width.
Icon Height	Set the icon’s height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Action	The action to be executed.
Action When Up	The action when pressed or released, default is False – action when pressed, True – action when released.
Action Sub	The SUB function that is to be called after the button is operated. It is selected from the edited Basic SUB function list.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. Examples

- **Example 1: assign value 1 for register**

Step 1: select register type and No.

Step 2: for “action”, select “Reset Bit” (set as 0).



Property	Value
TextLib	
Format text(0)	Reset
Format text(1)	Reset
Action	
Action	Reset Bit
Action when up	False

Step 3: check the effect: when set bit register value as 0 (that is, MODBUS_BIT(0) = 0), at this time, the object shows state 1, when released, MODBUS_BIT(0) is still 0, and object shows state 0 at the same time.

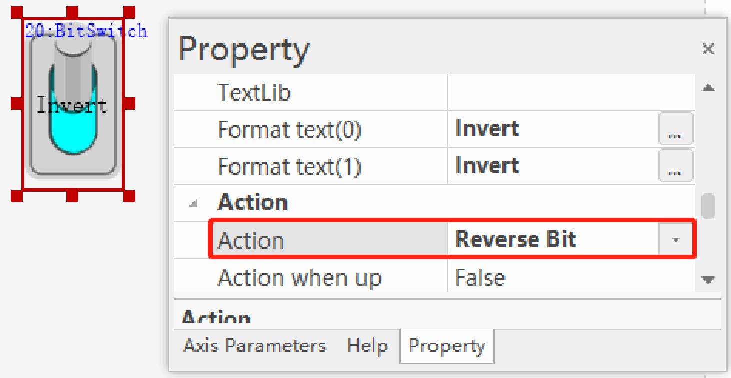
If you selected “True” for “Action when up”, that is, after the object pressed and released, MODBUS_BIT(0)=0, MODBUS_BIT(0) value keeps value 0.

(if you select “Set bit” for action, that is, set register as 1, which is inversed to “Reset Bit”.)

- **Example 2: Invert register value**

Step 1: select register type and No.

Step 2: select “reverse bit” for “action”.



Property

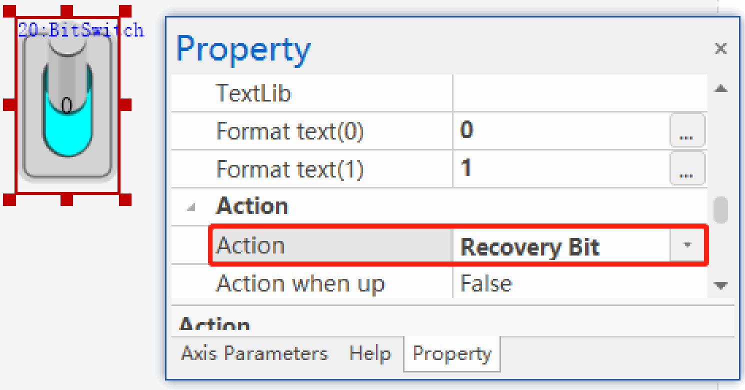
TextLib	
Format text(0)	Invert
Format text(1)	Invert
Action	
Action	Reverse Bit
Action when up	False

Step 3: check the effect: if initial value of MODBUS_BIT(0) is 0, when the component is pressed, the value will be inversed, that is, MODBUS_BIT(0)=1, when pressed again, MODBUS_BIT(0)=0.

- **Example 3: When pressed, Set Bit (1), when released, Reset Bit (0).**

Step 1: select register type and No.

Step 2: select “Recovery Bit” for “action”.



Property

TextLib	
Format text(0)	0
Format text(1)	1
Action	
Action	Recovery Bit
Action when up	False

Step 3: check the effect: when pressed, bit register MODBUS_BIT(0)=1, when released, bit register MODBUS_BIT(0)=0.

E. Example 4: Call SUB (Please refer to 4.3.16.)

4.3.24. Word Switch

A. What is It?

Set word register address's value and show corresponding state according to component action. That is, when register value is set according to action, it will make state change correspondingly. When register value is 0 / 1, showing text 0 / 1.

B. How to Use?

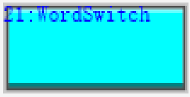
Click "Tool Box", select "Word Switch" from "Input / Toggle / Button". Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed info of different states. And set "action" type. It will set word register value according to action type.

Notes:

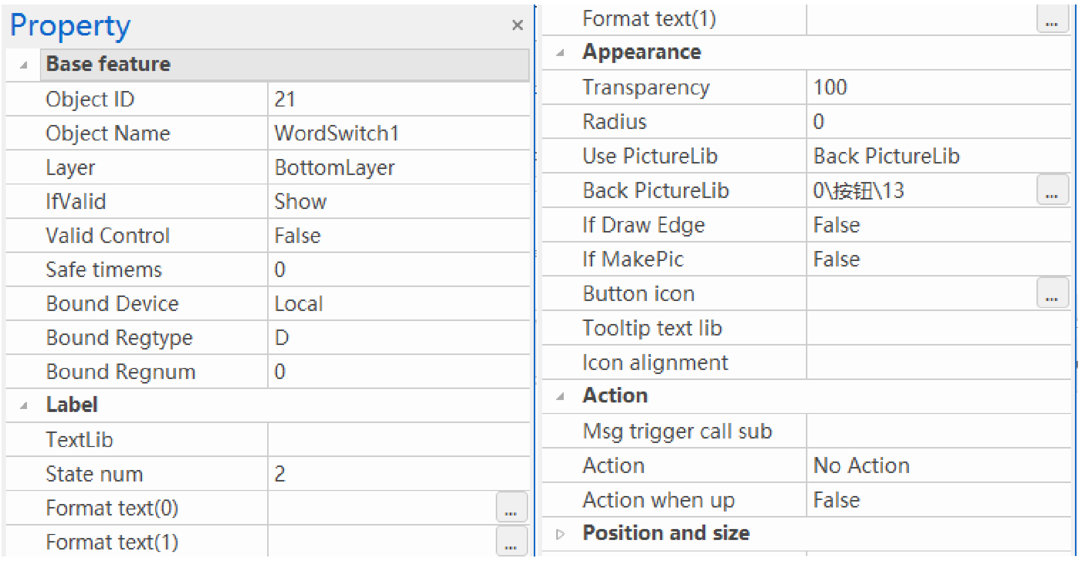
a. It also supports Call Sub to achieve state switching, but when no action is set, no way to do that through directly pressed, you need to modify register value and switch manually.

b. You can set how many states for texts in Property – state num, the range is 1-256.

c. It is recommended to select "word register control", default is D0, when the register value is 0, showing text 0, when the register value is 1, showing text 1, and so on.



C. "Property" Description



Property

Base feature	
Object ID	21
Object Name	WordSwitch1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Label	
TextLib	
State num	2
Format text(0)	...
Format text(1)	...
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\13 ...
If Draw Edge	False
If MakePic	False
Button icon	...
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
Action	No Action
Action when up	False
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list

Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows "Text".
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Button Icon	Select the icon from the picture library.
Icon Width	Set the icon's width.
Icon Height	Set the icon's height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Action	The action to be executed.
Action When Up	The action when pressed or released, default is False – action when pressed, True – action when released.
Action Sub	The SUB function that is to be called after the button is operated. It is selected from the edited Basic SUB function list.
Action Data	Write assigned value into the register after the button is operated.

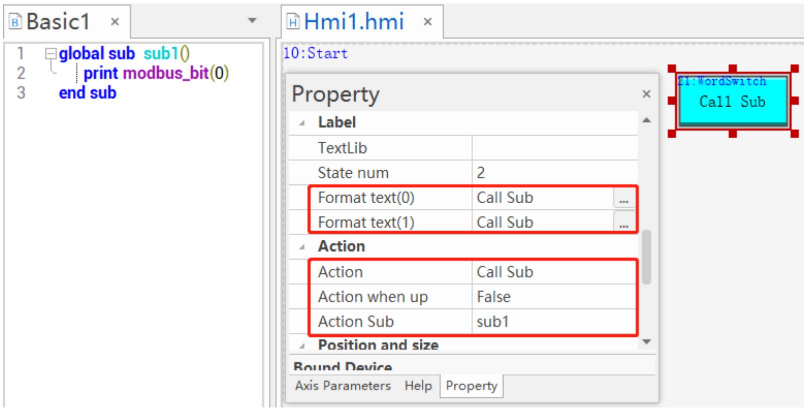
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. Examples

➢ Example 1: Call Sub

Step 1: edit one global SUB function in Basic file.

Step 2: for “action”, select “Call Sub”, for Action Sub, select corresponding SUB function name.



```
global sub sub1()
    print modbus_bit(0)
end sub
```

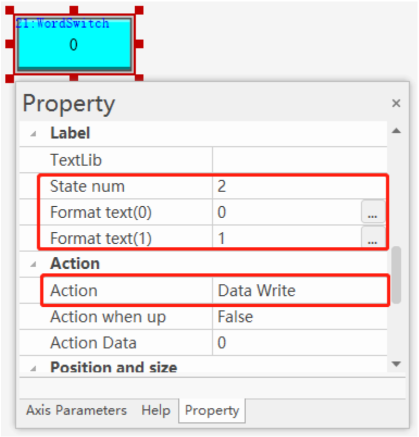
Property	
Label	
TextLib	
State num	2
Format text(0)	Call Sub
Format text(1)	Call Sub
Action	
Action	Call Sub
Action when up	False
Action Sub	sub1
Position and size	
Round Device	

Step 3: check the effect: when the component is pressed, execute SUB function of Basic.

➢ Example 2: Write data into register

Step 1: select register type and No.

Step 2: select “Data Write” for “action”, and set action data for register value, for example, set it as 1.



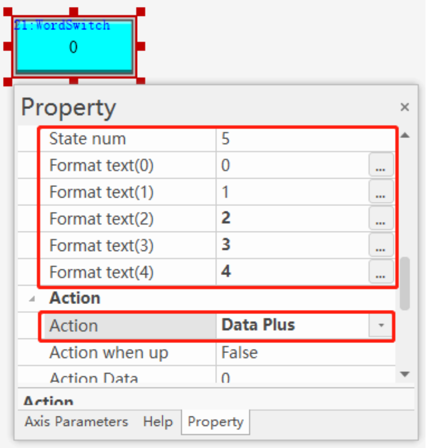
Property	
Label	
TextLib	
State num	2
Format text(0)	0
Format text(1)	1
Action	
Action	Data Write
Action when up	False
Action Data	0
Position and size	

Step 3: check the effect: when pressed, write data 1 to register MODBUS_REG(0), that is, MODBUS_REG(0)=1, and it shows text 1 at the same time.

- **Example 3: register original value + action data value**

Step 1: select register type and No.

Step 2: select “Data Plus” for “action”, and enter value in “action data”.



State num	5
Format text(0)	0
Format text(1)	1
Format text(2)	2
Format text(3)	3
Format text(4)	4
Action	
Action	Data Plus
Action when up	False
Action Data	0

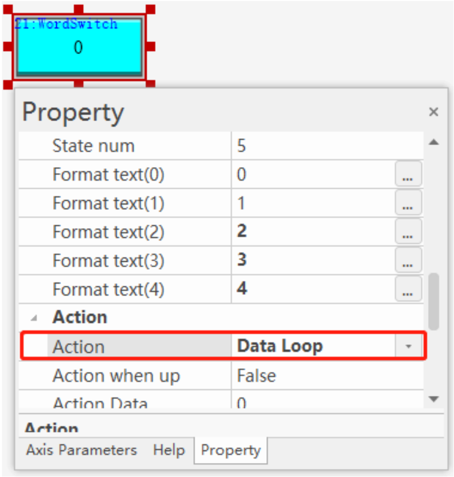
Step 3: check the effect: when the component is pressed once, MODBUS_REG(0)=original value + 1.

Note: when the register value > state numbers, the component will not be displayed, but the touch will still work. As shown in the figure above, when the register value is > 4, the component will not be displayed.

- **Example 4: register original value + action data value, switch between set states in loop**

Step 1: select register type and No.

Step 2: select “Data Loop” for “action”, and "Action Data" fills in the data to be increased.



State num	5
Format text(0)	0
Format text(1)	1
Format text(2)	2
Format text(3)	3
Format text(4)	4
Action	
Action	Data Loop
Action when up	False
Action Data	0

Step 3: check the effect:

After pressing the component, the value of MODBUS_REG(0) switches between 0-4.

If the initial value of MODBUS_REG(0) is > 2, pressing it once will automatically calculate and decrement to the set state number range, and then start switching between 0 and 2.

The register value cycles according to the number of states, for example:

- When the state number is 3 and the action data is 1, the corresponding register value switches between 0, 1, and 2.
- When the state number is 5 and the action data is 2, the corresponding register value switches between 0, 2, 4, 1, and 3.

4.3.25. Button

A. What is It?

It can switch state / window / keyboards according to set action. There are only two states, showing text 0 by default, but when pressed, it will show text 1, like one switch.

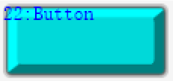
B. How to Use?

Click “Tool Box”, select “Button” from “Input / Toggle / Button”. Put the component at suitable position, then in its property window, enter needed content in the “format text”. And select needed action type in “action”. Then it will operate according to that configuration.

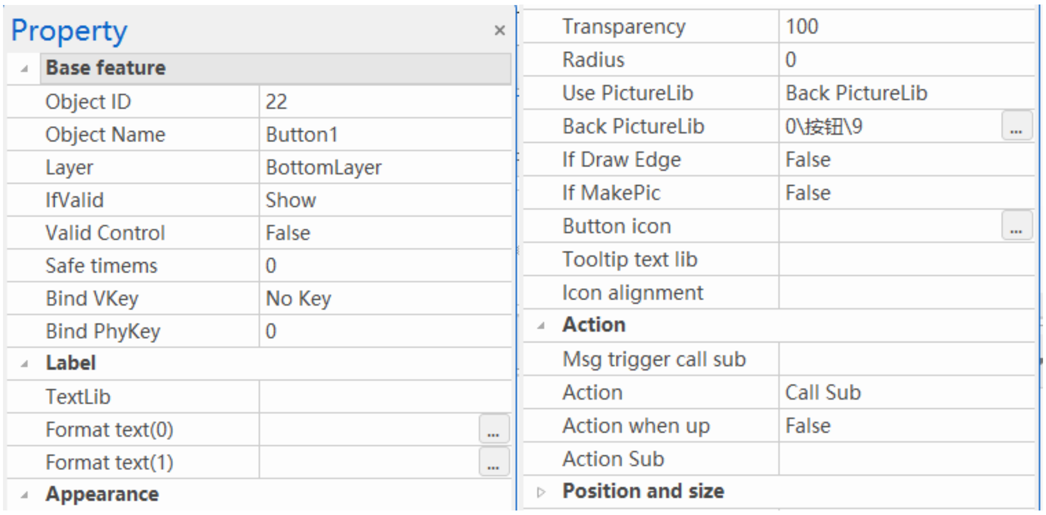
“Button” main functions: call basic function, open / close window, input button / character, switch keyboards. Select them through “property” – “action”.

Notes:

a. Button doesn’t support binding with register.



C. **"Property" Description**



Property

Base feature	
Object ID	22
Object Name	Button1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bind VKey	No Key
Bind PhyKey	0
Label	
TextLib	
Format text(0)	...
Format text(1)	...
Appearance	
Transparency	100
Radius	0
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9 ...
If Draw Edge	False
If MakePic	False
Button icon	...
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
Action	Call Sub
Action when up	False
Action Sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local

Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Label	
TextLib	Text library name, if no set, it shows "Text".
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
Button Icon	Select the icon from the picture library.
Icon Width	Set the icon's width.
Icon Height	Set the icon's height.
Icon Alignment	set the icon align position.
Icon Spacing	Set the spacing between the icon and the border.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Action	The action to be executed.
Action When Up	The action when pressed or released, default is False – action when pressed, True – action when released.
Action Sub	The SUB function that is to be called after the button is operated. It is selected from the edited Basic SUB function list.

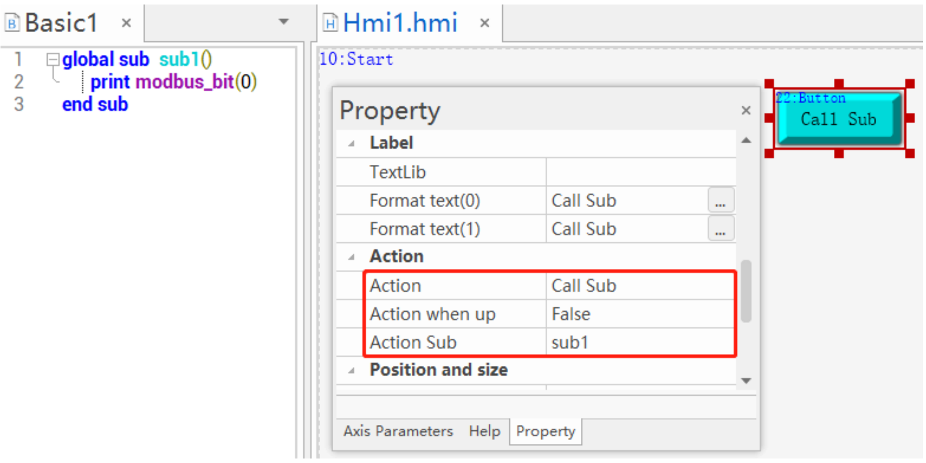
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. Examples

➢ Example 1: Call Sub

Step 1: set button's action as "Call Sub".

Step 2: then select which one sub function to be called, set it in "action sub"



```
global sub sub1()
    print modbus_bit(0)
end sub
```

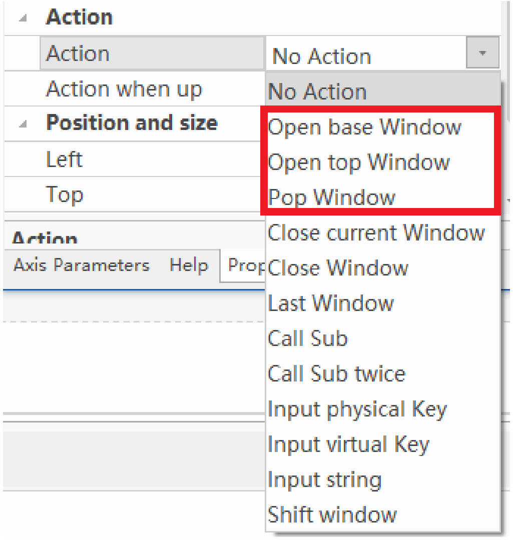
Property

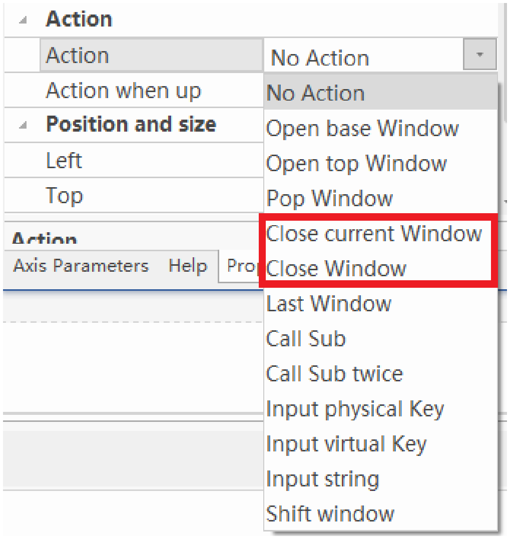
Label	
TextLib	
Format text(0)	Call Sub
Format text(1)	Call Sub
Action	
Action	Call Sub
Action when up	False
Action Sub	sub1
Position and size	

Step 3: check the effect: when the component is pressed, call Basic sub function "sub 1".

➢ Example 2: Open / Close Window

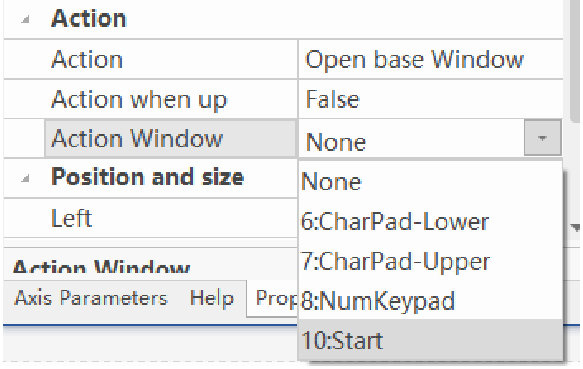
Step 1: in "action" selection of button, you can select below these:





Open Window Close Window

Step 2: take "open window" as the example, when you selected one among these three windows, then find action window, select which one window you want to open.



Action	
Action	Open base Window
Action when up	False
Action Window	None
Position and size	
Left	

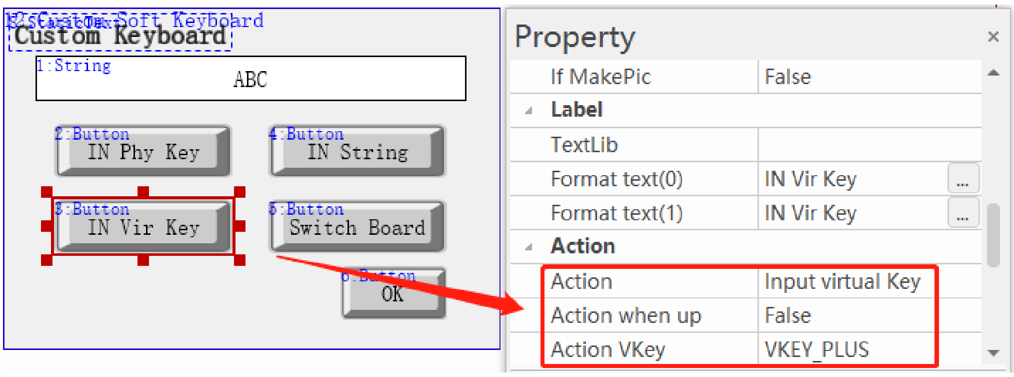
Step 3: check the effect: when pressed, window 10 will be opened.

- **Example 3: custom soft keyboard function**

Step 1: create one base window and one keyboard window (please refer to "New Window"), recommend the keyboard window < base window.

Step 2: in base window, add one "string" component, and in string property, select "true" for "Allow Edit", then select the new created soft keyboard window No.

Step 3: in keyboard window, add one "string" component and several button components. Also, set True for string "allow edit". For button property, select "Input physical key" / "input string" / "input virtual key" / "shift window", and bind key values / keyboard window / character string info.

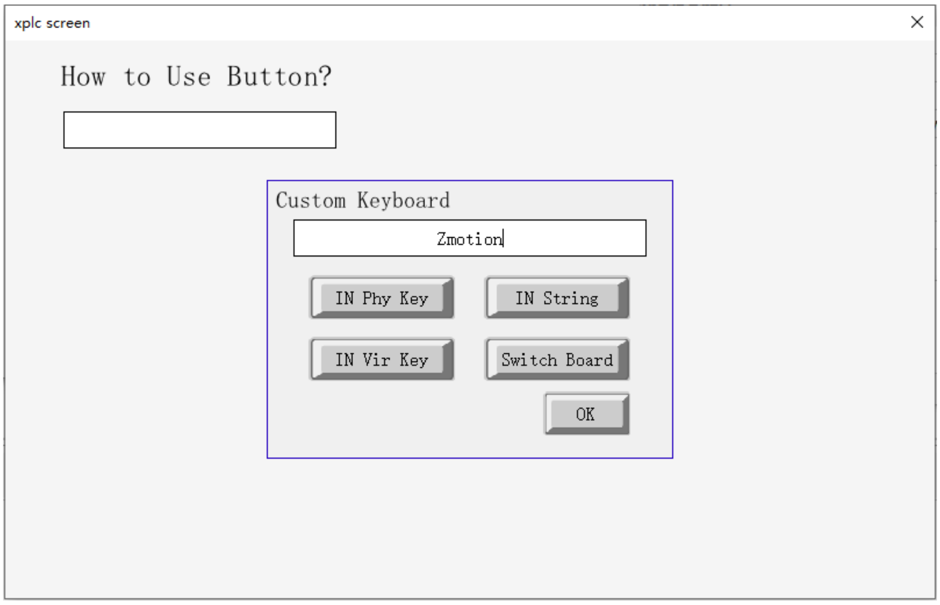


Property

If MakePic	False
Label	
TextLib	
Format text(0)	IN Vir Key
Format text(1)	IN Vir Key
Action	
Action	Input virtual Key
Action when up	False
Action VKey	VKEY_PLUS

Step 3: check the effect:

After downloading, click one button to trigger corresponding effect. For example, if press "IN String", entering info (Zmotion) in string component, if press "switch board", you can witch to other windows.



4.3.26. Key Button

A. What is It?

This can bind with virtual key / teach pendant’s physical key, that is, you can custom physical key actions.

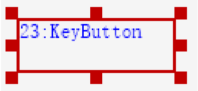
B. How to Use?

Click “Tool Box”, select “KeyButton” from “Input / Toggle / Button”. Put the component at suitable position, and in its property window, select “Bind VKey” / “Bind PhyKey” (only can choose one). Then select needed action type in “action”. In this way, you can achieve corresponding actions through virtual keys / real hardware buttons (for example, if it binds with one certain button of teach pendant key, select “Call Sub” for action, then when you press the teach pendant button, you can call corresponding sub function).

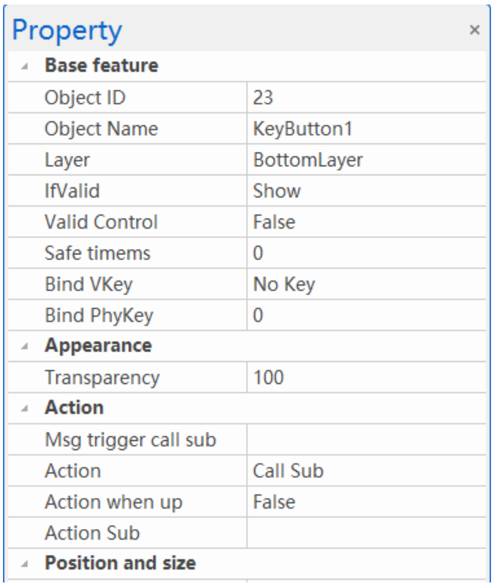
Notes:

a. This component only can be shown when in HMI editing interface, it will not be shown when real running.

b. For “bind phykey”, it needs to get button value. For the button value, please check teach pendant manual.



C. “Property” Description



Property

- Base feature

Object ID	23
Object Name	KeyButton1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bind VKey	No Key
Bind PhyKey	0

- Appearance

Transparency	100

- Action

Msg trigger call sub	
Action	Call Sub
Action when up	False
Action Sub	

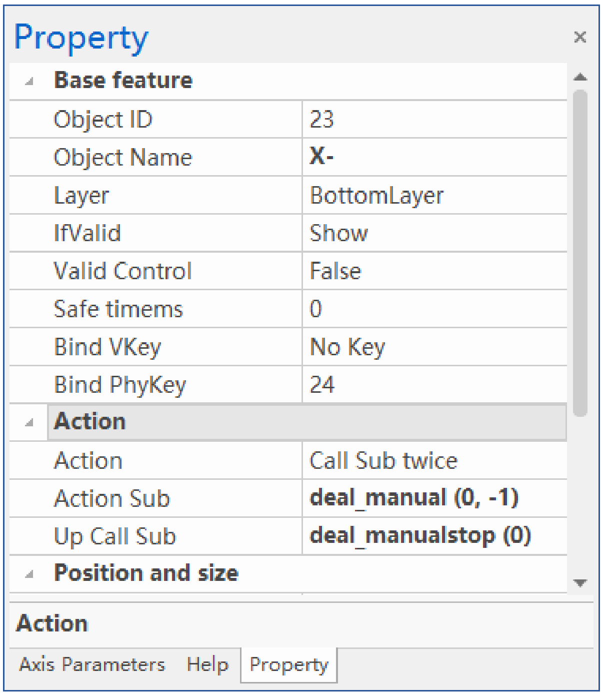
- Position and size

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list

	● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Safe timems	The min button time, the unit is ms
Bind VKey	Select which virtual key to be bound. Default: No Key
Bind PhyKey	Bind physical key on the teach pendant.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Action	The action to be executed.
Action when up	The action when pressed or released, default is False – action when pressed, True – action when released.
Action Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Action window	Which window you need to operate
Action VKey	Select the virtual keys, it is valid when the action is "Input virtual key". Default: not key.
Action String	Enter the string by soft keyboard window, it is valid when the action is "Input String".
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

Bind ZHD400X teach pendant physical key X- (corresponding physical key value is 24) with virtual key, and in property window, set which function will be called when pressed. Then, when you pressed the X-. SUB function will be executed.



Property

Base feature	
Object ID	23
Object Name	**X-**
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bind VKey	No Key
Bind PhyKey	24
Action	
Action	Call Sub twice
Action Sub	**deal_manual (0, -1)**
Up Call Sub	**deal_manualstop (0)**
Position and size	

Action

4.3.27. Radio Button

A. What is It?

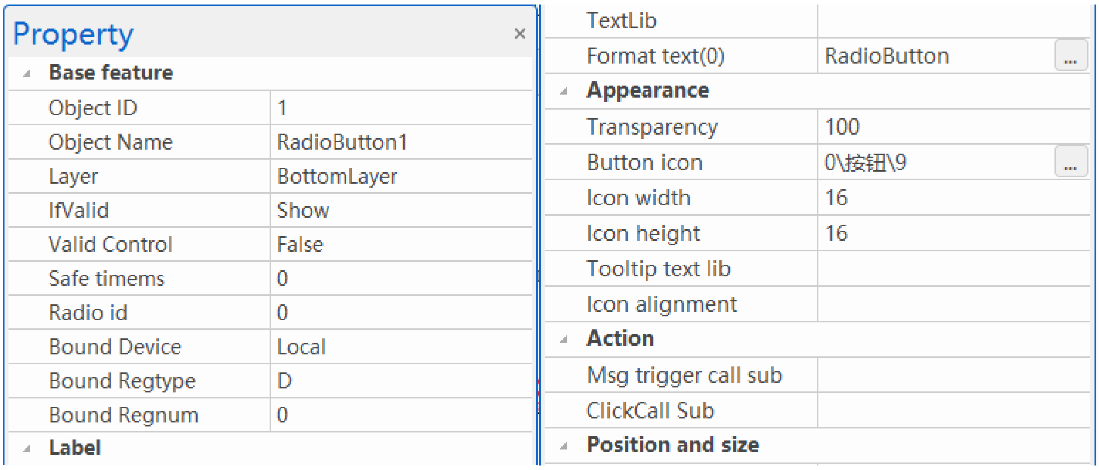
Use one [radio button] to bind with one same register, but only one button can be selected.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click “Tool Box”, select “RadioButton” from “Input / Toggle / Button”. Put the component at suitable position, and in its property window, set the needed appearance. Also, you could set the format text. At last, choose the subfunction that is to be called.

C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	RadioButton1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Radio id	0
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Label	
TextLib	
Format text(0)	RadioButton
Appearance	
Transparency	100
Button icon	0\按钮\9
Icon width	16
Icon height	16
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

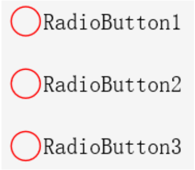
Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid @reg: select @ register’s variable
Safe timems	The min button time, the unit is ms
Radio ID	Set the button No.

	For one group of radio buttons (they bind with one same variable), they must specify different button ID. When clicking the radio button, the variable will assign the No., and the state will become “selected”.
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.
Bound Regnum	Set register No., the register type is not @.
Bound @Reg	Set @ register’s variable, the register type is @ type.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Button Icon	Set the button displayed icon. You could add the icon into picture library, then select it from the picture library. ○ ◉ (show the icon by default) Note: the picture will be shown in the left of the component, and in the middle (vertical), the width and height can be self-defined.
Icon Width	Set the icon’s width.
Icon Height	Set the icon’s height.
Icon Alignment	set the icon align position.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Label	
TextLib	The text library’s name, if no set, it will show format text.
Format Text (0)	Set the text and format that is displayed in radio button.
Action	
ClickCall Sub	Select the SUB function that will be called when clicking the button.
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.

Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

Step 1: here sets 3 radio buttons.



Step 2: the button ID of these 3 buttons are 0, 1, 2 from up to down, and they are bound with one same register DT (0).

Step 3: when one button is clicked, the button will be in “selected” state, and the bound register value will equal to the button ID, remain 2 buttons are still in “unselected” state.

4.3.28. Check Box

A. What is It?

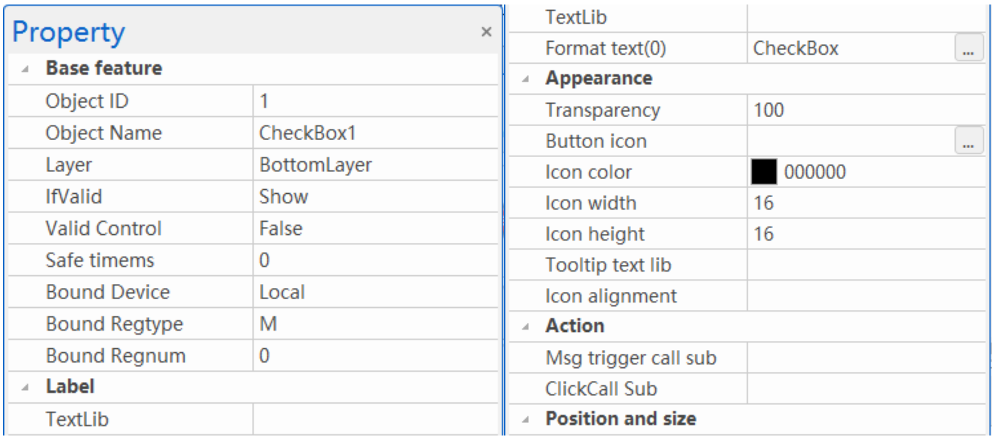
Several [check box] components are bound with different registers. Single-click the component to select it / cancel it.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click “Tool Box”, select “Check Box” from “Input / Toggle / Button”. Put the component at suitable position, and in its property window, set the needed appearance. Also, you could set the format text. At last, choose the subfunction that is to be called.

C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	CheckBox1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	M
Bound Regnum	0
Label	
TextLib	
Format text(0)	CheckBox
Appearance	
Transparency	100
Button icon	
Icon color	■ 000000
Icon width	16
Icon height	16
Tooltip text lib	
Icon alignment	
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid @reg: select @ register’s variable
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.

Bound Regnum	Set register No., the register type is not @.
Bound @Reg	Set @ register's variable, the register type is @ type.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Button Icon	Set the button displayed icon. You could add the icon into picture library, then select it from the picture library. ○ ◉ (show the icon by default) Note: the picture will be shown in the left of the component, and in the middle (vertical), the width and height can be self-defined.
Icon Color	Set the icon's color. Default: RGB (0, 0, 0).
Icon Width	Set the icon's width.
Icon Height	Set the icon's height.
Icon Alignment	set the icon align position.
Tooltip Text Lib	Select the text contents from the text library.
Tooltip Text	The text that is shown when the mouse stops at the component.
Label	
TextLib	The text library's name, if no set, it will show format text.
Format Text (0)	Set the text and format that is displayed in radio button.
Action	
ClickCall Sub	Select the SUB function that will be called when clicking the button.
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

When there is one group of check boxes, each one is bound with different register (variable). And when one box is clicked, it will be in “selected” state, remain boxes also can be selected.

4.3.29. Spin Button

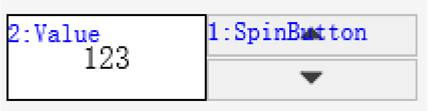
A. What is It?

Provide a pair of up and down arrow buttons [spin button], you can click it to increase or decrease the set value of a specified numerical element. And this is usually used in conjunction with a “value show” component.

Note: when using this control, please make sure RTHMI version be V1.4.0 or above & RTSys version be V1.3.02 or above.

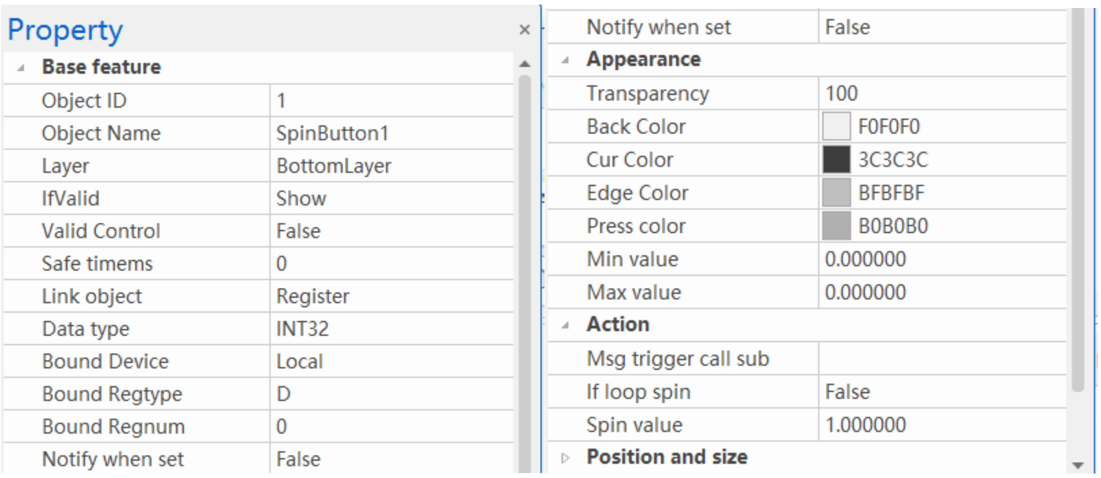
B. How to Use?

Click “Tool Box”, select “Value Show” from “Input / Toggle / Button”. Put the component at suitable position, and set the max value, min value, the spin value. At last, bind the same register as “value show” components’.



(left: value show component; right: spin button component)

C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	SpinButton1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Link object	Register
Data type	INT32
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Notify when set	False

Notify when set	False
Appearance	
Transparency	100
Back Color	F0F0F0
Cur Color	3C3C3C
Edge Color	BFBFBF
Press color	B0B0B0
Min value	0.000000
Max value	0.000000
Action	
Msg trigger call sub	
If loop spin	False
Spin value	1.000000
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid @reg: select @ register's variable
Safe timems	The min key press time, the unit is ms.
Link Object	Select the link method, register / control.
Link Control ID	Specify the "value show" component (in same window), which is

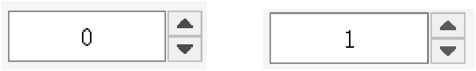
	valid when "link object is control".
Data Type	Set "value" component data type, default is INT32. Please make it consistent with bound register data type. It is valid when "link object is register"
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.
Bound Regnum	Set register No., different values can be obtained by register, then different states of component can be controlled.
Bound @Reg	Set @ register's variable, the register type is @ type.
Notify when set	Notify BIT after modification (ON / OFF), default is False, if True, it will notify the register.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Back Color	Set the background color.
Cur Colorr	Set the current icon color.
Edge Color	Set the edge color, True – show the set color
Press Color	Set the color displayed when the up and down arrows are pressed.
Min value	The lower limit.
Max value	The upper limit.
Action	
Msg Trigger Call Sub	Sets the SUB function to be called when a component configuration switch message is triggered. It is matched with Hmi_CtrlMsgCall command, SUB is triggered only when related message is opened.
If Loop Spin	Whether to cyclically accumulate/decrease.
Spin Vaue	Set the (each accumulate/decrease) value when If Loop Spin is True.
Position & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

Bind the “value show” component with register D(0), also, bind “spin button” with register D(0), and set the spin value as 1. That is, each time the up/down arrow is pressed, the value stored in the corresponding register is incremented/decremented by 1.

Effect:

- Not Pressed (left) & Upper Arrow Pressed (right):



4.3.30. Value Show

A. What is It?

It can edit and show value, also put the value into bound register.

B. How to Use?

Click “Tool Box”, select “Value Show” from “Input / Toggle / Button”. Put the component at suitable position, and in its property window, bind register and No., select whether to use “allow edit” function to call keyboard. Also, set others for the value, like, data type, character, decimal part, etc.

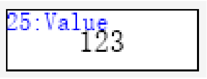
Common Functions of the “Value Show”:

*show register content, only can show non-character type data.

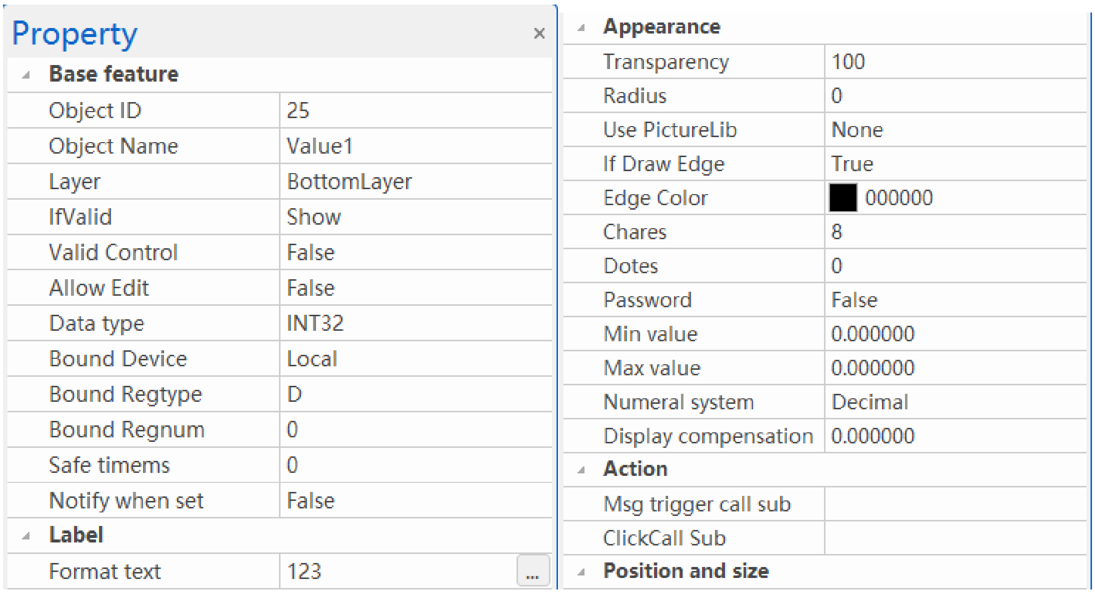
*call keyboard window to set bound register value directly.

*call SUB function.

Notes: when you set as “True” for “allow edit”, and after selecting call keyboard window, call sub can’t be used.



C. “Property” Description



Property

Base feature	
Object ID	25
Object Name	Value1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Allow Edit	False
Data type	INT32
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Safe timems	0
Notify when set	False
Label	
Format text	123

Appearance	
Transparency	100
Radius	0
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 000000
Chares	8
Dotes	0
Password	False
Min value	0.000000
Max value	0.000000
Numeral system	Decimal
Display compensation	0.000000
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.

Allow Edit	Allow it to input data? Default is False, if you select True, it will call the software keyboard to input.
Data Type	Set "value" component data type, default is INT32.
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.
Bound Regnum	Set register No., different values can be obtained by register, then different states of component can be controlled.
Safe timems	The min key press time, the unit is ms.
Notify when set	Notify BIT after modification (ON / OFF), default is False, if True, it will notify the register.
Label	
Format text	Open the setting window, you can set the content of component, and the number of texts is determined by state num.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
If Draw Edge	Whether to draw the edge
Edge Color	Set the edge color, True – show the set color
Chares	Set how long the character of component can operate, default is 8
Dotes	Dotes, the default is 0
Password	Default is False, when True, the component will show "*".
Min value	IN bottom limit
Max value	IN upper limit
Numerial System	Set the numerial system. The default is decimal, and this setting only applies when displaying integers. Options include binary, octal, decimal, and hexadecimal.
Display Compensation	Set the auto error compensation value, it is valid for folating type. Floating-point calculations introduce errors, causing discrepancies in the displayed value. For example, -0.0000001 will be displayed as -0.000. Active error compensation eliminates these common errors. Range: -1.0000000~1.000000
Action	

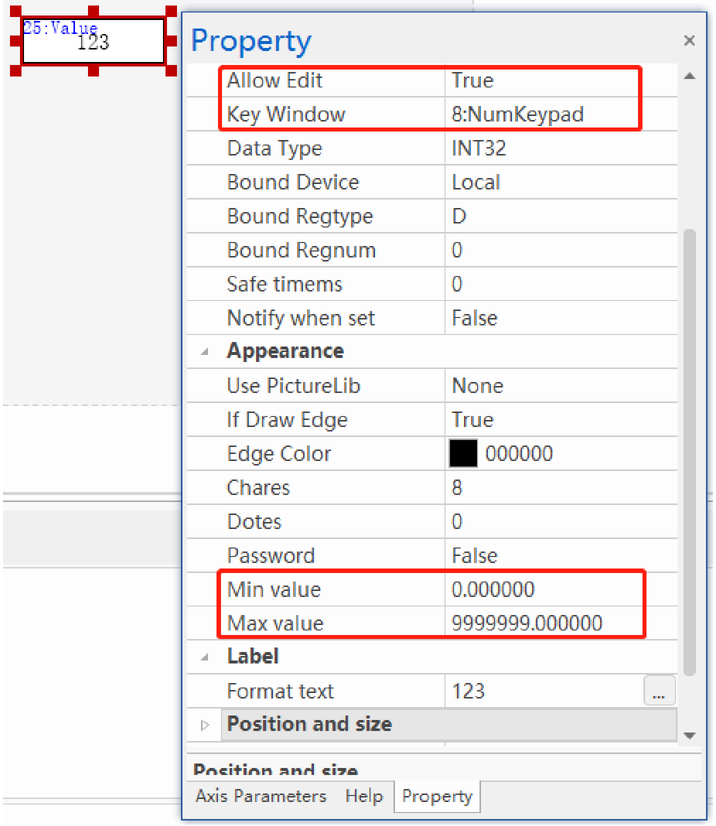
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCall Sub	Call Basic function.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

Step 1: select the keyboard window to be called for “allow edit”.

Step 2: set register type and No.

Step 3: set data type, max value, min value, etc. as needed.

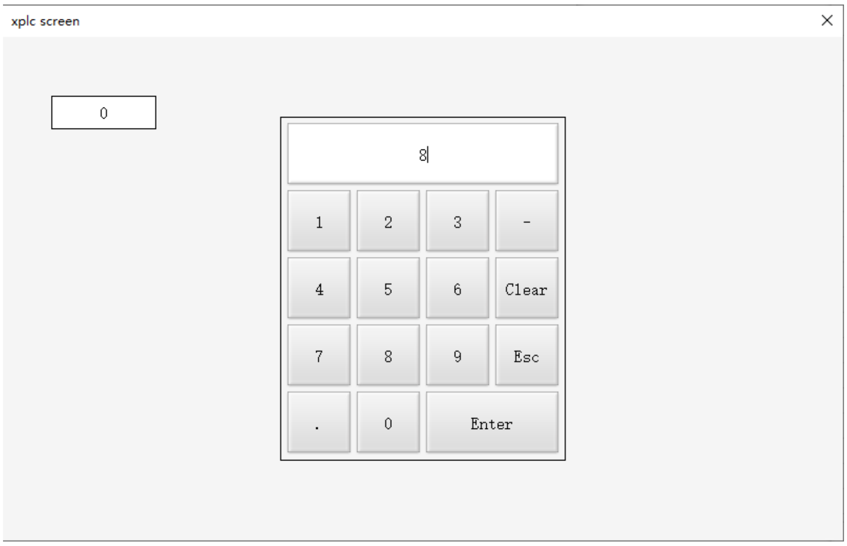


Allow Edit	True
Key Window	8:NumKeypad
Data Type	INT32
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Safe timems	0
Notify when set	False
Appearance	
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 000000
Chares	8
Dotes	0
Password	False
Min value	0.000000
Max value	9999999.000000
Label	
Format text	123
Position and size	

Effect:

The “value” component shows bound register value at first, MODBUS_REG(0)=32. Then, it will real-time get register value and refresh component content.

After downloading the program, it will run, then you can click the “value 123” component to open the keyboard window 8, and you can enter data to modify the register value.



4.3.31. String Show

A. What is It?

It can edit and show all data that belong to character sting type.

B. How to Use?

Click “Tool Box”, select “String Show” from “Input / Toggle / Button”. Put the component at suitable position, and in its property window, bind register and No., select whether to use “allow edit” function to call keyboard. Also, set others for the value, like, character, multi-line showing, Call sub, etc.

Notes:

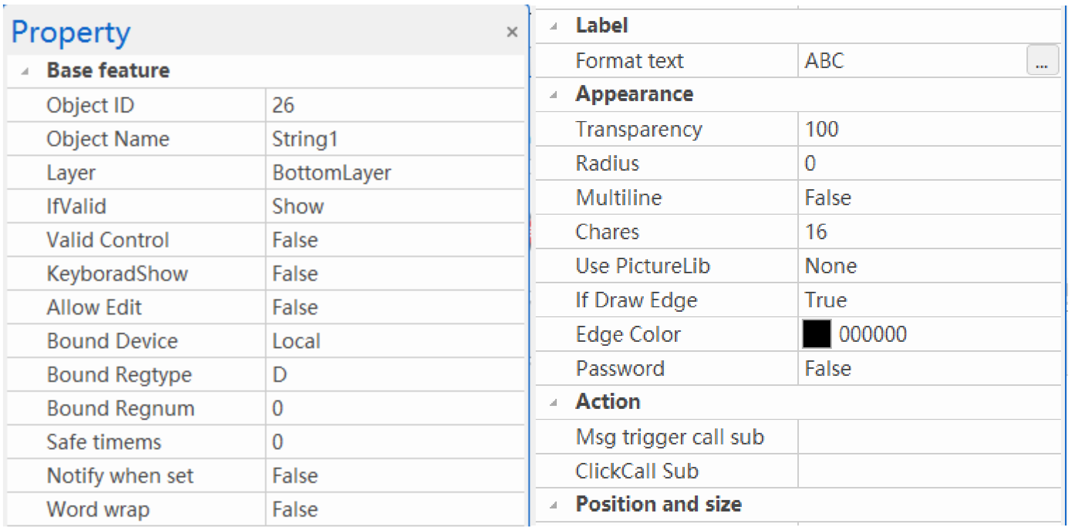
a. The data of register type must be character string type.

b. For “call keyboard window”, you can input data and show them. When the register type is custom @ and variables are not arrays, keyboard window can’t be called and edited.

c. Set “True” for “allow edit”, when selected the called keyboard window, no way to call sub.

d. When you opened “multiline” and “allow edit”, if no keyboard window pops up while using, you can input directly by physical key.

C. “Property” Description



Property

Base feature	
Object ID	26
Object Name	String1
Layer	BottomLayer
IfValid	Show
Valid Control	False
KeyboradShow	False
Allow Edit	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Safe timems	0
Notify when set	False
Word wrap	False

Label	
Format text	ABC
Appearance	
Transparency	100
Radius	0
Multiline	False
Chares	16
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 000000
Password	False
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown.

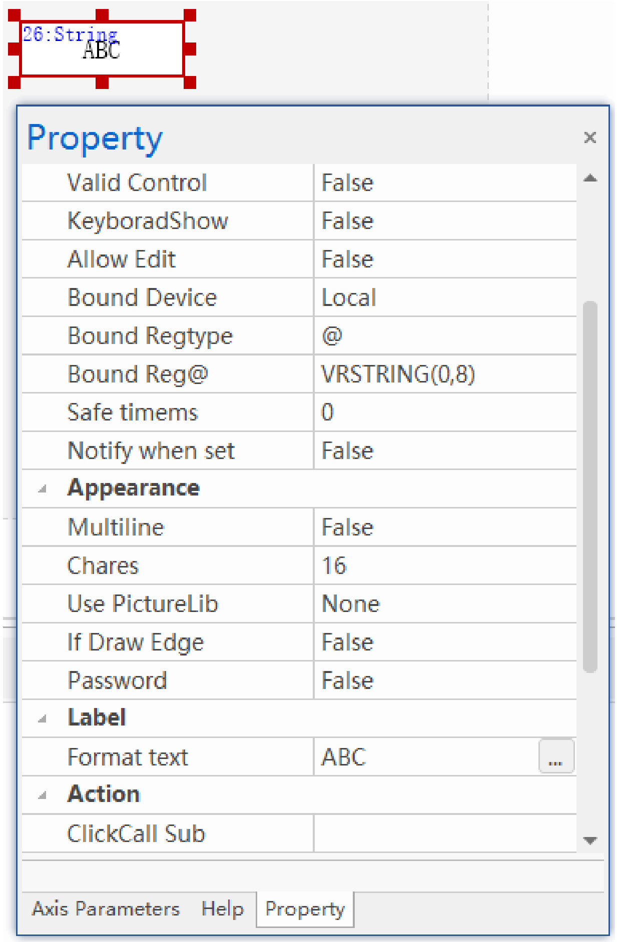
	● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
KeyboardShow	Display "enter" by soft keyboard. Generally it is used for keyboard window.
Allow Edit	Allow it to input data? Default is False, if you select True, it will call the software keyboard to input.
Key Window	When select "True", you can choose keyboard window to be called by the staring component.
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.
Bound Regnum	Set register No., different values can be obtained by register, then different states of component can be controlled.
Safe timems	The min key press time, the unit is ms.
Notify when set	Notify BIT after modification (ON / OFF), default is False, if True, it will notify the register.
Word Wrap	Does the text display automatically wrap?
Label	
Format text	Open the setting window, you can set the content of component, and the number of texts is determined by state num.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Radius	Set the component's chamfering radius, default is 0.
Use Picture Lib	None / use picture library / use back picture
If Draw Edge	Whether to draw the edge
Edge Color	Set the edge color, True – show the set color
Multiline	When character is multi-line character, select True.
Chares	Set how long the character of component can operate, default is 16
Use Picture Lib	None / use picture library / use back picture
Password	Default is False, when True, the component will show "*".
Action	

Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCall Sub	Call Basic function.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

➢ Example 1: show character string saved in register

Step 1: set register type and No., show character string type data.



Valid Control	False
KeyboradShow	False
Allow Edit	False
Bound Device	Local
Bound Regtype	@
Bound Reg@	VRSTRING(0,8)
Safe timems	0
Notify when set	False
Appearance	
Multiline	False
Chares	16
Use PictureLib	None
If Draw Edge	False
Password	False
Label	
Format text	ABC
Action	
ClickCall Sub	

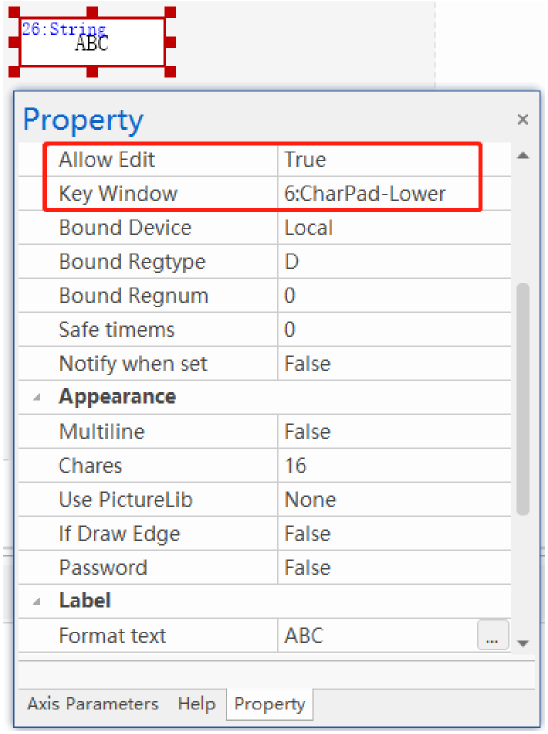
Effect:

When VRSTING (0,8) = “ABC”, the component shows “register saved string ABC”.

- **Example 2: custom component format text**

Step 1: select which keyboard window to be called for “allow edit”.

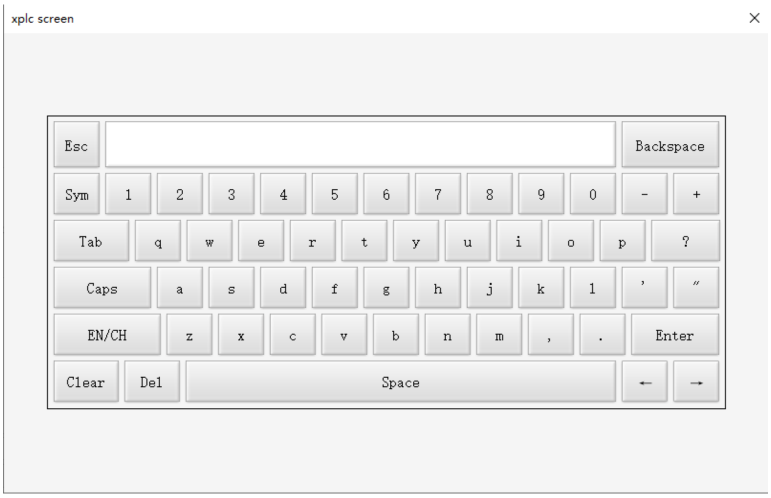
Step 2: set register type and No.



Allow Edit	True
Key Window	6:CharPad-Lower
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Safe timems	0
Notify when set	False
Appearance	
Multiline	False
Chares	16
Use PictureLib	None
If Draw Edge	False
Password	False
Label	
Format text	ABC

Effect:

After downloading, the program is running, when click the component, it will pop up the soft keyboard window, then you can enter needed information, then click “Enter”.


xplc screen
Esc
Backspace
Sym 1 2 3 4 5 6 7 8 9 0 - +
Tab q w e r t y u i o p ?
Caps a s d f g h j k l ' "
EN/CH z x c v b n m , . Enter
Clear Del Space ← →

4.3.32. Slip

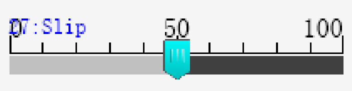
A. What is It?

A component that can be dragged to change the value within a certain range. For example, a slider can be used to zoom in or out, adjust the magnification, or act as a slider switch. The specific application depends on the actual needs.

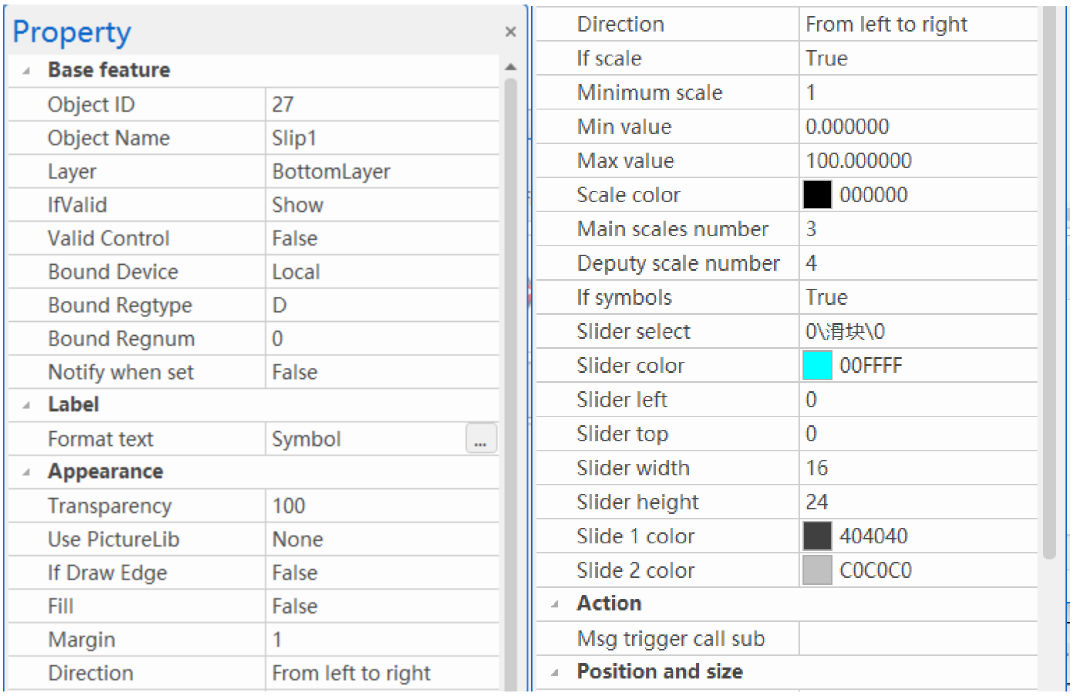
B. How to Use?

Click "Tool Box", select "String Show" from "Input / Toggle / Button". Put the component at suitable position, and in its property window, bind register and No., then it can control data by register variables.

Note: it is only valid in ZMC4XX controllers and above, and valid ZHD500X. For ZHD300X and ZHD400X, you only need to update the firmware.



C. "Property" Description



Property

Base feature	
Object ID	27
Object Name	Slip1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Notify when set	False
Label	
Format text	Symbol
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	False
Fill	False
Margin	1
Direction	From left to right

Direction	From left to right
If scale	True
Minimum scale	1
Min value	0.000000
Max value	100.000000
Scale color	■ 000000
Main scales number	3
Deputy scale number	4
If symbols	True
Slider select	0\滑块\0
Slider color	■ 00FFFF
Slider left	0
Slider top	0
Slider width	16
Slider height	24
Slide 1 color	■ 404040
Slide 2 color	■ C0C0C0
Action	
Msg trigger call sub	
Position and size	

Property	Description

Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.
Bound Regnum	Set register No., different values can be obtained by register, then different states of component can be controlled.
Notify when set	Notify BIT after modification (ON / OFF), default is False, if True, it will notify the register.
Label	
Format Text	Set component's display text.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
If Draw Edge	Whether to draw the edge

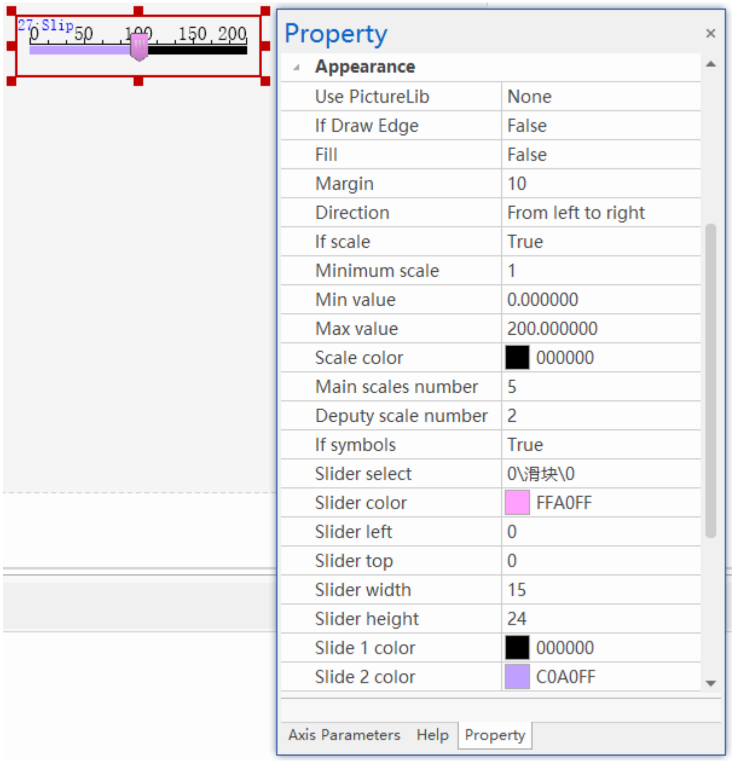
Fill	Whether to fill in the color
Margin	Adjust the distance from component to this object's framework, the border value is larger, the diatance is bigger.
Direction	Select slide direction, there are 4 directions: from left to right, from right to left, from up to bottom, from bottom to up.
If scale	Select True to show min scale
Minimum scale	Set the scale of each slide dragging
Min value	Min initial value of scale (+ & -)
Max value	Max initial value of scale (+ & -)
Scale color	Select the color of scale showing, default is black
Main scales number	Set the number of long scale segments of the slider
Deputy scales number	Set the number of short scale segments of the slider
If symbols	Whether to show the scale
Slider select	Select slider's format
Slider color	Select slider's color
Slider left	Set silder's horizontal position, when it is 0, Center the slider horizontally
Slider top	Set silder's vertical position, when it is 0, Center the slider vertically.
Slider width	Set silder's width
Slider height	Set silder's height
Slide 1 color	The color of the right rail of the slider
Slide 2 color	The color of the left rail of the slider
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Postion & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. For Example:

➢ Example 1: basic configuration of slider scale

Select the register type to store the value of the slider changes. The size can be adjusted by dragging the slider directly. And the number of scales and divisions displayed by the slider can be set by the main scale and sub-scale. After determining the minimum and maximum values, set the number of lines inserted in the main scale to 5 as needed, and then set the number of lines inserted in the sub-scale to 2, so as to divide the single grid into 3 small grids. Select the progress bar color, slider color and style according to your preferences.

For detail routine, please refer to View Zoom.



Appearance

Use PictureLib	None
If Draw Edge	False
Fill	False
Margin	10
Direction	From left to right
If scale	True
Minimum scale	1
Min value	0.000000
Max value	200.000000
Scale color	■ 000000
Main scales number	5
Deputy scale number	2
If symbols	True
Slider select	0\滑块\0
Slider color	■ FFA0FF
Slider left	0
Slider top	0
Slider width	15
Slider height	24
Slide 1 color	■ 000000
Slide 2 color	■ C0A0FF

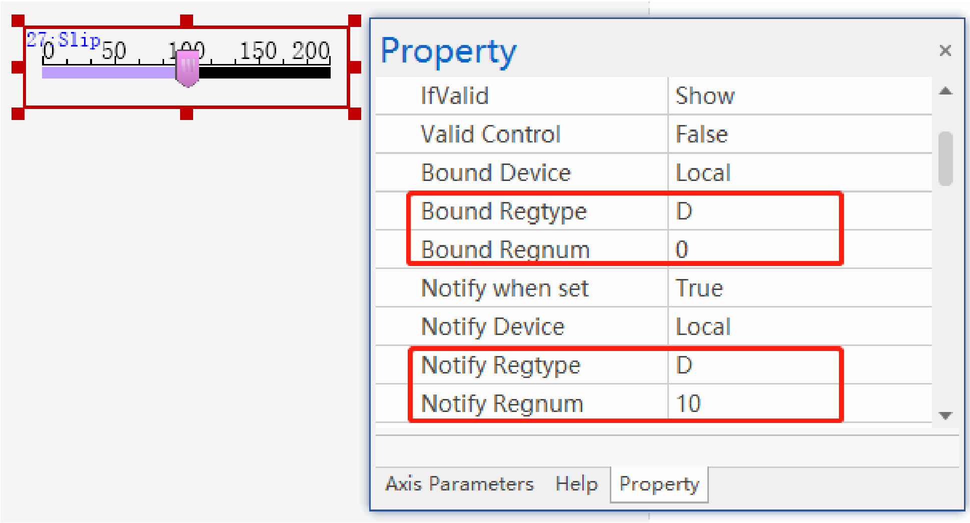
➢ Example 2: bind register with slip

Step 1: add “slip” component

Step 2: in its property, please select suitable register type and enter register No., here, select DT0.

(Note: when you select TRUE for notify when set, notified device No. can’t be same as device No. register)

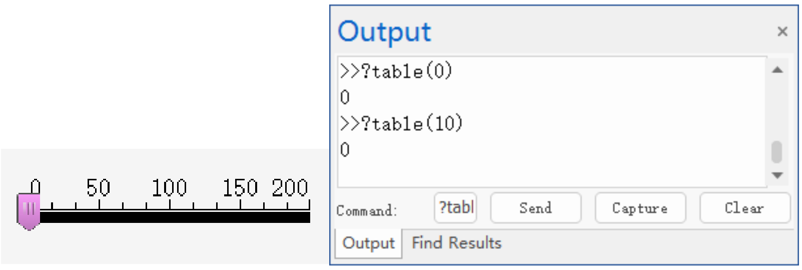
Step 3: in Basic function, set register initial value, if not set, default value is 0.



IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Notify when set	True
Notify Device	Local
Notify Regtype	D
Notify Regnum	10

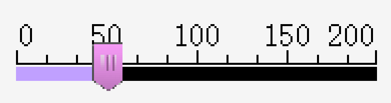
Effect:

Initial value is 0 by default, so table(0) printed value is 0, and slider is not operated, the notify register table(10) also is 0



```
>>?table(0)
0
>>?table(10)
0
```

When you moved the slide to 500, then table(0) will refresh as 50, and the table(10) changes as 1, which means the modification takes effect.



4.3.33. Report View

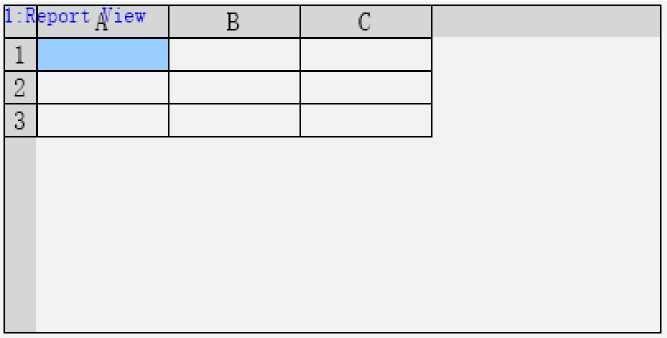
A. What is It?

[Report View] component can show several groups of data in the form format, for showing and managing report data. For editing it, it can enter character and Chinese.

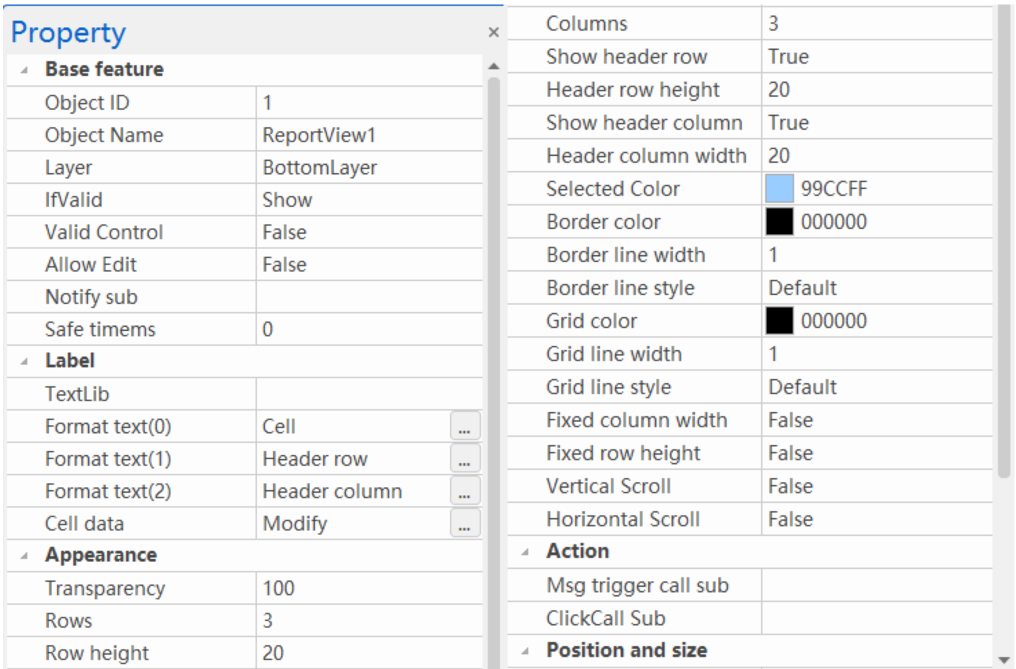
Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

B. How to Use?

Click “Tool Box”, select “Report View” from “Input / Toggle / Button”. Put the component at suitable position, set rows and columns, whether to edit, whether to show header row / column and other properties. Then, click “Cell Data”, you can edit the report content. When set, saved and downloaded, you can check and edit report data.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	ReportView1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Allow Edit	False
Notify sub	
Safe timems	0
Label	
TextLib	
Format text(0)	Cell
Format text(1)	Header row
Format text(2)	Header column
Cell data	Modify
Appearance	
Transparency	100
Rows	3
Row height	20
Columns	3
Show header row	True
Header row height	20
Show header column	True
Header column width	20
Selected Color	99CCFF
Border color	000000
Border line width	1
Border line style	Default
Grid color	000000
Grid line width	1
Grid line style	Default
Fixed column width	False
Fixed row height	False
Vertical Scroll	False
Horizontal Scroll	False
Action	
Msg trigger call sub	
ClickCall Sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.

Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Allow Edit	Allow to input data for the report view? Default is False, if you select True, it will call software keyboard.
KeyWindow	Select the software keyboard window that will be used when editing. It can be used when Allow Edit is True.
Safe Timems	The min key press time, the unit is ms.
Label	
Format text 0 (Cell)	Set the text format. Text name can't be modifyed.
Format text 1 (Header row)	Set the format of report form head row. Text name can't be modifyed.
Format text 2 (Header column)	Set the format of report form head column. Text name can't be modifyed.
Cell data	Edit the content. ● **Chares:** max displaying characters, max is 255. ● **Data type:** select which data type to be limited, string / numeric. When String is selected, no other configurations. When Numeric is selected, below properties are valid: **Dotes:** dotes range, 0-13. **Range limitation:** whether to check, when checked, Max & Min are

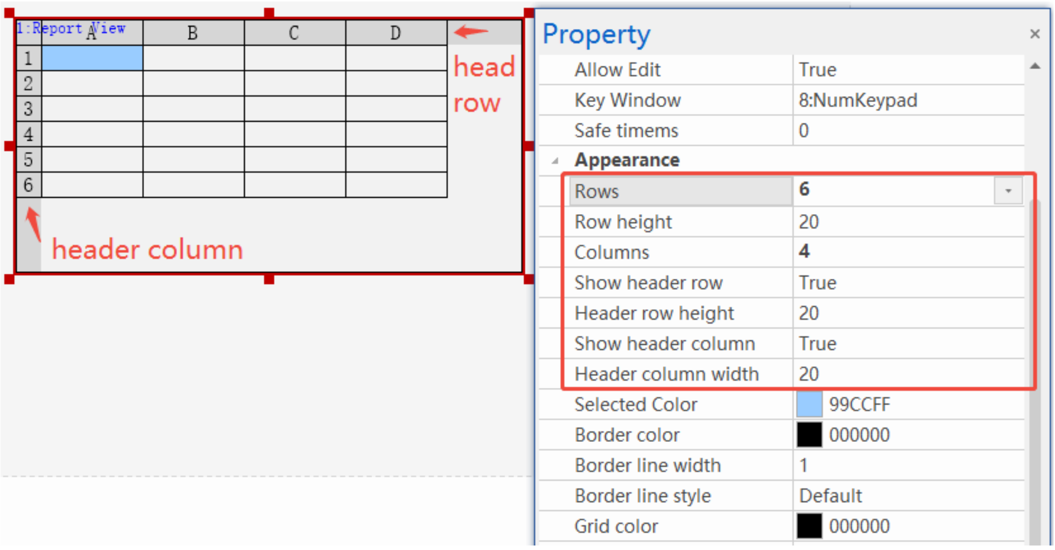
	valid, and max > min.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Rows	Set how many rows of the form: not include the header row, max 512.
Rows Height	The row height of each row (1~n). Modify each row's height: drag the downloaded HMI interface to set. Modify each row's width: in "cell data" – "header title setting" – "width" to set / drag the downloaded HMI interface to set.
Columns	Set how many columns: not include the header column, max is 16.
Show header row	True: show the form head row (default). False: not show
Header row height	Set the height of the header row.
Show header column	True: show the form head column (default). False: not show
Header column width	Set the width of the header column.
Selected color	Set the color of the selected part.
Border color	Set the form border color, the default is black.
Border line width	Set the form border line width.
Border line style	Set the form border line style, the default is solid line.
Grid color	Set the grid line color, the default is black.
Grid line width	Set the grid line width, default is 1, max is 20.
Grid line style	Set the grid line stype, default is the solid line.
Fixed column width	True: column width is fixed. False: scale the column wdith (default).
Fixed row height	True: row height is fixed. False: scale the row height (default).
Vertical scroll	Set whether to use the vertical scroll bar
Horizontal scroll	Set whether to use the horizontal scroll bar
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
ClickCall Sub	SUB function that was defined by BASIC will be called when pressed. The SUB must be GLOBAL function.
Position & Size	

Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

D. For Example

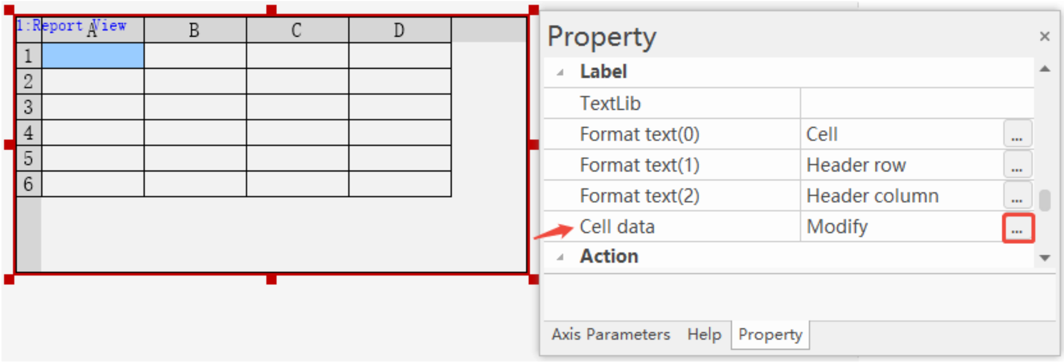
a. In report view property – appearance, set rows, columns, row height and column width (not include head row and head column).

b. Set whether the form is editable, whether to show head row / column, row height (column height), select border color. Here, set True for “allow edit”, and show head row and column.

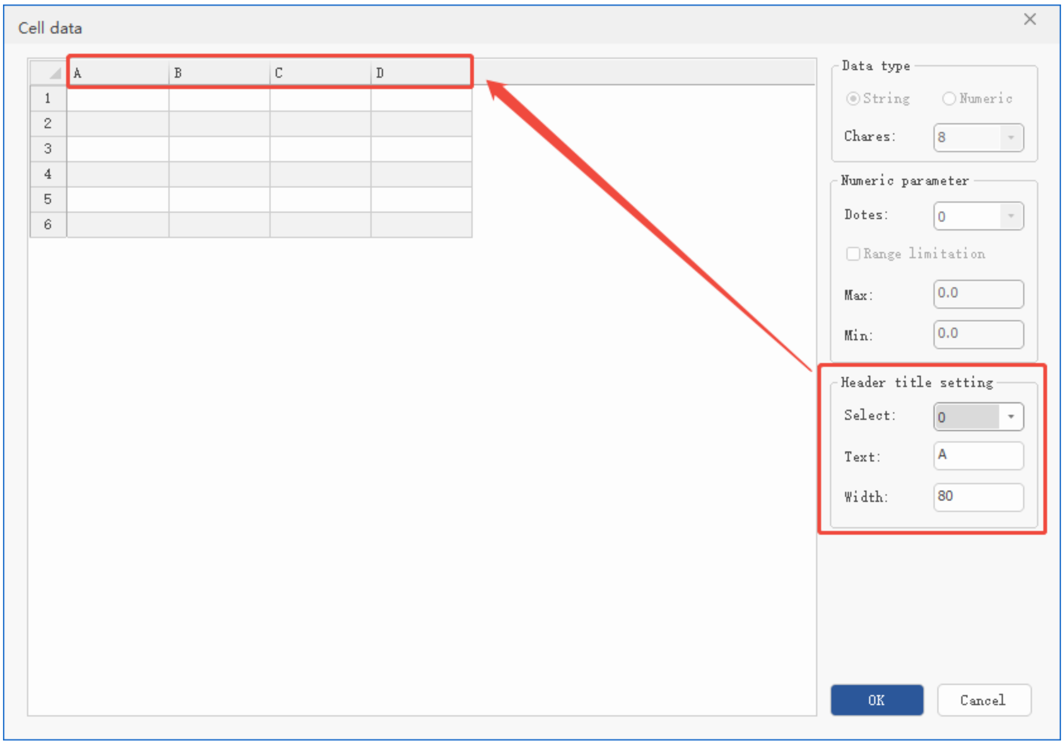


Allow Edit	True
Key Window	8:NumKeypad
Safe timems	0
Appearance	
Rows	6
Row height	20
Columns	4
Show header row	True
Header row height	20
Show header column	True
Header column width	20
Selected Color	99CCFF
Border color	000000
Border line width	1
Border line style	Default
Grid color	000000

c. Click “cell data”, set the header (header row) text.



Label	
TextLib	
Format text(0)	Cell
Format text(1)	Header row
Format text(2)	Header column
Cell data	Modify
Action	



● String ○ Numeric

Chares: 8

Dotes: 0

□ Range limitation

Max: 0.0

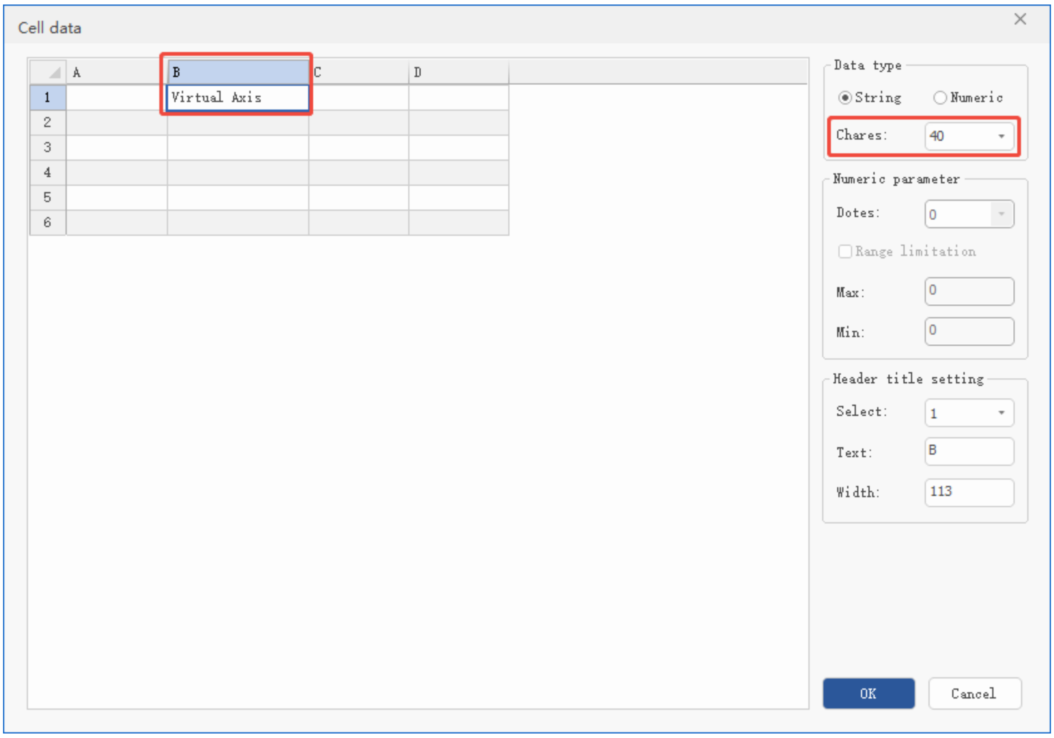
Min: 0.0

Select: 0

Text: A

Width: 80

d. Edit the content. In data type, when you select “string”, you only need to set characters. When you select “numeric”, you can set the dotes, minimal value and maximum value. If dotes is set, but no dotes is edited or 0 is edited, the parameter’s dotes is only shown in downloaded simulation interface.



Virtual Axis

● String ○ Numeric

Chares: 40

Dotes: 0

□ Range limitation

Max: 0

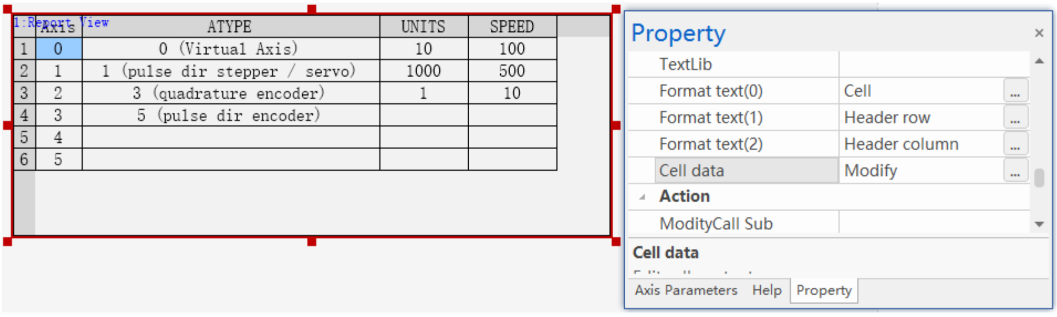
Min: 0

Select: 1

Text: B

Width: 113

e. In HMI window, report view is shown as below.



1:Report View

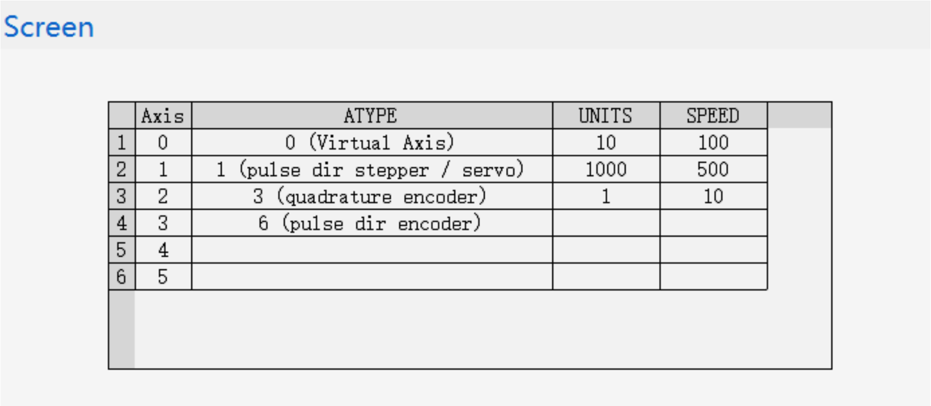
	Axis	ATYPE	UNITS	SPEED
1	0	0 (Virtual Axis)	10	100
2	1	1 (pulse dir stepper / servo)	1000	500
3	2	3 (quadrature encoder)	1	10
4	3	5 (pulse dir encoder)		
5	4			
6	5			

Property

TextLib	
Format text(0)	Cell
Format text(1)	Header row
Format text(2)	Header column
Cell data	Modify
Action	
ModityCall Sub	

Cell data

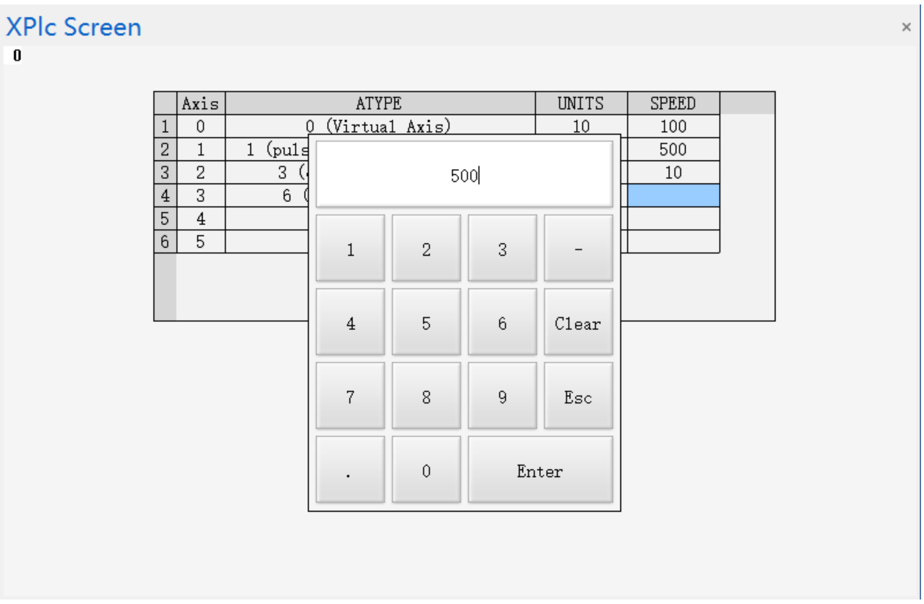
f. Connect to controller, download HMI file into controller, then, you can see the showing effect in xplc screen window. When the "fixed column width (row heigh)" is false, in downloaded simulation interface, you can put the mouse at header row to adjust the each column's width, or put the mouse at the header column to adjust the each row's height.



Screen

	Axis	ATYPE	UNITS	SPEED
1	0	0 (Virtual Axis)	10	100
2	1	1 (pulse dir stepper / servo)	1000	500
3	2	3 (quadrature encoder)	1	10
4	3	6 (pulse dir encoder)		
5	4			
6	5			

Like above, double-click one, you can edit it. Head row and column contents only can be set and edited in property cell data, it can't online edit!

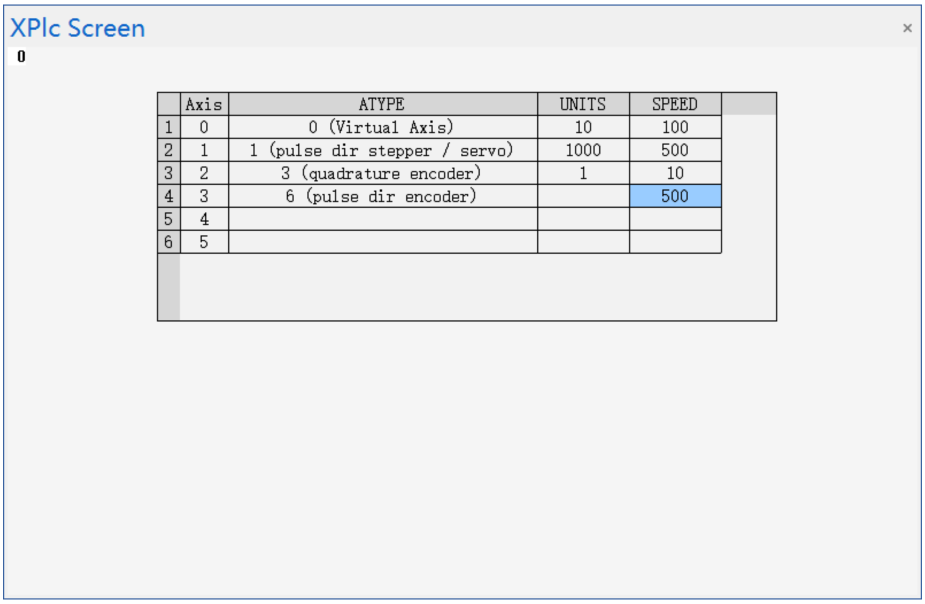


XPlc Screen

0

	Axis	ATYPE	UNITS	SPEED
1	0	0 (Virtual Axis)	10	100
2	1	1 (puls		500
3	2	3 (		10
4	3	6 (		
5	4			
6	5			

500



	Axis	ATYPE	UNITS	SPEED
1	0	0 (Virtual Axis)	10	100
2	1	1 (pulse dir stepper / servo)	1000	500
3	2	3 (quadrature encoder)	1	10
4	3	6 (pulse dir encoder)		500
5	4			
6	5			

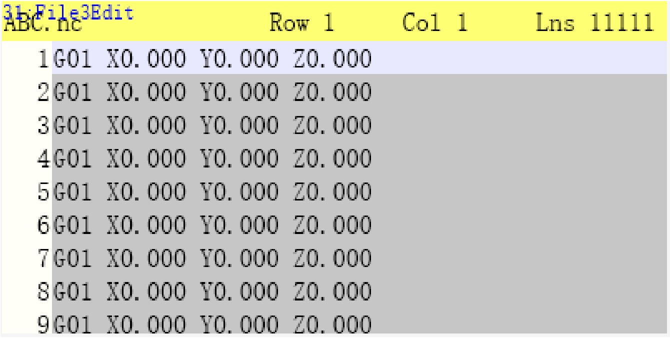
4.3.34. File 3 Edit

A. What is It?

It can develop file 3 program in HMI interface, specifically, edit and show the program. It needs using together with FILE3 (Z3P) commands.

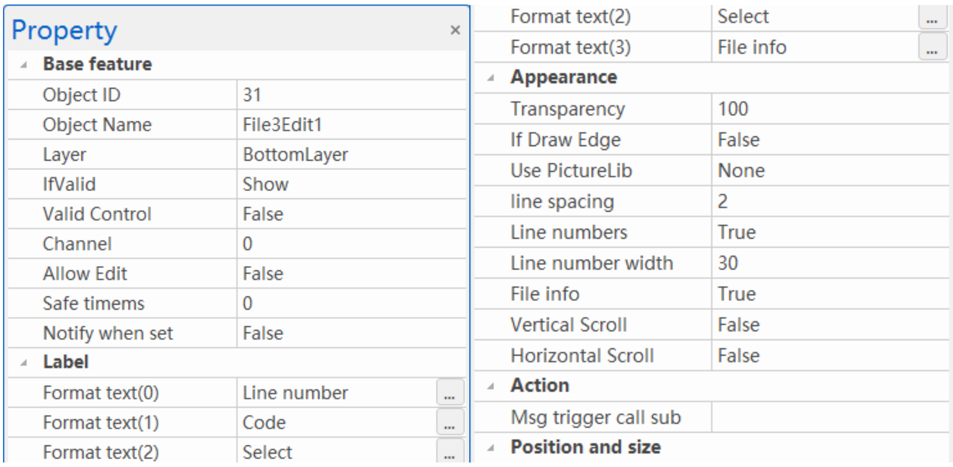
B. How to Use?

Click “Tool Box”, select “Report View” from “Input / Toggle / Button”. Put the component at suitable position, and select the channel No. for matching with other components and BASCI program. Then it can import and edit the program in real-time (for editing, select true for Allow Edit in its property window).



```
ABC.nc          Row 1     Col 1     Lns 11111
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
```

C. "Property" Description



Property

Base feature	
Object ID	31
Object Name	File3Edit1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Channel	0
Allow Edit	False
Safe timems	0
Notify when set	False
Label	
Format text(0)	Line number
Format text(1)	Code
Format text(2)	Select
Format text(3)	File info
Appearance	
Transparency	100
If Draw Edge	False
Use PictureLib	None
line spacing	2
Line numbers	True
Line number width	30
File info	True
Vertical Scroll	False
Horizontal Scroll	False
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Allow Edit	Allow to input data for the report view? Default is False, if you select True, it will call software keyboard.
Channel	Set the file 3 (Z3P) channel No., 3 channels at most: 0 / 1 /2
KeyWindow	Select the software keyboard window that will be used when editing.

	It can be used when Allow Edit is True.
Safe Timems	The min key press time, the unit is ms.
Notify when set	Notify BIT after modification (ON / OFF), default is False, if True, it will notify the register.
Label	
Format text (0)	Set the line number style.
Format text (1)	Set the progamming code style.
Format text (2)	Set the progamming select style.
Format text (3)	Set file information row style.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge
Use Picturer Lib	None / use picture library / use back picture
Line Spacing	Set the spacing between rows, default is 2.
Line Numbers	Whether to show the row No.
Line Number Width	Set the width of the row No., valid when "Line Numbers" is True.
File Info	Whether to show the title row.
Vertical Scroll	Set whether to use the vertical scroll bar
Horizontal Scroll	Set whether to use the horizontal scroll bar
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

For Quick Start CAD, please refer to Chapter VI Use CAD & File 3.

4.3.35. Pie Chart

A. What is It?

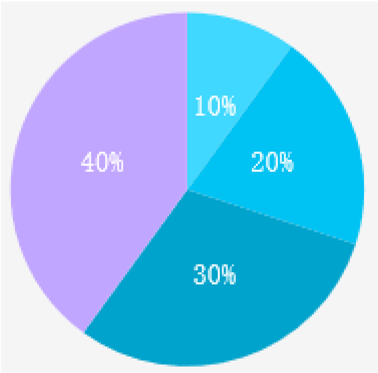
Used to show one group of register value's each channel ratio in the form of pie chart.

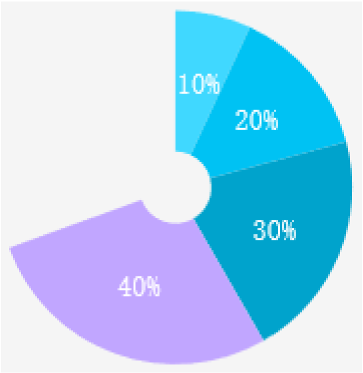
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

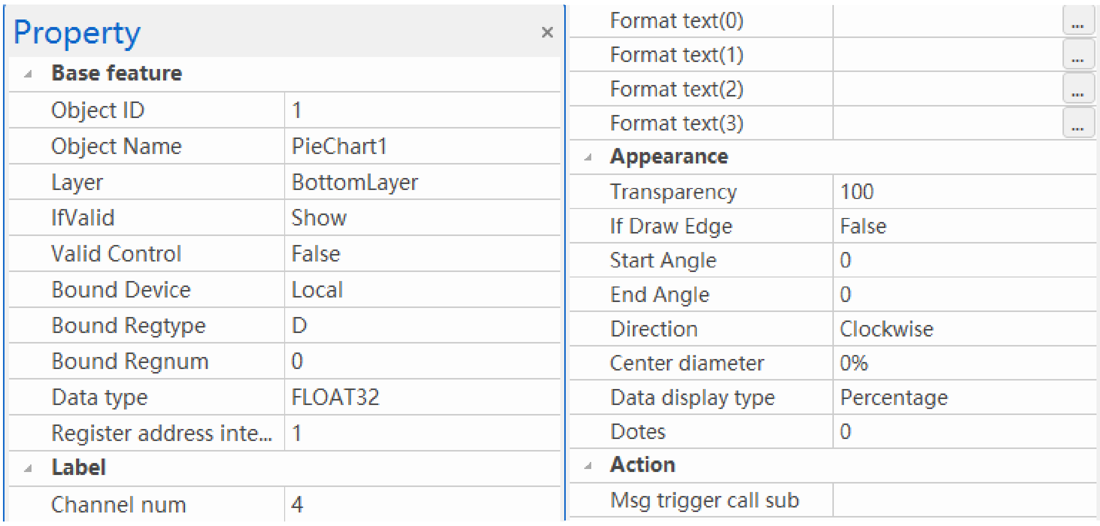
Click "Tool Box", select "Pie Chart" from "Data Visualization". Put the component at one suitable position, and in the property window, select needed register type, address, data type, and register address spacing.

Note: the space between register addresses should be consistent with the space between register each channel.





C. "Property" Description



Base feature	
Object ID	1
Object Name	PieChart1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Data type	FLOAT32
Register address inte...	1
Label	
Channel num	4
Format text(0)	...
Format text(1)	...
Format text(2)	...
Format text(3)	...
Appearance	
Transparency	100
If Draw Edge	False
Start Angle	0
End Angle	0
Direction	Clockwise
Center diameter	0%
Data display type	Percentage
Dotes	0
Action	
Msg trigger call sub	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Data Type	Set the progress value's data type, default is FLOAT (floating type 32-bit), generally make it consistent with register type.
Register Address Interval	Set the interval of bound registers.
Label	
Channel Num	Set how many channels displayed.
Format text	Set the text displayed, you can set each channel's color, font color,etc.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc.

	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whetehr to draw the edge, default is False.
Edge Color	Set the edge color, it can be set when "If Draw Edge" is True.
Start Angle	Set the pie chart's staring speed. 0~359. When the start angle = end angle, which means one full circle.
End Angle	Set the pie chart's end speed. 0~359. When the start angle = end angle, which means one full circle.
Direction	Set the pie chart's direction, clockwise / anti-clockwise.
Center Diameter	Set the circle center size. When it is 0, it is circle chart, when it is not 0, it is ring chart.
Data Display Type	Set the data display type, there are value / percent.
Dotes	Set the decimal dotes part, 0~6, default is 0.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

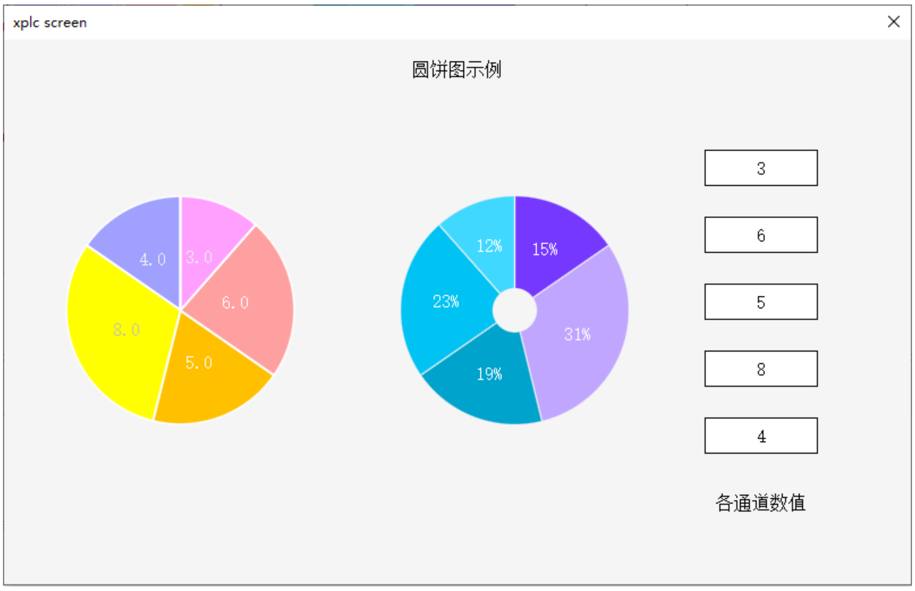
D. For Example

Put two "pie charts" and multiple "value show" elements in the HMI window; the first pie chart displays values clockwise, and the second pie chart displays percentages counterclockwise.

The "value show" components are bound to the register TABLE (DT). Different components are assigned register addresses with intervals of 10, i.e., TABLE(0), TABLE(10), TABLE(20), and so on, and the data is stored in the corresponding register.

The "pie charts" components are bound to the same registers as the "value show" components, with register address intervals set to 10, and the number of channels is set to the same number as the number of value show elements (5 in this example). For other properties, can be configured according to requirements, such as setting the data display type to value or percentage.

Effect:



4.3.36. Bar Chart

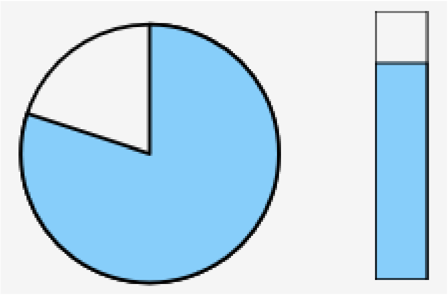
A. What is It?

The data in the register is represented by a bar chart as a percentage. Each bar chart represents only one object, and multiple bar charts can be combined to form a histogram.

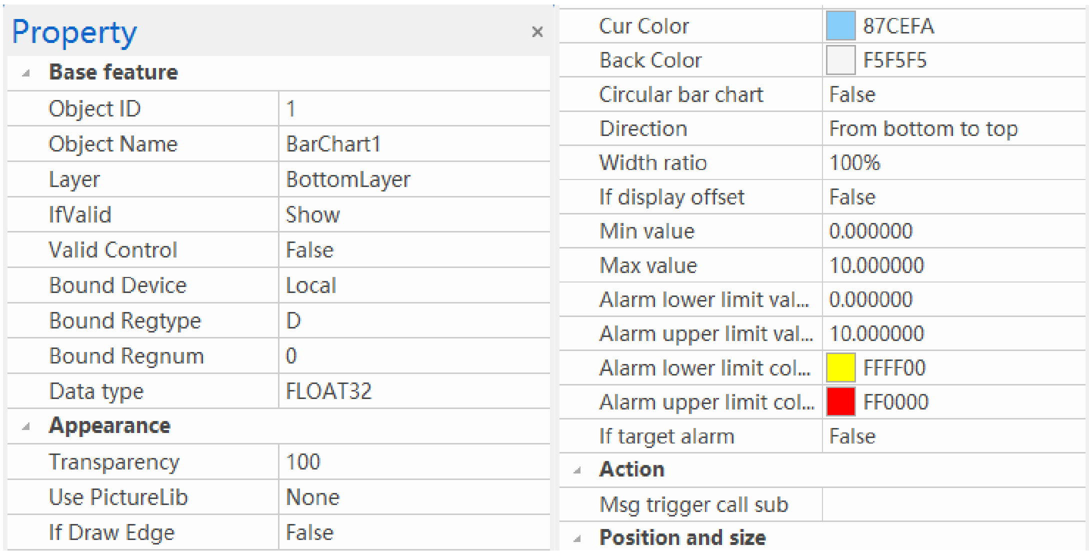
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click “Tool Box”, select “Bar Chart” from “Data Visualization”. Put the component at one suitable position, and in the property window, select needed register type, address, alarm limits, and the displayed colors (please check the example).



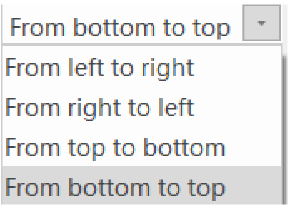
C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	BarChart1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Data type	FLOAT32
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	False
Cur Color	🟦 87CEFA
Back Color	⬜ F5F5F5
Circular bar chart	False
Direction	From bottom to top
Width ratio	100%
If display offset	False
Min value	0.000000
Max value	10.000000
Alarm lower limit val...	0.000000
Alarm upper limit val...	10.000000
Alarm lower limit col...	🟨 FFFF00
Alarm upper limit col...	🟥 FF0000
If target alarm	False
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.

<table>
<tr><td>Bound Device</td><td>Assigned device, default is local</td></tr>
<tr><td>Bound Regtype</td><td>Select register type, you can select from the list</td></tr>
<tr><td>Bound Regnum</td><td>Set register No., values obtained from register, then control different states of component.</td></tr>
<tr><td>Data Type</td><td>Set the progress value’s data type, default is FLOAT (floating type 32-bit), generally make it consistent with register type.</td></tr>
<tr><td colspan="2"><b>Appearance</b></td></tr>
<tr><td>Transparency</td><td>Set the component’s transparency, which includes current color, background color, edge color, etc.
Range: 0% ~ 100% (default 100%)
0% -- fully transparent, 100% -- fully show</td></tr>
<tr><td>Use Picture Lib</td><td>None / use picture library / use back picture</td></tr>
<tr><td>If Draw Edge</td><td>Whether to draw the edge.</td></tr>
<tr><td>Edge Color</td><td>Set the edge color. It is valid when “If Draw Edge” is True.</td></tr>
<tr><td>Cur Color</td><td>Set the color of the loading progress.</td></tr>
<tr><td>Back Color</td><td>Set the color of the unloaded progress.</td></tr>
<tr><td>Circular Bar Chart</td><td>Convert now progress bar to circular type?
False: default, keep current bar type.
True: convert to circular progress, and you can set below:
<table>
<tr><td>Start Angle</td><td>Set the pie chart’s staring speed. 0~359. When the start angle = end angle, which means one full circle.</td></tr>
<tr><td>End Angle</td><td>Set the pie chart’s end speed. 0~359. When the start angle = end angle, which means one full circle.</td></tr>
<tr><td>Direction</td><td>Set the pie chart’s direction, clockwise / anti-clockwise.</td></tr>
<tr><td>Center Diameter</td><td>Set the circle center size.</td></tr>
</table></td></tr>
<tr><td>Direction</td><td>Set the bar chart’s loading directions, there are 4 choices:

</td></tr>
<tr><td>Width Ratio</td><td>Set the percentage between displayed width and component width.</td></tr>
<tr><td>If Display Offset</td><td>Whether to make the display offet? If True, it will offset from original position.</td></tr>
<tr><td>Min Value</td><td>Set the minimal value of the bar chart.</td></tr>
<tr><td>Max Value</td><td>Set the maximum value of the bar chart.</td></tr>
</table>

<table>
<tr><td>Alarm Lower Limit Value</td><td>Set the minimal value for alarm. When the data in register < this value, the bar chart' color will be "alarm lower limit color".</td></tr>
<tr><td>Alarm Upper Limit Value</td><td>Set the maximum value for alarm. When the data in register > this value, the bar chart' color will be "alarm upper limit color".</td></tr>
<tr><td>Alarm Lower Limit Color</td><td>Set the color for "Alarm Lower Limit Value".</td></tr>
<tr><td>Alarm Upper Limit Color</td><td>Set the color for "Alarm Upper Limit Value".</td></tr>
<tr><td>If Target Alarm</td><td>Whether to show alarm when it reached the target value?
Flase: No
True: yes, and you should set below parameters.
<table>
<tr><td>Target Value</td><td>Set the alarm target value.</td></tr>
<tr><td>Allowable Error</td><td>Set the allowable error of the target value.</td></tr>
<tr><td>Target Alarm Color</td><td>Set the color.</td></tr>
</table></td></tr>
<tr><td colspan="2">Action</td></tr>
<tr><td>Msg Trigger Call Sub</td><td>Set the SUB function that is to be called when the message is triggered by configured component triggering mode.
It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened</td></tr>
<tr><td colspan="2"><b>Position &amp; Size</b></td></tr>
<tr><td>Left</td><td rowspan="2">Horizontal and vertical starting position of the component (&lt;horizontal & vertical resolution)</td></tr>
<tr><td>Top</td></tr>
<tr><td>Width</td><td>Width of the component</td></tr>
<tr><td>Height</td><td>Height of the component</td></tr>
</table>

D. For Example

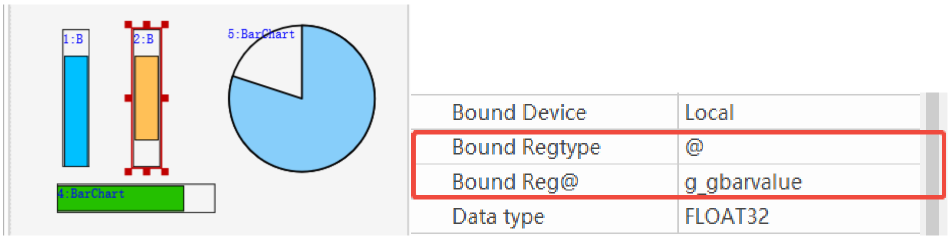
Step 1: add "bar chart" and "timer" components, and you could choose bar type or circle type, set needed alarm min and max values, colors, target alarm, etc.

Step 2: edit Basic program:

```
GLOBAL dim g_gbarvalue 'define progress value variable
GLOBAL dim g_bifplus    'increment / decrement Flag
GLOBAL dim g_bifauto    'auto-flag
g_gbarvalue = 0
g_bifplus = 1
g_bifauto = 1
end
GLOBAL sub sub_timer()
    if g_bifauto=1 then
```

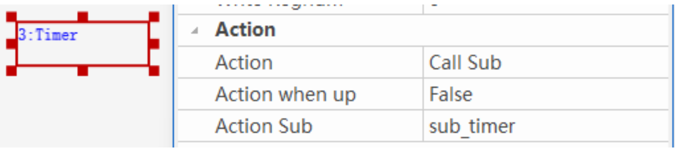
```
        if g_bifplus=1 then
            g_gbarvalue = g_gbarvalue + 1
            if g_gbarvalue > 100 then     'accelerate to max value, then it starts to decelerate
                g_gbarvalue = 100
                g_bifplus = 0
            endif
        else
            g_gbarvalue = g_gbarvalue - 1
            if g_gbarvalue < 0 then       'decelerate to min value, then it starts to accelerate
                g_gbarvalue = 0
                g_bifplus = 1
            endif
        endif
    endif
end sub
```

Step 3: bind register / variable in “bar chart” property window (here uses variable “g_gbvarvalue”).



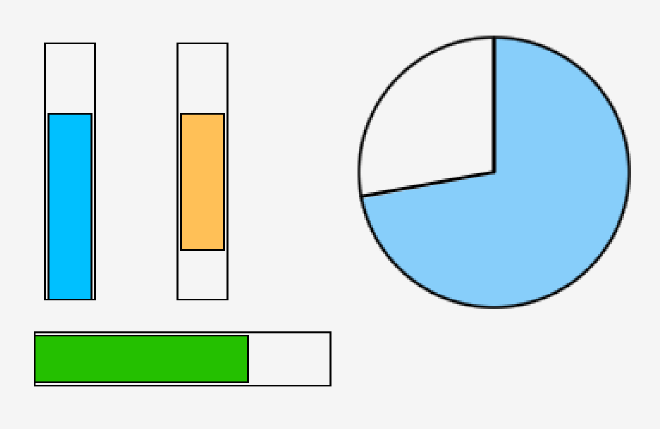
Bound Device	Local
Bound Regtype	@
Bound Reg@	g_gbarvalue
Data type	FLOAT32

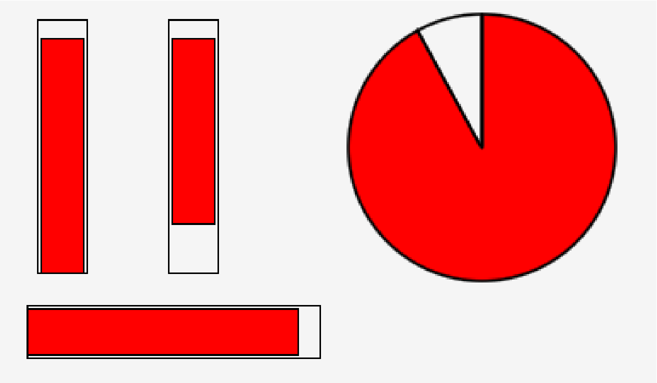
Step 4: for the timer component, set “callsub” in property “action”, and choose corresponding subfunction name (here uses sub_timer).



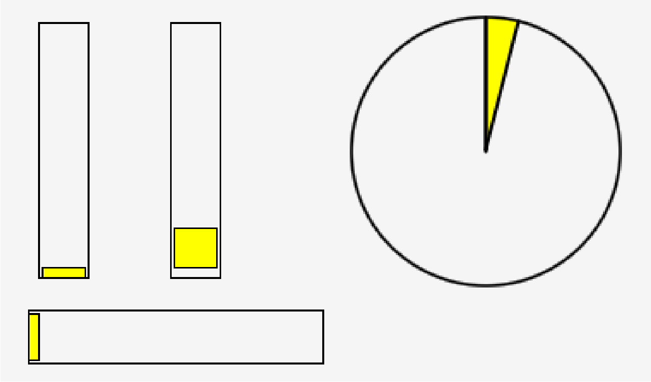
Action	
Action	Call Sub
Action when up	False
Action Sub	sub_timer

Step 4: after downloading into controller / simulator, *check the effect.*





(when it reaches alarm upper limit value)



(when it reaches alarm lower limit value)

4.3.37. XY Chart

A. What is It?

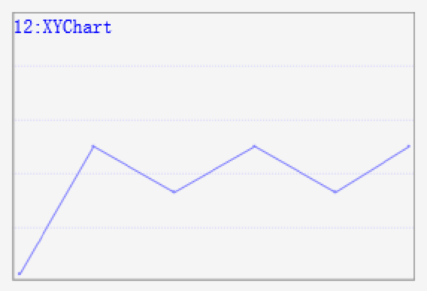
Display 2D coordinate XY data points as curves. Each data point contains X and Y values, they are written using the HMI_XYChartWrite. It supports displaying up to 16 sets of curves simultaneously, which is convenient for you to observe and analyze the various registers' data.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

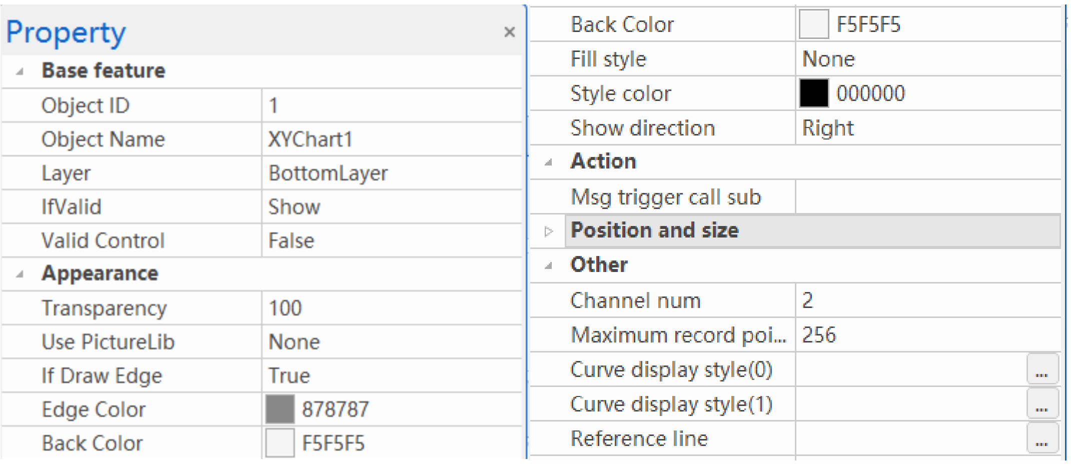
B. How to Use?

Click "Tool Box", select "XY Chart" from "Data Visualization". Put the component at one suitable position, and in the property window, set the appearance and channel numbers, also, set each channel property separately, including displayed format, color, min & max ranges.

Note: there is no scale, if you need the scale displaying, please use together with “Dynamic Scale”.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	XYChart1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True
Edge Color	878787
Back Color	F5F5F5
Back Color	F5F5F5
Fill style	None
Style color	000000
Show direction	Right
Action	
Msg trigger call sub	
Position and size	
Other	
Channel num	2
Maximum record poi...	256
Curve display style(0)	...
Curve display style(1)	...
Reference line	...

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading

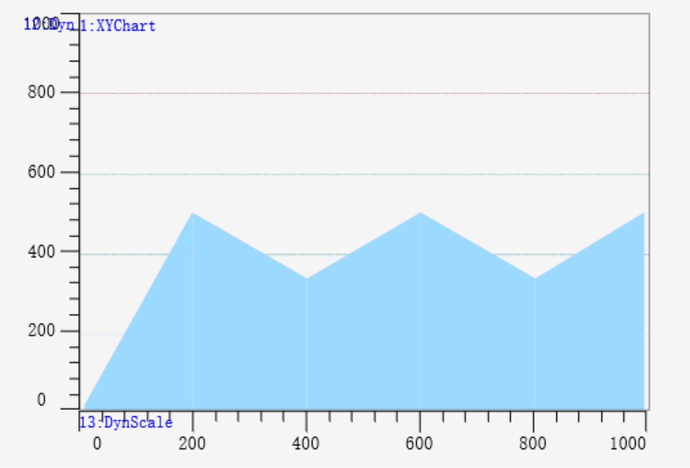
	● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the edge.
Edge Color	Set the edge color. It is valid when “If Draw Edge” is True.
Back Color	Set the background color of the component.
Fill Style	Choose the background fill style, color gradient is supported.
Style Color	Set the fill style’s color.
Show Direction	Set the XY curve’s displayed color, there are 4 kinds: right, left, up, and down.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Postion & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	

Width	Width of the component
Height	Height of the component
Others	
Channel Num	Set the channel numbers, up to 16.
Maximum Record Point Count	Set the max points that can be recorded in each channel. Range: 2~1024.
Curve Display Style (0)	Set the display style of each channel's curve.
Reference Line	Set the background reference curve for the XY curve.

D. For Example

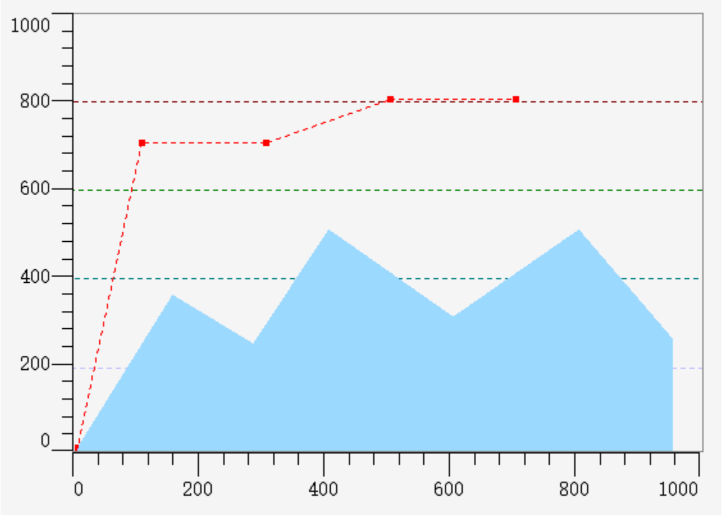
Step 1: add "XY chart" and "Dynamic Scale" components, make them as XY coordinates graphic. And set needed channel numbers and max points. Also, you could set the style for each channel separately. *Here, 2 curve channels are set, and channel 0 shows in the way of mapping towards axis X, channel 1 shows in the way of drawing one line by points.*

Step 2: for "dynamic scale" component, please add 2 straight types, one is used as Y coordinate (starting value is 1000, end value is 0), another is used for X coordinate (starting value is 0, end value is 1000), and set the main scale line width is 6, set the sub width is 4.



Step 3: edit Basic program, use HMI_XYChartClear command to clear data, then use HMI_XYChartWrite command to write data. After that, download into controller / simulator.

```
HMI_XYChartClear(10, 1, -1)
        "write default XY data
        HMI_XYChartWrite(10, 1, 0, 0, 0)
        HMI_XYChartWrite(10, 1, 0, 150, 350)
        HMI_XYChartWrite(10, 1, 0, 280, 240)
        HMI_XYChartWrite(10, 1, 0, 400, 500)
        HMI_XYChartWrite(10, 1, 0, 600, 300)
        HMI_XYChartWrite(10, 1, 0, 800, 500)
        HMI_XYChartWrite(10, 1, 0, 950, 250)
        HMI_XYChartWrite(10, 1, 1, 0, 0)
        HMI_XYChartWrite(10, 1, 1, 101, 700)
        HMI_XYChartWrite(10, 1, 1, 300, 700)
        HMI_XYChartWrite(10, 1, 1, 500, 800)
        HMI_XYChartWrite(10, 1, 1, 700, 800)
end
```



4.3.38. Trend Chart

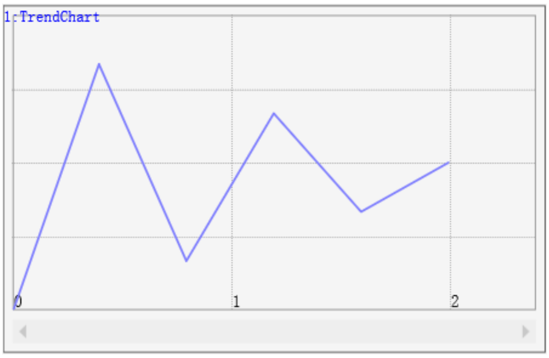
A. What is It?

The component makes the real-time collected data into a graph using continuous line segments for data analysis. It needs to be used in conjunction with "resource sampling library".

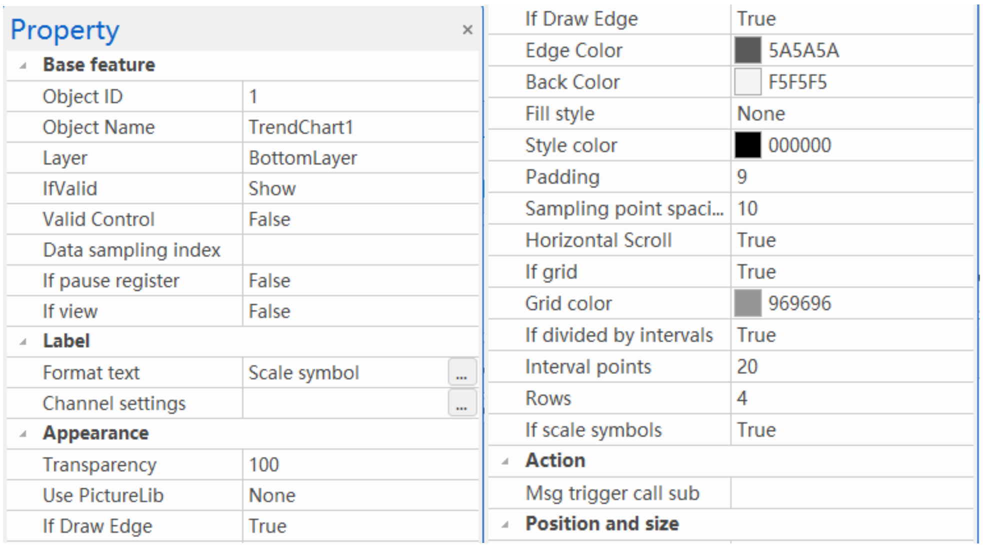
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click "Tool Box", select "Trend Chart" from "Data Visualization". Put the component at one suitable position, and in the property window, set the appearance, and select the existed resource sampling index. If you need to use "pause" to control register, then set "If pause register" as True, and select bound register. If you need to view needed data, set “If view” as True, and bind the register.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	TrendChart1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Data sampling index	
If pause register	False
If view	False
Label	
Format text	Scale symbol
Channel settings	
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True

If Draw Edge	True
Edge Color	■ 5A5A5A
Back Color	□ F5F5F5
Fill style	None
Style color	■ 000000
Padding	9
Sampling point spaci...	10
Horizontal Scroll	True
If grid	True
Grid color	■ 969696
If divided by intervals	True
Interval points	20
Rows	4
If scale symbols	True
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading.

	● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register's variables.
Data Sampling Index	Select the data sampling index from the resource sampling library.
If Pause Register	Whether to open "pause" to control the register.
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., when the register type is not @.
Bound Reg@	Set register No., when the register type is @.
If View	Whether to open "view & detect" function. Flase: No True: Yes, then one vertical detection line will be made, and the data above the line will be output to assigned register.
View Line Color	Select the view line color.
Write Device	Set the device No., default is Local.
Write Regtype	Select the register type from the list.
Write Regnum	Select register No. starting address, it can be multiple.
Label	
Format Text	Set the component displayed text.
Channel Settings	➢ Color: set the displayed color of the channel segment ➢ Line Width: set the line width of the channel segment ➢ Line Type: set the line type of the channel segment ➢ Min: set the channel displayed minimal range ➢ Max: set the channel displayed maximum range
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%)

	0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the edge.
Edge Color	Set the edge color. It is valid when “If Draw Edge” is True.
Back Color	Set the background color of the component.
Fill Style	Choose the background fill style, color gradient is supported.
Style Color	Set the color for the filled background.
Padding	Set the distance between component displayed contents and component edge.
Sampling Point Spacing	Set the pixel distance between two sampling points.
Horizontal Scroll	Whether to use horizontal scroll, which is displayed at the same width below the component graphic area, used to control the movement of the group line chart.
If Grid	Whether to show the grid.
Grid Color	Set the grid color.
If Divided by Intervals	Whether the axis X is divided by the interval.
Interval Points	Set how many sampling points between 2 vertical grid lines.
Rows	Set the grid rows. Divide the vertical width of the component into N equal parts, drawing a total of N-1 horizontal lines. Range: 1~500. Default: 4.
If Scale Symbols	Whether to show the horizontal coordinate scale symbol.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

D. For Example: please check Chapter VIII How to Use Trend Chart

4.3.39. Data Group Show

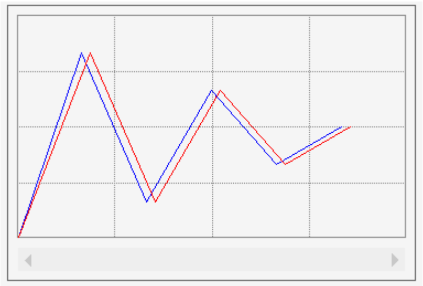
A. What is It?

A data group (or block) refers to data within a contiguous set of addresses. Axis X = address, axis Y = data. The [Data Group Show] can simultaneously display the contents of multiple data groups in a line chart format for easy observation.

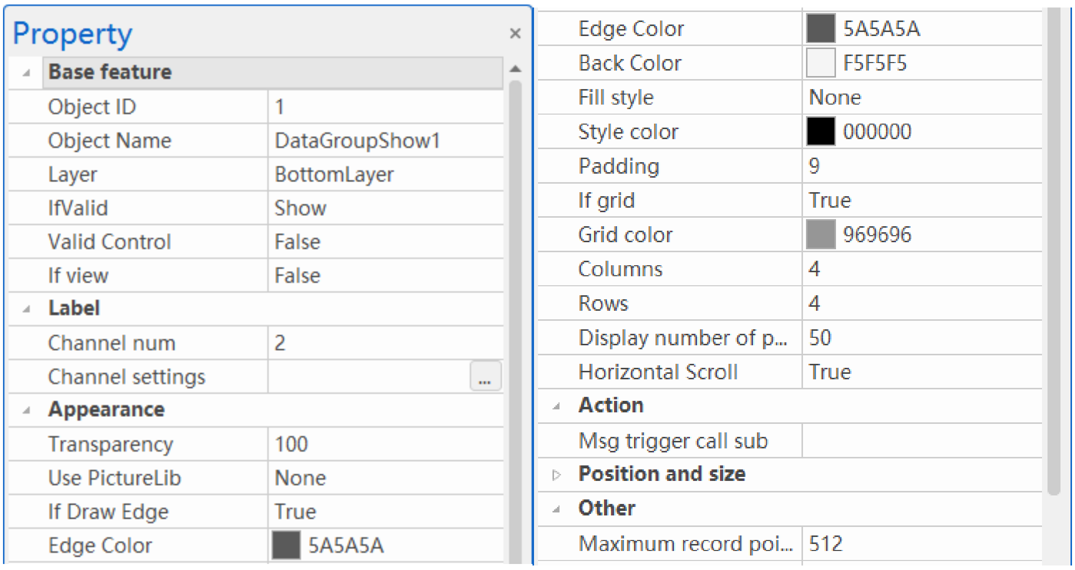
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click “Tool Box”, select “Data Group Show” from “Data Visualization”. Put the component at one suitable position, and in the property window, set the appearance. If you need to view needed data, set “If view” as True, and bind the register. At last, set the channel numbers, etc.



C. “Property” Description



Base feature	
Object ID	1
Object Name	DataGroupShow1
Layer	BottomLayer
IfValid	Show
Valid Control	False
If view	False
Label	
Channel num	2
Channel settings	...
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 5A5A5A
Back Color	□ F5F5F5
Fill style	None
Style color	■ 000000
Padding	9
If grid	True
Grid color	■ 969696
Columns	4
Rows	4
Display number of p...	50
Horizontal Scroll	True
Action	
Msg trigger call sub	
Position and size	
Other	
Maximum record poi...	512

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register's variables.
If View	Whether to open "view & detect" function. Flase: No True: Yes, then one vertical detection line will be made, and the data above the line will be output to assigned register.
View Line Color	Select the view line color.
Write Device	Set the device No., default is Local.
Write Regtype	Select the register type from the list.
Write Regnum	Select register No. starting address, it can be multiple.
Label	
Channel Num	Set the displayed channel numbers. Max is 12.
Channel Settings	➢ Color: set the displayed color of the channel segment ➢ Line Width: set the line width of the channel segment

	➢ Line Type: set the line type of the channel segment ➢ Min: set the channel displayed minimal range ➢ Max: set the channel displayed maximum range
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the edge.
Edge Color	Set the edge color. It is valid when "If Draw Edge" is True.
Back Color	Set the background color of the component.
Fill Style	Choose the background fill style, color gradient is supported.
Style Color	Set the color for the filled background.
Padding	Set the distance between component displayed contents and component edge.
If Grid	Whether to show the grid.
Grid Color	Set the grid color.
Columns	Set the grid columns. Divide the horizontal width of the component into N equal parts, drawing a total of N-1 horizontal lines. Range: 1~500. Default: 4.
Rows	Set the grid rows. Divide the vertical width of the component into N equal parts, drawing a total of N-1 horizontal lines. Range: 1~500. Default: 4.
Display Number of Points	Sets the maximum number of data points of graphics that can be displayed on a single page for a given element.
Horizontal Scroll	Whether to use a horizontal scroll bar. It displays the component graphic area at a constant width, used to control the movement of the grouped line chart.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB

	function is only triggered when related message is opened.
Postion & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component
Other	
Maximum Record Point Count	Set the max points that can be recorded in each channel. Range: 2~1024.

D. For Example:

Step 1: in the Basic program, set two global variables: g_iCurChannel (current channel) and g_iWriteVal (write value). And write functions for initializing group test data, writing data, and clearing data.

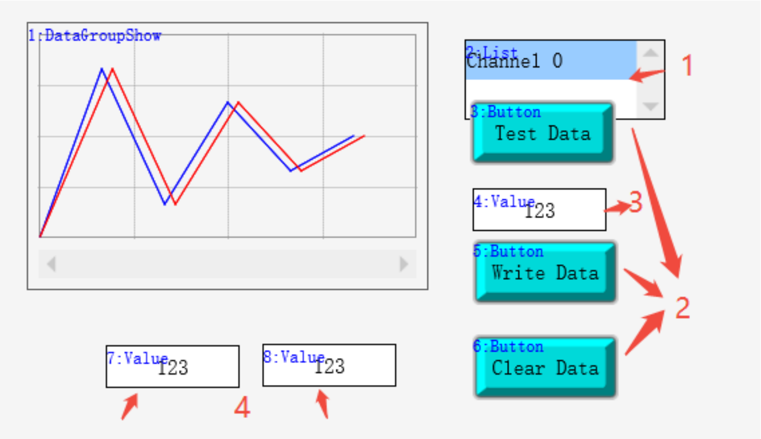
```
global dim g_iCurChannel
global dim g_iWriteVal
g_iCurChannel = 0
end
""""test, initialize group data
GLOBAL sub sub_TestInitGroupData()
local i
    for i=0 TO 100
    table(1000 + i) = SIN(PI*i/10 + g_iCurChannel*PI/4) * 500
    ' table(1000 + i) = ABS(((i*TICKS) MOD 1000))
        NEXT
HMI_GDataWrite(10, 1, g_iCurChannel, 1000, 100)
end sub
""""write data
GLOBAL sub sub_WriteData()
    HMI_GDataWrite(10, 1, g_iCurChannel, g_iWriteVal)
end sub
""""clear data
GLOBAL sub sub_ClearData()
HMI_GDataClear(10, 1, g_iCurChannel)
end sub
```

Step 2: in HMI interface, add “data group show” component, and in its property window, set corresponding appearance, set “if view” as True, and please select the register starting No. for writing corresponding view data. Also, for “Channel Num”, here, set it as 2.

Step 3: in HMI interface, add one "list" component, and bind it with customized register g_iCurChannel to select the channel.

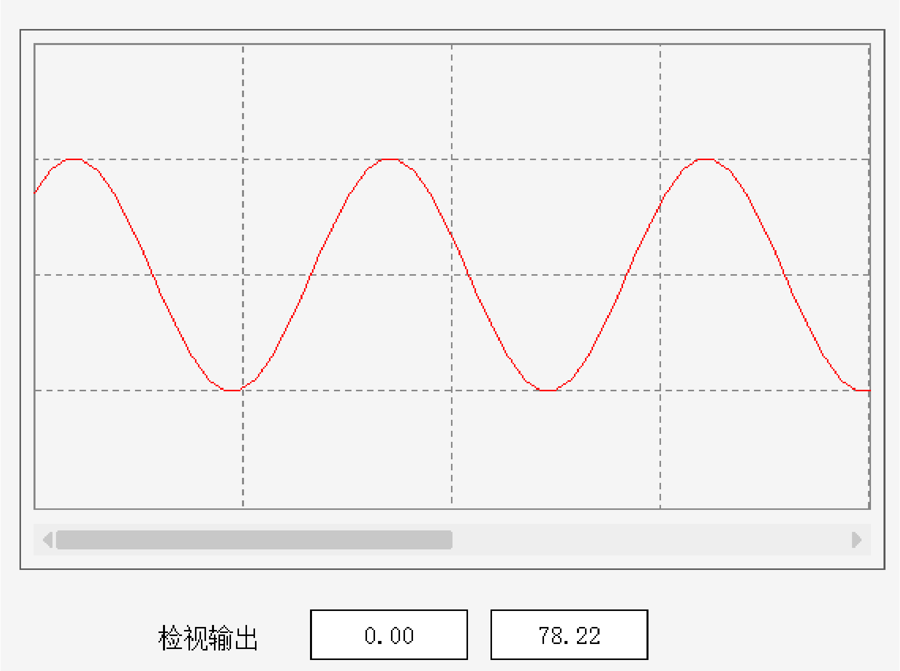
Step 4: in HMI interface, add 3 "button" components to call corresponding SUB functions.

Step 5: in HMI interface, add 3 "value show" components, bind one with "g_iWriteVal" to write data, bind another 2 with DT (0) and DT (1) separately to show 2 channel's view data.



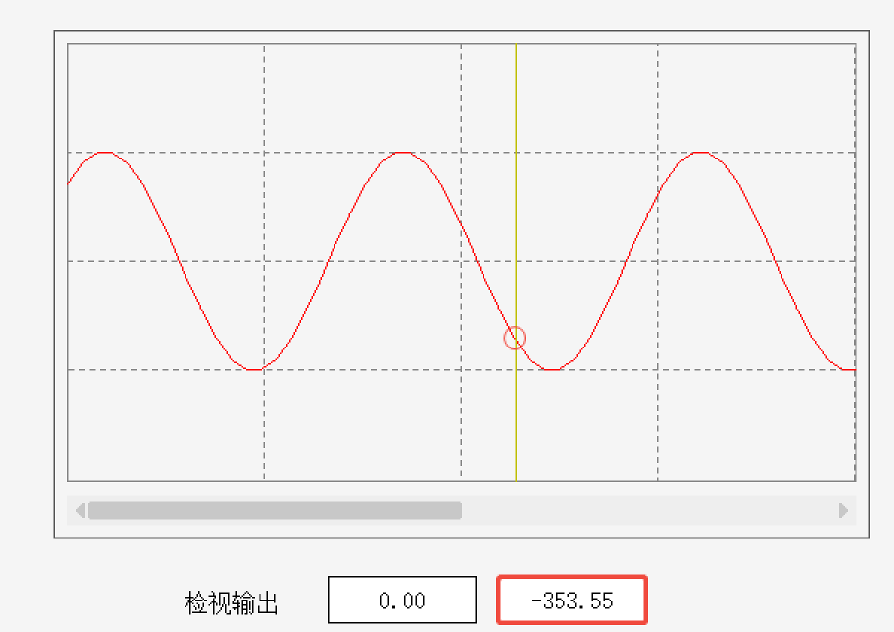
Effect:

1) select channel, click "test data", it will show one data group relation diagram. And the first "view data" value is channel 0 data, the second is channel 1 data.

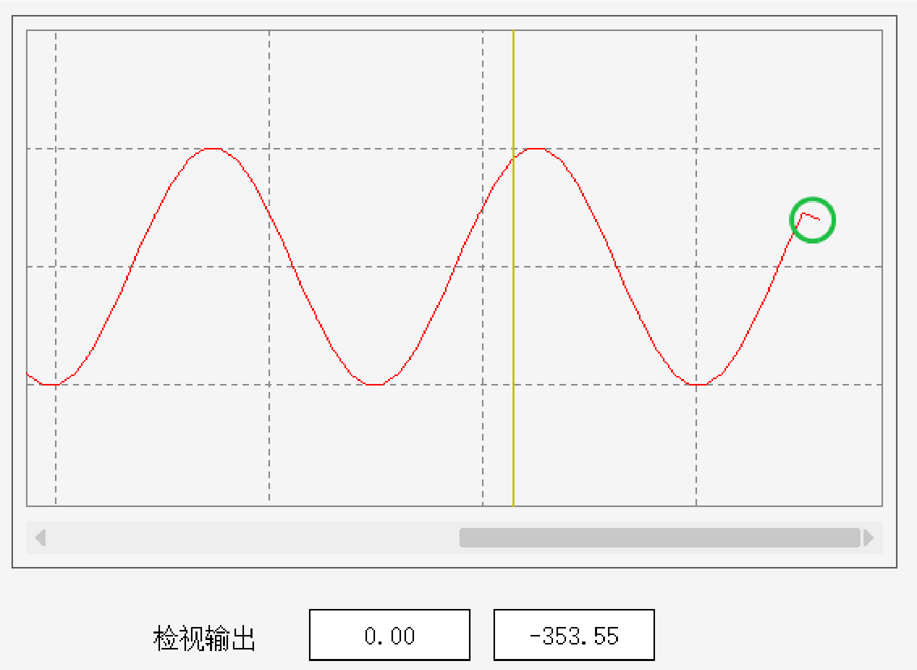


2) Click any position, one yellow line will appear, and the intersection data of the yellow line and

the graphic will be saved to the corresponding register.



3) It can import new data through “write data” button.



4) It can clear data through “clear data” button.

4.3.40. CAD

A. What is It?

Used to show vector graphics.

Note: it is only valid when your controller has Linux system (VPLC5XX series).

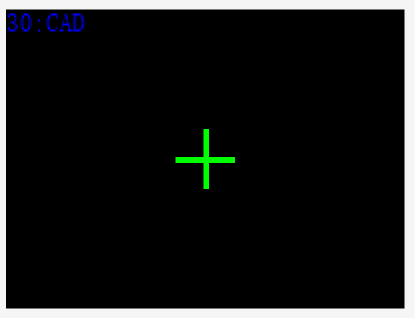
B. How to Use?

Click “Tool Box”, select “CAD” from “Data Visualization”. Put the component at one suitable position, and select the channel No. for importing file 3 (there are channel 0 and 1), then set layer, path, empty motion color, etc. For import / close graphic, please use together with CAD commands.

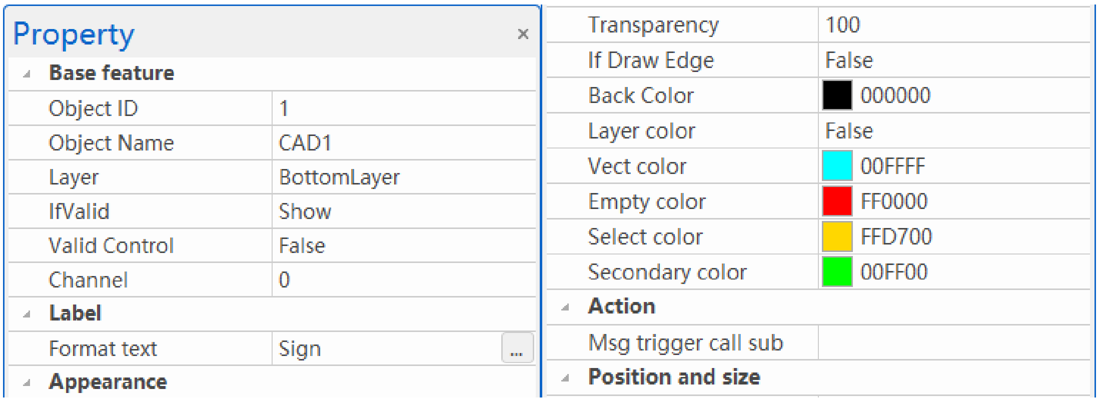
Notes:

a. The vector graphic must be saved into flash folder.

b. Now supported graphic format: .dxf/.ai/.dst/.nc/.cnc.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	CAD1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Channel	0
Label	
Format text	Sign
Appearance	
Transparency	100
If Draw Edge	False
Back Color	⬛ 000000
Layer color	False
Vect color	🟦 00FFFF
Empty color	🟥 FF0000
Select color	🟨 FFD700
Secondary color	🟩 00FF00
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list

	● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Channel	Set channel No., only 2 chanels 0 / 1.
Label	
Format Text	Set the component displayed text’s format.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whetehr to draw the edge, default is False.
Back Color	Select background color, default is black.
Layer Color	Whether to use layer color, when TRUE, different layers are with different colors.
Vect Color	Select trajectory color.
Empty Color	Select empty motion color (The displacement between incoherent areas is empty displacement.)
Select Color	Select the color
Secondnary Color	Select auxuliary color.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

For Quick Start CAD, please refer to Chapter VI How to Import Vector Graphic by CAD.

4.3.41. Meter

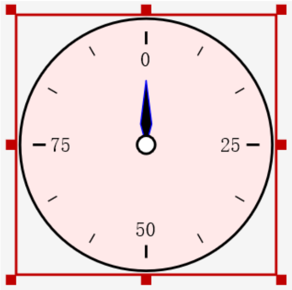
A. What is It?

Used to point out the current data value in the register in the way of the instrument chart.

Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

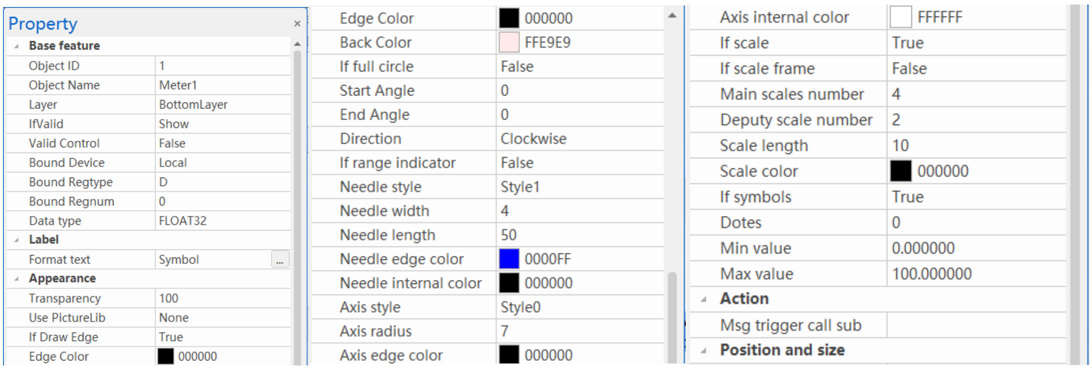
B. How to Use?

Click “Tool Box”, select “Meter” from “Data Visualization”. Put the component at one suitable position, and in the property window, bind the register, and set starting angle, end angle, etc., also, please select whether to show the rangle indicator (if rangle indicator).



Note: width and height are forced to be the same. If the width is modified, the height will change accordingly, but the height cannot be modified.

C. “Property” Description



Property

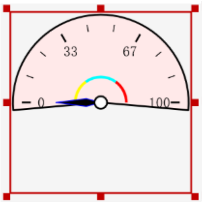
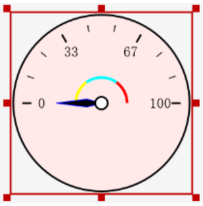
Base feature	
Object ID	1
Object Name	Meter1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Data type	FLOAT32
Label	
Format text	Symbol
Appearance	
Transparency	100
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 000000

Edge Color	■ 000000
Back Color	□ FFE9E9
If full circle	False
Start Angle	0
End Angle	0
Direction	Clockwise
If range indicator	False
Needle style	Style1
Needle width	4
Needle length	50
Needle edge color	■ 0000FF
Needle internal color	■ 000000
Axis style	Style0
Axis radius	7
Axis edge color	■ 000000

Axis internal color	□ FFFFFF
If scale	True
If scale frame	False
Main scales number	4
Deputy scale number	2
Scale length	10
Scale color	■ 000000
If symbols	True
Dotes	0
Min value	0.000000
Max value	100.000000
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.

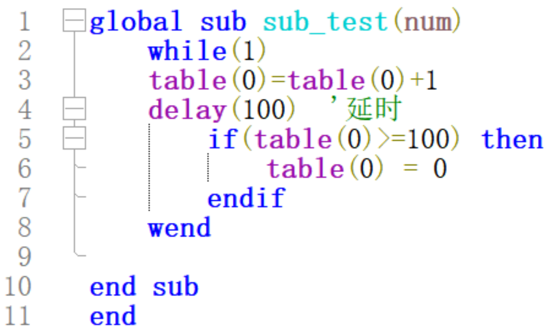
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register's variables.
Data Type	Set the value's data type, default is FLOAT32 (floating type 32-bit), generally please make it consistent with register type.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Edge	Whether to draw the egde.
Edge Color	Set the edge color.
Back Color	Set the background color of the component.
If Full Circle	Whether to make the component show one full circle.

	 No  Yes
Start Angle	Set the starting and end angles. They are 0° (default) when it is at 12:00, the range is 0~359. And when they are equal, one full circle.
End Angle	
Direction	Set the direction, clockwsie / anti-clockwise.
If Rangle Indicator	Whether to show the range mark. If Yes (True), there are 4 features: ➢ Lower Limit Value: set the lower limit value, default is 30. The lower limit range: min value ~ lower limit value. ➢ Upper Limit Value: set the upper limit value, default is 60. The lower limit range: upper limit value ~ max value. ➢ Lower Limit Color: set the range indicator (lower) color ➢ Upper Limit Color: set the range indicator (upper) color ➢ Color in Range: set the range indicator (normal range) color ➢ Display Radius: set the range indicator's displayed radius ➢ Diaply Width: set the range indicator's displayed width
Needle Style	Set the needle style, there are 5 styles. Default style is 1.
Needle Width	Set the meter needle width.
Needle Length	Set the meter needle length.
Needle Edge Color	Set the meter needle's edge color.
Needle Internal Color	Set the meter needle's internal color.
Axis Style	Set the meter axis center style, there are 2 styles.
Axis Radius	Set the meter axis center radius.
Axis Edge Color	Set the meter axis edge color.
Axiss Internal Color	Set the meter axis internal color.
If Scale	Set whether to show the scale.
If Scale Frame	Set whether to show the scale frame
Main Scales Number	Set the main scale numbers. Range: 0~100. Default is 4.
Deputy Scales Number	Set the deputy scale numbers. Range: 0~100. Default is 2.
Scale Length	Set the main scale length. The deputy scale length is 4/5 of the main scale, which can't exceed the radius of the component. Default: 10.
Scale Color	Set the scale showing ccolor, default RGB (0, 0, 0).
If Symbols	Set whether to show the scale symbol of the main scale.
Dotes	Set the decimal part dotes. Range: 0~8. Default: 0.

Min Value	Set the meter’s minimal value. Default is 0.
Max Value	Set the meter’s maximum value. Default is 100.
Label	
Format Text	Set the component label’s font and format.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position and Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

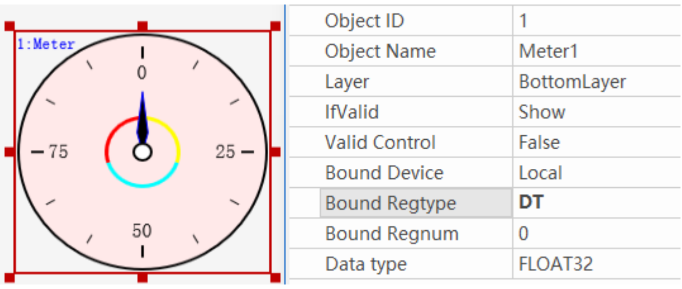
D. For Example:

Step 1: in Basic program, edit one loop, table(0) increments cyclically every 100ms.



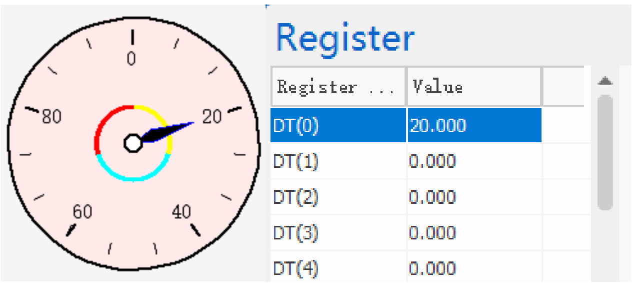
```
global sub sub_test(num)
    while(1)
    table(0)=table(0)+1
    delay(100)  ’延时
        if(table(0)>=100) then
            table(0) = 0
        endif
    wend

end sub
end
```

Step 2: in HMI interface, add “meter” component, and in its property window, bind it to table(0).



Object ID	1
Object Name	Meter1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	DT
Bound Regnum	0
Data type	FLOAT32

Effect: the meter display value will change as the bound register value changes.



Register

Register ...	Value
DT(0)	20.000
DT(1)	0.000
DT(2)	0.000
DT(3)	0.000
DT(4)	0.000

4.3.42. Dynamic Scale

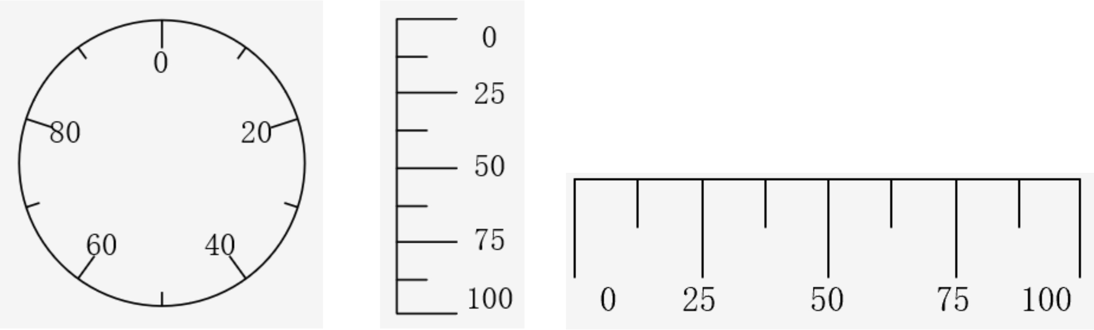
A. What is It?

Used to display different scale styles and dynamically adjust the scale range to provide scales for other components. And there are main scale and deputy scale.

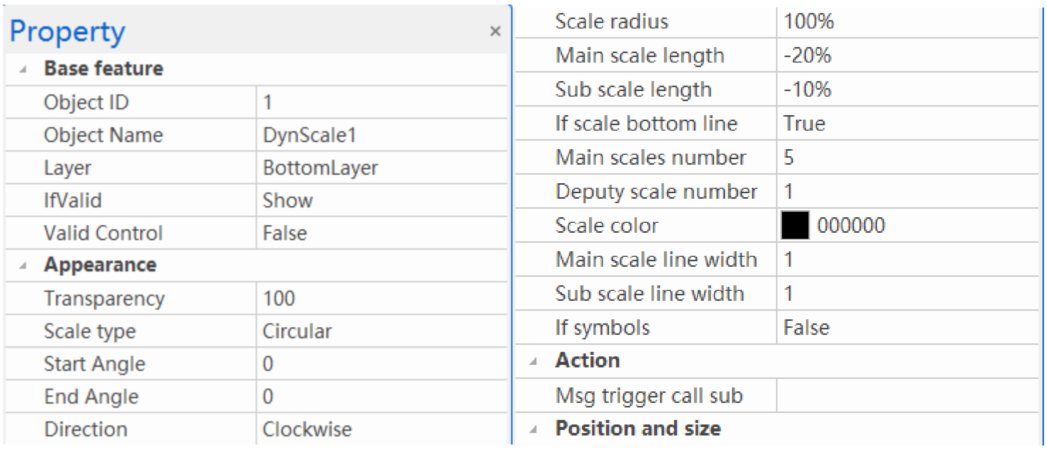
Note: please use RTSys V1.3.01 and above version, and RTHmi should be V1.3.1 and above.

B. How to Use?

Click "Tool Box", select "Dynamic Scale" from "Data Visualization". Put the component at one suitable position, and in the property window, set the scale type, including direction, main scale, deputy scale.



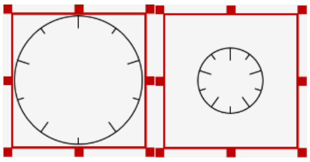
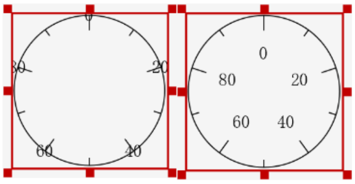
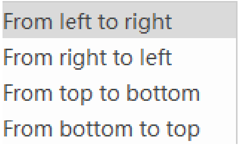
C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	DynScale1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Scale type	Circular
Start Angle	0
End Angle	0
Direction	Clockwise
Scale radius	100%
Main scale length	-20%
Sub scale length	-10%
If scale bottom line	True
Main scales number	5
Deputy scale number	1
Scale color	■ 000000
Main scale line width	1
Sub scale line width	1
If symbols	False
Action	
Msg trigger call sub	
Position and size	

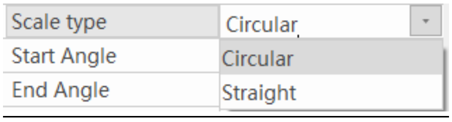
Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register's variables.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc.

	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Scale Type	There are 2 scale types. 1) **Circular** (below properties can be set) ● Start & End angle: set the starting and end angles. They are 0° (default) when it is at 12:00, the range is 0~359. And when they are equal, one full circle. ● Direction: set the direction, clockwsie / anti-clockwise. ● Scale radius: set the scale radius, range: 0% ~ 100%, defaulit is 100. For example, radius = 100% VS = 50%.  ● Main scale length: set the main scale length, range (can be decimal part): -100% (towards inside) ~ 100% (torwards outside), the default is -20%. ● Sub scale length: set the sub scale length, range (can be decimal part): -100% (towards inside) ~ 100% (torwards outside), the default is -10%. ● Symbol Radius: set the radius symbol position. Range: 0% ~ 100%. Default is 70%. For Example: radius = 100% VS = 50%.  2) **Straight** (below properties can be set) ● Direction: there are 4 choices.  From left to right From right to left From top to bottom From bottom to top
If Scale Bottom Line	Whether to draw the scale bottom line. True: yes, the type and width refer to main scale configuration.
Main Scales Number	Set the main scale numbers. Range: 2~100. Default is 5.
Deputy Scales Number	Set the deputy scale numbers. Range: 0~100. Default is 1.

Scale Color	Set the scale showing ccolor, default RGB (0, 0, 0).
Main Scale Line Width	Set the main scale line width. Range: 1~20. Default is 1.
Sub Scale Line Width	Set the deputy scale numbers. Range: 1~20. Default is 1.
If Symbols	Set whether to show the scale symbol of the main scale. When True: ● Chares: set character numbers. Range: 1~128. Default: 4. ● Dotes: set the decimal part dotes. Range: 0~8. Default: 0. ● Start Value: set the scale starting value. Default: 0 ● End Value: set the scale end value. Default: 100
Label	
Format Text	Set the component label's font and format.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position and Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

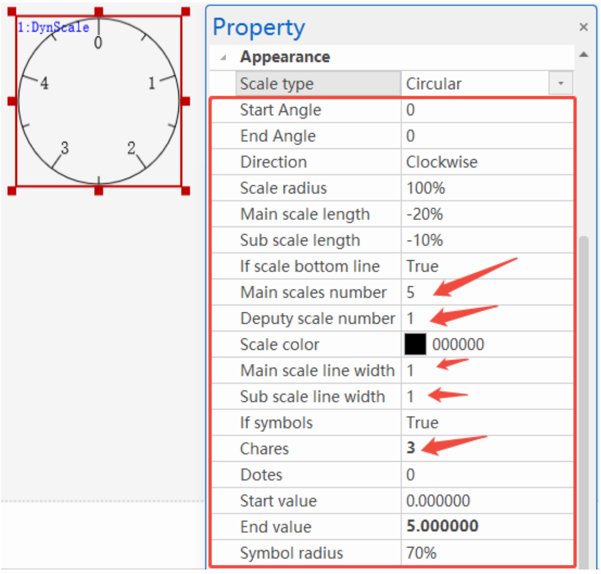
D. For Example:

Step 1: add one "dynamic scale" component, and the type is circular.



Scale type	Circular
Start Angle	
End Angle	

Step 2: set other properties.

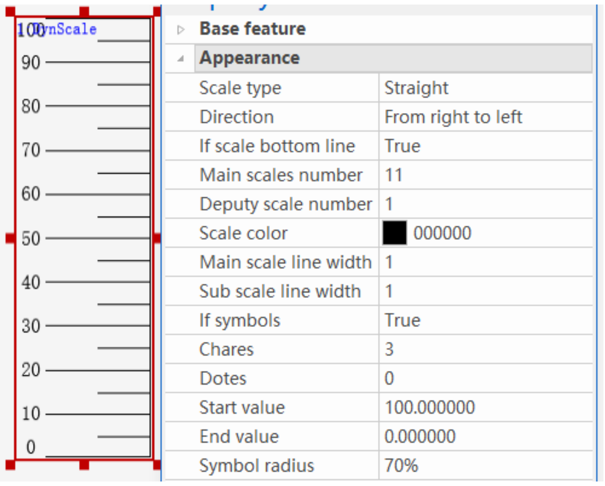


Property

Appearance

Scale type	Circular
Start Angle	0
End Angle	0
Direction	Clockwise
Scale radius	100%
Main scale length	-20%
Sub scale length	-10%
If scale bottom line	True
Main scales number	5
Deputy scale number	1
Scale color	■ 000000
Main scale line width	1
Sub scale line width	1
If symbols	True
Chares	3
Dotes	0
Start value	0.000000
End value	**5.000000**
Symbol radius	70%

Step 3: modify the type as straight, and modify its main scale as 11, range becomes 100~0.

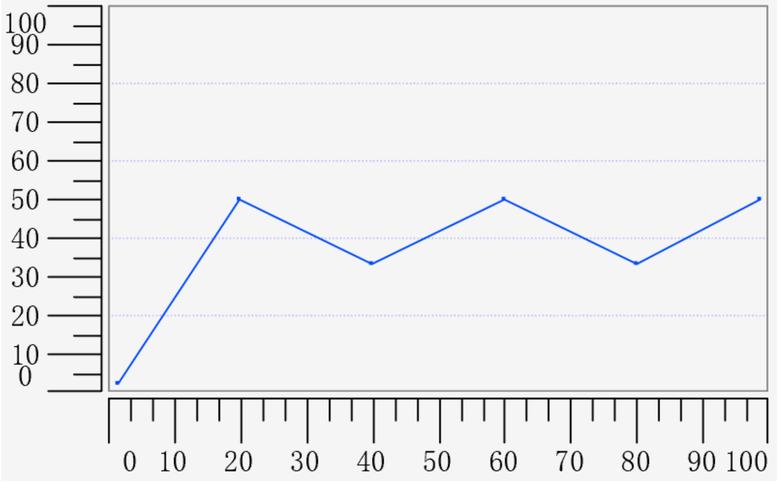


Base feature

Appearance

Scale type	Straight
Direction	From right to left
If scale bottom line	True
Main scales number	11
Deputy scale number	1
Scale color	■ 000000
Main scale line width	1
Sub scale line width	1
If symbols	True
Chares	3
Dotes	0
Start value	100.000000
End value	0.000000
Symbol radius	70%

Effect: set scale for one XY chart.



4.3.43. List

A. What is It?

Used to show multiple states through list. The number of list rows can be set by state num (1-256).

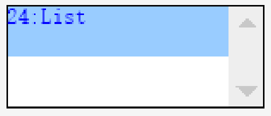
B. How to Use?

Click "Tool Box", select "Dynamic Scale" from "System Navigation". Put the component at suitable position, and in its property window, bind register and No., then set "state num" for the component, enter needed list text. And this component can be together with HMI_LISTTEXTS command.

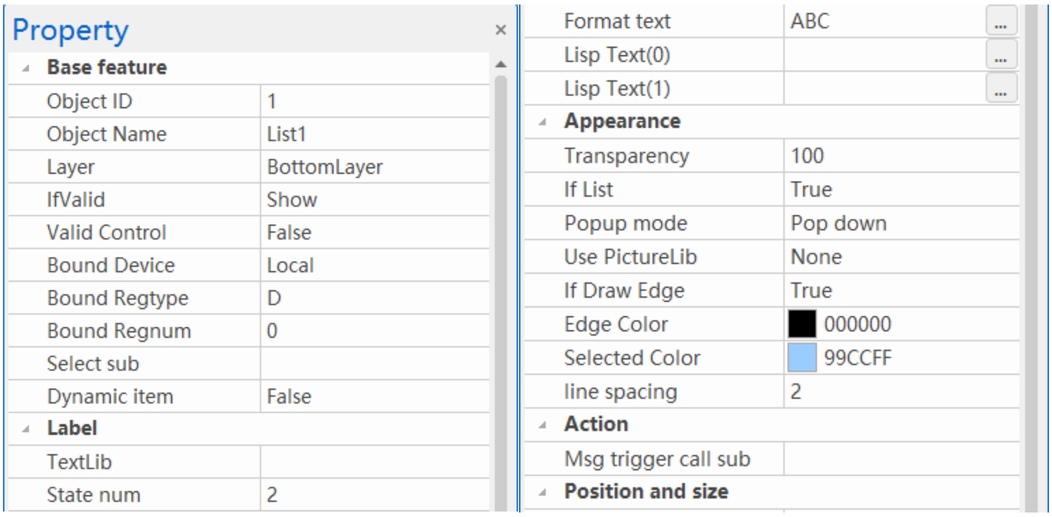
Notes:

a. When the height is too low to show all list items, after running, there is one scroll bar to help you view and select needed items.

b. When you select one row text, bound register value corresponds to text state value.



C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	List1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Select sub	
Dynamic item	False
Label	
TextLib	
State num	2
Format text	ABC
Lisp Text(0)	
Lisp Text(1)	
Appearance	
Transparency	100
If List	True
Popup mode	Pop down
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 000000
Selected Color	■ 99CCFF
line spacing	2
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	

Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Bound Device	Assigned device, default is local.
Bound Regtype	Set register type, there is one register list.
Bound Regnum	Set register No., different values can be obtained by register, then different states of component can be controlled.
Select sub	Select which function to be called
Dynamic item	Dynamically read and modify list items. When using HMI_LISTTEXTS command to set read list, this must be opened.
Label	
TextLib	Text library name
State num	The number of object states (0-255)
Format text	Set space label format
Format Text (0)	Text of list
Format Text (1)	Text of list
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc.

	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If List	Set list component as list-based or drop-down list, default is True, when you changed into False, it will pop up “drop down”.
Popup mode	It can pop up or pop down, default is pop down.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the edge
Selected Color	Set selected part’s color
Line Spacing	List line spacing, equal spacing above and below, default is 2
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Object horizontal starting position, don’t exceed X resolution.
Top	Object vertical starting position, don’t exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

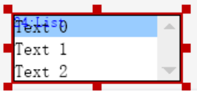
Step 1: select register type and No.

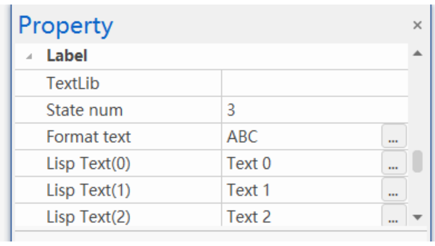
Step 2: select “state num”, and select corresponding texts under each state.

Step 3: you can switch the list line to change bound register’s state value, also, you can modify register state value to switch selected line.

Step 4: list text content also can be set by command “HMI_LISTTEXTS”, but you need set “dynamic item” as True. Command configuration priority > component property configuration.

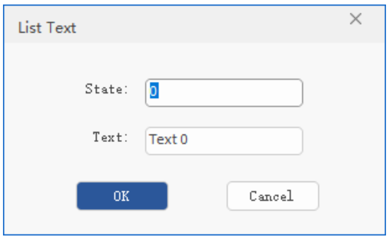
Step 5: state showing color, line spacing, popup mode can be set through property.





Property

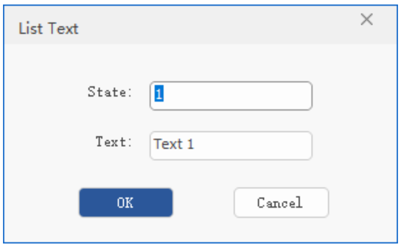
Label	
TextLib	
State num	3
Format text	ABC
Lisp Text(0)	Text 0
Lisp Text(1)	Text 1
Lisp Text(2)	Text 2



List Text

State: 0

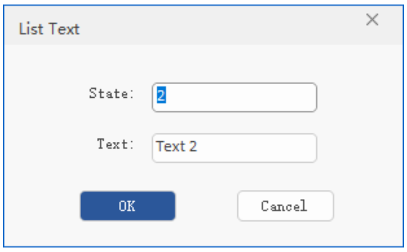
Text: Text 0



List Text

State: 1

Text: Text 1



List Text

State: 2

Text: Text 2

Effect:

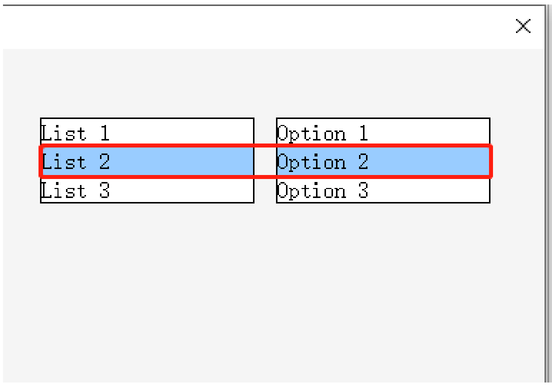
When text 0 line is pressed, MODBUS_REG(0)=0 (value of the first state 0). When text 1 line is pressed, MODBUS_REG(0)=3 (value of the second state 1). When text 2 line is pressed, MODBUS_REG(0)=1 (value of the second state 2), and selected line showing color is set color. Inversely, if you want make MODBUS_REG(0)=0, text 0 line will be selected.

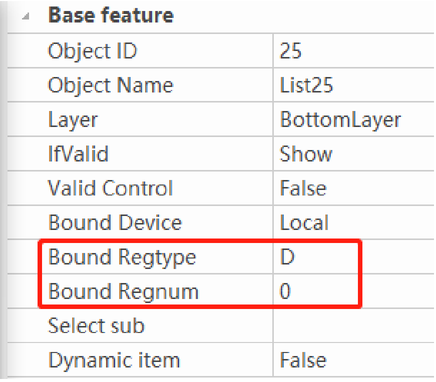
If state values are the same, no way to switch between same states.

Special Note:

When setting multiple lists, pay attention to whether the register variables bound to each list are the same. When selecting list options, if the register variables set are the same, the lists will affect each other and the same options will be selected synchronously. If the register variables set are different, each list can select options independently without affecting each other. See the figure below.

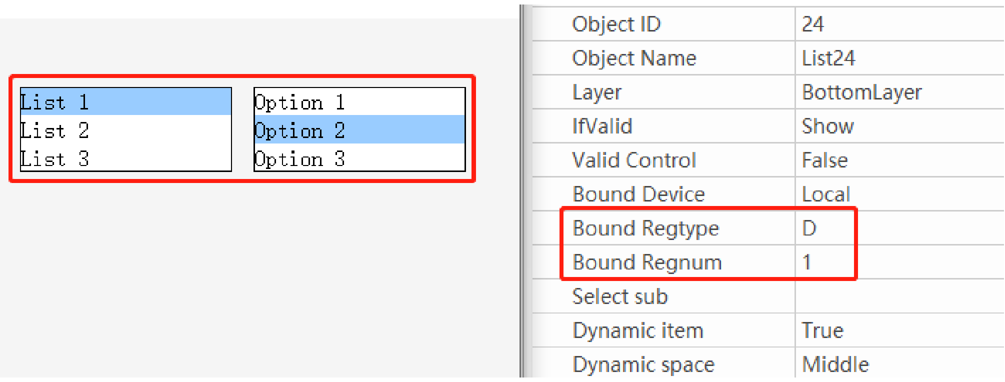
When two lists' bound registers are D0, after running, when you select "list item" 2, "option 2" also will be selected synchronously.





Base feature	
Object ID	25
Object Name	List25
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Select sub	
Dynamic item	False

When one list binds with register D0, another list binds with register D1, after running, when you select list 1, another you can select option 1 / 2 / 3.



Object ID	24
Object Name	List24
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	1
Select sub	
Dynamic item	True
Dynamic space	Middle

4.3.44. Menu

A. What is It?

[Menu] component is similar to windows menu bar, it only can embed 5 layers. You can set whether to enable or not for each menu (if it is disabled, shown in gray).

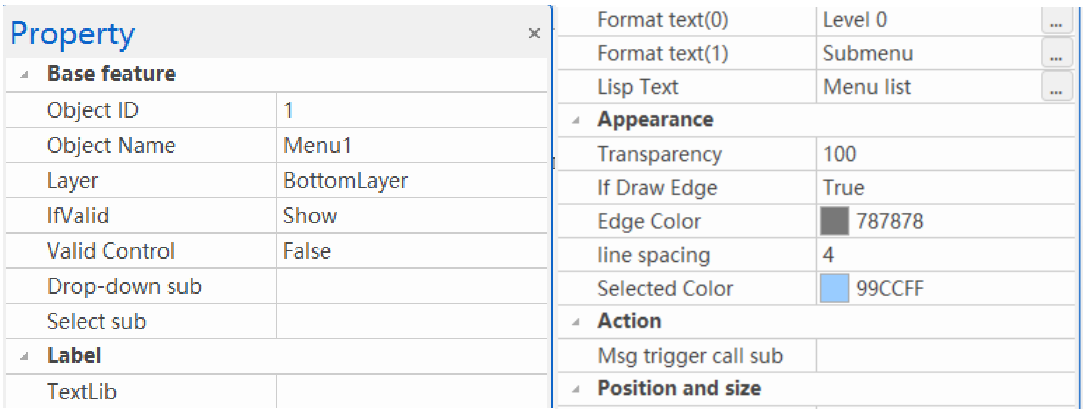
When clicking one menu item, corresponding SUB action will be triggered, also, you can set the state (selected or not). The menu item position where the mouse stops is in highlight state. What's more, you can add the delimiter in any menu item.

Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

B. How to Use?

Click "Tool Box", select "Menu" from "System Navigation". Put the component at suitable position, open menu property, and set column text content, then edit corresponding global SUB function in Basic file according to each item menu No., then each item's operations can be achieved. At last, in menu property, call edited SUB function to complete menu function.

C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	Menu1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Drop-down sub	
Select sub	
Label	
TextLib	
Format text(0)	Level 0
Format text(1)	Submenu
Lisp Text	Menu list
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	787878
line spacing	4
Selected Color	99CCFF
Action	
Msg trigger call sub	
Position and size	

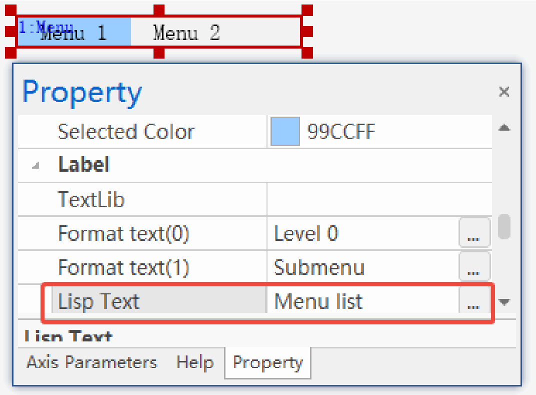
Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	The object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Drop-down sub	Trigger and call SUB function that was defined by Basic when pulling down, the SUB function must be GLOBAL type. Note: it can manage sub-menu gray state, selected state in this action sub.

Select sub	When the menu item area pops up, click and select any one valid menu item to trigger corresponding SUB, the SUB function must be GLOBAL type. Note: the action function can determine the last clicked selected item and bind the corresponding action SUB_XXXX. However, when defining the action, an ID parameter needs to be passed in, such as SUB_XXXX(ID), where ID represents the last clicked selected menu item No.
Label	
TextLib	The text library name, if it is empty, the text label will be used.
Format text 0 (Level 0)	The showing format of the main menu button. RTSys transferred control size is the menu button size, then menu item size depends on menu item numbers, that is, corresponding property will be calculated and shown automatically. The text name can’t be modified.
Format text 1 (Submenu)	Showing format of sub menu item, including all sub menu items. This format text’s background color is used for the whole pop up menu area, the alignment method only can be left alignment. The text name can’t be modified.
Lisp Text (Menu list)	Set the text of each menu item.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge for the sub-menu.
Edge Color	Select the edge color, default is gray.
Line spacing	Set the space between rows of sub-item, for example, font size = 16, line space = 4, then, each row height = 16+2*4=24. Default: 4. Range: 0~100.
Selected Color	Set the highlight background color (default: blue) for the main menu and all sub menu items, that is, when the mouse puts at one certain menu button or one menu item, the background color will be highlightened.
Action	

Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

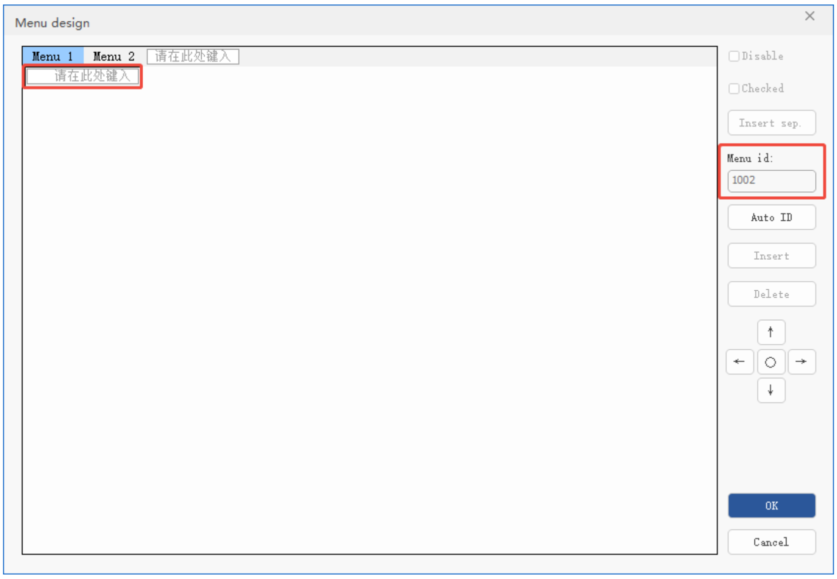
D. For Example

Step 1: in menu property, open "lisp text".



Selected Color 99CCFF
Format text(0) Level 0
Format text(1) Submenu
Lisp Text Menu list

Step 2: edit menu item, each item has one unique menu No., and they are numbered in order.

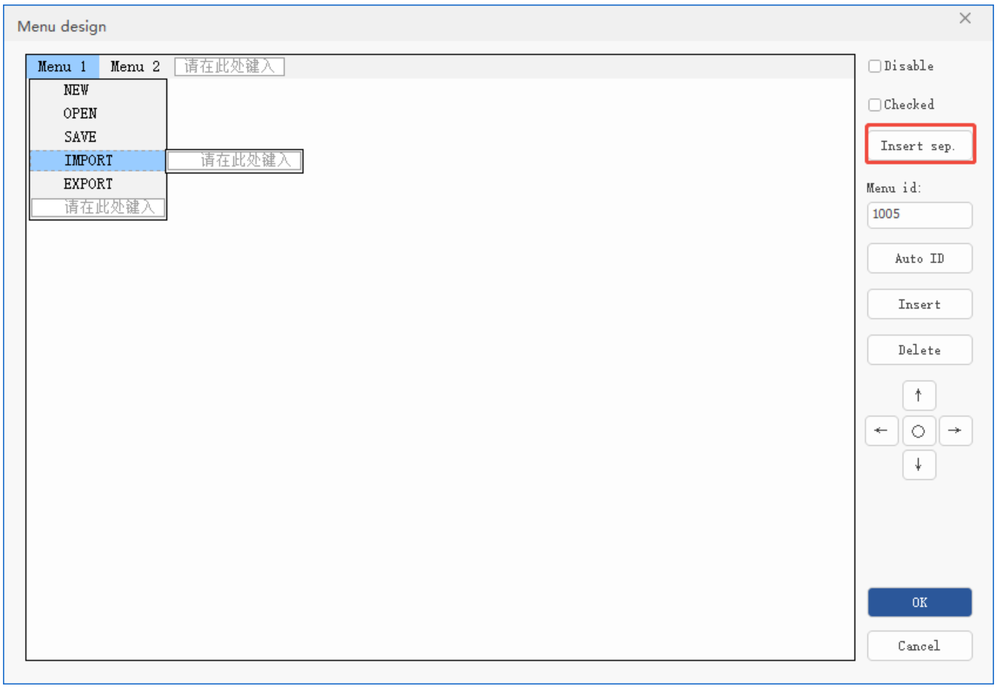


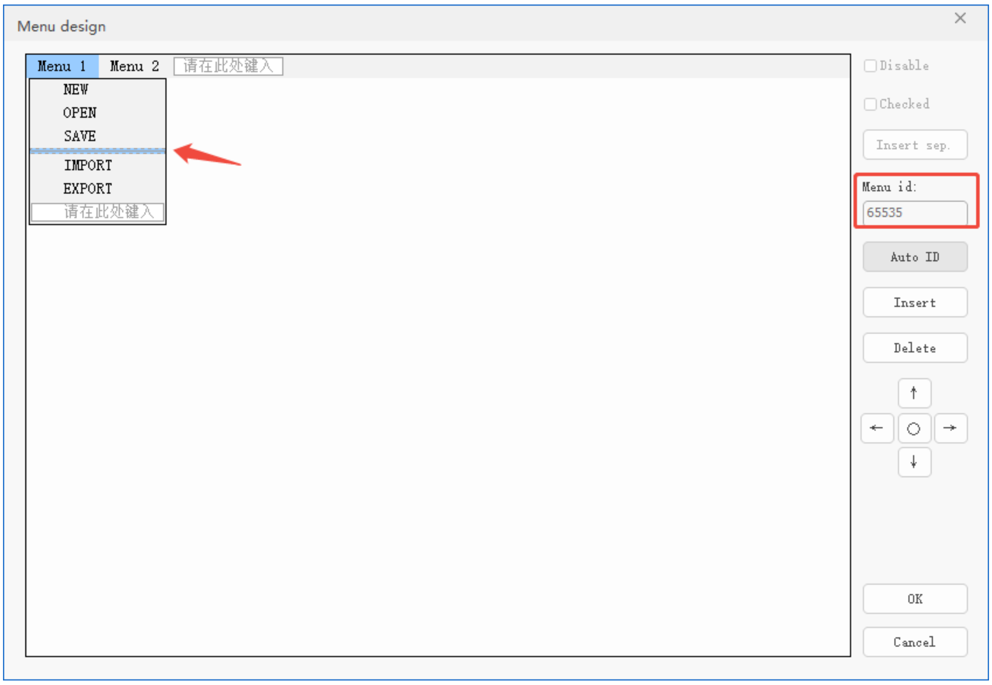
Menu id: 1002

Note: menu No. can be set manually or automatically, but used No. can’t be used again! the max value is 65535. “Auto ID” function will automatically find No. that are not used. For manual number, no way to detect directly.

a) Insert one Delimiter

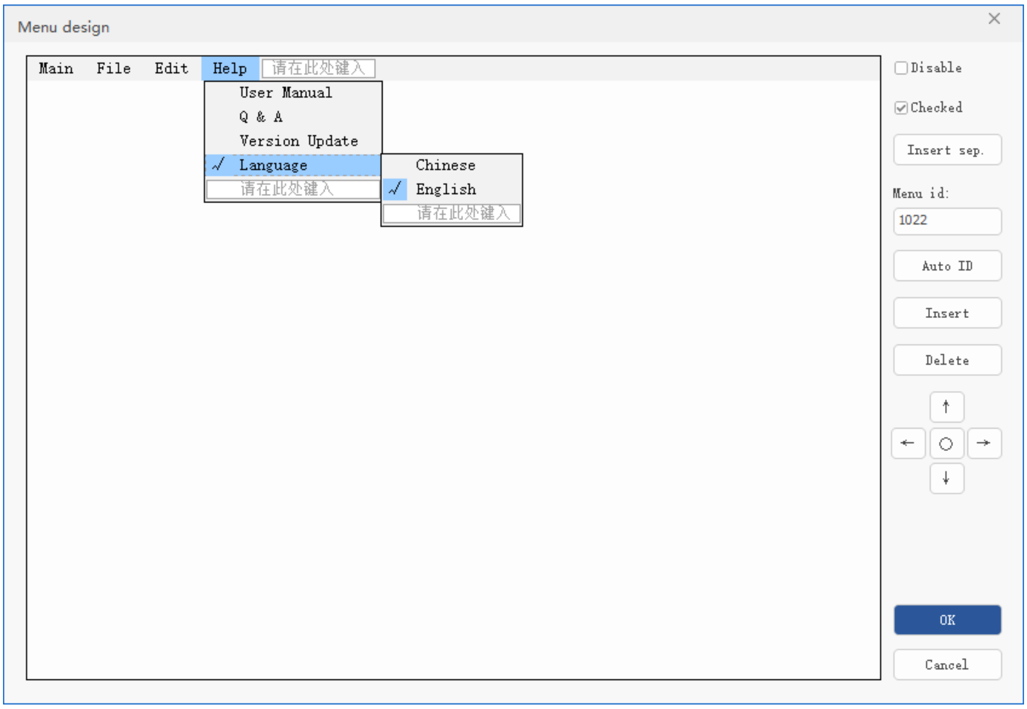
Select one item, click “insert sep”, then the delimiter will be inserted into above the selected item, the menu No. is fixed value 65535.





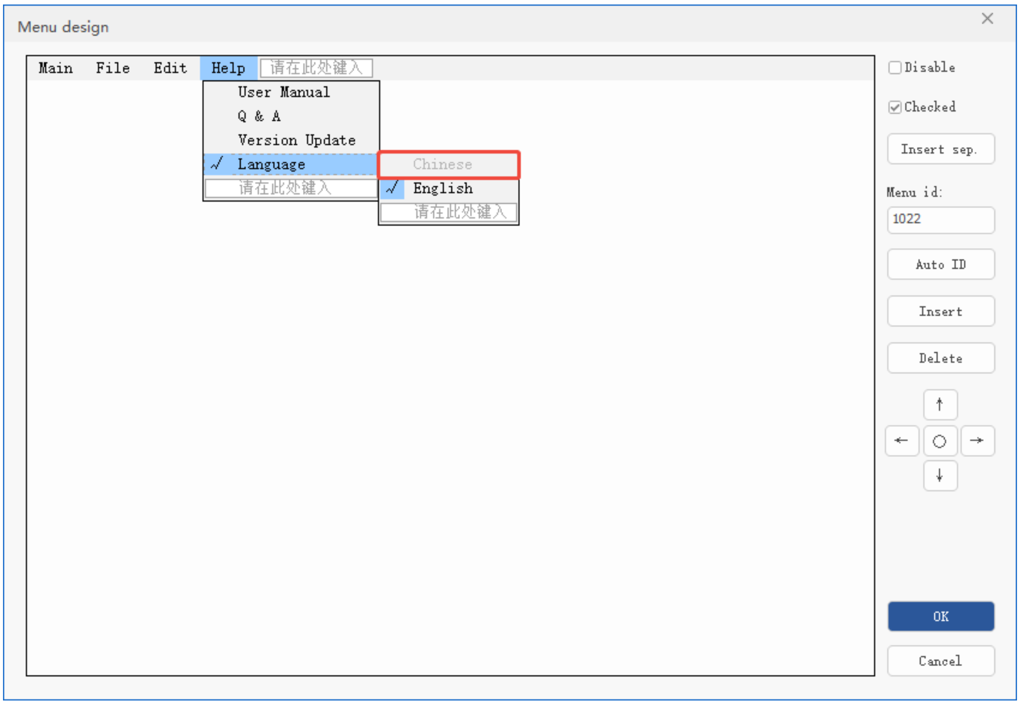
b) Set Checked State

Select one item, and click “checked”, then one “√” will appear in the front of selected item. Also, you can set parameters through control operation command HMI_MENUITEM.



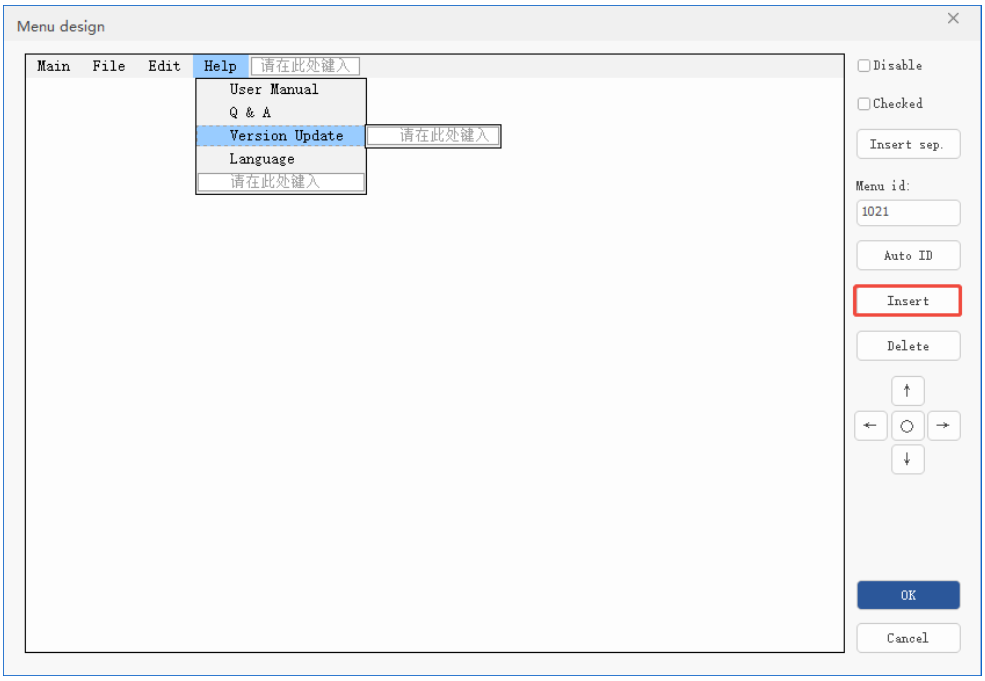
c) Disable one Item

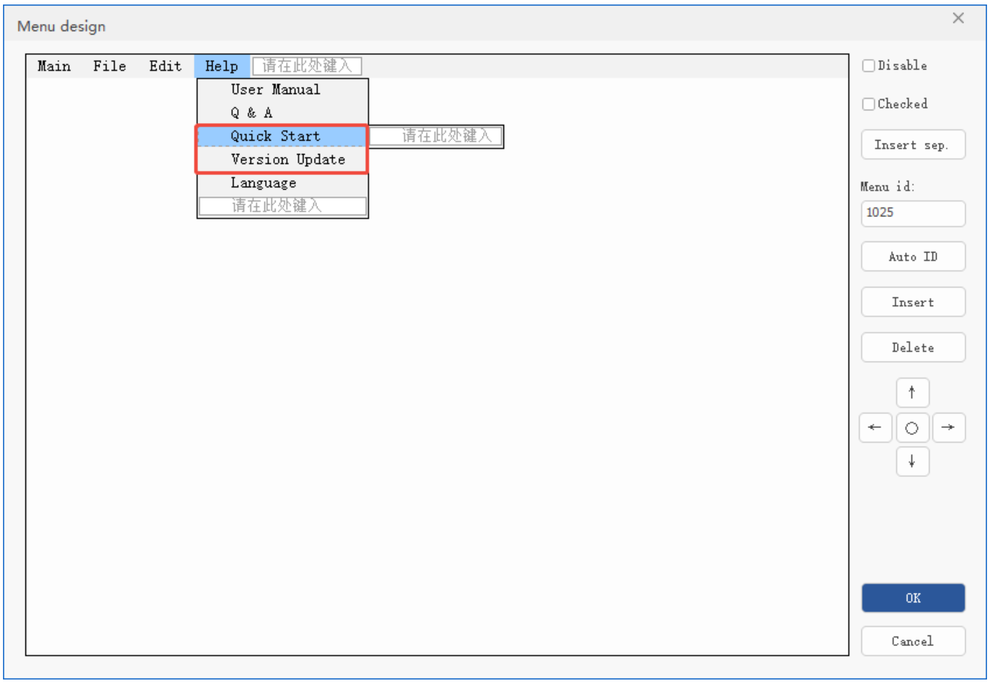
Select one item, then check “disable”, then the selected one will become gray. In this way, in real operation, no way to select it. You also can set through HMI_MENUITEM command.



d) Insert one Item

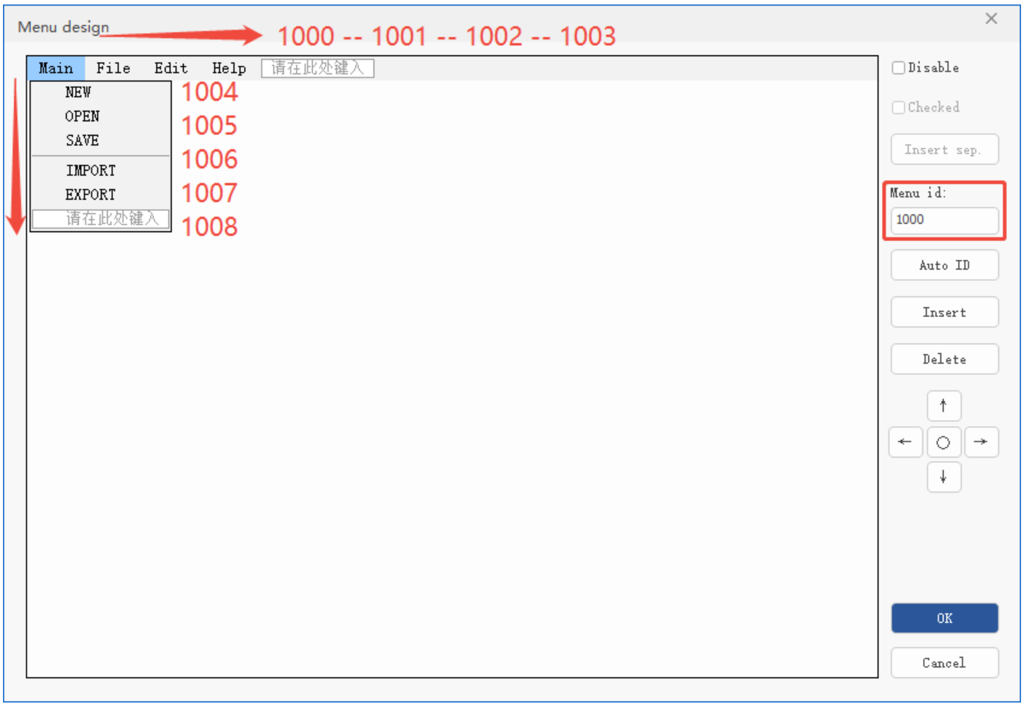
Select one item, then check “insert”, new one item will be inserted above it.



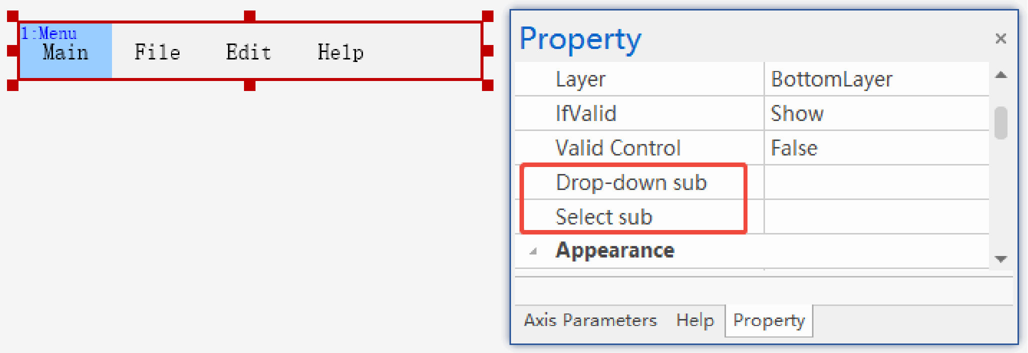


e) Auto ID

After edited, click Auto ID, system will number main item from the left to right. Then, it will number sub menu item from left to right and from the up to bottom.



Step 3: when all are edited, click OK. In HMI interface, it only shows the main menu item. According to requirements, select corresponding SUB function in menu basic property. The command is HMI_MENUITEM.

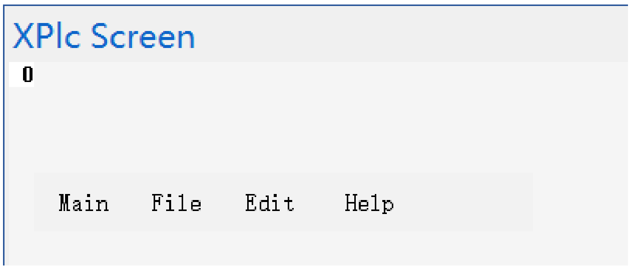


Property	
Layer	BottomLayer
IfValid	Show
Valid Control	False
Drop-down sub	
Select sub	
Appearance	

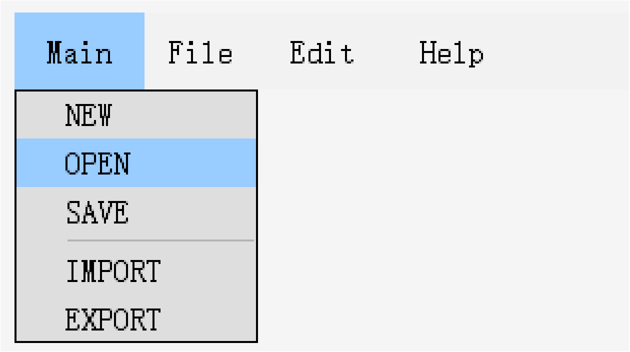
Example of Select Sub:

```
end        "main program ends

'selected sub for the menu item (some functions)
'transfer one parameter (menu No.)
global sub Sub_MenuSelect(menuid)
        ?"Menu Item Clicked",menuid
        'process menu item clicking event
        if menuid = 1005 then
                HMI_SHOWWINDOW(11) "open the "main" menu
        elseif menuid = 1027 then
                HMI_SHOWWINDOW(12) "open file browser
        elseif menuid >= 1031 AND menuid <= 1034 then "select the language

        "it only can select one language.
        HMI_MENUITEM(1031, 0, 0)
        HMI_MENUITEM(1032, 0, 0)
        HMI_MENUITEM(1033, 0, 0)
        HMI_MENUITEM(1034, 0, 0)
        HMI_MENUITEM(menuid, 0, 1)
endif

end sub
```

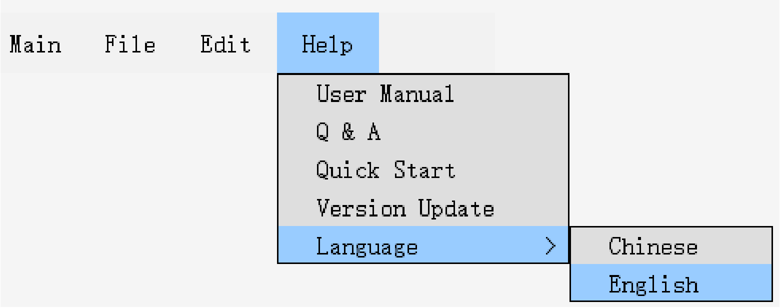
Step 4: connect to controller, download into the controller. It will be shown in xplc screen.



a) open Main

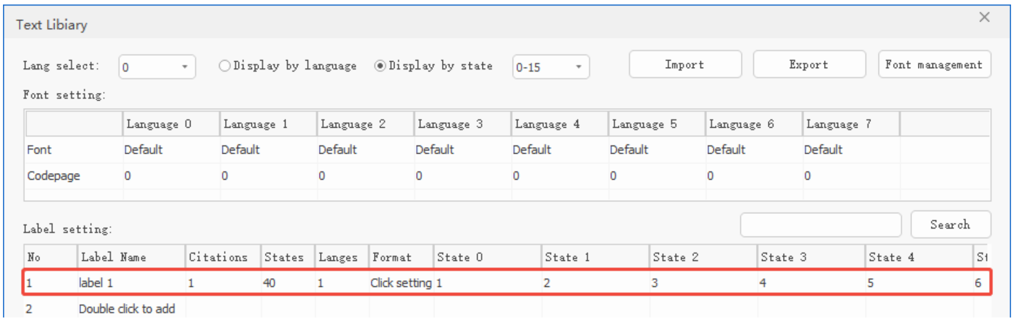


b) Select Language



It only can select one language, disabled menu item can’t be selected.

Step 5: if it uses text library for the menu control, then contents will be filled into each menu item according to the layer structure (the delimiter is also as one text library state).



Text Libiary

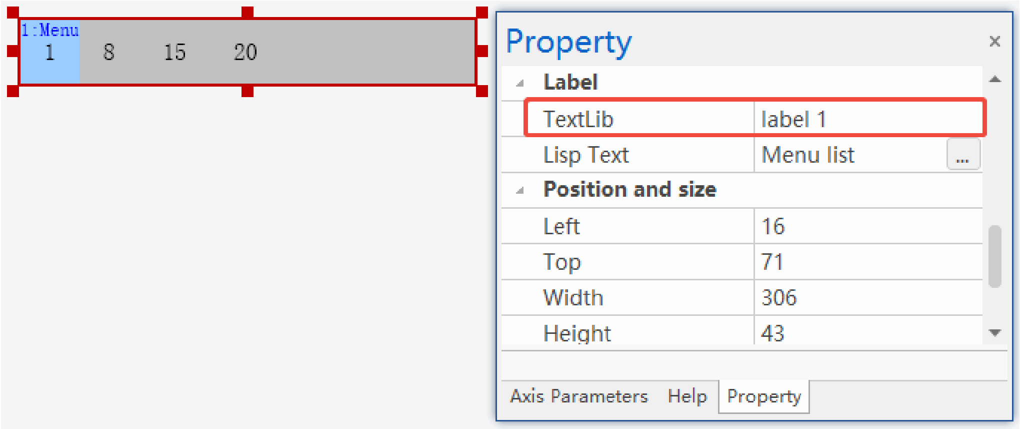
Lang select: 0 ○Display by language ●Display by state 0-15

Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting:

No	Label Name	Citations	States	Langes	Format	State 0	State 1	State 2	State 3	State 4	St
1	label 1	1	40	1	Click setting	1	2	3	4	5	6
2	Double click to add										



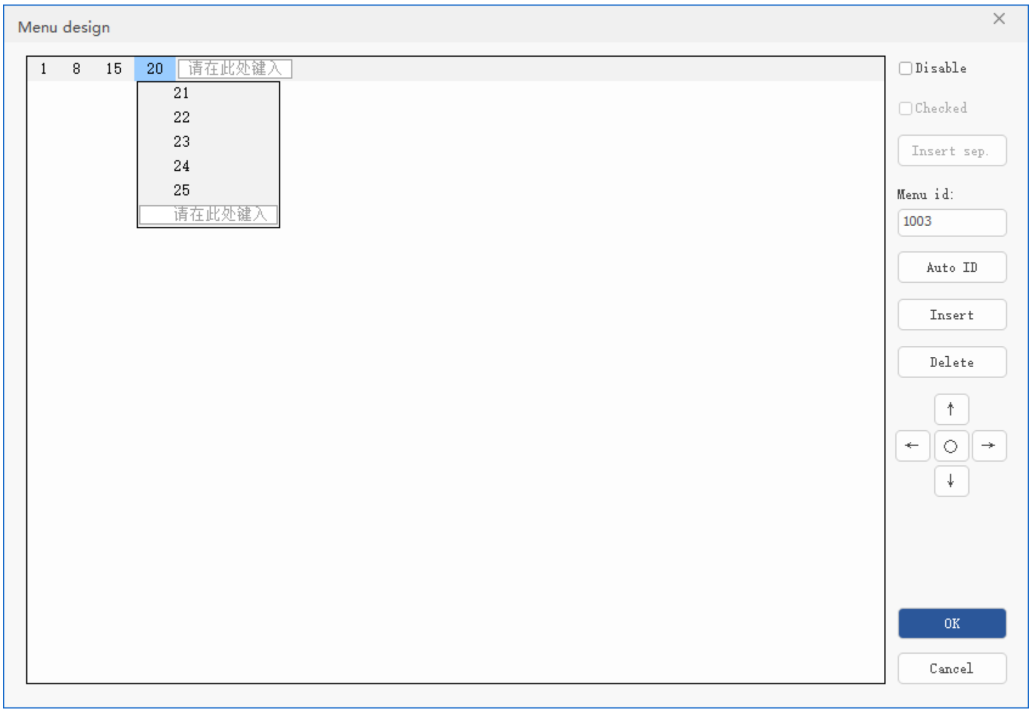
Property

Label

TextLib	label 1
Lisp Text	Menu list

Position and size

Left	16
Top	71
Width	306
Height	43



Menu design

□Disable

□Checked

Menu id: 1003

4.3.45. Tree

A. What is It?

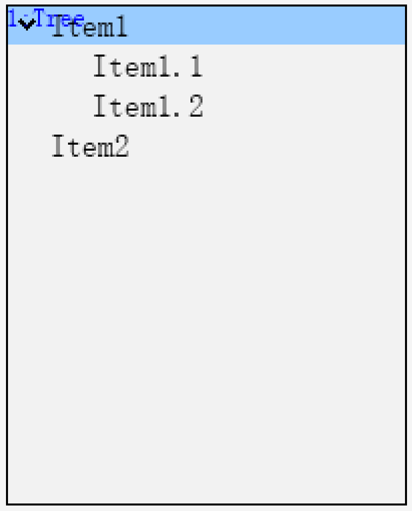
[Tree] component shows all items (the tree nodes of the first layer) in the tree shape structure. It supports clicking corner icons to expand/collapse subtrees, and clicking tree node content to trigger actions. When the number of tree nodes exceeds the display range, the vertical scroll bar is automatically enabled.

Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

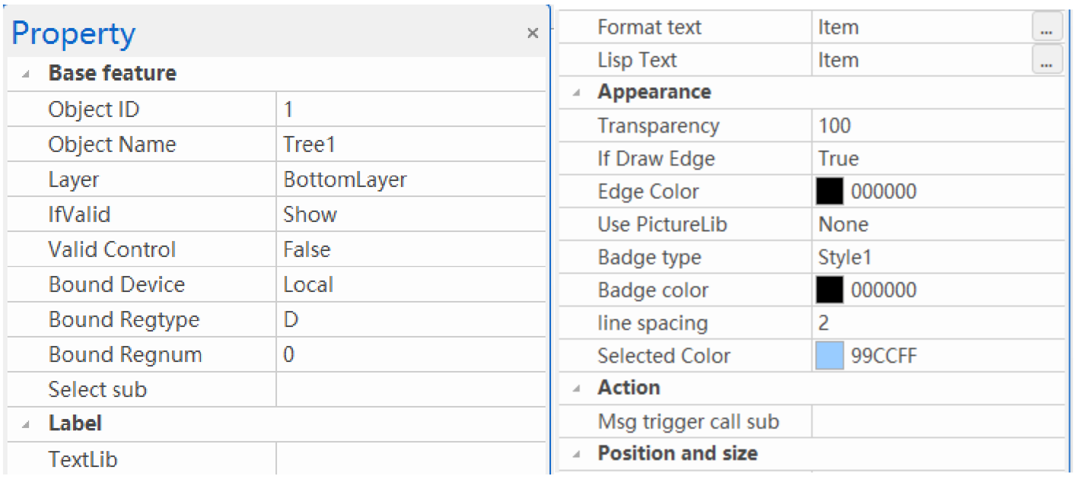
For current “tree” component, it can’t dynamic modification, addition, and clearing of tree nodes in the simulation interface after downloading. All tree nodes can only be designed in the Hmi file.

B. How to Use?

Click “Tool Box”, select “Tree” from “System Navigation”. Put the component at suitable position, design the structure of the tree diagram in the list text, and bind each tree node to a unique ID. When a tree node is selected, its associated register (state) value will be automatically set to the ID. When the register value is modified, the tree node corresponding to the ID will be selected. Through the "Select Call Sub" in the tree property, call the global SUB function of the corresponding action to realize the function of triggering the action by clicking the tree node content.



C. “Property” Description



Property

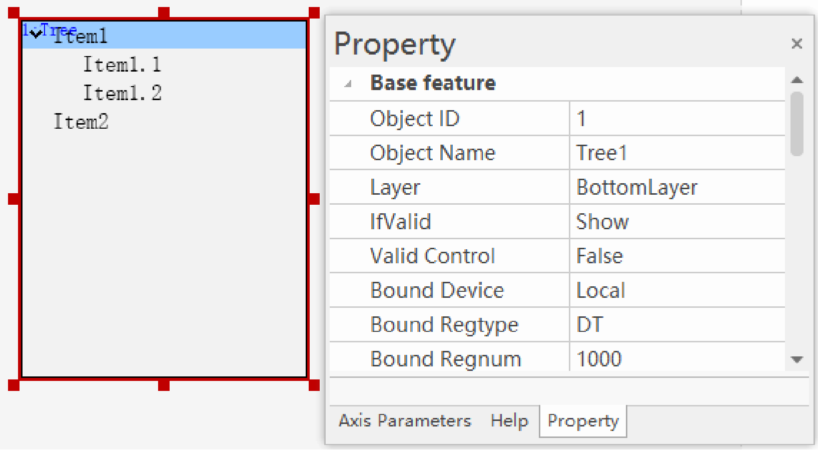
Base feature	
Object ID	1
Object Name	Tree1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	D
Bound Regnum	0
Select sub	
Label	
TextLib	
Format text	Item
Lisp Text	Item
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	■ 000000
Use PictureLib	None
Badge type	Style1
Badge color	■ 000000
line spacing	2
Selected Color	🟦 99CCFF
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Bound Device	Assign the device, default is local.
Bound Regtype	Select the register type, there is one list, you can select direcly.
Bound Regnum	The corresponding register No., different values of register can be obtained to control different components’ states.

Select Sub	Select the SUB function that is to be callled when corresponding item is clicked, the SUB must be GLOBAL type.
Label	
TextLib	The text library name, if it is empty, the text label will be used.
Lisp text (Item)	The displayed text and ID of all tree (parent and child) nodes. The currently selected item will be associated with the tree node ID No.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge. If TRUE, draw it.
Edge Color	Set the edge color, it is valid when "If Draw Edge" is True.
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
Badge Style	The display type of the parent tree corner mark, style 1 ~ style 3
Badge Color	The color of the corner mark.
Line spacing	Set the space between rows of sub-item, for example, font size = 16, line space = 4, then, each row height = 16+2*4=24. Default: 2. Range: 0~100.
Selected Color	Set the highlight background color (default: blue) for the tree and all tree items, that is, when the mouse puts at one certain tree node, the background color will be highlightened.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Postion & Size	
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

D. For Example

Step2: in "tree" property window, select register type and No. (here selects DT register and set the No. as 1000, that is, bound register address is **TABLE(1000)**).



1:Tree

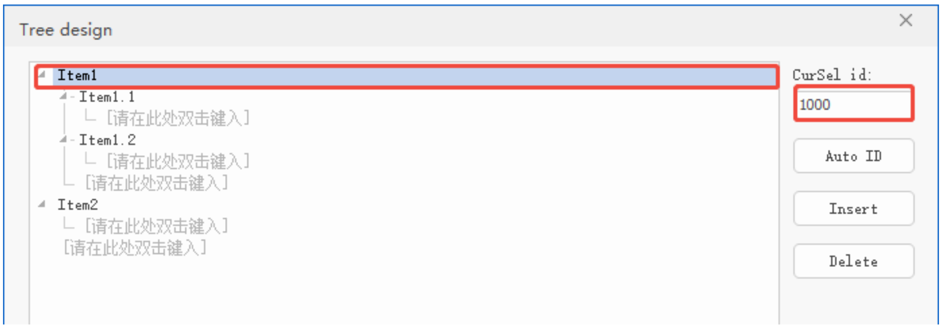
Item1
Item1.1
Item1.2
Item2

Property

Base feature

Object ID	1
Object Name	Tree1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	DT
Bound Regnum	1000

Step 2: in tree property – lisp text, set the tree branch structure and edit the content. Each tree node binds with one unique ID, ID of tree nodes are numbered according to editing sequence. After edited, if you click "Auto ID", then they will be numbered from the up to bottom.

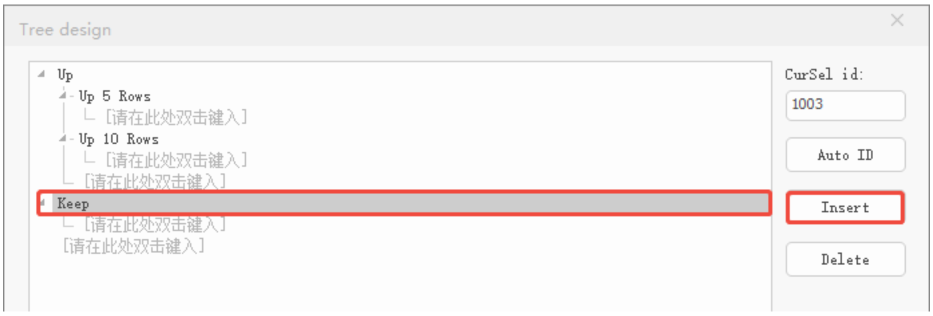


Tree design

Item1
Item1.1
〔请在此处双击键入〕
Item1.2
〔请在此处双击键入〕
〔请在此处双击键入〕
Item2
〔请在此处双击键入〕
〔请在此处双击键入〕

CurSel id: 1000

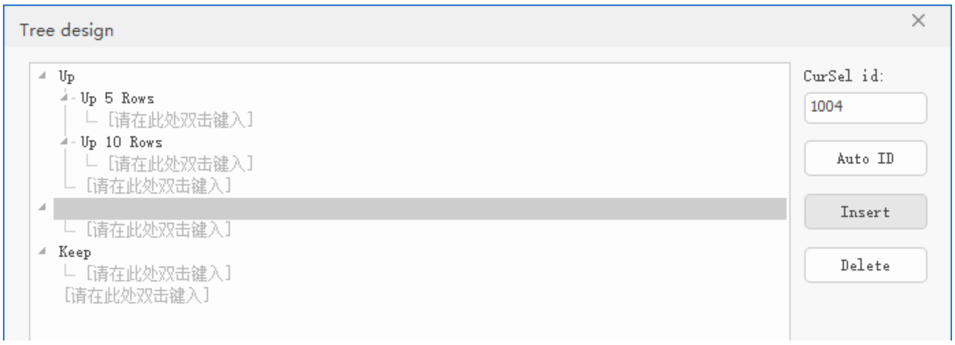
a) insert: add one same level item above selected item



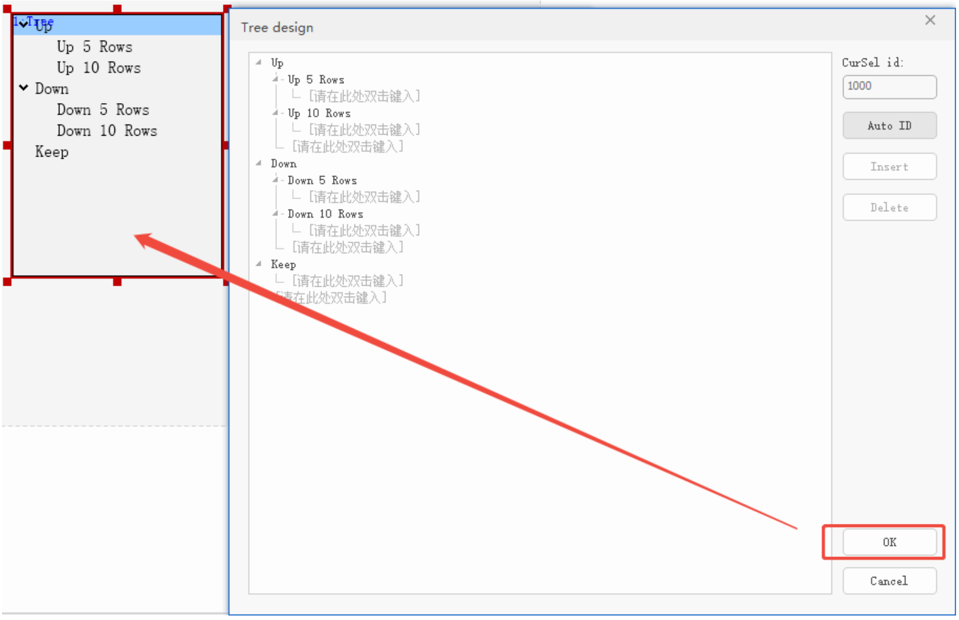
Tree design

Up
Up 5 Rows
〔请在此处双击键入〕
Up 10 Rows
〔请在此处双击键入〕
〔请在此处双击键入〕
Keep
〔请在此处双击键入〕
〔请在此处双击键入〕

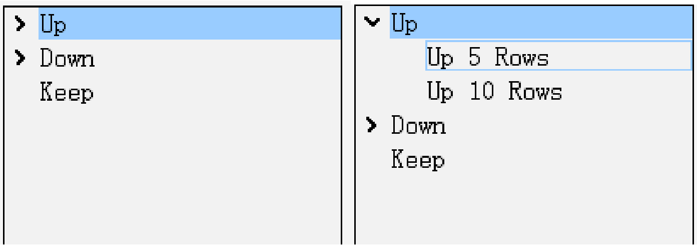
CurSel id: 1003



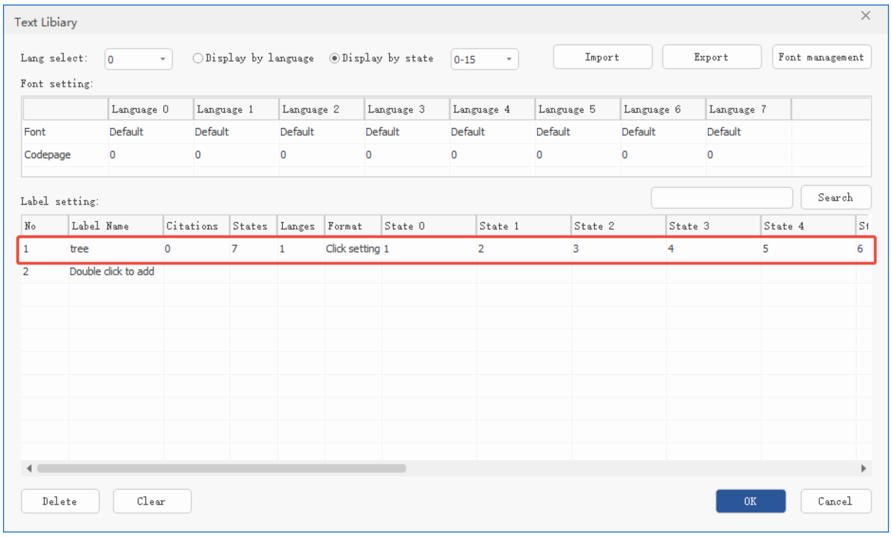
Step 3: after edited, click OK.



Step 4: connect to controller, download into controller, open xplc screen. It only shows the first layer content, you can click the arrow > to show all sub items.



Step 6: if it uses text library for the menu control, then contents will be filled into each menu item according to the layer structure.



Text Libiary

Lang select: 0 ○Display by language ●Display by state 0-15 Import Export Font management

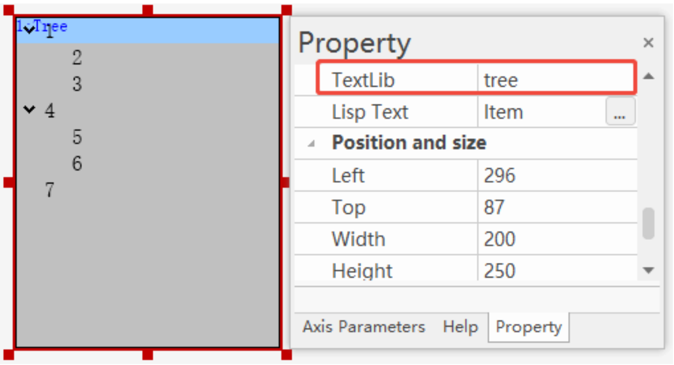
Font setting:

	Language 0	Language 1	Language 2	Language 3	Language 4	Language 5	Language 6	Language 7
Font	Default	Default	Default	Default	Default	Default	Default	Default
Codepage	0	0	0	0	0	0	0	0

Label setting: Search

No	Label Name	Citations	States	Langes	Format	State 0	State 1	State 2	State 3	State 4	St
1	tree	0	7	1	Click setting	1	2	3	4	5	6
2	Double click to add										

Delete Clear OK Cancel



Property

TextLib	tree
Lisp Text	Item
Position and size	
Left	296
Top	87
Width	200
Height	250

Axis Parameters Help Property

Step 6: for state showing color, line spacing, pop up method, please set in property.

Step 7: for usage of SUB calling by tree, please refer to Chapter VIII [File Browser Usage].

4.3.46. File Browser

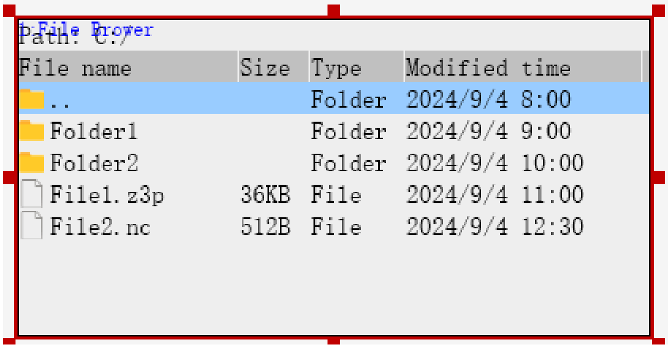
A. What is It?

[File Browser] component shows current content, the file information is shown in the form format. It needs to assign the displaying format (file name, file size, modified time, file type are needed). Double click the folder, corresponding files are shown, then double click it to open it (for calling assigned SUB).

Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

B. How to Use?

Click “Tool Box”, select “File Browser” from “System Navigation”. Put the component at suitable position, call assigned global SUB program.

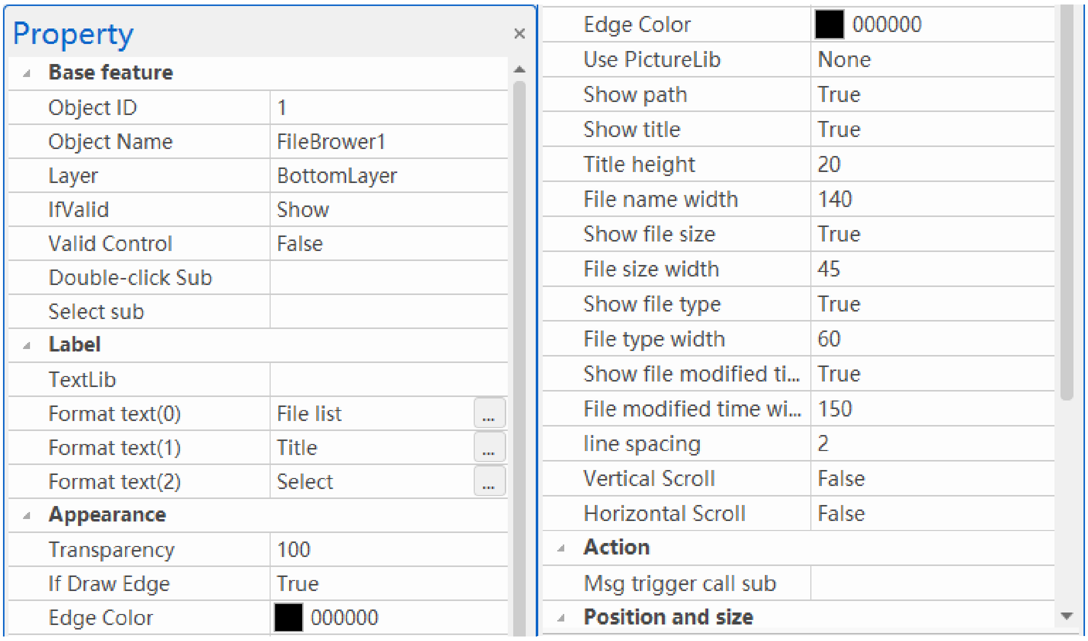


File Brower

Path: C:/

File name	Size	Type	Modified time
..		Folder	2024/9/4 8:00
Folder1		Folder	2024/9/4 9:00
Folder2		Folder	2024/9/4 10:00
File1.z3p	36KB	File	2024/9/4 11:00
File2.nc	512B	File	2024/9/4 12:30

C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	FileBrower1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Double-click Sub	
Select sub	
Label	
TextLib	
Format text(0)	File list
Format text(1)	Title
Format text(2)	Select
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	■ 000000
Use PictureLib	None
Show path	True
Show title	True
Title height	20
File name width	140
Show file size	True
File size width	45
Show file type	True
File type width	60
Show file modified ti...	True
File modified time wi...	150
line spacing	2
Vertical Scroll	False
Horizontal Scroll	False
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components.

	● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Double-click Sub	Trigger and call SUB function that was defined by Basic when double-clicking, the SUB function must be GLOBAL type.
Select sub	Trigger and call SUB function that was defined by Basic when the component is selected, the SUB function must be GLOBAL type.
Label	
TextLib	Select the text library form the list.
Format text 0 (File list)	Set the font format of the file list row, including the configuration of current path row, file name can't be modified.
Format text 1 (Title)	Set the font format of the file list title row, file name can't be modified.
Format text 2 (Select)	Set the font format of the file list selected row, file name can't be modified.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge.
Use PictureLib	Select from none / background picture lib / background picture.
Show path	Whether to show the file path, default is True.
Show title	Whether to show the title, default is True.
Title height	The title height

File name width	The file name’s column width
Show file size	Whether to show file size, default is True.
File size width	The file size’s column width
Show file type	Whether to show file type, default is True.
File type width	The file type’s column width
Show the modified time	Whether to show file modification time, default is True.
File modified time width	File modified time’s column width
Line spacing	Show the space of each row between up and down, including file list and current path row, for example, font size = 16, line space = 4, then, each row height = 16+2*4=24. Default: 2. Range: 0~100.
Vertical scroll	Whether to use the vertical scroll bar
Horizontal scroll	Whether to use the horizontal scroll bar
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

For initial file 3 editor usage, please refer to Chapter VIII [File Browser Usage].

4.3.47. Group Box

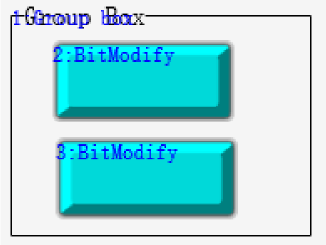
A. What is It?

This is a container element used to provide identifiable grouping for other elements within the same window. You can use it to subdivide the window's content by function. And it can display group titles and borders.

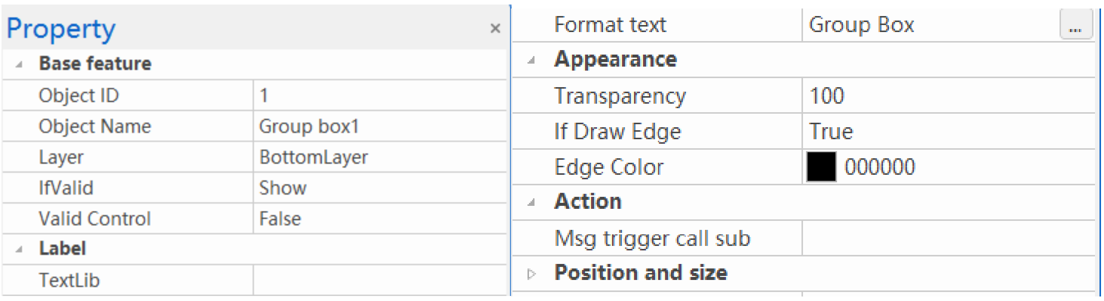
Note: when using this control, please make sure RTHMI version be V1.4.0 or above & RTSys version be V1.3.02 or above.

B. How to Use?

Click “Tool Box”, select “Group Box” from “System Navigation”. Put the component at suitable position, and set the format text, then put same type components into corresponding group boxes.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	Group box1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Label	
TextLib	
Format text	Group Box
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	■ 000000
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading

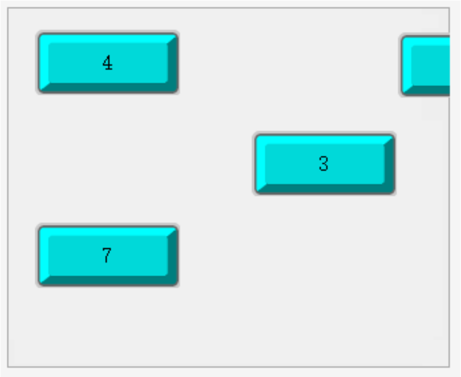
	● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Label	
TextLib	The text library's name. if not set manually, it is shown as "text lib".
Format Text	Set the text and format. Horizontal alignment only supports 1 and -1 (all negative values refer to -1, and all positive values refer to 1), and vertical alignment is not supported.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge. If TRUE, draw it.
Edge Color	Set the edge color, it is valid when "If Draw Edge" is True.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

D. For Examples:

➢ Example 1: Load the Component

Add new component / drag components fully into the group box (that is, the rectangular area of the container needs to completely enclose the element), it is only valid when using mouse to drag it (not include "move component by keyboard", "copy component", "align component", "move the group box", etc.).

- ✓ When this component is added, the container is empty, regardless of whether there are other components within the selected range.
- ✓ Loaded components will automatically adjust their hierarchy to match that of the container component. It is recommended to place the container and its child components at the same level.
- ✓ The portion of a sub-element that extends beyond the container element is not displayed; only the portion within the container is shown.



➢ Example 2: Unload the Component

Delete component / drag components outside the the group box (that is, drag the component out of the gray range). Please note it can't unload the component by operations, like, move & align the components by keyboard, and the part of the component that is outside the container will not be displayed. If the component fully moves out the container in non-mouse-drag way, it still is the container inner component, which only can be selected, will not be displayed.

➢ Example 3: Move the Group Box

While moving the group box, inner components are moved together.

➢ Example 4: Delete the Group Box

When the group box is deleted, all components inside the group box will be unloaded, and will be displayed.

4.3.48. Tab Control

A. What is It?

This is a container element that defines the same area of a window as multiple pages. When the option is selected, the relevant information or element (tab page) will be displayed. And each page contains a set of information or a group of elements, and each tab page has a corresponding button. Clicking the option button switches between tab pages.

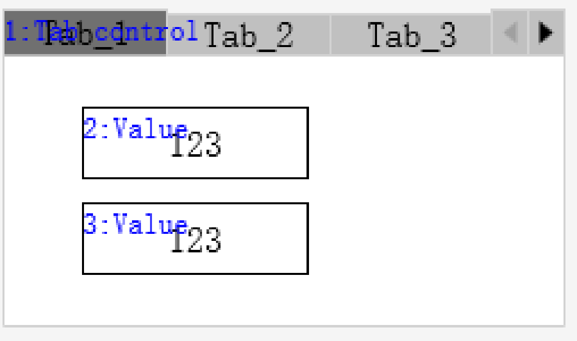
Note: when using this control, please make sure RTHMI version be V1.4.0 or above & RTSys version be V1.3.02 or above.

B. How to Use?

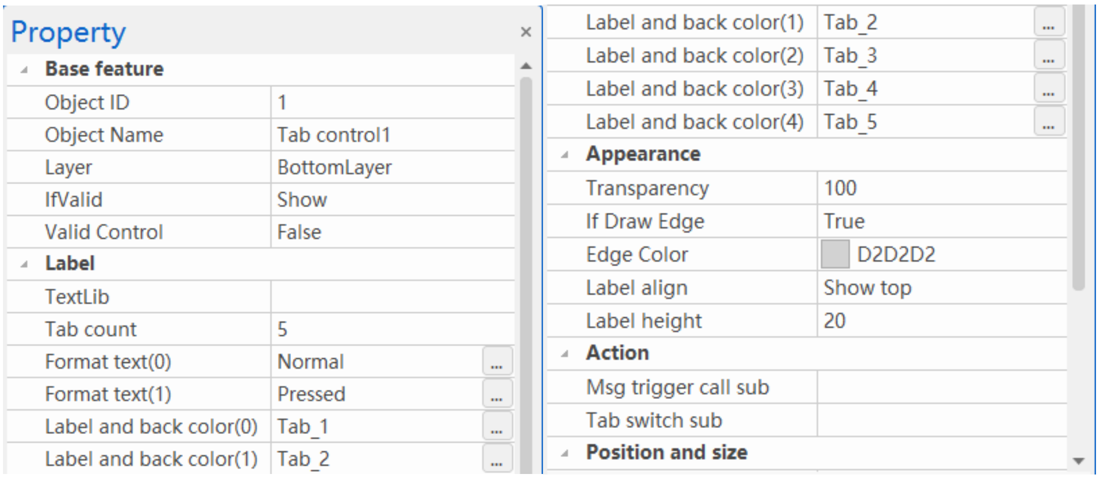
Click “Tool Box”, select “Tab Control” from “System Navigation”. Put it at suitable position, and put needed components into tab’s different pages.

The tab’s each option page can load multiple components (loaded components only can be displayed on assigned tab page), max 16 pages.

- Elements loaded on tabs not currently displayed will be hidden and will not be refreshed.
- Furthermore, when there are too many tabs and some cannot be displayed, two arrows will appear on the far right of the tab; clicking the arrows will switch tabs.



C. “Property” Description



Property

Base feature	
Object ID	1
Object Name	Tab control1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Label	
TextLib	
Tab count	5
Format text(0)	Normal
Format text(1)	Pressed
Label and back color(0)	Tab_1
Label and back color(1)	Tab_2
Label and back color(1)	Tab_2
Label and back color(2)	Tab_3
Label and back color(3)	Tab_4
Label and back color(4)	Tab_5
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	▨ D2D2D2
Label align	Show top
Label height	20
Action	
Msg trigger call sub	
Tab switch sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Label	
TextLib	The text library's name. if not set manually, it is shown as "text lib".
Tab Count	Set how many tabs.
Format Text (0/1)	Set the tab button display format:

	0 – the style when not pressed. 1 – the style when pressed.
Label and Back Color	Set the tab's label text and background color.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge. If TRUE, draw it.
Edge Color	Set the edge color, it is valid when "If Draw Edge" is True.
Label Align	Set the label alignment postion.
Label Height	Set the label height.
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Tab Switch Sub	Set the SUB function that is to be called when you switches tabs.
Postion & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

D. For Examples:

Please refer to Group Box – Load the Component.

4.3.49. Plane

A. What is It?

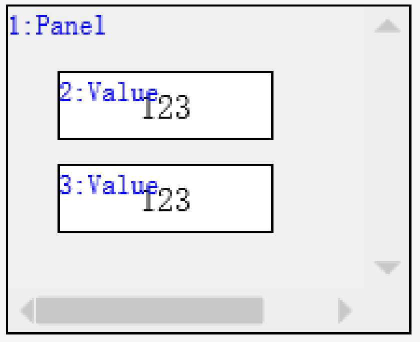
This is a container element that allows users to group all elements within a window and categorize the space. It supports custom container size, and the display area can be dragged using a scroll bar, and the interface can be scrolled by clicking the scroll bar.

[Panel is similar to group box, except that group box elements can display titles, while panel elements do not display titles but can have scrollbar controls].

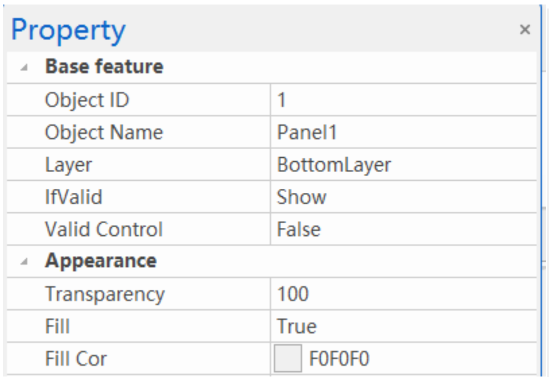
Note: when using this control, please make sure RTHMI version be V1.4.0 or above & RTSys version be V1.3.02 or above.

B. How to Use?

Click "Tool Box", select "Panel" from "System Navigation". Put it at suitable position, and put same types of components into the panel.

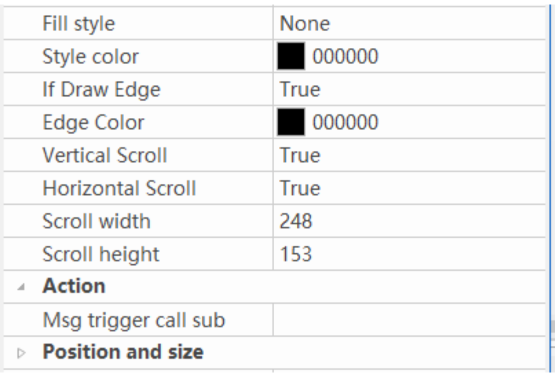


C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	Panel1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Fill	True
Fill Cor	□ F0F0F0



Fill style	None
Style color	■ 000000
If Draw Edge	True
Edge Color	■ 000000
Vertical Scroll	True
Horizontal Scroll	True
Scroll width	248
Scroll height	153
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and

	covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Fill	Whether to fill in the color.
Fill Cor	Set the filled color.
Fill Style	Select the filled style.
Style Color	Set the filled style's color.
If Draw Edge	Whether to draw the edge. If TRUE, draw it.
Edge Color	Set the edge color, it is valid when "If Draw Edge" is True.
Vertical Scroll	Whether to show the vertical scroll
Horizontal Scroll	Whether to show the horizontal scroll
Scroll Width	Set the scroll width
Scroll Height	Set the scroll height
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.

Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

D. For Examples:

Please refer to Group Box – Load the Component.

4.3.50. Timer

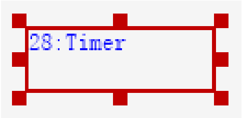
A. What is It?

It can refresh regularly and repeat actions.

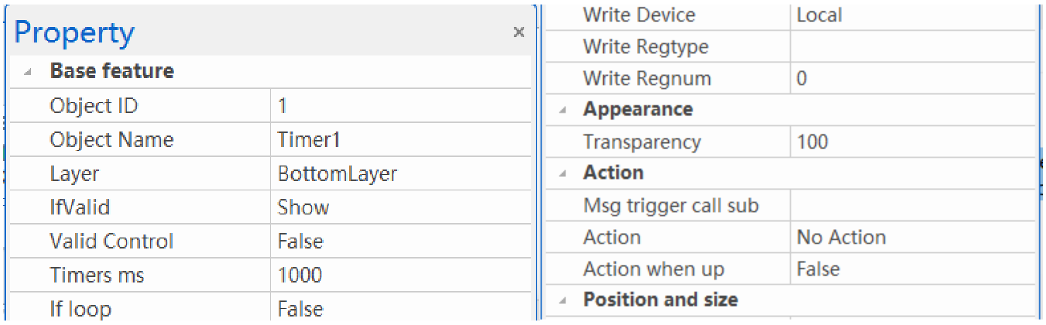
B. How to Use?

Click "Tool Box", select "Timer" from "System Navigation". Put the component at suitable position, and in its property window, set timer (to enable the spacing time) and the "action" (the repeated action). For "action", it also can call SUB function / assign the register.

Note: HMI interface is shown while editing, that is, it won't be shown in real running.



C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	Timer1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Timers ms	1000
If loop	False
Write Device	Local
Write Regtype	
Write Regnum	0
Appearance	
Transparency	100
Action	
Msg trigger call sub	
Action	No Action
Action when up	False
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Timers ms	The min button pressed time, the unit is ms, and it only supports integer, not support floating.
If loop	Whether to loop the timer, default is False
Write Device	Device No., default is LOCAL.
Write Regtype	Select register type, there is one list, you can select.
Write Regnum	Select register No., values can be obtained by register, then control component's different states.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is

	triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Action	Select what kind of action you need, please refer to 4.2.2
Action when up	The action when pressed or released, default is False – action when pressed, True – action when released.
Action data	Wite specified value for register when pressed
Action Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Postion & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

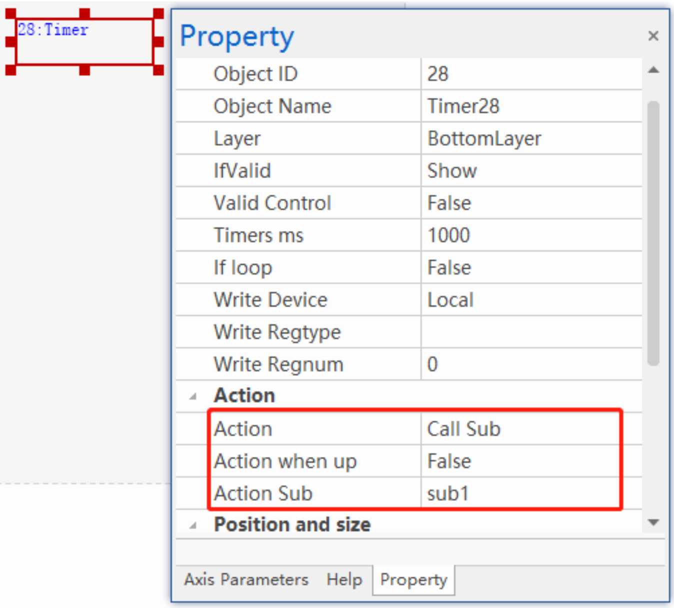
D. For Examples:

➢ Example 1: Call SUB Function

Step 1: in Basic file, edit one global sub function.

Step 2: in property, select Call Sub for Action, and select corresponding sub name.

Step 3: fill how long the time space for calling sub in “timer ms”.



Object ID	28
Object Name	Timer28
Layer	BottomLayer
IfValid	Show
Valid Control	False
Timers ms	1000
If loop	False
Write Device	Local
Write Regtype	
Write Regnum	0
Action	
Action	Call Sub
Action when up	False
Action Sub	sub1
Position and size	

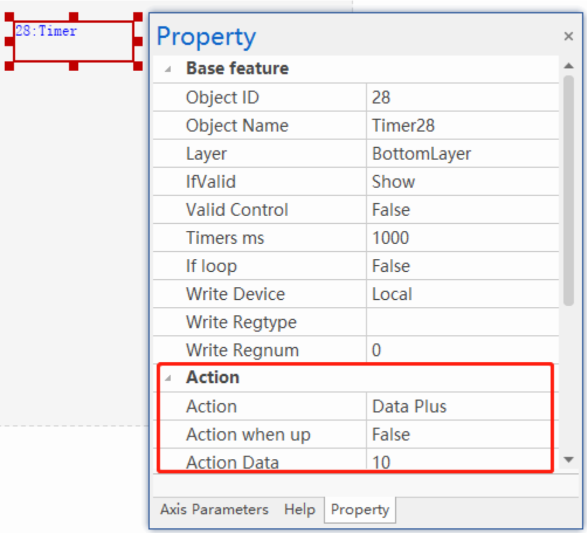
Effect: after HMI running, the object will not be shown, it will call one sub function "sub1" once each span 1000ms.

- **Example 2: register original value + action data value**

Step 1: select register type and No.

Step 2: select "data plus", enter the data of increasing for register each time in "action data".

Step 3: in "timer ms", fill in the time interval at which the register is incremented.



Base feature	
Object ID	28
Object Name	Timer28
Layer	BottomLayer
IfValid	Show
Valid Control	False
Timers ms	1000
If loop	False
Write Device	Local
Write Regtype	
Write Regnum	0
Action	
Action	Data Plus
Action when up	False
Action Data	10

Effect:

Every 1000ms, the value of MODBUS_REG(0) = the original value + 10.

--How to Check Register Value--

Method 1: click menu Tool – Register, select bound register type and address, then click read.

Method 2: in "output" window, enter print + bound register address and state value, and click "send", for example, enter "print MODBUS_BIT(0) / ? MODBUS_BIT(0)".

4.3.51. Scheduling

A. <u>What is It?</u>

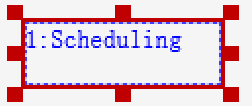
It can be used to set schedules, to execute specified actions at preset times, making it suitable for planning weekly routines. A schedule can specify a start time, start action, end time, and end action.

Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

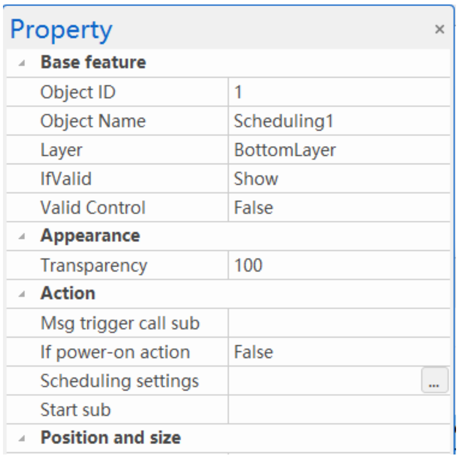
B. How to Use?

Click “Tool Box”, select “Scheduling” from “System Navigation”. Put the component at suitable position, and in its property window, whether to make it execute after powered on. In addition, in scheduling settings, set the starting time, end time, and corresponding sub actions. **In this way, it will execute starting sub action at starting time, and execute end sub action at end time.**

Note: HMI interface is shown while editing, that is, it won’t be shown in real running. Generally, one component is with one scheduling, recommend put at base window.



C. “Property” Description

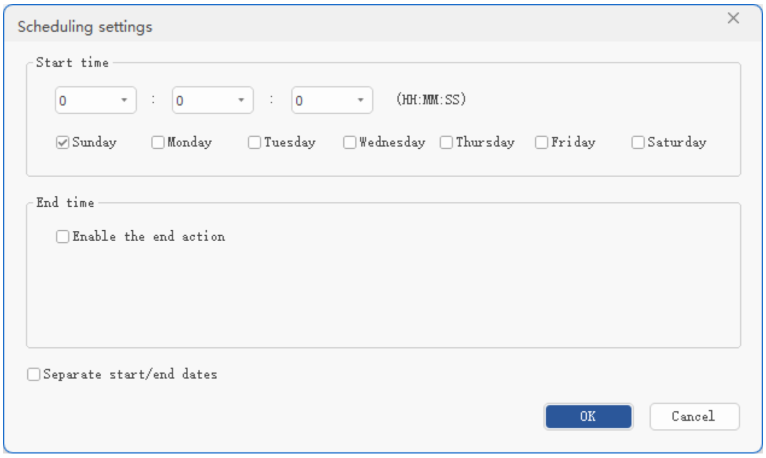
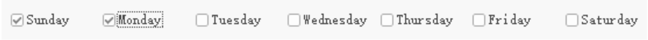
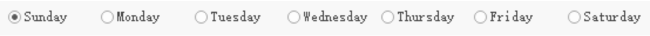


Property

Base feature	
Object ID	1
Object Name	Scheduling1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Appearance	
Transparency	100
Action	
Msg trigger call sub	
If power-on action	False
Scheduling settings	...
Start sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer

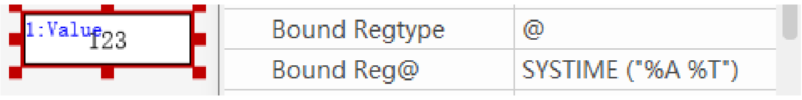
	● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register variables.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
If Power-On Action	Whether to make it execute action after powered-on. That is, when HMI is opened, starting action and end action will be triggered according to the time. ➢ True: if the HMI start time is within the scheduled range, the start action will be executed immediately. ➢ False: if the HMI start time is not within the scheduled range, the termination action will be executed immediately. Note: if the end action is not enabled, the end time will not be recognized, and therefore the end action will not be executed.

Scheduling Settings	 Scheduling settings Start time 0 : 0 : 0 (HH:MM:SS) ☑Sunday □Monday □Tuesday □Wednesday □Thursday □Friday □Saturday End time □Enable the end action □Separate start/end dates ● Separate start / end dates: whether to set the starting date and the end date separately. ✓ True: the start and end dates can be set at two different times during the week. Note: this mode applies to scheduling within a day, so if the end time is earlier than the start time, the end action will not be executed until the next day. ✓ False: selected weekdays have same start and end dates. ● Start Date: select the starting date. ✓ when “separate” is False, each day of the week corresponds to 1 option, and it can select multiple options.  ☑Sunday ☑Monday □Tuesday □Wednesday □Thursday □Friday □Saturday ✓ when “separate” is True, each day of the week corresponds to 1 option, only can be single-selection.  ● Sunday ○Monday ○Tuesday ○Wednesday ○Thursday ○Friday ○Saturday ● Start Time: set the starting time. ✓ HH (hour): MM (minute): SS (second) [integer only] ● Enable End Action: whether to use end action **If True: (when “separate” is True, this is also True)** ✓ End Date: select the end date (single-option). ✓ End Time: set the end time, same as above start time.
Start Sub	The sub function that will be called when it is at the starting time.
End Sub	The sub function that will be called when it is at the end time.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	

Width	Width of the component
Height	Height of the component

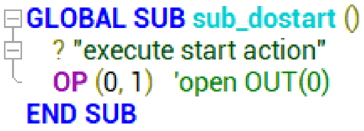
D. For Example:

Step 1: add one "value show" component, bind register to obtain & show now weekday & time.

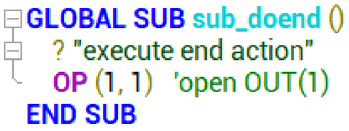


Bound Regtype	@
Bound Reg@	SYSTIME ("%A %T")

Step 2: in Basic program, edit the start sub and end sub functions.

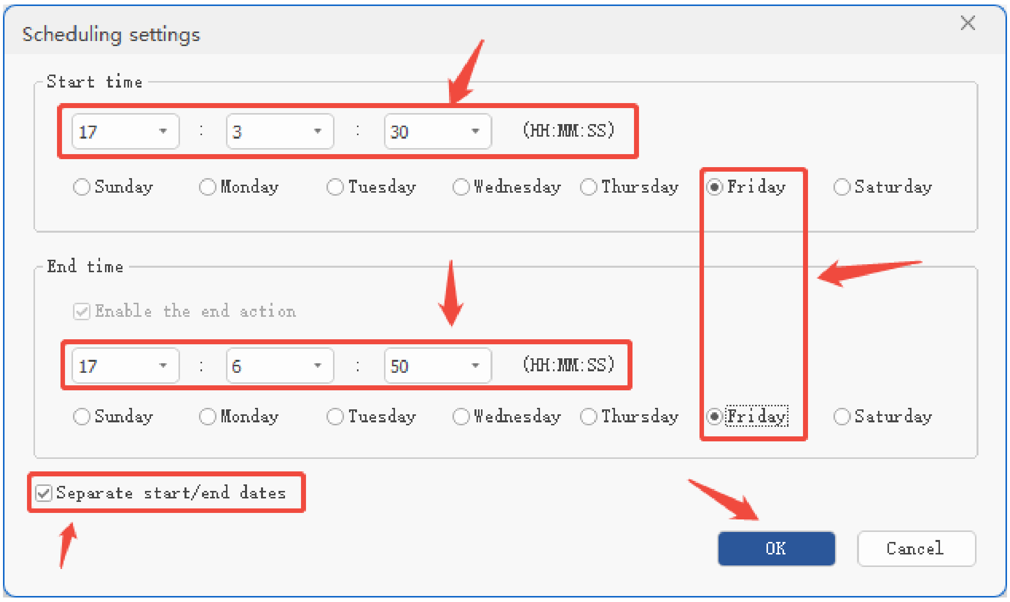


```
GLOBAL SUB sub_dostart ()
    ? "execute start action"
    OP (0, 1)   'open OUT(0)
END SUB
```



```
GLOBAL SUB sub_doend ()
    ? "execute end action"
    OP (1, 1)   'open OUT(1)
END SUB
```

Step 3: add one "scheduling" component, in its property window, open "scheduling settings" window, check "separate start / end dates", then set the starting date and time as "Friday, 17:3:30", set the end date and time as "Friday, 17:6:50".



Scheduling settings

Start time

17 : 3 : 30 (HH:MM:SS)

○ Sunday ○ Monday ○ Tuesday ○ Wednesday ○ Thursday ● Friday ○ Saturday

End time

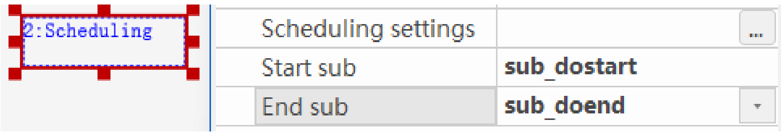
☑ Enable the end action

17 : 6 : 50 (HH:MM:SS)

○ Sunday ○ Monday ○ Tuesday ○ Wednesday ○ Thursday ● Friday ○ Saturday

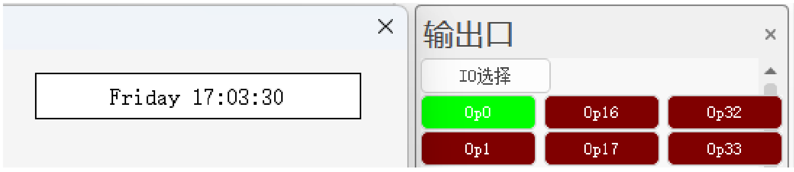
☑ Separate start/end dates

Step 4: in "scheduling" property window again, select the start sub and end sub that have been edited in Basic (step 2).

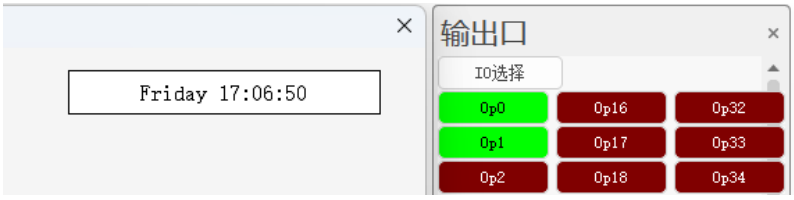


Scheduling settings	...
Start sub	sub_dostart
End sub	sub_doend

Step 5: when it arrives starting time (Friday, 17:3:30), it will execute start action, open OP 0.


Friday 17:03:30

Step 6: when it arrives starting time (Friday, 17:6:50), it will execute end action, open OP 1.


Friday 17:06:50

4.3.52. Custom

A. What is It?

Used to define the behavior by calling a Basic program. It works with the program to dynamically draw on the display screen, and the drawing area is determined by the size of the component.

B. How to Use?

Click "Tool Box", select "Custom" from "System Navigation". Put the component at suitable position, adjust the size to determine drawing area. Then, in BASIC file, please define global refresh sub and draw sub at first, then in the property window, select corresponding refresh sub & draw sub.

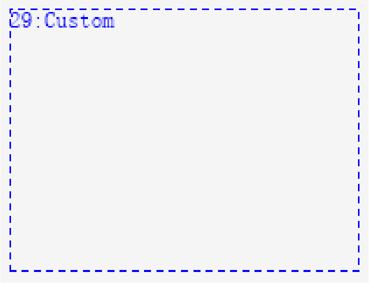
Note: for the coordinates of the drawing sub, the zero is related to the upper left corner of the component.

3 special properties of "custom" component:

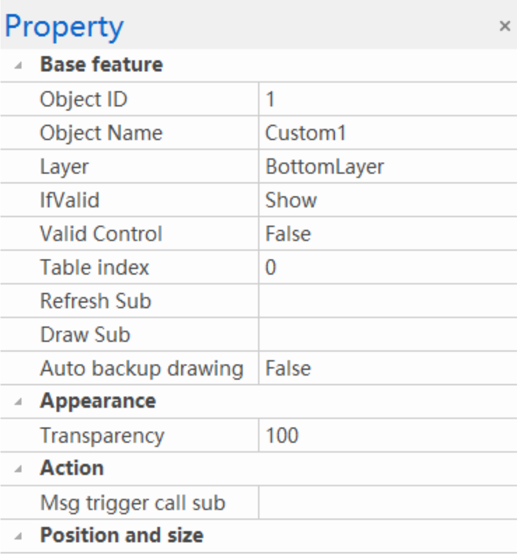
(1) Table Index: usually used for TABLE No., which can indicate where is now component data in TABLE.

(2) Refresh Sub & Draw Sub: it indicates component's SUB functions.

(3) For usage example, please refer to Development Example – Quick Start – HMI.



C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	Custom1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Table index	0
Refresh Sub	
Draw Sub	
Auto backup drawing	False
Appearance	
Transparency	100
Action	
Msg trigger call sub	
Position and size	

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading.

	● Hide: not show after downloading ● Show & Disable: show but can’t use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh.
Table Index	Indicate the TABLE position where the data of the current component is located
Refresh Sub	Be called cyclicly to determine to redraw.
Draw Sub	It is called automatically when needs to draw.
Auto Backup Drawing	Automatically back up the last drawing? The drawing will be redrawn when the SER_REDRAW command is used actively next time; if the SER_REDRAW command is not used actively, the drawing will not be redrawn and the backup drawing will be displayed directly.
Appearance	
Transparency	Set the component’s transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Postion & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

4.3.53. Basic Extend

A. What is It?

[Basic Extend] is an advanced version of "custom" component. It not only allows users to write custom initialization sub-units, refresh sub-units, and drawing sub-units, but also allows you to configure member properties (including numeric, TABLE, TABLE spaces) for components. Therefore, this facilitates you in developing independent custom components.

Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

B. How to Use?

Click "Tool Box", select "Basic Extend" from "System Navigation". Put the component at suitable position, adjust the size to determine drawing area. Then, in BASIC file, please define global refresh sub and draw sub at first, then in the property window, select corresponding refresh sub & draw sub.

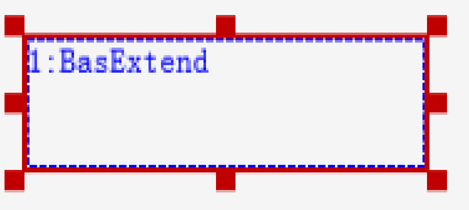
Note: for the coordinates of the drawing sub, the zero is related to the upper left corner of the component.

3 special properties of "custom" component:

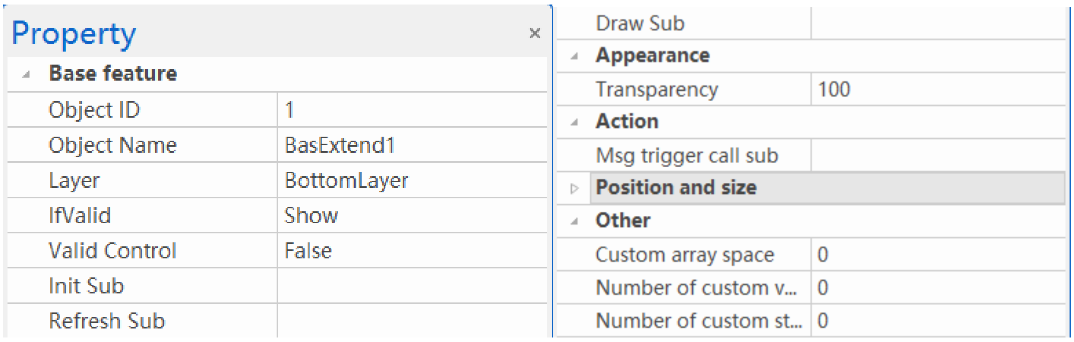
(1) Table Index: usually used for TABLE No., which can indicate where is now component data in TABLE.

(2) Refresh Sub & Draw Sub: it indicates component's SUB functions.

(3) For usage example, please refer to Development Example – Quick Start – HMI.



C. "Property" Description



Property

Base feature	
Object ID	1
Object Name	BasExtend1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Init Sub	
Refresh Sub	
Draw Sub	
Appearance	
Transparency	100
Action	
Msg trigger call sub	
Position and size	
Other	
Custom array space	0
Number of custom v...	0
Number of custom st...	0

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register's variables.
Init Sub	Initializa sub action that is called to execute once when initiializing.
Refresh Sub	Be called cyclicly to determine to redraw.
Draw Sub	It is called automatically when needs to draw.
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc.

	Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
Postion & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component
Other	
Custom Array Space	Set the customized variable & array length that can be read. After that, the extended component allocates corresponding array space for developers to store some global variables of the component. One unit represents one custom variable. Range: 0~1024. Default: 0
Number of Values	Set the number of custom values. After that, you can access (read & write) in the extended component SUB function by HMI command. Range: 0~64. Default: 0
Number of Strings	Set the number of custom strings. After that, you can access (read & write) in the extended component SUB function by HMI command. Range: 0~64. Default: 0

Note: custom array spaces, custom numeric attributes, and custom string numeric values can be accessed by HMI_CONTROLATTR, HMI_CONTROLATTRS commands. See the command descriptions for details.

D. Example

For “basic extend” usage, please refer to Chapter VIII [Manual Draw].

4.3.54. Recipe View

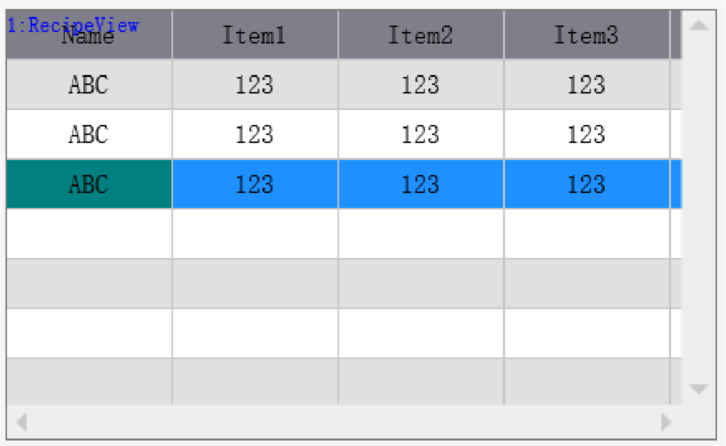
A. What is It?

[Recipe View] is used together with “recipe library” to show, select, edit the recipe library.

Note: when using this control, please make sure RTHMI version be V1.4.0 or above & RTSys version be V1.3.02 or above.

B. How to Use?

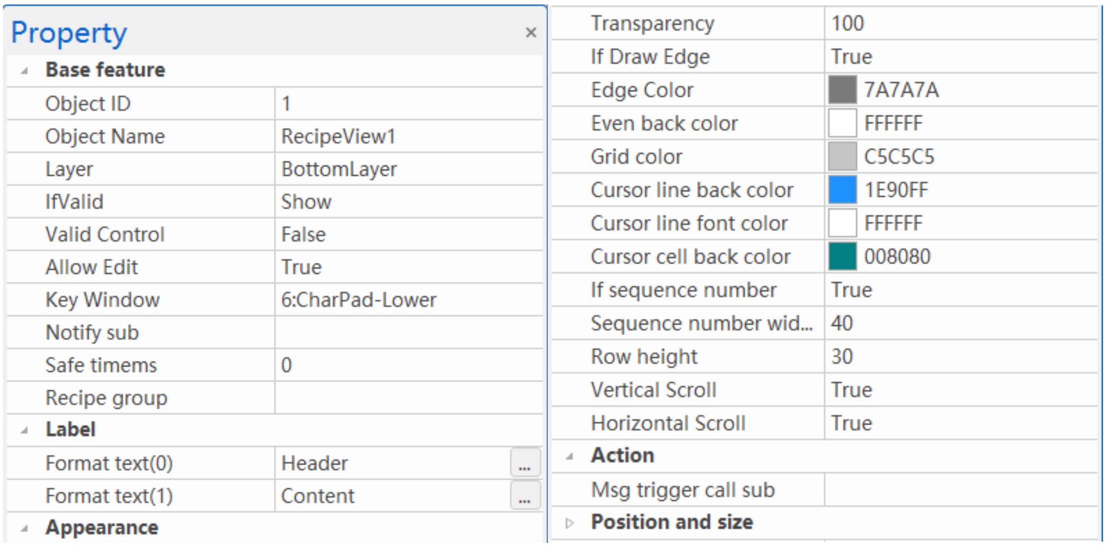
Click “Tool Box”, select “Recipe View” from “System Navigation”. Put the component at suitable position, and select corresponding recipe group in property window.



1:RecipeView

Name	Item1	Item2	Item3
ABC	123	123	123
ABC	123	123	123
ABC	123	123	123

C. “Property” Description



Property

Base feature

Object ID	1
Object Name	RecipeView1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Allow Edit	True
Key Window	6:CharPad-Lower
Notify sub	
Safe timems	0
Recipe group	

Label

Format text(0)	Header
Format text(1)	Content

Appearance

Transparency	100
If Draw Edge	True
Edge Color	7A7A7A
Even back color	FFFFFF
Grid color	C5C5C5
Cursor line back color	1E90FF
Cursor line font color	FFFFFF
Cursor cell back color	008080
If sequence number	True
Sequence number wid...	40
Row height	30
Vertical Scroll	True
Horizontal Scroll	True

Action

Msg trigger call sub	

Position and size

Property	Description
Base Feature	
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer

	● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: the middle layer ● BottomLayer: the bottom layer (default)
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: 0 = not show, 1 = displayed, 2 = displayed but will not refresh. ● Valid reg@: select @ register's variables.
Allow Edit	Whether the item is allowed to edit.
Key Window	Select the keyboard window No. that is popped up when double-clicking the item.
Notify Sub	Trigger SUB when modifying the recipe data by recipe view.
Safe Timems	The min key press time, the unit is ms.
Recipe Group	Select displayed recipe group.
Label	
Format Text (0)	Set the text format and background for the table header row.
Format Text (1)	Set the text format and background for the table other rows (odd).
Appearance	
Transparency	Set the component's transparency, which includes current color, background color, edge color, etc. Range: 0% ~ 100% (default 100%) 0% -- fully transparent, 100% -- fully show
If Draw Edge	Whether to draw the edge. If TRUE, draw it.
Edge Color	Set the edge color, it is valid when "If Draw Edge" is True.
Even Back Color	Set the background color for even rows
Grid Color	Set the grid color.

Cursor Line Back Color	Set selected row's background color.
Cursor Line Font Color	Set selected row's font color.
Cursor Cell Back Color	Set selected cell's background color.
If Sequence Number	Whether to show the sequence No.
Sequence Number Width	Set the display width of the column containing the serial number (usually the first column).
Row Height	Set each row's displayed height.
Vertical Scroll	Whethet to show vertical scroll
Horizontal Scroll	Whethet to show horizontal scroll
Action	
Msg Trigger Call Sub	Set the SUB function that is to be called when the message is triggered by configured component triggering mode. It matches with HMI_WINMSGCALL command. And SUB function is only triggered when related message is opened.
Position & Size	
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

D. Example

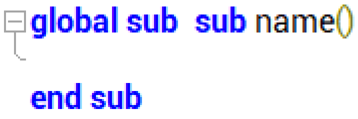
For "recipe view" usage, please refer to Chapter VIII [Recipe View].

Chapter V Call Basic Function in HMI

This chapter mainly introduce "how to call Basic function CALL SUB in HMI" and several kinds of function calling.

➢ **General Ways to Call Sub Function:**

(1) In Basic file, define global (GLOBAL) sub function, the syntax is shown below, and please note sub name and the () must be English character.



```
global sub  sub name()
end sub
```

For Example: GLOBAL SUB redraw() END SUB.

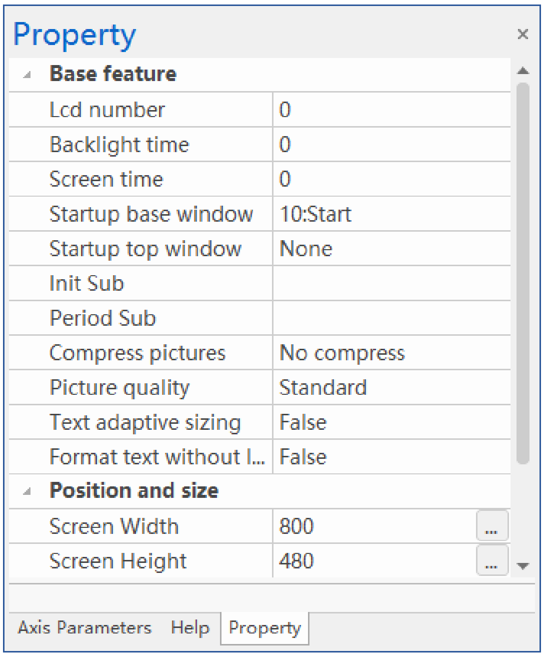
(2) In defined SUB function, edit this function's program.

(3) Open / build HMI file, and add corresponding component, then open its property window, for "action", select Call Sub, then in action sub, select corresponding sub to be called.

(4) If it is HMI setting or custom component, it can be called directly in property "xx function".

5.1. Call Basic Function SUB in HMI Setting

In menu "HMI setting", you can open below window, Init sub and period sub are optional.



Property

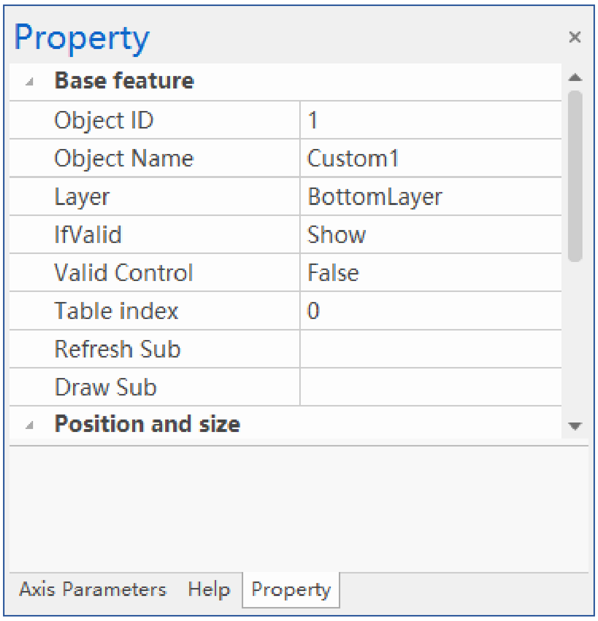
Base feature	
Lcd number	0
Backlight time	0
Screen time	0
Startup base window	10:Start
Startup top window	None
Init Sub	
Period Sub	
Compress pictures	No compress
Picture quality	Standard
Text adaptive sizing	False
Format text without l...	False
Position and size	
Screen Width	800
Screen Height	480

- **Init Sub:** when powered on, it is the function that is only called once. Generally write initial related parameter definitions into defined initialization global SUB function, and it will be called in HMI setting.
- **Period Sub:** when powered on, it is the function that is called cyclically.
- Note: init sub and period sub both select SUB function defined by GLOBAL in Basic.

5.2. Call Basic Function SUB in Custom Component

For “Custom” component, it can add draw sub and refresh sub, and these two sub functions are global functions defined in Basic.

- **Draw Sub:** it will be automatically called for drawing, in this sub, draw by calling DRAW / DRAWEX related functions, and the zero point of the drawing function is the upper left corner of the custom component.
- **Refresh Sub:** periodically call to determine whether to redraw (the system will automatically call the refresh sub periodically), refresh the drawing area, and specify which area to refresh by calling the SET_REDRAW command.



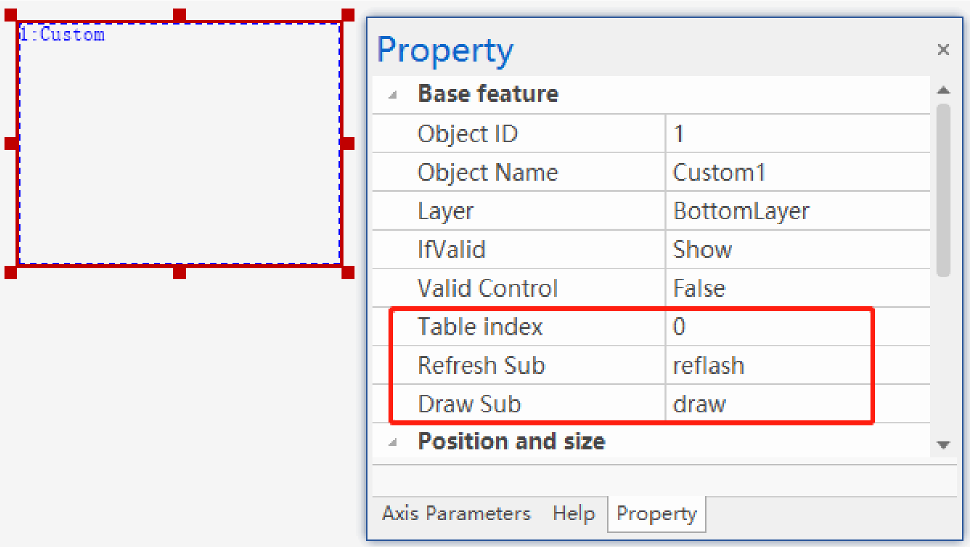
Base feature

Object ID	1
Object Name	Custom1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Table index	0
Refresh Sub	
Draw Sub	

Position and size

- **Examples:**

Example 1: in HMI window, create one Custom component, open the property window, set Table index, draw sub, refresh sub.



Property

Base feature

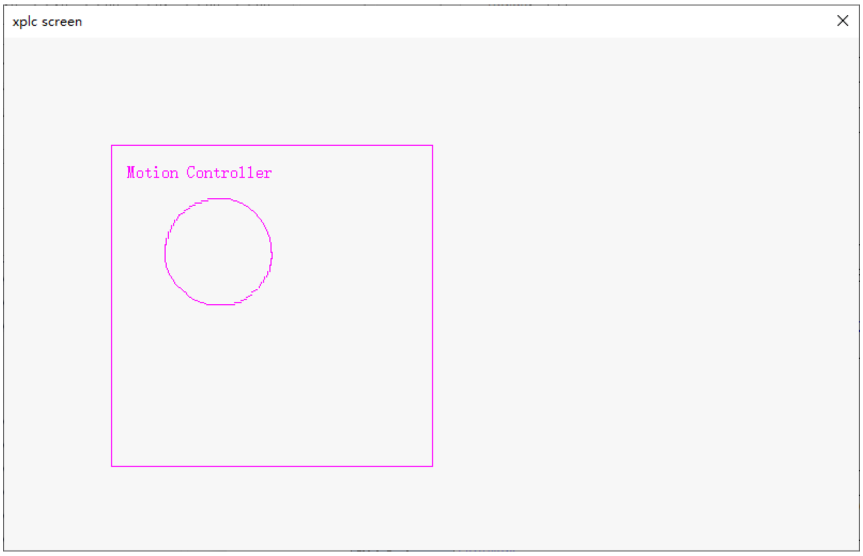
Object ID	1
Object Name	Custom1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Table index	0
Refresh Sub	reflash
Draw Sub	draw

Position and size

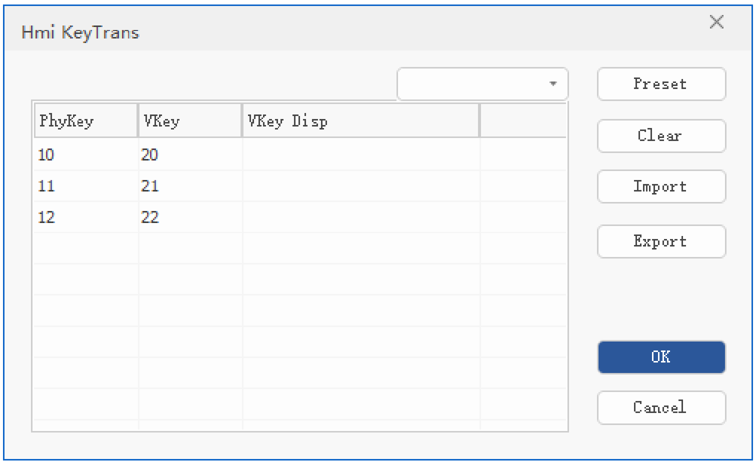
--Basic Program--

```
GLOBAL SUB reflash()            'refresh function
      SET_REDRAW
END SUB
GLOBAL SUB draw()               'draw function
      SET_COLOR(RGB(255,0,255))      'set color
      DRAWRECT(0,0,300,300)          'draw the edge in the custom component
      DRAWTEXT(10,10, "Motion Controller") 'show string in the custom component
      DRAWARC(100,100,50, 0, PI*2)    'draw one full circle
END SUB
```

--Effect--

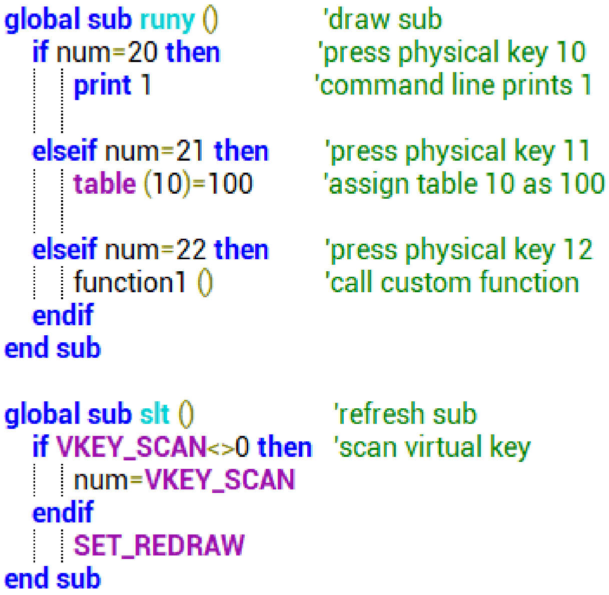


Example 2: bind virtual key 20, 21, 22 with physical key 10, 11, 12.



PhyKey	VKey	VKey Disp
10	20	
11	21	
12	22	

--Basic Program--



```
global sub runy ()          'draw sub
    if num=20 then          'press physical key 10
        print 1             'command line prints 1

    elseif num=21 then      'press physical key 11
        table (10)=100      'assign table 10 as 100

    elseif num=22 then      'press physical key 12
        function1 ()        'call custom function
    endif
end sub

global sub slt ()             'refresh sub
    if VKEY_SCAN<>0 then      'scan virtual key
        num=VKEY_SCAN
    endif
        SET_REDRAW
end sub
```

--Effect--

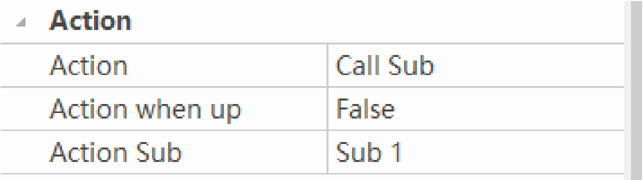
When physical key 10 pressed, command line prints 1.

When physical key 11 pressed, assign TABLE(10) as 100.

When physical key 12 pressed, call sub function1, its function can be customized.

5.3. Call Basic Function SUB in Component

Hmi components can select "call sub" in action to achieve Basic calling. And many components can do that, for example, BUTTON. After call sub, in “action sub”, you can select defined global SUB in BASIC, then, when the component is pressed or released, SUB can be called.



Action	
Action	Call Sub
Action when up	False
Action Sub	Sub 1

Chapter VI HMI Basic Commands

6.1. HMI Common Commands

6.1.1. RUN – Open File Task

Type	Task Instructions
Description	**Start a new task to execute a file on controller.** Restart the same task that will report error. Multi-task running instructions： END: Present task ends normally. STOP: Stop assigned files. STOPTASK Stop assigned tasks HALT: Stop all tasks. RUN Start file execution RUNTASK Start task that executes on one SUB or on mark
Grammar	RUN "filename"[, tasknum] filename: program file name, it can't add expansion name for BAS file tasknum: Task NO., find first valid task NO. in default mode. RUN "file.HMI", TASKID, [LCDNUM] filename: program file name, it can't add expansion name for BAS file tasknum: Task Number
Controller	General
Example	**RUN** "aaa", 1 'start task 1 to run aaa.bas file

6.1.2. SCAN_EVENT – Scan Data State Changes

<table>
<tr><td><b>Type</b></td><td>Input and output functions</td></tr>
<tr><td><b>Description</b></td><td><b>Scan data changes.</b>
<table>
<tr><th>Return Value</th><th>Description</th></tr>
<tr><td>1</td><td>Data from 0 to non-0</td></tr>
<tr><td>-1</td><td>Data from non-0 to0</td></tr>
<tr><td>0</td><td>BOOL no changes of data</td></tr>
</table>
</td></tr>
</table>

Grammar	event = SCAN_EVENT (condition) condition: express data condition
Controller	General
Example	WHILE 1 IF SCAN_EVENT(IN(0))>0 THEN 'trigger rising edge of IN0 PRINT “IN0 ON” ELSELF SCAN_EVENT(IN(0))<0 THEN 'trigger falling edge of IN0 PRINT “IN0 OFF” ENDIF WEND

6.1.3. SET_XPLCTERM – Set Touch Screen ON State

Type	Display operations
Description	**Read the state of display screen.**
Grammar	SET_XPLCTERM = value value: 0 – default, not to ON 1 – automatically open full screen XPLCTERM HDMI version
Controller	It is only valid in 5xx series controllers.
Example	SET_XPLCTERM = 1 ‘automatically open full screen XPLCTERM

6.1.4. SYSTEM – System Time

Type	System Time
Description	**Get system current time.**
Grammar	string = SYSTEM (style) style: time format, “%Y/%m/%d %H:%M:%S” => 2021/08/14/ 11:03 return time character string time format: %a: abbreviation for day of the week %A: full name for the day of the week %b: abbreviation for month

%B: full name of the month

%c: standard date time string

%C: first two digits of the year

%d: day of the month in decimal

%D: month/day/year

%e: decimal day of the month in a two-character field

%F: year-month-day

%g: last two digits of year, using week-based year

%G: year, using week-based year

%h: abbreviated month name

%H: 24-hour clock hour

%I: 12-hour clock hour

%j: decimal day of the year

%m: decimal month

%M: decimal minutes

%n: new line character

%p: display equivalent of local AM or PM

%r: 12 hours of time

%R: displays hours and minutes in 24-hour format: hh:mm

%S: decimal seconds

%t: horizontal tab

%T: displays hours, minutes and seconds: hh:mm:ss (24-hour format)

%u: day of the week, with Monday being the first day (values from 1 to 7, Monday is1)

%V: week of the year, using week-based years

%w: day of the week in decimal (values from 0 to 6, Sunday is 0)

%U: week of the year, with Sunday as the first day (values from 0 to 53)

%W: week of the year, with Monday as the first day (values from 0 to 53)

%x: standard date string

%X: standard time string

%y: decimal year without century (values from 0 to 99)

%Y: decimal year with century part

	%z, %Z time zone name, or null if no time zone name can be obtained. %%: percent sign
Controller	It is valid in 4xx series and 5xx series controllers or above.
Example	**Example 1: ?SYSTIME("%Y/%m/%d %H:%M:%S")** 2021/11/12 11:30 **Example 2: ? "Date:", SYSTIME("%x")** ? "day of the week:", SYSTIME("%u") The printed results are as follows: Data: 07/13/22 Day of the week: 3

6.2. HMI Syntax Commands

6.2.1. DMSET – Assign in Array Area

Type	Grammar Instructions
Description	**Assign value for array area.**
Grammar	DMSET arrayname(pos, size, data) pos： starting index size： length data: array to be set
Controller	General
Example	**DMSET** TABLE(0,10,2) 'assign value in the array part FOR i=0 TO 9 PRINT "TABLE",i, TABLE(i) 'print array NEXT **DMSET** TABLE(0,10,3) 'assign value in the array part FOR i=0 TO 9 PRINT "TABLE",i, TABLE(i) 'print array NEXT
Instructions	ZINDEX_LABEL, ZINDEX_ARRAY, ZINDEX_CALL

6.2.2. ZINDEX_LABEL – Create Index Pointer

Type	Grammar Instructions.
Description	**Build pointer index for SUB function or function, then it can call behind through index pointer.**
Grammar	Pointer = zindex_label (subname) subname: array or SUB name
Controller	General
Example	DIM arr1(100) ‘define array Arr1(0,1) ‘assign 1 to array address 0 Pointer = ZINDEX_LABEL (arr1) ‘build index pointer PRINT ZINDEX_ARRAY (Pointer) (0) ‘access array, print the first bit data of array
Instructions	ZINDEX_CALL, ZINDEX_ARRAY

6.2.3. ZINDEX_ARRAY – Access Array

Type	Grammar Instructions.
Description	**Access array through index pointer.**
Grammar	var = ZINDEX_ARRAY (Pointer) (index) pointer: pointer index generated from ZINDEX_LABEL index: array index
Controller	General
Example	DIM arr1(100) ‘define array Arr1(0,1) ‘assign 1 to array address 0 Pointer = ZINDEX_LABEL (arr1) ‘build index pointer PRINT ZINDEX_ARRAY (Pointer) (0) ‘access array, print the first bit data of array
Instructions	ZINDEX_CALL, ZINDEX_LABEL

6.2.4. ZINDEX_CALL – Call SUB Function

Type	Grammar Instructions.
Description	**Call SUB function through index pointer.**
Grammar	ZINDEX_CALL (zidnex) (subpara, …)

	zidnex: index pointer generated from ZINDEX_LABEL subpara: sub parameters calling
Controller	General
Example	Pointer = ZINDEX_LABEL (sub1) 'build index pointer ZINDEX_CALL (Pointer) (2) 'call function End SUB sub1 (a) PRINT a END SUB
Instructions	ZINDEX_LABEL, ZINDEX_ARRAY

6.2.5. ZINDEX_VAR – Operate Pointer Variables

Type	Grammar Instructions.
Description	**Operate variables through index pointer.**
Grammar	ZINDEX_VAR (zindex) zidnex: z pointer generated from ZINDEX_LABEL
Controller	General

Example	zindex= ZINDEX_LABEL(varname)
	ZINDEX_VAR(zindex)=value
	VAR2 = ZINDEX_VAR(zindex)

Instructions	ZINDEX_LABEL, ZINDEX_ARRAY

6.2.6. ZINDEZX_MARK – Set Pointer Mark No.

Type	Grammar Instructions.
Description	**Set the label to the label, so that the array index of the label can be recorded.**
Grammar	ZINDEX_MARK(labelname) = mark varmark = ZINDEX_MARK(labelname) labelname: If different files have definitions, note that different results are called in different files mark: mark
Controller	Valid in controllers that support ZV function or 5xx series or above.
Example	dim var1 dim arr1(10),MarkArr(10)

	dim ArrIndex,VarIndex arr1(0,1,2,3,4,5,6,7,8,9,10) VarIndex = zindex_label(var1) ‘get variables pointer, array pointer ArrIndex = zindex_label(arr1) zindex_mark("VarIndex") = 0 ‘set mark number of variable pointer and array pointer zindex_mark("ArrIndex") = 1 markarr(zindex_mark("VarIndex")) = VarIndex ‘save pointer into array subscript specified by pointer array through gained mark Number markarr(zindex_mark("ArrIndex")) = ArrIndex zindex_var(MarkArr(zindex_mark("VarIndex"))) = 777 ‘access position variables and arrays related to pointer array through mark Number ?zindex_var(MarkArr(0)) ?var1 ?zindex_array(MarkArr(zindex_mark("ArrIndex")))(9) ?zindex_array(MarkArr(1))(9) end
Instructions	ZINDEX_LABEL, ZINDEX_ARRAY

6.2.7. ZINDEX_STRUCT – Get / Access Structure Variables

Type	Grammar Instructions.
Description	**Access structural variables or arrays through pointer after getting the pointer of structural variables.**

Grammar	index = ZINDEX_LABEL(structvarname)
	ZINDEX_STRUCT(structname,index).item = var
	var = ZINDEX_STRUCT(structname,index).item

Controller	Valid in controllers that support ZV function or 5xx series or above.

Example	/
Instructions	ZINDEX_LABEL, ZINDEX_ARRAY

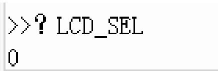
6.2.8. ZINDEX_AVOBJ – Get Object Index Data

Type	Grammar Instructions.
Description	**Get the data of object index.**
Grammar	ZINDEX_ZVOBJ (index)
Controller	Valid in controllers that support ZV function or 5xx series or above.
Example	ZVOBJECT img ZV_READIMAGE(img,"logo.png",0) ‘read one image index = ZINDEX_LABEL(img) ‘get index of image object ZV_LATCH(ZINDEX_ZVOBJ(index), 0) ‘get the data of image index, and display it in latch channel 0
Instructions	ZINDEX_LABEL, ZINDEX_ARRAY

6.3. HMI Display Commands

6.3.1. LCD_SEL – Set / Get LCD No. of Task No.

Type	Syntax Command
Description	Set/get the LCD No. bound to the corresponding task No., the task No. used by the online command is the last one.
Grammar	LCD_SEL[(hmilcdnum)]=lcdnum hmilcdnum: current LCD No. lcdnum: LCD No. that is to be set num = LCD_SEL[(hmilcdnum)] num: obtained LCD No. hmilcdnum: current LCD No.
Controller	Controllers that support RTHMI.
Example	➢ **Example 1:** LCD_SEL(23)=1 'set current LCD No. as 1 from 0 LCD_SEL=1 'set current LCD No. as 1 from 0

	➢ **Example 2:** ? LCD_SEL(23) 'print current LCD No. ? LCD_SEL 'print current LCD No.  >>? LCD_SEL 0

6.3.2. LCD_FEATURE – Read Displayer Features

Type	Display instructions
Description	**Read features of displayer. It is valid when controller and displayer both support this function.**
Grammar	var = LCD_FEATURE(lcdnum, featurenum) lcdnum: 0 – displayer Number featurenum: feature number 0: type, 0 – displayer built in controller, 1 – computer term, -1 – disconnected, -2 – not exist, 300 – 300X, 400 – 400X, 701 – 7 inch touch screen 1: width, it is the physical width of displayer, 0 – changeable width range 2: height, it is the physical height of displayer, 0 – changeable height range
Controller	Controllers that support RTHMI.
Example	PRINT LCD_FEATURE(0,0) 'Print display type PRINT LCD_FEATURE(0,1) 'Print display physical width PRINT LCD_FEATURE(0,2) ‘Print display physical height

6.3.3. LCD_LEDSTATE – Control LED State

Type	System Screen Parameters
Description	Set LED state on the teach pendant.. This function is valid when the controller and displayer both support.
Grammar	LCD_LEDSTATE(lcdnum) = VAR lcdnum: 0 – displayer No Set bit by bit, default is 1, RUN LED is ON.
Controller	Controller supports RTHMI, and its firmware version is 230801 or above.
HMI	ZHD500X

Example	LCD_LEDSTATE(0)=3 ‘RUN & ALM leds are ON.

6.3.4. LCD_WDOGTIME – Displayer Offline Process Time

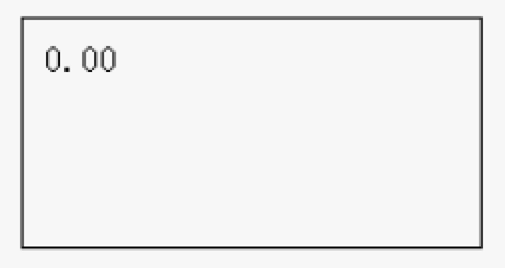
Type	System Screen Parameters
Description	Set the screen offline processing time, the unit is ms. When the screen is operated without communication during this time, the emergency stop switch (physical key number = 5) is automatically pressed, and when it is equal to 0, this function is not enabled. It needs to be supported by the firmware version above 20180404 of the controller 5 series, and supported by the standard firmware version 20170721 of the 4 series.
Grammar	LCD_WDOGTIME(lcdnum) = time lcdnum: 0 – displayer Number time: time (ms)
Controller	Controllers that support RTHMI.
Example	LCD_WDOGTIME(0) = 100

6.3.5. LCD_CONNECT – LCD No. Config

Type	Teach pendant settings.
Description	This is used in HMI side when the communication between controller and teach pendant is built successfully. Then, set the needed controller LCD (HMI) No., default is 0. When the controller supports multiple HMI, you can set it as other values. After that, please restart the HMI.
Grammar	VAR1 = LCD_CONNECT LCD__CONNECT = VAR1
Controller	Controllers that support RTHMI.

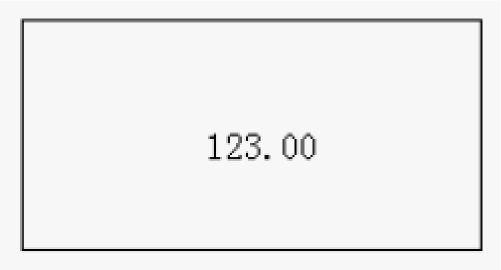
Example	LCD_CONNECT = 1
	LCD_CONNECT = 0

6.3.6. DRAWNUM – Show Value in Custom Component

Type	Display instructions.
Description	It shows one value. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DrawNum(x,y,value,n,dot) x, y: the coordinate position of the upper left corner of the display area value: default display value n: total length digits, including decimal point and sign bit. When N is set to a negative number, it means right alignment dot: how many digits to take after the decimal point
Controller	Controllers that support RTHMI.
Example	DRAWRECT(0,0,200,100) 'draw the edge in "Custom" object DRAWNUM(10,10,0,4,2) 'displayed 0.00 at (10,10) in the upper left corner of the "custom" object  0.00
Instructions	DRAWNUM2

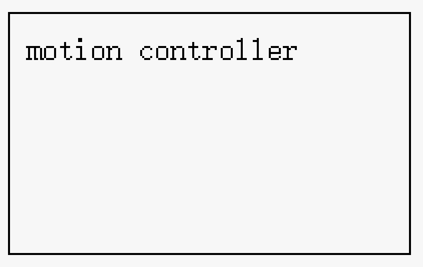
6.3.7. DRAWNUM2 – Show Value in Specified Position

Type	Display instructions.
Description	It shows one value, and assign one box to display. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DrawNum2(x1,y1, x2, y2, align, value, n, dot [,drawrect]) x1, y1: show the coordinate position of upper left corner x2, y2: show the coordinate position of bottom right corner align: align selection 0 – center alignment

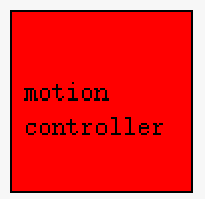
	>0 – left alignment, the value represents the distance from the left <0 – right alignment, absolute value represents the distance from the right value: default display value n: total length digits, including decimal point and sign bit. When N is set to a negative number, it means right alignment dot: how many digits to take after the decimal point drawrect: it is optional, 0 – not to draw edge (default), 1 – draw the edge
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,100) ‘draw the edge in “Custom” object DRAWNUM2(10,10,200,100,0,123,6,2) ‘displayed in the middle of the two coordinate positions, keep two decimal places, display result: 123.00  123. 00
Instructions	DRAWNUM

6.3.8. DRAWTEXT – Draw String

Type	Display instructions.
Description	It shows one string, it can be Chinese. SRTING can be used as character string expression. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DrawText(x,y,STRING) X, Y: the coordinate position of the upper left corner of the display area STRING: the displayed string
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,120) ‘draw the edge in “Custom” object DRAWTEXT (10,10,”motion controller) ‘display text “motion controller” on “Custom” object.

	 motion controller
Instructions	DRAWTEXT2

6.3.9. DRAWTEXT2 – Draw String

Type	Display instructions.
Description	It shows one character string, and shows it on assigned box. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DrawText2(x1,y1, x2,y2, align, STRING[,drawrect])) x1, y1: show the coordinate position of upper left corner x2, y2: show the coordinate position of bottom right corner align: align selection 0 – center alignment >0 – left alignment, the value represents the distance from the left <0 – right alignment, absolute value represents the distance from the right STRING: displayed character string drawrect: 0 – not to draw edge (default), 1 – draw the edge
Controller	Controllers that support ZHMI.
Example	DRAWTEXT2 (10,10,100,100,0,"motion controller",1) 'display text on "Custom" object, and draw the edge for specified area  motion controller
Instructions	DRAWTEXT

6.3.10. DRAWLIBTEXT – Show Text Library String

Type	Display instructions.

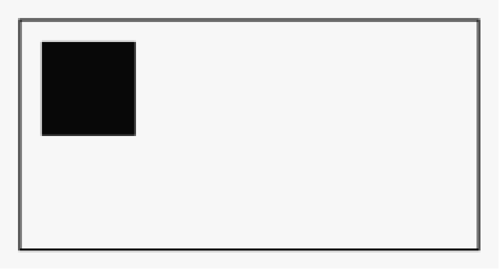
Description	It shows character string content of text. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWLIBTEXT(x,y,state, "textname") X, Y: the coordinate position of the upper left corner of the display area state: the state of text lib textname: name of text lib
Controller	Controllers that support ZHMI.
Example	DRAWLIBTEXT (10,10,0, "text1")
Instructions	DRAWTEXT2, DRAWTEXT, DRAWLIBTEXT2

6.3.11. DRAWLIBTEXT2 – Show Text Library String

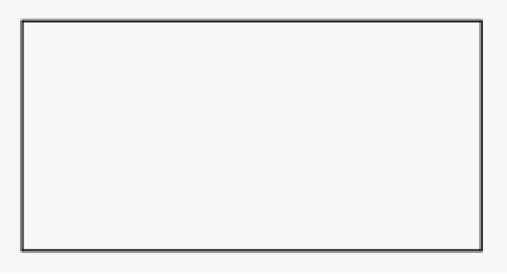
Type	Display instructions.
Description	It shows character string content of text. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWLIBTEXT2(x1,y1,x2,y2,align,state, "textname"[,drawrect]) x1, y1: show the coordinate position of upper left corner x2, y2: show the coordinate position of bottom right corner align: align selection 0 – center alignment >0 – left alignment, the value represents the distance from the left <0 – right alignment, absolute value represents the distance from the right state: the state of text lib textname: name of text lib drawrect: 0 – not to draw edge (default), 1 – draw the edge
Controller	Controllers that support ZHMI.
Example	DRAWLIBTEXT2 (10,10,100,50,0)
Instructions	DRAWTEXT2, DRAWTEXT, DRAWLIBTEXT

6.3.12. DRAWREVERSE – Draw Square

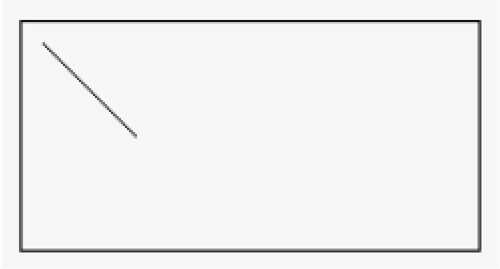
Type	Display instructions.

Description	Draw one filled (black) box. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWREVERSE(x1,y1,x2,y2) x1, y1: show the coordinate position of upper left corner x2, y2: show the coordinate position of bottom right corner
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,100) 'draw the edge in "Custom" object DRAWREVERSE(10,10,50,50) 'Fill the black box 

6.3.13. DRAWRECT – Draw Rectangle

Type	Display instructions.
Description	Draw one box. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWRECT(x1,y1,x2,y2) x1, y1: show the coordinate position of upper left corner x2, y2: show the coordinate position of bottom right corner
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,100) 'draw the edge in "Custom" object 
Instructions	DRAWEX_RECT, DRAWEX_ELLIPSE

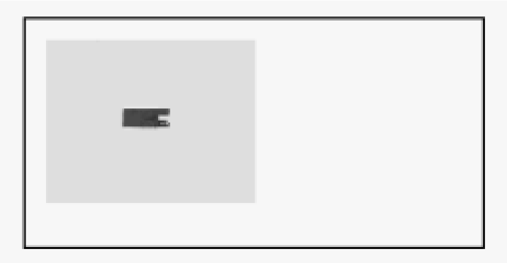
6.3.14. DRAWLINE – Draw Segment

Type	Display instructions.
Description	Draw one line. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWLINE (x1,y1,x2,y2) x1, y1: starting point coordinate position of line x2, y2: ending point coordinate position of line
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,100) 'draw the edge in "Custom" object DRAWLINE(10,10,50,50) 'draw the line 
Instruction	DRAWEX_LINE, DRAWARC

6.3.15. DRAWCLEAR – Clear Content in Specified Area

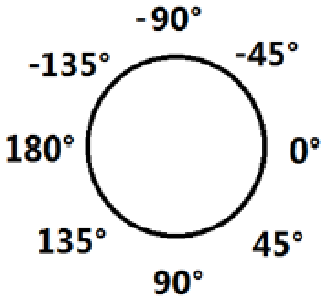
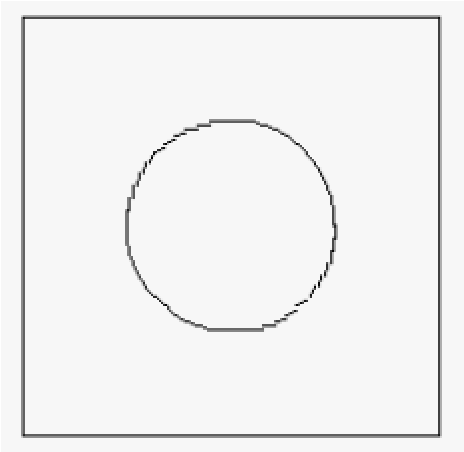
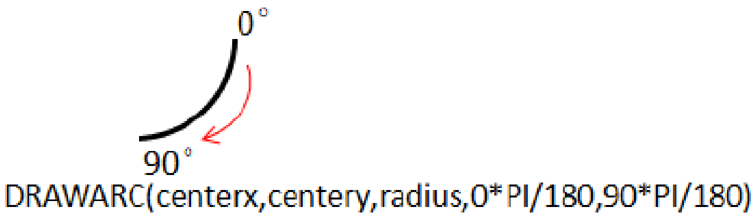
Type	Display instructions.
Description	Clear assigned reading area. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWCLEAR ([x1,y1,x2,y2]) x1, y1: clear the coordinate position of upper left corner x2, y2: clear the coordinate position of bottom right corner all is cleared when there is no parameter.
Controller	Controllers that support ZHMI.
Example	DRAWCLEAR (10,10,50,50)

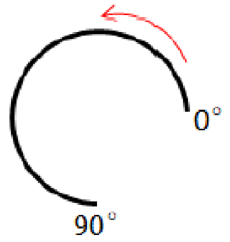
6.3.16. DRAWPIC – Insert PNG File

Type	Display instructions.
Description	To draw a picture, the picture file must be added to the project first, and the picture should be added in the file view. Note that the picture takes up more space, and do not add unnecessary pictures to the project. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWPIC (Picname, x1,y1[,x2,y2]) picname: picture file name x1, y1: starting point coordinate position of line x2, y2: ending point coordinate position of line
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,100) 'draw the edge in "Custom" object DRAWPIC("1.bmp",10,10,100,80) 'add the picture 
Instruction	DRAWLIBPIC

6.3.17. DRAWARC – Draw Arc

Type	Display instructions.
Description	Draw the arc. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWARC(centx, centy, radius, startangle, endangle) centx, centy: the position of the center of the circle radius: radius startangle: starting angle, in radian unit (formula: radian=angle*PI/180) endangle: end angle, in radian unit

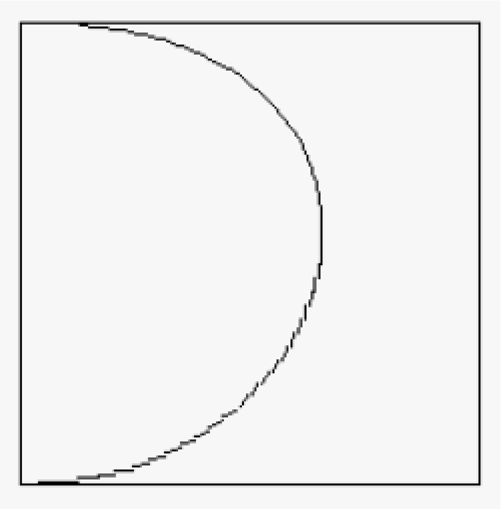
	Description of angles drawn:  Draw an arc from a start angle to an end angle: If start angle < end angle: draw an arc clockwise If start angle > end angle: draw an arc counterclockwise
Controller	Controllers that support ZHMI.
Example	**Example 1: draw one full circle** DRAWRECT(0,0,200,100) 'draw the edge in "Custom" object DRAWARC(100,100,50,0,PI*2) 'draw one full circle  **Example 2: draw the arc**  DRAWARC(centerx,centery,radius,0*PI/180,90*PI/180) **Example 3: When it is necessary to draw a superior arc segment, add 2*pi to the smaller angle between the starting angle and the ending angle.**

	 DRAWARC(centerx,centery,radius,0*PI/180+2*PI,90*PI/180)
Instruction	DRAWLINE, DRAWEX_ARC

6.3.18. DRAWLIBPIC – Insert PNG Picture

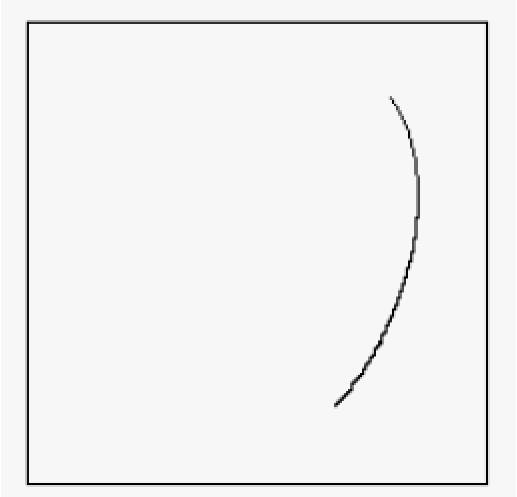
Type	Display instructions.
Description	To draw a picture, the picture file must be added to the project first, and the picture should be added in the file view. Note that the picture takes up more space, and do not add unnecessary pictures to the project. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWLIBPIC (piclibnamePath, state, x1,y1[,x2,y2]) piclibnamePath: the path of the picture library to be loaded 1. Under the system picture library: System\secondary directory name\the name of the loaded picture 2. User picture library: User\secondary directory name\picture library name state: the state number of the picture x1, y1: the coordinate position of the upper left corner of the display area x2, y2: the coordinate position of the lower right corner of the display area
Controller	Controllers that support ZHMI.
Example	DRAWLIBPIC ("User\default\LOGO",0,100,100,300,300) 'display the 0 state named LOGO in the picture library in the area (100,100) to (300,300)
Instruction	DRAWPIC

6.3.19. DRAWBEZIER – Draw Bezier Curve

Type	Display instructions.
Description	To draw the Bezier. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWBEZIER(x1,y1,x2,y2,x3,y3,x4,y4) x1,y1: 1st control point x2,y2: 2nd control point x3,y3: 3rd control point x4,y4: 4th control point
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,200) ‘draw the edge in “Custom” object DRAWBEZIER(0,0,200,0,150,200,0,200) ‘draw the Bezier 
Instruction	DRAWEX_BEZIER, DRAWBSPLINE, DRAWEX_BSPLINE

6.3.20. DRAWBSPLINE – Draw B Type Spline Curve

Type	Display instructions.
Description	To draw B spline curve This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWBSPLINE(x1,y1,x2,y2,x3,y3,x4,y4) x1,y1: 1st control point

	x2,y2: 2nd control point x3,y3: 3rd control point x4,y4: 4th control point
Controller	Controllers that support ZHMI.
Example	DRAWRECT(0,0,200,200) ‘draw the edge in “Custom” object DRAWBSPLINE(0,0,200,0,150,200,0,200) ‘draw the B spline curve 

6.3.21. DRAWDTLIST – Draw Graphics

Type	Display instructions.
Description	This is draw instruction to speed up TABLE data drawing. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWDTLIST (dtstart, imax, ispace, fxstart, fystart, fxlevel, fylevel, imode, [drawtype], [TYPE1, TYPE2, TYPE3, TYPE4]) dtstart: table starting position, pointing to the row type of the first row line type: It is defined by the user. You can set 0 to be a point or a straight line, and 1 to be a point or a straight line. imax: the number of rows (the number of groups of the set table) The number of rows is not necessarily equal to the number of table groups set, and can be filled in according to the actual situation. ispace: row interval (that is, the difference between the starting positions of each table, if the imode parameter is 1, the row interval generally cannot be <2, if the imode is offset by N data, the minimum row interval needs to be increased by N accordingly)

fxstart: The offset of the X coordinate of the upper left corner, which is offset relative to the X coordinate set in tbale (>0—offset to the left, <0—offset to the right, =0 does not offset)

fystart: The offset of the Y coordinate of the upper left corner, which is offset relative to the Y coordinate set in tbale (>0—upward offset, <0—downward offset, =0 no offset)

fxlevel: the numerical ratio of the X coordinate in the table

fylevel: the numerical ratio of the Y coordinate in the table

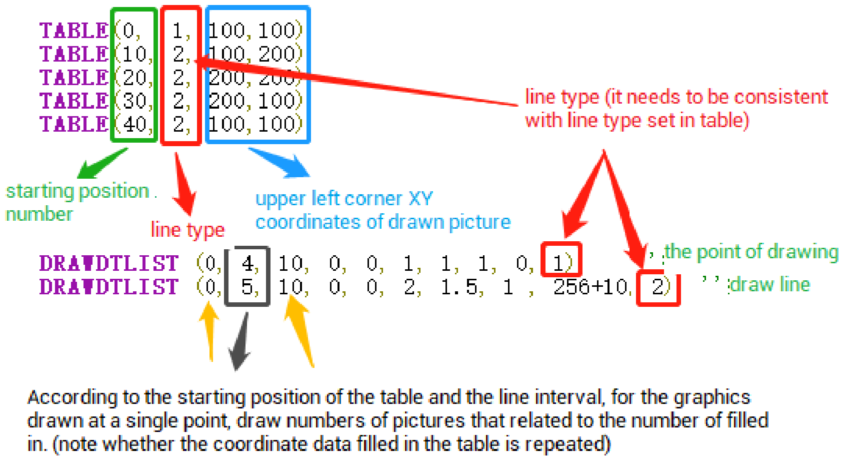
imode: the storage format

Value	Description
10	DSB reads the stored format, at this time, the position of X and the row type are offset by 3 data
1-9	At this time, it means that the position of X in the table is offset by N data from the row type, and the data of XY is next to each other
Others	Invalid

drawtype: display type

Value		Description
0 (default)		Directly draw one point
N*256+ sub display mode N = radius / line width (N: 1 ≤ N ≤ 16)	N*256+ 1	Single point solid arc
	N*256+ 2	Single point hollow arc
	N*256+ 3	Single point solid square
	N*256+ 4	Single point hollow square
	N*256+ 5	Single point cross
	N*256+ 6	Single point fork
	N*256+ 9	Single point display, draw a dotted line between the previous point, N is only is 1.
	N*256+ 10	Draw a straight line between two points before and after, must be of the same type
	N*256+ 11	Set the starting point, middle point and end point of the arc, and automatically calculate and display the arc (in this mode, the radius cannot be adjusted by

<table>
<tr>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>N)</td></tr>
<tr><td></td><td>N*256+ 12</td><td>Set the end point, middle point and starting point of the arc, and automatically calculate and display the arc (in this mode, the radius cannot be adjusted by N)</td></tr>
<tr><td></td><td>N*256+ 13</td><td>The whole circle, calculate and display the circle with the previous 2 points (in this mode, the radius cannot be adjusted by N)</td></tr>
<tr><td></td><td>N*256+ 19</td><td>The sequence is the isolated point, the end point of the straight line, the end point of the arc, the end point of the full circle, and the teaching routine is quickly displayed</td></tr>
<tr><td></td><td>N*256+ 90</td><td>Numerical control system is reserved, the type defined by the standard is G00 G01 G02 G03, etc., and the row type is based on the design of the numerical control system.</td></tr>
</table>
TYPE1~TYPE4: the type of line to be drawn, at most 4 line types can be filled in at a time for the same drawing (that is, this parameter can be filled with up to 4 line types in a DRAWDTLIST command), and the graph must be drawn corresponding to the line type set in the table .
</td>
</tr>
<tr>
<td><b>Controller</b></td>
<td>Controllers that support ZHMI (It is recommended to use the latest firmware).</td>
</tr>
<tr>
<td><b>Example</b></td>
<td>The row type data is stored in the first position in the table. In this example, 1 means a single point, 2 means a straight line, the X coordinate is stored in the second position, and the Y data is stored in the third position. If the XY coordinates are not filled, the default is 0, the XY coordinates here are relative to the “custom” object, and the coordinates of the custom object take the upper left corner as the origin (0,0).</td>
</tr>
</table>

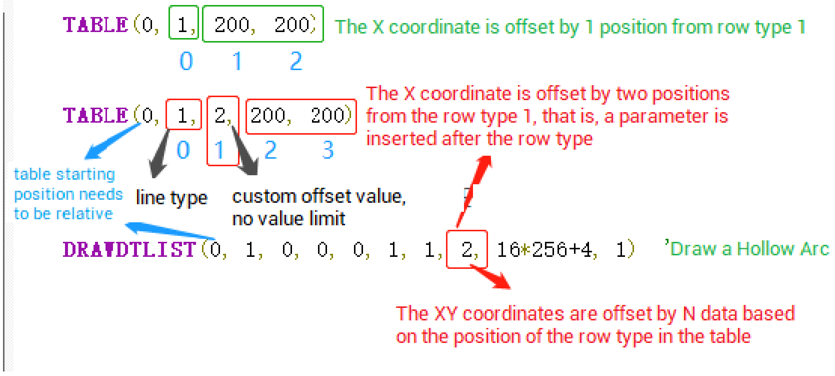


```
TABLE(0,  1, 100, 100)
TABLE(10, 2, 100, 200)
TABLE(20, 2, 200, 200)
TABLE(30, 2, 200, 100)
TABLE(40, 2, 100, 100)

DRAWDTLIST (0, 4, 10, 0, 0, 1, 1,   1 , 0,      1)
DRAWDTLIST (0, 5, 10, 0, 0, 2, 1.5, 1 , 256+10, 2)
```

line type (it needs to be consistent with line type set in table)
starting position number
line type
upper left corner XY coordinates of drawn picture
' ,the point of drawing
' ' :draw line

According to the starting position of the table and the line interval, for the graphics drawn at a single point, draw numbers of pictures that related to the number of filled in. (note whether the coordinate data filled in the table is repeated)

How to use imode (1-9) storage format for parameters:



```
TABLE(0, 1, 200, 200)
TABLE(0, 1, 2, 200, 200)
DRAWDTLIST(0, 1, 0, 0, 0, 1, 1, 2, 16*256+4, 1)
```

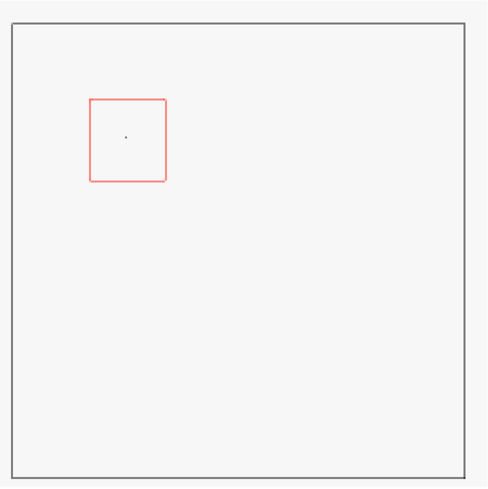
The X coordinate is offset by 1 position from row type 1
0
1
2
The X coordinate is offset by two positions from the row type 1, that is, a parameter is inserted after the row type
0
1
2
3
table starting position needs to be relative
line type
custom offset value, no value limit
'Draw a Hollow Arc
The XY coordinates are offset by N data based on the row type in the table

Example 1: draw one point

TABLE(0, 1, 100, 100)

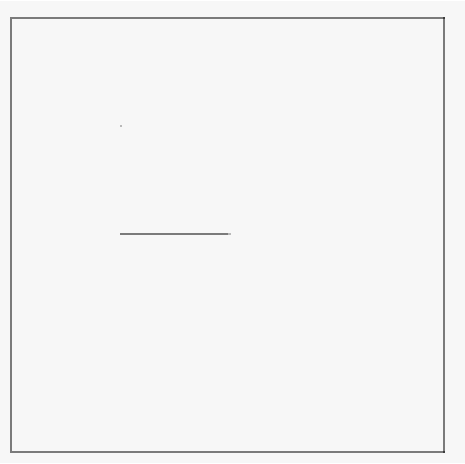
DRAWDTLIST (0, 1, 10, 0, 0, 1, 1, 1, 0, 1)

'draw a point (the position of the point is in the rectangle frame, the rectangle in the figure is not an instruction to draw, it is only a reminder)



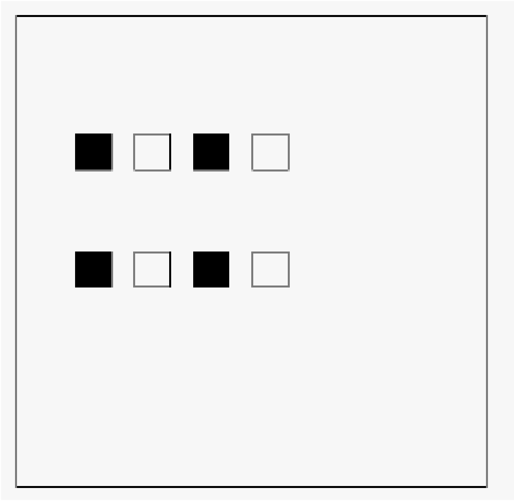
Example 2: draw a segment of straight line

```
TABLE(10,2,100,200)
TABLE(20,2,200,200)
DRAWDTLIST(10,2,10,0,0,1,1,1,256+10,2)
                                        'draw a straight line (solid line)
```



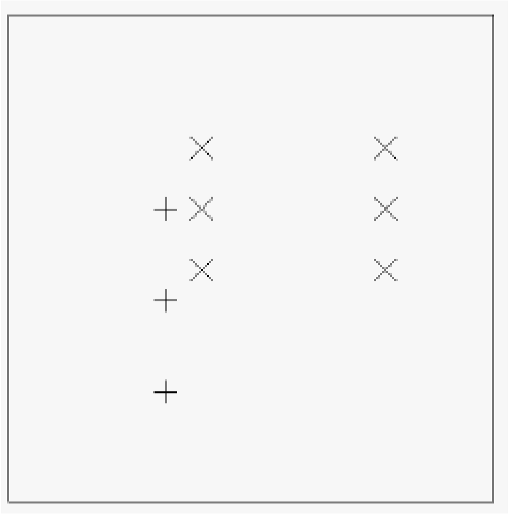
Example 3:

```
TABLE(0,2,100,100)
TABLE(10,2,100,200)
TABLE(20,2,200,200)
TABLE(30,2,200,100)
DRAWDTLIST(0,4,10,0,0,1,1,1,16*256+4,2)  'draw a hollow rectangle
DRAWDTLIST(0,4,10,40,0,1,1,1,16* 256+3,2) 'draw a solid rectangle
```



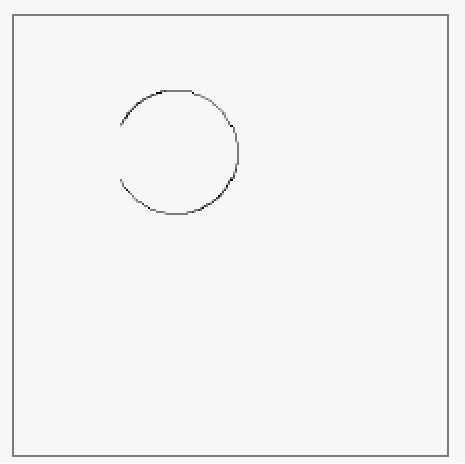
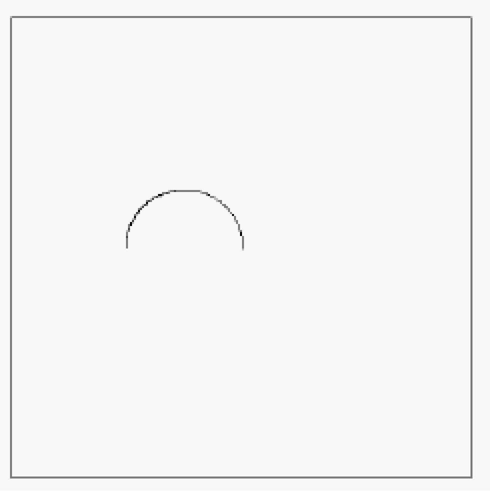
Example 4:

```
TABLE(0,1,100,100)
TABLE(3,1,100,150)
TABLE(6,1,200,150)
TABLE(9,1,200,100)
TABLE(12,1,100,200)
TABLE(15,1,200,200)
DRAWDTLIST(0,6,3,0,0,1.5,1,1,10*256+6,1)
            'draw 6 forks (X coordinate expanded by 1.5 times)
DRAWDTLIST(9,3,3,80,0,1,1.5,1,10*256+5,1)
            ‘draw 3 crosses (Y coordinate scaled up by 1.5 and X coordinate
            offset left by 80)
```



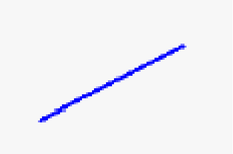
Example 5:

```
TABLE(0,1,100,100)       'the starting point of the arc
TABLE(10,1,200,150)      'the middle point of the arc
TABLE(20,1,100,150)      'the end point of the arc
DRAWDTLIST(0,3,10,0,0,1,1,1,10*256+11,1)
            'draw an arc (the way of drawing an arc with three points)
```

	 **Example 6: draw the arc** TABLE(0,1,200,200) TABLE(10,1,150,150) TABLE(20,1,100,200) DRAWDTLIST(0,4,10,0,0,1,1,1,16*256+12,1) 'draw a semicircle 
Instructions	DRAWLINE

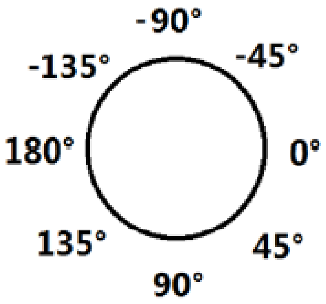
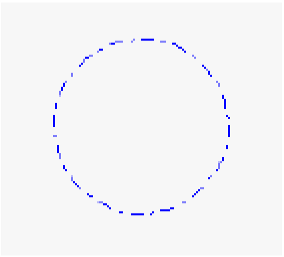
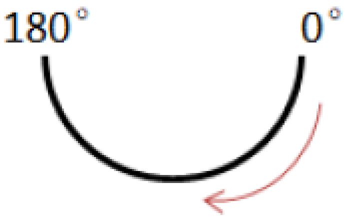
6.3.22. DRAWEX_LINE – Draw Segment (with format)

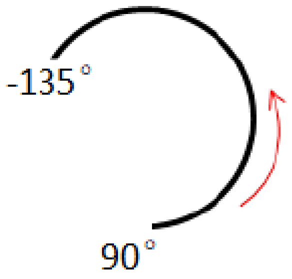
Type	Drawing instructions.
Description	Draw a straight line. The difference with DRAWLINE is that DRAWEX_LINE can set the line width and line type, but DRAWLINE does not support it. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom control, but the DRAW related commands are unlimited.

	This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_LINE(x1,y1,x2,y2) x1, y1: the coordinate position of the starting point of the line x2, y2: the coordinate position of the end point of the line
Controller	Controllers that support ZHMI (4xx series controllers and above)
Example	SETEX_LINE(3,0) 'set line width SET_COLOR(RGB(0,0,255)) 'set line color DRAWEX_LINE(100,100,200,50) 'draw the straight line 
Instruction	DRAWLINE, SETEX_LINE, SET_COLOR

6.3.23. DRAWEX_ARC – Draw Arc (with format)

Type	Drawing instructions.
Description	Draw circular/elliptical arcs. The difference with DRAWARC is that DRAWEX_ARC can set the line width and line type, but DRAWARC does not support it. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom control, but the DRAW related commands are unlimited. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_ARC(centx, centy, radius, startangle, endangle) centx, centy: the position of the center of the circle radius: radius startangle: starting angle, in radian unit (formula: radian=angle*PI/180) endangle: end angle, in radian unit Description of angles drawn:

	 Draw an arc from a start angle to an end angle: If start angle < end angle: draw an arc clockwise If start angle > end angle: draw an arc counterclockwise
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	**Example 1: draw the full circle** SETEX_LINE(1,3) ‘set line style DRAWEX_ARC(200,200,50,50,0,2*PI) ‘draw one full circle  **Example 2:** SETEX_LINE(3,0) 'set line width DRAWEX_ARC(200,200,50,50,0*PI/180,PI) ‘draw semicircle clockwise  **Example 3:** SETEX_LINE(2,0) 'set line width DRAWEX_ARC(200,200,50,50,90*PI/180,-135*PI/180) 'draw an arc counterclockwise

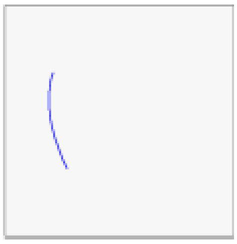
	
Instruction	DRAWARC, DRAWEX_LINE, SETEX_LINE, SET_COLOR

6.3.24. DRAWEX_BEZIER – Draw Bezier (with format)

Type	Drawing instructions
Description	Draw Bezier curves. The difference with DRAWBEZIER is that DRAWEX_BEZIER can set the line width and line type, but DRAWBEZIER does not support it. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom control, but the DRAW related commands are unlimited. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWEX_BEZIER(x1,y1,x2,y2,x3,y3,x4,y4) x1,y1: 1st control point x2,y2: 2nd control point x3,y3: 3rd control point x4,y4: 4th control point
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SETEX_LINE(1,1) 'set the line type to dotted line DRAWEX_BEZIER(200,0,350,100,100,300,200,400) 'draw a Bezier curve
Instruction	DRAWBEZIER, DRAWBSPLINE, SETEX_LINE, SET_COLOR

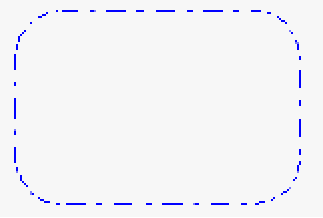
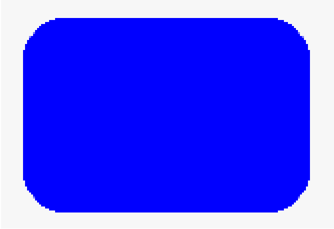
6.3.25. DRAWEX_BSPLINE – Draw B Type Spline Curve (with format)

Type	Drawing instructions
Description	Draw a B-spline curve. The difference with DRAWBSPLINE is that

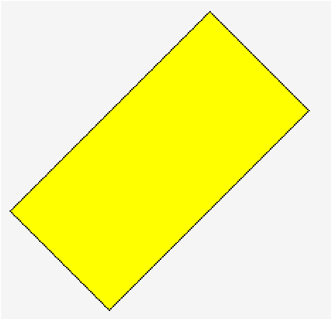
	DRAWEX_BSPLINE can set the line width and line type, but DRAWBSPLINE does not support it. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom control, but the DRAW related commands are unlimited. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_BSPLINE(x1,y1,x2,y2,x3,y3,x4,y4) x1,y1: 1st control point x2,y2: 2nd control point x3,y3: 3rd control point x4,y4: 4th control point
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SETEX_LINE(2,0) 'set line width SET_COLOR(RGB(0,0,255)) 'set line color DRAWEX_BSPLINE(200,0,50,100,100,300,200,400) 'draw B spline curve 
Instructions	DRAWBSPLINE, DRAWBEZIER, DRAWEX_BEZIER, SETEX_LINE, SET_COLOR

6.3.26. DRAWEX_RECT – Draw Rounded Corners Rectangle (with format)

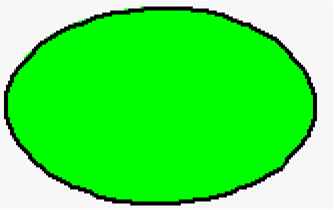
Type	Drawing instructions
Description	Draw a rounded rectangle in the specified area. The difference with DRAWRECT is that DRAWEX_RECT supports setting line width, line type and filling drawing, while DRAWRECT does not. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom control, but the DRAW related commands are unlimited.

	This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_RECT(x1,y1,x2,y2,round [,iffill]) x1, y1: show the coordinate position of upper left corner x2, y2: show the coordinate position of bottom right corner round: chamfer radius iffill: whether to fill, default is 0 (not to fill) 0 – not to fill, non 0 – fill (note: the one drawn later may block the one drawn before, so it is recommended to draw the fill first and then draw the border when selecting fill)
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	**Example 1:** SETEX_LINE(1,3) 'set line type SET_COLOR(RGB(0,0,255)) 'set line color DRAWEX_RECT(50,50,200,150,25) 'draw a rounded rectangular border with a radius of 25  **Example 2:** SET_COLOR(RGB(255,255,255),RGB(0,0,255)) 'set fill color DRAWEX_RECT(50,50,200,150,20,1) 'draws a fillable rounded rectangle 
Instructions	DRAWRECT, DRAWEX_ELLIPSE, DRAWEX_POLYGON, SETEX_LINE, SET_COLOR, DRAWEX_POLYGON2

6.3.27. DRAWEX_ROTRECT – Draw Rotary Rectangle (with format)

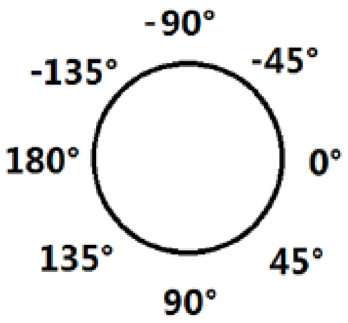
Type	Drawing instructions
Description	Draw a rotary rectangle in the specified area. This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_ROTRECT(cx, cy, w, h, angle [, iffill]) cx, cy: rotary rectangle's center coordinate position w, h: rotary rectangle's width and height angle: rectangle's rotary angle, the unit is radian, anticlockwise. iffill: whether to fill, default is 0 (not to fill) 0 – not to fill, non 0 – fill (note: the one drawn later may block the one drawn before, so it is recommended to draw the fill first and then draw the border when selecting fill)
Controller	Controllers that support RTHMI. (4XX series controllers and above)
Example	For Example: SET_COLOR(RGB(0,0,0),RGB(0,255,0)) DRAWEX_ROTRECT(150,150,200,100,45*PI/180, 1) 'draw the rectangle with the radian of 45 degrees anticlockwise. DRAWEX_ROTRECT(150,150,200,100,45*PI/180, 0) 'draw the rectangle framework with the radian of 45 degrees anticlockwise. 
Instructions	DRAWEX_RECT, SETEX_LINE, SET_COLOR

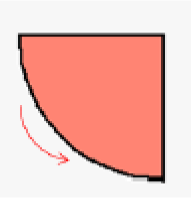
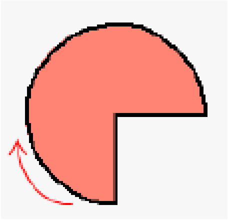
6.3.28. DRAWEX_ELLIPSE – Draw Ellipse Rectangle (with format)

Type	Drawing instructions
Description	Draw an ellipse in the specified area. This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_ELLIPSE(centx, centy, radiusx, radiusy [, iffill]) centx, centy: the position of the center of the circle radiusx: the radius in the X direction radiusy: the radius in the Y direction iffill: whether to fill, default is 0 (not to fill) 0 – not to fill, non 0 – fill (note: the one drawn later may block the one drawn before, so it is recommended to draw the fill first and then draw the border when selecting fill)
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SETEX_LINE(2,0) ‘set a solid line with a line width of 2 SET_COLOR(RGB(0,0,0),RGB(0,255,0)) ‘set the border color of the ellipse to black and fill it with green DRAWEX_ELLIPSE(100,150,80,50,1) 'draw a fillable ellipse DRAWEX_ELLIPSE(100,150,80,50,0) 'draw an ellipse border 
Instructions	DRAWEX_RECT, DRAWEX_SECTOR, SETEX_LINE, SET_COLOR

6.3.29. DRAWEX_SECTOR – Draw Sector (with format)

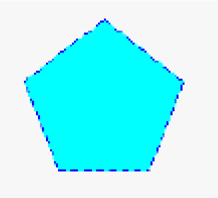
Type	Drawing instructions

Description	Draw a sector in the specified area. This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
Grammar	DRAWEX_SECTOR(centx, centy, radiusx, radiusy, startangle, endangle [, iffill]) centx, centy: the position of the center of the circle radiusx, radiusy: the radius size startangle: start angle, in radians endangle: end angle iffill: whether to fill, default is 0 (not to fill) 0 – not to fill, non 0 – fill (note: the one drawn later may block the one drawn before, so it is recommended to draw the fill first and then draw the border when selecting fill) Angle description of drawing:  Draw a sector from the start angle to the end angle: If start angle < end angle: draw a sector clockwise If start angle > end angle: draw a sector counterclockwise
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	**Example 1:** SETEX_LINE(2,0) 'set line width SET_COLOR(RGB(0,0,0),RGB(250,128,114)) 'set border color and fill color DRAWEX_SECTOR(200,150,100,100,PI,90*PI/180,1) 'draw sector border

	DRAWEX_SECTOR(200,150,100,100,PI,90*PI /180,0) 'draw a fillable sector counterclockwise  **Example 2: To draw a high-quality sector, the starting angle +PI*2** SETEX_LINE(2,0) 'set line width SET_COLOR(RGB(0,0,0),RGB(250,128,114)) 'set border color and fill color DRAWEX_SECTOR(200,150,50,50,PI/2+2*PI,4*PI,1) 'draw sector border DRAWEX_SECTOR (200,150,50,50,PI/2+2*PI,4*PI,0) ' ‘draw a fillable sector clockwise 
Instructions	DRAWEX_ELLIPSE, SETEX_LINE, SET_COLOR

6.3.30. DRAWEX_POLYGON – Draw Polygon (with format)

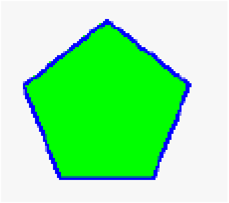
Type	Drawing instructions
Description	Draw a polygon in the specified area. This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWEX_POLYGON(points [, x1,y1,..,xn,yn] [, iffill]) points: the number of polygon points, which determines how many sets of xy data there are behind (the start and end coordinates of the closed figure need to be passed twice, and the total number of coordinates that need to be

	passed in = the number of polygon points + 1) xn,yn: the x and y coordinates of the nth point iffill: whether to fill, default is 0 (not to fill) 0 – not to fill, non 0 – fill (note: the one drawn later may block the one drawn before, so it is recommended to draw the fill first and then draw the border when selecting fill) Special Description If drawing a closed polygon, such as drawing a quadrilateral, you need to pass 5 points, and the start point = end point, otherwise it is an ordinary polyline segment (cannot be filled). If the drawn filled polygon is not displayed, please check whether it is a closed shape. The number of sides of a polygon is limited, and the current version supports up to 32 sides.
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SETEX_LINE(1,2) 'set the line type to dotted line SET_COLOR(RGB(0,0,255),RGB(0,255,255)) 'set border color and fill color DRAWEX_POLYGON(6,100,90,145,125,125,175,75,175,55,125,100,90,1) 'draw a fillable pentagon DRAWEX_POLYGON(6,100,90,145,125,125,175,75,175,55,125,100,90) 'draw a pentagonal border 
Instructions	DRAWRECT, DRAWEX_RECT, SETEX_LINE, SET_COLOR

6.3.31. DRAWEX_POLYGON2 – Draw Polygon (save in TABLE)

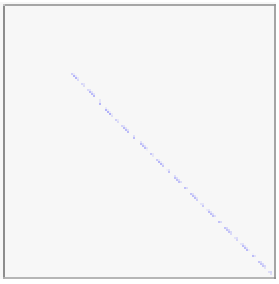
Type	Drawing instructions
Description	Draw a polygon in the specified area.

	This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWEX_POLYGON2(points, tableindex [, iffill]) points: the number of polygon points, which determines how many sets of xy data there are behind (the start and end coordinates of the closed figure need to be passed twice, and the total number of coordinates that need to be passed in = the number of polygon points + 1) tableindex: the table subscript to store the starting point data (x1,y1,..xn,yn) iffill: whether to fill, default is 0 (not to fill) 0 – not to fill, non 0 – fill (note: the one drawn later may block the one drawn before, so it is recommended to draw the fill first and then draw the border when selecting fill) Special Description If drawing a closed polygon, such as drawing a quadrilateral, you need to pass 5 points, and the start point = end point, otherwise it is an ordinary polyline segment (cannot be filled). If the drawn filled polygon is not displayed, please check whether it is a closed shape. The number of sides of a polygon is limited, and the current version supports up to 32 sides.
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SETEX_LINE(2,0) 'set line width to 2, solid line type SET_COLOR(RGB(0,0,255),RGB(0,255,0)) 'set border color and fill color TABLE(0)=100 ‘x1 TABLE(1)=90 ‘y1 TABLE(2)=55 ‘x2 TABLE(3)=120 ‘y2 TABLE(4)=75 ‘x3 TABLE(5)=175 ‘y3

	TABLE(6)=125 ‘x4 TABLE(7)=175 ‘y4 TABLE(8)=145 ‘x5 TABLE(9)=125 ‘y5 TABLE(10)=100 ‘x6 TABLE(11)=100 ‘y6 DRAWEX_POLYGON2(6,0,1) 'draw a fillable pentagon DRAWEX_POLYGON2(6,0) 'draw a pentagonal border 
Instructions	DRAWRECT, DRAWEX_RECT, SETEX_LINE, SET_COLOR

6.3.32. SETEX_LINE – Set Segment Property

Type	Display instructions
Description	Drawing line width setting (this command is used together with DRAWEX commands) All DRAWEX commands support line attributes. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	SET_LINE (width [, type]) width: line segment width (range: 1-20) type: line segment type (only solid line when line width>1) 0 - solid line (default) 1 - Long dashed line DASH: —————— 2 - Dotted line DOT: ------ 3 – DASHDOT: –·–·–·–·–·–· 4 – DASHDOTDOT: ··–··–··–··–··
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SETEX_LINE(1,3) 'set the line width to 1 and the line type to DASHDOT SET_COLOR(RGB(0,0,255)) DRAWEX_LINE(100,100,400,400)

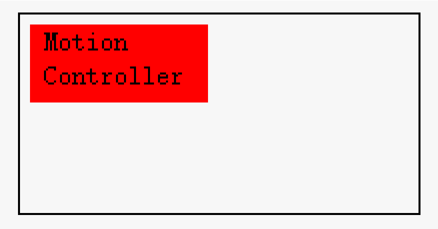
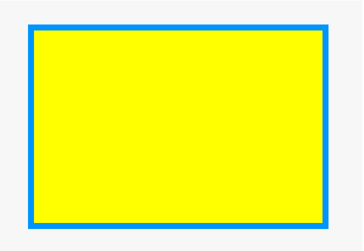
	
Instructions	SET_COLOR

6.3.33. SET_FONT – Set Font

Type	Display instructions
Description	Font setting, the built-in 16*16 Chinese font is used by default, and the English font is 16*8. But when the size is inconsistent with the size of the font library, scaling will occur, if you need your own font, please use zfontmaker to make a special font file.
Grammar	SET_FONT(width, height, [fontname]) width: font width, English is automatically halved height: font height fontname: the font file name used, if not set, the current font will not be modified
Controller	Controllers that support ZHMI.
Example	SET_FONT(16,16, "16X16 font file small five.zft")
Instructions	SET_COLOR

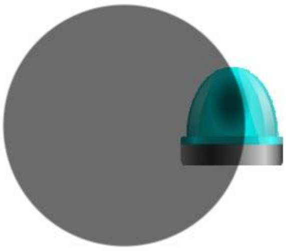
6.3.34. SET_COLOR – Set Color

Type	Display instructions
Description	Specify the color used by the draw command, if the color is not set, the default is black. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	SET_COLOR (cor[,backcor]) cor: line segment color, the default is black. backcor: the background color of DRAWTEXT, it is transparent (-1) when it is not filled
Controller	Controllers that support ZHMI.
Example	**Example 1:**

	SET_COLOR(RGB(255,255,255)) ‘set as white **Example 2:** SET_COLOR(RGB(0,0,0),RGB(255,0,0)) ‘the color is black, and the background of the string displayed by DRAWTEXT is red. DRAWRECT(0,0,200,100) ‘draw the edge in “Custom” object DRAWTEXT(10,10, "motion controller") ‘show character string in “Custom” object  **Example 3:** SET_COLOR(RGB(0,150,255),RGB(255,255,0)) 'blue line, yellow fill background DRAWEX_RECT(50,50,200,150,0,1) ‘draw a filled rectangle inside a “custom” object DRAWEX_RECT(50,50,200,150,0) ‘draw a rectangular border inside a “custom” object 
Instructions	RGB

6.3.35. SETEX_ALPHA – Set Drawing Transparency

Type	Display instructions
Description	Set the draw transparency. For RTHMI_V1.3.0, this command is valid for all DRAWEX drawing commands, scrolling commands and draw commands (DRAWPIC,

	DRAWLIBPIC).
Grammar	SETEX_ALPHA (alpha) Alpha: the transparency value, 0 – not transparent, 100 - full transparent.
Controller	Controllers that support RTHMI.
Example	SETEX_ALPHA (50) 'set the transparency as 50 DRAWEX_ELLIPSE(150, 150, 100, 100, 100, 1) 

6.3.36. SET_REDRAW – Redraw

Type	Display instructions
Description	Set the specified area to be redrawn, used in the "refresh sub" of the custom object This command can only be used in the "refresh sub" of the custom object, please refer to the "custom" object.
Grammar	SET_REDRAW ([x, y, width, height]) X, Y: the coordinate position of the upper left corner of the display area width, height: the width and height of the area Draw the entire region with no arguments.
Controller	Controllers that support ZHMI.
Example	SET_REDRAW 'redraw all regions

6.3.37. RGB – Color Property

Type	Display instructions
Description	Generate one color.
Grammar	COR = RGB(R,G,B) RGB: represents the color of the three channels of red, green, and blue R, G, B: the color of each object 0-255
Controller	Controllers that support ZHMI.
Example	RGB(255,255,0) 'pure yellow
Instructions	SET_COLOR

6.3.38. HMI_LANG – Switch Text Library Lanaguage

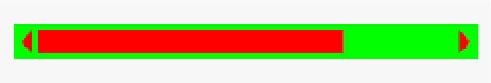
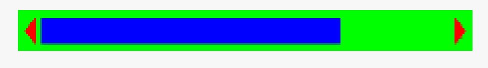
Type	Display instructions
Description	Select the language version, the text library can automatically switch languages, see text library routines
Grammar	HMI_LANG = ilang language number 0-7
Controller	Controllers that support ZHMI.
Example	HMI_LANG=0 ‘Select the content of text library language number 0
Instructions	/

6.3.39. SCROLLBAR_FREE – Release Scroll Bar

Type	Display instructions
Description	Release the scroll bar. Delete and release the created scroll bar, and the released scroll bar ID will be invalid in calling the refresh sub and drawing sub in the custom control.
Grammar	SCROLLBAR_FREE (id) id: the unique number of the scroll bar
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	/

6.3.40. SCROLLBAR_INIT – Scroll Initial Bar

Type	Display instructions
Description	Initializes a scrollbar with the specified ID. For specific usage, please refer to the scroll bar usage routine.
Grammar	SCROLLBAR_INIT(id,x1,y1,width,height,maxlines,showlines[forecor,backcor ,presscor]) id = SCORLLBAR_INIT (x1, y1, width, height, maxlines, showlines, [forecor,backcor,presscor]) id: the unique number of the scroll bar, if the id parameter is not passed, the available ID will be automatically obtained and returned As the return value, return -1 to indicate that there is no free ID, and the automatic initialization fails width, height: scroll bar width and height maxlines: total number of lines showlines: display the number of lines

	forecor: the color of the scroll bar, which can be defaulted backcor: scroll bar background color, can be default presscor: scroll bar press color, can be default Special Description: If width > height, it is a horizontal scroll bar, otherwise it is a vertical scroll bar.
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SCROLLBAR_INIT(0, 0, 0, 21, 330, maxlines, showlines, RGB(255,0,0), RGB(0,255,0), RGB(0,0,255)) Scroll bar display effect:  Show effect when scroll bar is pressed: 
Instructions	SCROLLBAR_POS, SCROLLBAR_REFLASH, SCROLLBAR_DRAW

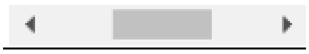
6.3.41. SCROLLBAR_POS – Get / Set Scroll Value

Type	Refresh instructions
Description	Get the set scroll value Handle the scroll bar refresh event, control the scroll bar to drag up and down, scroll. This command can only be used in the refresh function of custom object. For specific usage, please refer to the scroll bar usage routine.
Grammar	Auto refresh scrollbar event value = SCROLLBAR_POS(id) id: the unique number of the scroll bar value: return the current scroll value, return -1 means no refresh 'set scroll value SCROLLBAR_POS (id, value) id: the unique number of the scroll bar value: set the current scroll value compulsively
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	Refresh suhb value = SCROLLBAR_POS(0) ? "current scroll value: "value

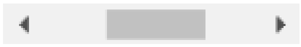
Instructions	SCROLLBAR_INIT, SCROLLBAR_REFLASH, SCROLLBAR_DRAW

6.3.42. SCROLLBAR_REFLASH – Refresh Scroll Bar

Type	Refresh instructions
Description	Refresh scroll bar. Handle the scroll bar refresh event, control the scroll bar to drag up and down, scroll. This command can only be used in the refresh function of custom object. For specific usage, please refer to the scroll bar usage routine.
Grammar	‘automatically refresh the scrollbar event, automatically trigger redrawing SCROLLBAR_RELASH (id) id: the unique number of the scroll bar ‘When the scroll bar event is triggered, the user can judge whether to redraw according to the return value bifredraw = SCROLLBAR_RELASH (id) id: the unique number of the scroll bar bifredraw: whether to redraw, 0 or 1
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	**Example 1:** 'refresh sub SCROLLBAR_RELASH (0) ‘mouse events will automatically trigger SET_REDRAW 'draw sub SCROLLBAR_DRAW (0) **Example 2:** ‘refresh sub (need to be called by the refresh sub of the “custom” object) global sub sub_reflash() bifredraw = SCROLLBAR_RELASH (0) 'manually judge whether to redraw according to the return value if bifredraw > 0 then ? "current scroll value:" SCROLLBAR_POS (0)

	SET_REDRAW endif end sub ‘draw sub (need to be called through the draw sub of the “custom” object) global sub sub_redraw() SCROLLBAR_DRAW (0) end sub 
Instructions	SCROLLBAR_INIT, SCROLLBAR_POS, SCROLLBAR_DRAW

6.3.43. SCROLLBAR_DRAW – Draw Scroll Bar

Type	Drawing instruction.
Description	Scroll bar drawing instructions. This command can only be used in the draw sub of the custom object. For specific usage, please refer to the scroll bar usage routine. The display of the scroll bar is not limited to the custom control, please calculate the spatial position before use to avoid abnormal use.
Grammar	SCROLLBAR_DRAW (id) id: the unique number of the scroll bar
Controller	Controllers that support ZHMI. (4XX series controllers and above)
Example	SCROLLBAR_DRAW (0) 
Instructions	SCROLLBAR_INIT, SCROLLBAR_POS, SCROLLBAR_REFLASH

6.3.44. DRAWEX_QRCODE – Draw QR-Code

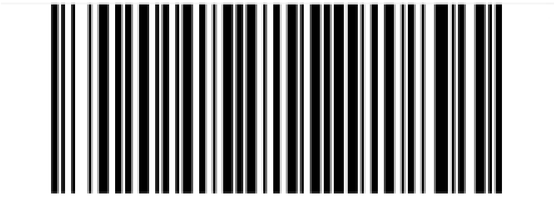
Type	Drawing instruction.
Description	Draw one QR-code in assigned area. This only can be used in “custom” component’s draw sub.
Grammar	DrawEx_QRCode (x, y, w, h, strCode, adjust) x,y: the left upper coordinates of the displayed area w,h: the width and height of the displayed area strCode: the character string that is to show corresponding QR-code. adjust: the correction standard.

	= 0, L (7%): 7% words & codes can be corrected. = 1, M (15%): 15% words & codes can be corrected. = 2, Q (25%): 25% words & codes can be corrected. = 3, H (30%): 30% words & codes can be corrected.
Controller	Controllers that support RTHMI

Example	SET_COLOR(RGB(0,0,0))
	'draw the QR-Code
	DrawEx_QRCode(10, 10, 100, 100, "123456 正运动", 3)

6.3.45. DRAWEX_BARCODE – Draw Bar-Code

Type	Drawing instruction.
Description	Draw one bar-code in assigned area. This only can be used in "custom" component's draw sub.
Grammar	DrawEx_BarCode (x, y, w, h, strCode [,mode]) x,y: the left upper coordinates of the displayed area w,h: the width and height of the displayed area strCode: the character string that is to show corresponding bar-code. mode: encode mode, default is 0 = 0, Code128 encode Others are reserved.
Controller	Controllers that support RTHMI

Example	SET_COLOR(RGB(0,0,0))
	'draw the bar-Code
	DrawEx_BarCode(10, 10, 100, 300, "123456789")
	

6.3.46. DRAWEX_TABVECT – Draw from TABLE (Draw in Bulk)

Type	Drawing instruction.
Description	Draw a table graphic in the specified area, and the graphic data is stored in the specified table position according to a certain format. This only can be used in "custom" component's draw sub.
Grammar	DrawEx_TabVect(tablein, offx, offy, scalex, scaley) tablein: TABLE address offx, offy: show graphic offset (>0: offset in left / up; <0: offset in right / bottom; =0: not offset) scalex: display the scaling ratio in X; 1.0 = no scaling. scaley: display the scaling ratio in Y; 1.0 = no scaling.
Controller	Controllers that support RTHMI

Example	'encapsulate the table's graphical data, and fill needed graphic into TABLE
	"local iTableStart iTableCur" according to assigned format.
	iTableStart = 20000 'TABLE starting subscript
	' the total size, enter it at last
	iTableCur = iTableStart + 1
	
	'the first item, draw one filling rectangle "rect = (20,20,200,200)", R=10, there are 5 parameters, which occupy 2+5 size
	table(iTableCur, 354, 5, 20, 20, 200, 200, 10)
	iTableCur = iTableCur + 7
	
	'the second item, draw one rectangle "rect = (20,20,200,200)", R=10, there are 5 parameters, which occupy 2+5 size
	table(iTableCur, 304,5, 20, 20, 200, 200, 10)
	iTableCur = iTableCur + 7
	
	'the third item, draw one rotary rectangle "rect = (220,220,200,200)", the rotary radian = 45*PI/180, there are 5 parameters, which occupy 2+5 size
	table(iTableCur, 305,5, 220, 220, 200, 200, 45*PI/180)
	iTableCur = iTableCur + 7

	'at last, enter the total size table(iTableStart) = iTableCur-iTableStart 'draw, display the graphs stored in the table. Display the graphs stored in the table.

6.3.47. TABLE Graphic Format

➢ **Data Format Reference:**

tabid	**tabid + 1**	**tabid + 2**	**tabid + 3 ~ tabid + 3 + N**
TableSize	DataType	ParaSize	Para1~ParaN
…	**…**	**…**	**…**
…	DataTye	ParaSize	Para1~ParaN

- The first TABLE value is TableSize, that is, the TABLE graphic data storage size ("TableSize" is included).
- It starts to save the graphic data from the second TABLE value, and each graphic data format is [data type (DataType) + parameter numbers (ParaSize) + all parameters' values (Para1~ParaN)]. It can infer the address of the next motion data according to current graphic data, until all is read (check from Tablesize, also, type 0 can be used as end flag). For example: TableSize = 10

10	100	3	para1	para2	para3	101	2	para1	para2

"Note: the number of parameters for each segment of graphical data can be adjusted freely by the user (must be >= to the minimum number of parameters)."

➢ **Data Type Descriptions:**

DataType	**Tyep Description**	**Parameter Description**
0	End Mark	/
100~200 = continuous drawings: last drawing's end point = next drawing's starting point.		
100	Set the brush position	✓ ParaSize: the number of parameters N (>=2) ✓ Para1: the jumping position x ✓ Para2: the jumping position y ■ note: you can adjust as any parameter's numbers N (but it needs >= the min parameters numbers), redundant parameters are not processed, and

		below is the same.
101	Draw one line (the current brush position is the starting point)	✓ ParaSize: the number of parameters N (>=2) ✓ Para1: the jumping position x ✓ Para2: the jumping position y ✚ Draw one straight line from current brush position to position (Para1, Para2).
102	Draw one arc (the current brush position is the starting point)	✓ ParaSize: the number of parameters N (>=5) ✓ Para1, Para2: the arc end point coordinates ✓ Para3, Para4: the arc circle center coordinates ✓ Para5: 0 – anticlockwise, 1 – clockwise ✚ Draw one arc from current brush position to position (Para1, Para2), the circle center is (Para3, Para4), the arc direction is para5.
103	Draw one ellipse arc (the current brush position is the starting point)	✓ ParaSize: the number of parameters N (>=7) ✓ Para1, Para2: the ellipse end point coordinates ✓ Para3, Para4: the ellipse circle center coordinates ✓ Para5: 0 – anticlockwise, 1 – clockwise ✓ Para6, Para7: the ellipse radius of axis x and y ✚ Draw one ellipse arc from current brush position to position (Para1, Para2), the circle center is (Para3, Para4), the ellipse arc direction is Para5, and the ellipse radius is (Para6, Para7).
111	End point (the current brush position)	✓ ParaSize: the number of parameters N (>=0) ✚ Draw one point from current brush position.
301~400 = full graphic drawing (not continuous drawing)		
301	Draw one arc	✓ ParaSize: the number of parameters N (>=6) ✓ Para1, Para2: circle center coordinates ✓ Para3, Para4: the radius in x & y direction ✓ Para5: starting angle, the unit: radian ✓ Para6: end angle, the unit: radian
302	Draw one Bessel curve	✓ ParaSize: the number of parameters N (>=8) ✓ Para1, Para2: the 1st control point x1, y1 ✓ Para3, Para4: the 2nd control point x2, y2 ✓ Para5, Para6: the 3rd control point x3, y3

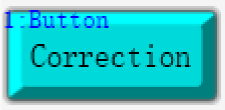
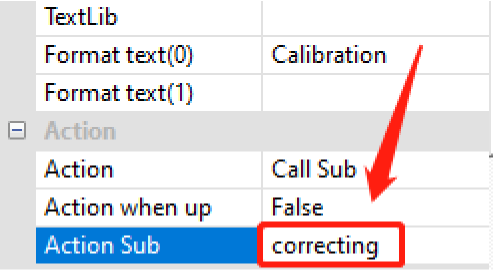
		✓ Para7, Para8: the 4th control point x4, y4
303	Draw one B spline curve	✓ ParaSize: the number of parameters N (>=8) ✓ Para1, Para2: the 1st control point x1, y1 ✓ Para3, Para4: the 2nd control point x2, y2 ✓ Para5, Para6: the 3rd control point x3, y3 ✓ Para7, Para8: the 4th control point x4, y4
304	Draw one fillet rectangle	✓ ParaSize: the number of parameters N (>=5) ✓ Para1, Para2: rectangle left upper position (x1, y1) ✓ Para3, Para4: rectangle left upper position (x2, y2) ✓ Para5: fillet radius
354 (filled type + 50)	Draw filled rounded rectangle	
305	Draw one rotary rectangle	✓ ParaSize: the number of parameters N (>=5) ✓ Para1, Para2: rotary rectangle center coordinates (cx, cy) ✓ Para3, Para4: rotary rectangle width and height (w, h) ✓ Para5: rotary angle, the unit: in radian, the direction is anticlockwise.
355 (filled type + 50)	Draw one filled rotary rectangle	
306	Draw one ellipse	✓ ParaSize: the number of parameters N (>=4) ✓ Para1, Para2: circle center coordinates x, y ✓ Para3, Para4: the radius in x & y direction
356 (filled type + 50)	Draw one filled ellipse	
307	Draw one sector	✓ ParaSize: the number of parameters N (>=6) ✓ Para1, Para2: circle center coordinates x, y ✓ Para3, Para4: the radius in x & y direction ✓ Para5: the starting angle, the unit: in radian ✓ Para6: the end angle, the unit: in radian
357 (filled type + 50)	Draw one filled sector	
308	Draw polylines / polygons	✓ ParaSize: the number of parameters N (>=2, N is even number) ✓ Para1, Para2: 1st point (x1, y1) ✓ Para3, Para4: 2nd point (x2, y2) ✓ ParaM-1, ParaM: the M / second point (x2, y2) ✓ Max 32 points
358 (filled type + 50)	Draw filled polygons	
>1000 = some properties configurations		
1000	Set relative / absolute mode	✓ ParaSize: the number of parameters N (>=1) ✓ Para1: 0 – absolute mode, 1 – relative mode (default is 0)

		✚ The drawing position coordinates will be calculated in a relative/absolute manner. The relative mode uses the current pen position as the reference coordinate.
1101	Modify the drawing’s color	✓ ParaSize: the number of parameters N (>=1~2) ✓ Para1: the brush color ✓ Para2: the brush filled color ✚ This is related to SET_COLOR.
1102	Modify segment’s property	✓ ParaSize: the number of parameters N (>=1~2) ✓ Para1: brush segment’s width (the brush size) ✓ Para2: brush segment’s type 0 - Solid line 1 - Long dashed line DASH: —————— 2 - Dashed line DOT: ------ 3 - DASHDOT: –·–·–·–·–·–· 4 - DASHDOTDOT: ··–··–··–··–··–·· ✚ This is related to SETEX__LINE.

6.4. HMI Touch Screen Instructions

6.4.1. TOUCH_ADJUST – Touch Screen Correction

Type	Touch Screen Instructions
Description	Perform touch screen calibration, do not refresh the screen at this time, the parameters will be automatically saved after calibration. Note: This command will only take effect when connected to the ZHD400X teaching box, 500X teaching box or the touch screen of a handheld box device that supports this command! **【Calibration method】** Method 1: Enter the touch adjustment interface by clicking the Touch Adjust function on the setting interface (continuously click on the upper left, upper right, lower left, lower right, upper left, upper right, lower left, and lower right

	to pop up the setting window and perform touch correction.) Method 2: After connecting the controller, trigger the calibration through the TOUCH_ADJUST command of the controller. Method 3: Without connecting the controller, press the 12 (F2) button without releasing it, and then press the 11 (F1) button. Without connecting the controller, press the 16 (F6) button, and continue to press the 11 (F1) button without releasing it
Grammar	TOUCH_ADJUST ()
Controller	Controllers that support ZHMI.
Example	**Example 1:** (1) Use one hmi button control  (2) In "Property", call sub function.  TextLib Format text(0) Calibration Format text(1) ⊟ Action Action Call Sub Action when up False Action Sub correcting (3) Basic sub function global sub correcting() TOUCH_ADJUST() end sub (4) To run the program item containing this instruction, click this button (5) Directly enter the touch calibration interface, and then press and hold the position of "+" to perform calibration. **Example 2: put it directly in the program initialization function** TOUCH_ADJUST () 'touch correcting
Instructions	/

6.4.2. TOUCH_SCAN – Scan Touching Action

Type	Touch Screen Instructions
Description	Scan the action of touch pressed, return 1 means scan to press, 1 means release, 0 means no change, save the X, Y coordinates of the corresponding position in the table. The touch screen is reserved to simulate the right and middle button functions of the mouse.
Grammar	action = TOUCH_SCAN (num, [,optkey]) num: table No. that stores the XY coordinates of the touch, and the X and Y coordinates are stored in table(num), table(num+1) respectively optkey: operate mouse key, optional, default 0 – mouse left key 0 – mouse left key 1 – mouse right key 2 – mouse middle key action: return mouse action, 1 – pressed, -1 – released, 0 – no change.
Controller	Controllers that support RTHMI.
Example	**Example 1:** IF TOUCH_SCAN(0) = 1 THEN ‘scan the operation of scanning, showing pressing position ?”pressed position is:” TABLE(0), TABLE(1) ENDIF IF TOUCH_SCAN(0) = -1 THEN ‘scan the operation of releasing ?”released” ENDIF **Example 2:** IF TOUCH_SCAN(0) = 1 THEN ‘scan the operation of scanning TIMES = TIME ENDIF IF TOUCH_SCAN(0) = -1 THEN ‘scan the operation of releasing ?”pressed time is:” TIME-TIMES ENDIF
Instructions	MOUSE_SCAN

6.4.3. TOUCH_STATE – Get Touching State

Type	Touch Screen Instructions
Description	Read touching state, >0 means pressed, 0 means released. Save touching related position X, Y in the table. The touch screen is reserved to simulate the right and middle button functions of the mouse.
Grammar	action = TOUCH_STATE (num, [,optkey]) num: table No. that stores the XY coordinates of the touch, and the X and Y coordinates are stored in table(num), table(num+1) respectively optkey: operate mouse key, optional, default 0 – mouse left key 0 – mouse left key 1 – mouse right key 2 – mouse middle key state: return mouse action, 1 – pressed, 0 – released
Controller	Controllers that support RTHMI.

Example	WHILE1
	IF SCAN_EVENT(TOUCH_STATE(0))>0 THEN
	?"pressed position is "TABLE(0),TABLE(1)
	ENDIF
	IF SCAN_EVENT(TOUCH_STATE(0))<0 THEN
	?”released”
	ENDIF
	WEND

Instructions	MOUSE_STATE

6.4.4. MOUSE_SCAN – Scan Mouse Action

Type	Touch Screen Instructions
Description	Scan the action of mouse pressed, return 1 means scan to press, 1 means release, 0 means no change, save the X, Y coordinates of the corresponding position in the table. The touch screen is reserved to simulate the right and middle button functions of the mouse.
Grammar	action = MOUSE_SCAN (num, [,optkey]) num: table No. that stores the XY coordinates of the touch, and the X and Y coordinates are stored in table(num), table(num+1) respectively

	optkey: operate mouse key, optional, default 0 – mouse left key 0 – mouse left key 1 – mouse right key 2 – mouse middle key action: return mouse action, 1 – pressed, -1 – released, 0 – no change.
Controller	Controllers that support RTHMI.
Example	**Example 1:** IF MOUSE_SCAN(0) = 1 THEN ‘scan the operation of scanning, showing pressing position ?”pressed position is:” TABLE(0), TABLE(1) ENDIF IF MOUSE_SCAN(0) = -1 THEN ‘scan the operation of releasing ?”released” ENDIF
Instructions	TOUCH_SCAN

6.4.5. MOUSE_STATE – Get Mouse State

Type	Touch Screen Instructions
Description	Read mouse state, >0 means pressed, 0 means released. Save touching related position X, Y in the table. The touch screen is reserved to simulate the right and middle button functions of the mouse.
Grammar	action = MOUSE_STATE (num, [,optkey]) num: table No. that stores the XY coordinates of the touch, and the X and Y coordinates are stored in table(num), table(num+1) respectively optkey: operate mouse key, optional, default 0 – mouse left key 0 – mouse left key 1 – mouse right key 2 – mouse middle key state: return mouse action, 1 – pressed, 0 – released
Controller	Controllers that support RTHMI.

Example	WHILE1
	IF SCAN_EVENT(MOUSE_STATE(0))>0 THEN
	?"pressed position is "TABLE(0),TABLE(1)
	ENDIF

	IF SCAN_EVENT(MOUSE_STATE(0))<0 THEN
	?"released"
	ENDIF
	WEND

Instructions	TOUCH_STATE

6.5. HMI Key Button Commands

6.5.1. MOUSE_WHEEL – Get Mouse Scroll Value

Type	Virtual key button instructions
Description	Get the value of the mouse wheel, this command can only be used in the refresh sub of the "custom" object.
Grammar	wheel = MOUSE_WHEEL (tabid) tabid: the X and Y coordinates of the mouse position are stored in table(tabid), table(tabid +1) wheel: return the value of the mouse wheel, -120 means scrolling down one space, 120 means scrolling up one space
Controller	Controllers that support ZHMI (it is valid in 5xx series or above and ZMC432 with the latest firmware version)

Example	wheelval = MOUSE_WHEEL(0)
	if wheelval <> 0 then
	wheelval = wheelval/120 'indicates the amount of scrolling, >0 scrolls up, <0 scrolls down
	endif

6.5.2. KEY_STATE – Get Physical Key State

Type	Virtual key button instructions
Description	Read the state of the physical button, 1-pressed. This command can only be used in the refresh sub of the "custom" object.
Grammar	KEY_STATE (keynum) keynum: key number
Controller	Controllers that support RTHMI.
Example	num =KEY_SCAN()

	IF KEY_STATE (num)=1 THEN 'scan the operation of pressing, and print key number ?"press" num ENDIF
Instructions	KEY_EVENT, KEY_SCAN

6.5.3. KEY_EVENT – Get Physical Key State Changes

Type	Virtual key button instructions
Description	Read the state change of the physical button, 1-pressed, -1-released, 0-unchanged. This command can only be used in the refresh sub of the "custom" object.
Grammar	KEY_EVENT (keynum) keynum: key number
Controller	Controllers that support RTHMI.

Example	num =KEY_SCAN()
	IF KEY_EVENT (num)=1 THEN 'scan the operation of pressing, and print key number
	?"press" num
	ENDIF

Instructions	KEY_STATE, KEY_SCAN

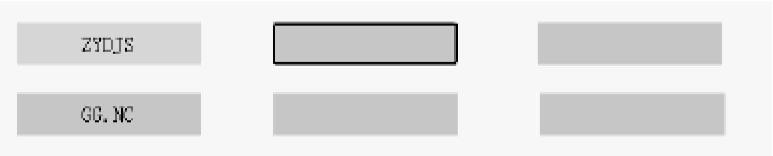
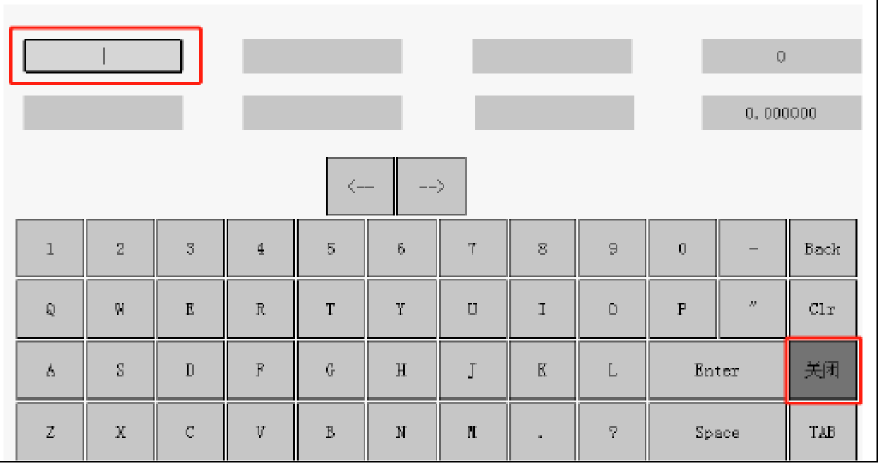
6.5.4. KEY_SCAN – Get Physical Key Encodes

Type	Virtual key button instructions
Description	Read the code of the currently pressed physical key, press it to return the key code, when it is released, it is the negative number of the key code, return 0 means there is no key state change. The code value of the physical key is determined by the hardware and cannot be modified in the program, but the virtual key bound to the physical key can be modified. This command can only be used in the refresh sub of the "custom" object.
Grammar	value = KEY_SCAN ()
Controller	Controllers that support RTHMI.
Example	num =KEY_SCAN()

	IF　KEY_EVENT (num)=1 THEN　　'scan the operation of pressing, and print key number
	　　?"press" num
	ENDIF

Instructions	KEY_STATE, KEY_SCAN

6.5.5. VKEY_MODE – ON Virtual Key IN Mode

Type	Virtual key button instructions
Description	Turn on/off the virtual key input mode. After the virtual key input mode is turned on, the "value" and "string" controls will get the focus (only one control gets the focus at the same time), and the control that gets the focus can directly receive the virtual key value, and get highlight prompt (the second control in the first line as follows) 
Grammar	VKEY_MODE (mode[, winid]) mode = VKEY_MODE() 　　mode: 0 - close virtual key input mode 　　　　1 - enable virtual key input mode 　　winid: specify the window number, the default is the current (top-level) window
Controller	Controllers that support RTHMI.
Example	VKEY_MODE(1)　　'open virtual key input mode VKEY_MODE(1,10) 　　　'specify window 10 to open virtual key input mode  0 0.000000 <-- --> 1 2 3 4 5 6 7 8 9 0 - Back Q W E R T Y U I O P " Clr A S D F G H J K L Enter 关闭 Z X C V B N M . ? Space TAB

Instructions	KEY_STATE, KEY_SCAN

6.5.6. VKEY_STATE – Set / Get Virtual Key State

Type	Virtual key button instructions
Description	Read the status of the virtual button, 1-pressed, 0-released. This command can only be used in the refresh sub of the “custom” object.
Grammar	VKEY_STATE (keynum,state) state=VKEY_STATE(keynum) keynum: virtual key number state: virtual key state
Controller	Controllers that support RTHMI.
Example	**Example 1:** 'set the state of the virtual key, select the action "Call Sub Twice" through the function key, and specify the function to be called when it is pressed and when it is released global sub VKeyPress_A() VKEY_STATE (65, 1) ‘virtual key ‘A’ is pressed end sub global sub VKeyRelease_A() VKEY_STATE (65, 0) ‘virtual key ‘A’ is released end sub **Example 2:** ‘get virtual key state in “custom” control refresh sub num =VKEY_SCAN() IF VKEY_STATE (num)=1 THEN ‘scan the operation of pressing, and print key number ?"press" num ENDIF
Instructions	VKEY_SCAN, VKEY_EVENT

6.5.7. VKEY_EVENT – Get Virtual Key State Changes

Type	Virtual key button instructions
Description	Read the status of the virtual button, 1-pressed, -1-released, 0-no change.

	This command can only be used in the refresh sub of the “custom” object.
Grammar	VKEY_EVENT (keynum) keynum: virtual key number
Controller	Controllers that support RTHMI.

Example	num =VKEY_SCAN()
	IF VKEY_EVENT (num)=1 THEN
	‘scan the operation of pressing, and print key number
	?"press" num
	ENDIF

Instructions	VKEY_SCAN, VKEY_STATE

6.5.8. VKEY_SCAN – Get Virtual Key Encodes

Type	Virtual key button instructions
Description	Read the code of the currently pressed virtual key, press it to return the key code, when it is released, it is the negative number of the key code, return 0 means there is no key state change. The code value of the virtual key is determined by the hardware and cannot be modified in the program, but the virtual key bound to the physical key can be modified. This command can only be used in the refresh sub of the “custom” object.
Grammar	value = VKEY_SCAN ()
Controller	Controllers that support RTHMI.
Example	Dim Curkey Curkey = VKEY_SCAN() ‘read the current key value (message code)
Instructions	VKEY_STATE, VKEY_EVENT

6.5.9. VKEY_INPUT – Input Virtual Key Info to Keyboard Window

Type	Virtual key button instructions
Description	Control simulates virtual key input in keyboard window.
Grammar	VKEY_INPUT (vkeynum) vkey: virtual key number
Controller	Controllers that support RTHMI.

Example	/
Instructions	VKEY_STATE, VKEY_EVENT

6.5.10. VKEY_IME – Get / Set Current Input Method

Type	Virtual key button instructions
Description	Set / get current input method type. Chinese input is only valid for virtual keys VKEY_a ~ VKEY_z, and is invalid for uppercase characters.
Grammar	ime = VKRY_IME() VKEY_IME(ime) ime: current input method type 0: default value, English input method 1: Chinese input method
Controller	Controllers that support RTHMI.
Example	VKEY_IME(1) ?VKEY_IME() output 1

6.5.11. ZSIMU_KEY – Simulate Physical Key

Type	Virtual key button instructions
Description	Simulate physical key.
Grammar	ZSIMU_KEY(keycode, state) keycode: physical key code state: pressed state, 1-pressed
Controller	Controllers that support RTHMI.
Example	ZSIMU_KEY(3, 1) ‘simulation physical key 3 is pressed
Instructions	ZSIMU_VKEY

6.5.12. ZSIMU_VKEY – Simulate Virtual Key

Type	Virtual key button instructions
Description	Simulate physical key.
Grammar	ZSIMU_VKEY(keycode, state) keycode: physical key code state: pressed state, 1-pressed
Controller	Controllers that support RTHMI.

Example	ZSIMU_VKEY(4, 1)　　‘simulation virtual key 4 is pressed
Instructions	ZSIMU_KEY

6.6. HMI Operation Commands

6.6.1. HMI_SHOWWINDOW – Show Assigned Window

Type	Show operations.
Description	Display the specified window. The soft keyboard window should be called in the relevant functions of the editing window, otherwise it is impossible to determine which window object needs to be edited. When in doubt, the topmost window is selected by default.
Grammar	HMI_SHOWWINDOW(winid, [showmode],[modifycontrol]) winid: window number showmode: display mode ZPLC_WIN_TYPE_AUTO = 0, the window mode specified in the HMI file ZPLC_WIN_TYPE_TOP = 1 ZPLC_WIN_TYPE_BOTTOM = 2 ZPLC_WIN_TYPE_BASE = 4, there can only be one base window, once switched all sub-windows are closed. ZPLC_WIN_TYPE_KEYBOARD = 5, the soft keyboard must be pop up window ZPLC_WIN_TYPE_POP = 6, pop-up window ZPLC_WIN_TYPE_MENU = 7, menu window, close automatically Modifycontrol: ZPLY_WIN_TYPE_KEBOARD, when the soft keyboard window type pops up, it corresponds to the object ID of the window to be edited.
Controller	Controllers that support RTHMI
Example	**Example 1:** HMI_SHOWWINDOW(13,6)　‘pop up window 13 **Example 2:** HMI_SHOWWINDOW(8,5,1)　‘pop up soft keyboard window 8 and relate to current window 1 object
Instructions	

6.6.2. HMI_CLOSEWINDOW – Close Window

Type	Show operations.
Description	Close specified window.
Grammar	HMI_CLOSEWINDOW(winid) winid: window number, default 0 - the window where the object called by the current function is located, other numbers - the window number in the HMI configuration
Controller	Controllers that support RTHMI.
Example	HMI_CLOSEWINDOW() ‘close the current window
Instructions	HMI_SHOWWINDOW

6.6.3. HMI_BASEWINDOW – Switch Base Window

Type	Show operations.
Description	Switch base windows.
Grammar	HMI_BASEWINDOW(winid) winid: the window number in the HMI configuration
Controller	Controllers that support RTHMI.
Example	HMI_BASEWINDOW = 11 ‘switch to base window 11
Instructions	HMI_SHOWWINDOW

6.6.4. HMI_FOCUSWINDOW – Window Focus Mode

Type	Show operations.
Description	Set /get whether the HMI window automatically switches when the mouse clicks the focus (the clicked window will automatically be displayed on the top layer) Note: Even if the pop-up window is switched to the top layer, it is always below the top window.
Grammar	HMI_FOCUSWINDOW(mode) mode= HMI_FOCUSWINDOW() mode: focus window switching mode 0: default, not switch 1: bit0=1, pop window (POP) follows with the mouse to click “switch focus” 2: bit1=1, top window (TOP) follows with the mouse to click

	"switch focus" Note: when pop window uses this command to switch the focus, which can't cover top window.
Controller	Controllers that support RTHMI.

Example	HMI_FOCUSWINDOW(1)
	'set pop window to follow the mouse to automatically switch the focus
	HMI_FOCUSWINDOW(2)
	'set top window to follow the mouse to automatically switch the focus
	HMI_FOCUSWINDOW(1+2)
	'set top and pop windows to follow the mouse to automatically switch the focus

6.6.5. HMI_LASTWINDOW – Last Clicked Window

Type	Show operations.
Description	Get the last clicked window
Grammar	winid = HMI_LASTWINDOW()
Controller	Controllers that support RTHMI.
Example	?HMI_LASTWINDOW() 'print the last clicked window ID

6.6.6. HMI_DEFAULTATTR – Set / Get HMI Inner Default Property

Type	HMI command
Description	Set / get HMI inner default property, it is valid for whole HMI. Note: This setting will take effect only if it is set before HMI initialization.
Grammar	value= HMI_DEFAULTATTR (strAttr) HMI_DEFAULTATTR (strAttr, value) strAttr: assign the property of the control to be operated ● "SBR_W": scroll bar width ● "SBR_F": front background color of scroll bar ● "SBR_B": background color of scroll bar ● "SBR_P": pressing color of scroll bar ● "BTN_RLM": button release mode 0 – default 1 – cancel trigger when the mouse away the control ● "DRAW_M": drawing mode

	0 – default lattice drawing, 1 - antialiased drawing ● "MONO_M": control monopoly mode (the control is with pop window, like, drop-down list, menu bar, etc.) 0 – not monopoly 1 – monopoly key 2 – monopoly mouse 3 – monopoly key & mouse (default) ● "RGB_GRAY": gray color ● "MINADAPT": text adaptive minimum font size ● "DMEMSIZE": dynamic memory size (bytes), default 0 After setting, the background image of the element can be specified to be loaded from the flash directory, but you need to ensure that there is enough dynamic memory available. Note: before HMI program opened, it must preset SBR_W, SBR_F, SBR_B, SBR_P and RGB_GRAY parameters well, therefore, you can add one function that is used to open HMI at the end of Basic program. In this way, needed parameters are configured correctly, then open HMI to make sure it works normally. value: property value
Controller	Controllers that support RTHMI.
Example	HMI_DEFAULTATTR ("SBR_W", 25) 'set scroll bar default width as 25 HMI_DEFAULTATTR ("SBR_P", RGB(255,0,0)) 'set scroll bar default pressing color

6.6.7. HMI_DEALINFO – Get HMI Processed Info

Type	HMI command
Description	Get current HMI processed information.
Grammar	value = HMI_DEALINFO (strAttr [,tabid]) strAttr: assign which one HMI information to be obtained. "CURWIN": current window No. "CURCTRL": current control No. "TOPWIN": the window No. that shows at the top level "EDITWIN": current editing input window "EDITCTRL": current editing input control "EDITNEXT": current focus' next editing control "EDITPREV": current focus' previous editing control "ACTWINS": get all currently opened windows. tabid: TABLE address where outputs window No. "CLKWIN": last click on the window

	"CLKCTRL": last click on the component value: value Note: "EDITWIN", "EDITCTRL", "EDITNEXT", "EDITPREV" parameters take effect only under the virtual key mode.
Controller	Controllers that support RTHMI
Example	**Example 1:** gobal dim aaa aaa = HMI_DEALINFO ("CURWIN") 'print current window No. ?aaa **Example 2: add "allow edit" control in HMI window** VKEY_MODE(1) 'open virtual key input mode ?HMI_DEALINFO ("EDITCTRL") 'print current editing control No. Print result will be: current editing control No. **Example 3: get how many windows that are opened now** Winnum = HMI_DEALINFO("ACTWINS", 100) ?"current opened window qty="winnum for i=0 to winnum -1 ?table(100+i) 'print all opened window No. Next The printed result is: 1 10 This means that only one window is currently opened, which is window 10.

6.6.8. HMI_CONTROLSIZEX – Get Component Width

Type	Show operations.
Description	Get object width.
Grammar	value= HMI_CONTROLSIZEX ([winid, controlid]) winid: the window number in the HMI file controlid: object number, the default is the current "custom" object width
Controller	Controllers that support RTHMI.
Example	PRINT HMI_CONTROLSIZEX(10,11) 'print the width of object 11 in window 10

Instructions	HMI_CONTROLSIZEY

6.6.9. HMI_CONTROLSIZEY – Get Component Height

Type	Show operations.
Description	Get object height.
Grammar	value= HMI_CONTROLSIZEY ([winid, controlid]) winid: the window number in the HMI file controlid: object number, the default is the current "custom" object height.
Controller	Controllers that support RTHMI.
Example	PRINT HMI_CONTROLSIZEY(10,11) 'print the height of object 11 in window 10
Instructions	HMI_CONTROLSIZEX

6.6.10. HMI_CONTROLDATA – Set / Get Custom Component Property

Type	Show operations.
Description	Get or set the special properties of "custom" object, specified in the HMI, multiple similar components can be distinguished through this.
Grammar	value= HMI_CONTROLDATA ([winid, controlid]) HMI_CONTROLDATA (winid, controlid) = value winid: the window number in the HMI file controlid: object number, the default is the current "custom" object
Controller	Controllers that support RTHMI.
Example	HMI_CONTROLDATA(10,1)=5 'set several custom objects as the same property HMI_CONTROLDATA(10,2)=5
Instructions	HMI_CONTROLSIZEX, HMI_CONTROLSIZEY

6.6.11. HMI_CONTROLBACK – Set / Get Assigned Component Background Color

Type	Show operations.

Description	Get or set the background color of "value" and "string" objects.
Grammar	value= HMI_CONTROLBACK ([winid, controlid]) HMI_CONTROLBACK (winid, controlid) = value winid: the window number in the HMI file controlid: object number, the default is the current "custom" object
Controller	Controllers that support RTHMI.
Example	HMI_CONTROLBACK(10,1)=RGB(255,255,0) 'yellow HMI_CONTROLBACK(10,1)=RGB(255,0,0) 'red
Instructions	RGB

6.6.12. HMI_CONTROLVALID – Set / Get Component Enable

Type	Show operations.
Description	Get or set the enable of the object, which can be specified in HMI.
Grammar	value= HMI_CONTROLVALID ([winid, controlid]) HMI_CONTROLVALID (winid, controlid) = value winid: the window number in the HMI file controlid: object number, the default is the current "custom" object value: when it is 1, object touching is valid, when it is 0, no touching effect.
Controller	Controllers that support RTHMI.
Example	HMI_CONTROLVALID(10,5)=0 'the fifth object of window 10 is invalid
Instructions	/

6.6.13. HMI_CONTROLSTRING – Get String Info

Type	Show operations.
Description	Get the string information of the string object or input display control. 5xx series 20180405 and above firmware support.
Grammar	string= HMI_CONTROLSTRING ([winid, controlid]) Hmi_ControlString (winid, controlid) = string winid: the window number in the HMI file controlid: object number, the default is the current "custom" object string: character string
Controller	Controllers that support RTHMI.
Example	RSTRING(0,10)= HMI_CONTROLSTRING (10, 2)

	'data of the second object in window 10 is read into VR
Instructions	/

6.6.14. Component "Value" Property Commands

Commands	Descriptions	Permission
"POSX"	The component's X position	RW
"POSY"	The component's Y position	RW
"SIZEX"	The component's width	RW
"SIZEY"	The component's height	RW
"STATE"	The component's state	RO
"FORE"	The component's current display color (maybe segment color, font color).	RW
"BACK"	The component's current background color (maybe base color, text background color, filled background color).	RW
"MIN"	The minimal value	RW
"MAX"	The maximum value	RW
"PWD"	Show the password (value, string)	RW
"DECS"	The decimal part (value)	RW
"POPUP"	The pop-up state (pull-down list)	RW
"CURSORX"	The component's input cursor row	RW
"CURSORY"	The component's input cursor column	RW
"FOCUS"	The component's focus state: 0 – non-focus, 1 – focus	RW
"MSELROWS"	Select multiple row numbers. +: select multi-row from this row to below rows. -: select multi-row from this row to upper rows. 0: no multiple selection Used together with "CURSORY" to get the cursor row, and you could know current selected row area.	RW
"CHANNEL"	Channel No.	RW
"EDIT"	If it can edit and input the text?	RW
"CODEPAGE"	The component's text code page	RW
"EXTVAL"	Customized value numbers	RW
"ARRAYNUM"	Customized array space	RO
"ARRAYVAL"	Access customized array each item's value	RW

Note: for multi-state attributes, you can append %n after the component attribute to access attributes for different states. For example, "BACK%0" represents the background color for state 0.

6.6.15. Component "String" Property Commands

Commands	Descriptions	Permission
"TEXT"	The component's state text	RW
"BACKPIC"	Background picture's name	RO
"PICLIB"	Picture library name	RO
"STRLIB"	Text library name	RO
"EXTSTR"	Customized string numbers Note: when writing to this attribute using a command, the content to be written must be less than or equal to the number of characters in the original content.	RW

Note: for multi-state attributes, you can append %n after the component attribute to access attributes for different states. For example, "TEXT%2" represents the component text under state 2.

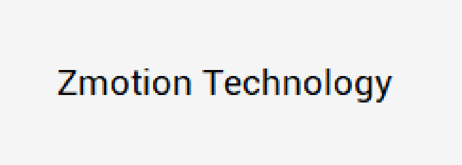
6.6.16. HMI_CONTROLATTR – Set / Get Component "Value" Property

Type	Show operations.
Description	Get/set the specified "value" property value. Note: before operating, make sure the operated control has the property.
Grammar	value=HMI_CONTROLATTR(strProperty [, winid, controlid]) HMI_CONTROLATTR(strProperty, value[, winid, controlid]) srtProperty: specify the control property to be operated, for details, please refer to 6.6.14. value: property value winid: the window No. in the HMI file, default is current window. controlid: object No., the default is the current "custom" object
Controller	Controllers that support RTHMI
Example	**Example 1:** HMI_CONTROLATTR("FORE",RGB(255,0,0),10,1) 'set the fore background color of window 10 first control as red

HMI_CONTROLATTR("BACK",RGB(255,255,255),10,1)

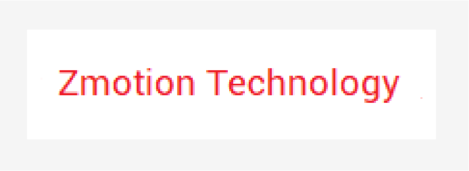
‘set the background color of window 10 first control as white

- The style before setting:



Zmotion Technology

- The style after setting:



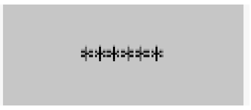
Zmotion Technology

Example 2:

VRSTRING(0,8) = "123456"

HMI_CONTROLATTR("PWD",123456,10,5)

'set the character of control 5 in window 10 as password display



******

Example 3:

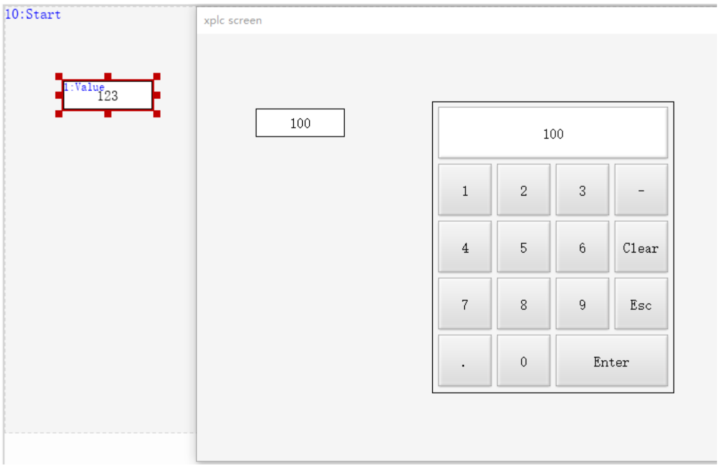
HMI_CONTROLATTR("MIN",10,10,6)

'set the minimum value of the value display control

HMI_CONTROLATTR("MAX",100,10,6)

'set the maximum value of the display control

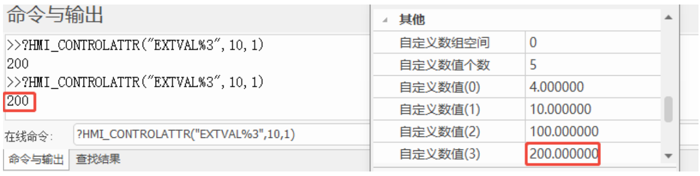
Note: The minimum value must be smaller than the maximum value to be valid, greater than or equal to all invalid. Also take into account the incoming values for Min and Max in the “Property”.



Example 4:

?HMI_CONTROLATTR("EXTVAL%3",10,1)

'read the data of “Basic Extend” customized value (3).



命令与输出

```
>>?HMI_CONTROLATTR("EXTVAL%3", 10, 1)
200
>>?HMI_CONTROLATTR("EXTVAL%3", 10, 1)
200
```

在线命令: ?HMI_CONTROLATTR("EXTVAL%3",10,1)

其他

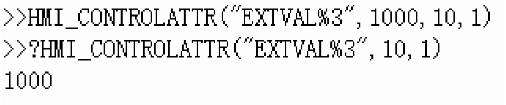
自定义数组空间	0
自定义数值个数	5
自定义数值(0)	4.000000
自定义数值(1)	10.000000
自定义数值(2)	100.000000
自定义数值(3)	200.000000

HMI_CONTROLATTR("EXTVAL%3",1000,10,1)

'set the data of “Basic Extend” customized value (3) as 1000.

?HMI_CONTROLATTR("EXTVAL%3",10,1)

'read the data of “Basic Extend” customized value (3) again.

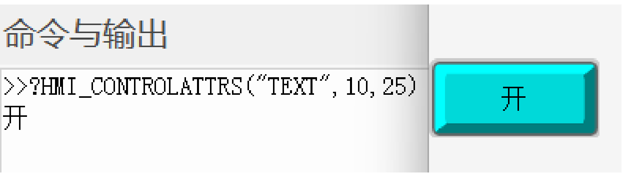
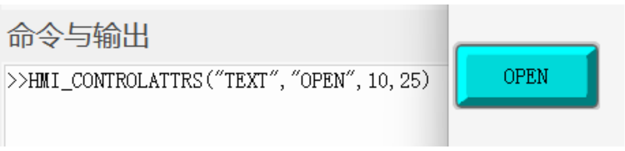


```
>>HMI_CONTROLATTR("EXTVAL%3", 1000, 10, 1)
>>?HMI_CONTROLATTR("EXTVAL%3", 10, 1)
1000
```

6.6.17. HMI_CONTROLATTRS – Set / Get Component “String” Property

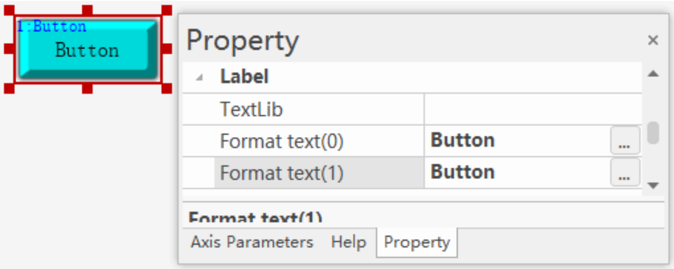
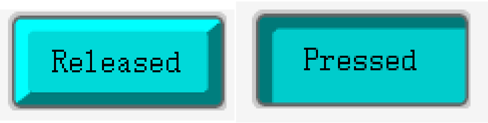
Type	Show operations.
Description	Get/set the specified “string” property value. Note: before operating, make sure the operated control has the property.

Grammar	string=HMI_CONTROLATTRS(strProperty [, winid, controlid])
	HMI_CONTROLATTRS(strProperty, string[, winid, controlid])
	srtProperty: specify the control property to be operated, for details, please refer to 6.6.15.
	string: the property’s string

	winid: the window No. in the HMI file, default is current window. controlid: object No., the default is the current object
Controller	Controllers that support RTHMI
Example	**Example 1:** ?HMI_CONTROLATTRS("TEXT",10,25) 'print the displayed text of component 25 that is in window 10  命令与输出 >>?HMI_CONTROLATTRS("TEXT",10,25) 开 **Example 2:** ?HMI_CONTROLATTRS("TEXT","OPEN",10,25) 'set the displayed text state of component 25 that is in window 10 as OPEN  命令与输出 >>HMI_CONTROLATTRS("TEXT","OPEN",10,25)

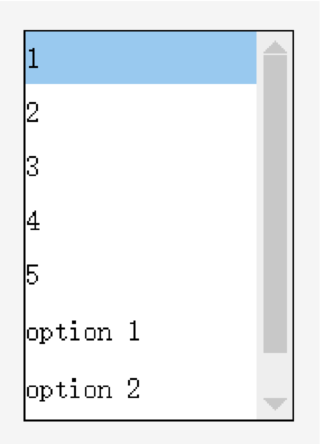
6.6.18. HMI_CONTROLTEXT – Get / Set Component Text

Type	Component operation command.
Description	Get / set the component's displayed text. It supports the component that is with all format's text (can be edited). Note: The length of the modified text cannot exceed the current control text, otherwise it will be truncated.
Grammar	strText = HMI_CONTROLTEXT (winid, controlid, state) HMI_CONTROLTEXT (winid, controlid, state, strText) winid: the window No. in the HMI file controlid: the component No. state: specify which state of the component text to modify. strText: modify the displayed text
Controller	Controllers that support RTHMI
Example	**Example 1:** HMI_CONTROLTEXT (10, 1, 0, "released") 'set text when button released HMI_CONTROLTEXT (10, 1, 1, "pressed") 'set text when button pressed --Before--

	 1:Button Button Property Axis Parameters Help Property Label TextLib Format text(0) Button Format text(1) Button Format text(1) --after-- Left: state 0, showing released. Right: state 1, showing pressed.  **Example 2:** Text=HMI_CONTROLTEXT (10, 1, 0) 'get the text when the button is in "released" state ? Text The print result will output "released".

6.6.19. HMI_LISTTEXTS – Modify Text of List Object

Type	Component operation command.
Description	Forcefully modify the text list of the "list" control. For specific application examples, see "Dynamic List". Note: To use this command, you must first enable the "Dynamic Item" property of the "List" control.
Grammar	HMI_LISTTEXTS (winid, controlid, strList, [,mode] [,lineheight])) winid: the window No. in the HMI file controlid: object No. strList: text list, separated by newlines mode: be optional, mode selection, default is 0 0 – cover method, clear original list item, write new list item again 1 – append method, add new list item on the basis of original list at the end, and the scroll bar automatically follows to the position after adding. Lineheight: be optional, force height of specified line, default -1: not assign. -1 – automatically calculate height, line height will change with list item numbers.

	=0 – use current line height >0 – force to assigned line height, when < min line height, set according to minimal line height. Min line height = word height + 2 * line space **Notes:** 1. The maximum number of characters in a string is 2049. 2. The list item modified by this instruction has a list number that increases line by line and cannot be specified. The same is true for the list item in the append operation. 3. In non-append operation mode, the scroll bar position of the list control will be automatically reset. In append operation mode, you can choose to keep the scroll bar in the append position. 4. The list control has a size limit for the dynamic list item buffer, with a maximum of 256 items. When the list item is filled with 256 items, the text of each list item must not exceed 12 characters.
Controller	Controllers that support RTHMI
Example	dim strList(100) = "option 1\n option 2\n option 3" HMI_LISTTEXTS (10, 3, strList, 1, 30) 'append new list item based on original list item, and assign the line height as 30 Display results:  1 2 3 4 5 option 1 option 2

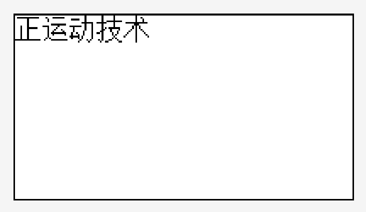
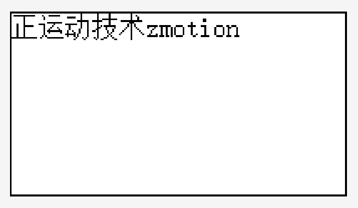
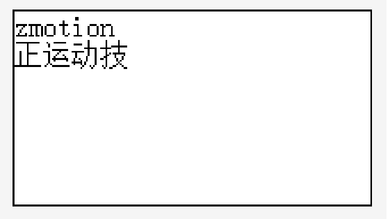
6.6.20. HMI_LISTITEM – Modify Assigned List Item Text

Type	Component operation command.
Description	Get / modify the assigned list item text of "list" control forcibly. Note: To use this command, you must first enable the "Dynamic Item" property of the "List" control.
Grammar	HMI_LISTITEM (winid, controlid, id, strText) strText = HMI_LISTITEM (wind, controlid, id)

	winid: the window No. in the HMI file controlid: object No. id: the list item ID to be modified strText: the list item text to be modified
Controller	Controllers that support RTHMI
Example	dim strList(100) = "option 1\n option 2\n option 3" Hmi_LISTTEXTS (10, 3, strList) ‘modify the list control text, automatically update the state numbers Hmi_LISTITEM (10, 3, 1, "TWO") ‘modify the second list item of component 3 as TWO

6.6.21. HMI_STRAPPEND – Object Text Append

Type	Component operation command.
Description	Append character string in the “string” component. Note: This command will only take effect when the component "Properties" is set to multi-line text status.
Grammar	HMI_STRAPPEND (strAppend, winid, controlid, mode [,ifNewLine]) strAppend: appended string winid: window No. in HMI file controlid: object No. mode: append mode (bit), the mode is determined by the combined value of bit0 and bit1. (When the string exceeds the maximum character limit, the overflow part of the total string after addition will be cleared according to the mode) bit0: overflow direction 0 - the overflow part is cleared from the end if the limit is exceeded, that is, the overflow part is not appended 1 - the overflow part is cleared from the beginning of the string bit1: append direction 0 - append from the end, the scroll bar automatically follows to the end of the text 1 - append from the beginning, the scroll bar automatically follows to the beginning of the text ifNewLine: whether to append a new line, the default is 0 (no) Note: appending a new line occupies one character space by default
Controller	Controllers that support RTHMI
Example	**Example 1: HMI_STRAPPEND ("zmotion",10, 2, 0, 0)** 'append a line of string to the end of the component text

	 正运动技术  正运动技术zmotion **Example 2: Limit the number of characters of the component text to 16, and append text to the component** HMI_STRAPPEND ("zmotion",10, 2, 2, 1) 'append a string to the beginning of the component text and append a new line The operation effect is as follows: only content with no more than 16 characters is displayed, and the overflow part is cleared from the end  zmotion 正运动技

6.6.22. HMI_IFMONO – Get Window Monopoly State

Type	Window operation command.
Description	Used in the “reflash” refresh sub of the custom object to determine whether the current object is monopolized by other windows, do not respond to mouse and button messages when monopolized, -1-monopolized, 0-not monopolized
Grammar	Value=HMI_IFMONO
Controller	Controllers that support RTHMI

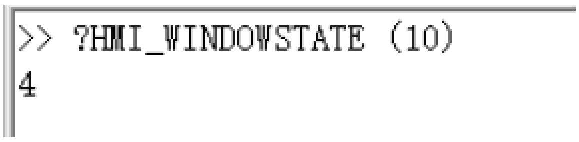
Example	GLOBAL SUB reflash()
	IF SCAN_EVENT(HMI_IFMONO)<0 THEN
	?"return to the custom object window"
	ENDIF
	IF SCAN_EVENT(HMI_IFMONO)>0 THEN
	?”leave custom object window"
	ENDIF
	END SUB

6.6.23. HMI_WINDOWSTATE – Get Window State

Type	Window operation command
Description	Get the window state. Firmware versions after 20161112 are supported.

Grammar	value = HMI_WINDOWSTATE (winid [, tablenum])
	winid: the window number in the HMI file
	tablenum: store the position and size of the window, and store posx, posy, sizex, sizey in sequence.
	The window type corresponding to the return value:
	ZPLC_WIN_TYPE_AUTO = 0, no display
	ZPLC_WIN_TYPE_TOP = 1, top window
	ZPLC_WIN_TYPE_BOTTOM = 2, bottom window
	ZPLC_WIN_TYPE_BASE = 4, base window
	ZPLC_WIN_TYPE_KEYBOARD = 5, soft keyboard.
	ZPLC_WIN_TYPE_POP = 6, pop-up window
	ZPLC_WIN_TYPE_MENU = 7, menu window, close automatically

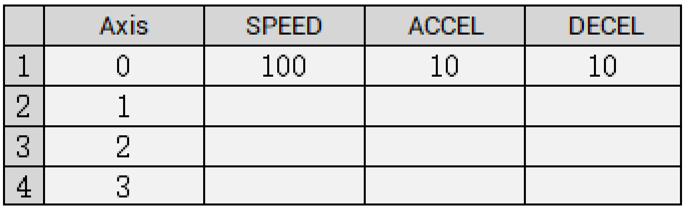
Controller	Controllers that support ZHMI

Example	
	`>> ?HMI_WINDOWSTATE (10)`
	`4`
	Read result: 4, which means window 10 is the base window.

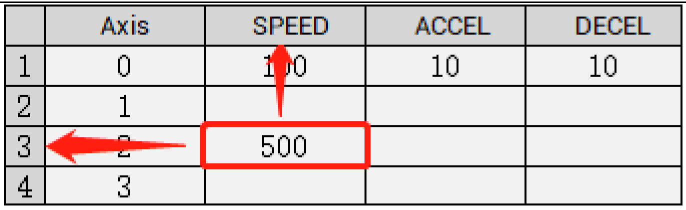
6.6.24. HMI_MOVEWINDOW – Move Assigned Window

Type	Window operation command
Description	Move specified window. Firmware versions after 20161112 are supported.
Grammar	HMI_MOVEWINDOW (winid, posx, posy [, sizex, sizey]) winid: window No. posx: horizontal coordinates posy: vertical coordinates sizex: horizontal size sizey: vertical size
Controller	Controllers that support RTHMI
Example	HMI_MOVEWINDOW (11,100,100) 'change the display position of window 11 to (100,100)

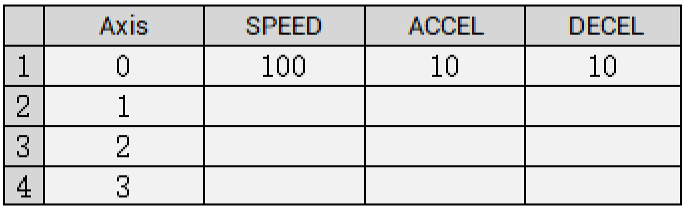
6.6.25. HMI_TABLEVALUE – Set / Get Table Value

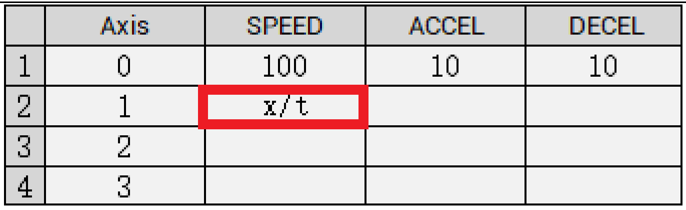
Type	"report view" control operation commands
Description	Set / get assigned value of table. When the table items are assigned as numeric type, it will call this command to set and get data. When getting the value, it will only return to the selected first one cell value in multiple selection.
Grammar	**Set / get table value, it only can write one single data.** HMI_TABLEVALUE (winid, controlid, row, col, value) value = HMI_TABLEVALUE (winid, controlid, row, col) winid: window No. in HMI file controlid: component No. row: table assigned row, -1 means select all rows. col: table assigned column, -1 means select all columns value: table assigned item's value When several are selected, same value is written into selected item. When getting the data, not support multiple selection. **Set / get table value in bulk, write into / read from table.** HMI_TABLEVALUE (winid, controlid, row, col, tabid, ifset) tabid: table starting position that saves table value ifset: whether to write, 0-reading, 1-writing Note: when the selected cursor row and column equal to -1 at the same time, which means no select (not select all). This command only can select one cell / one row / one column. Row No. and column No. start coding from 0 (head row / head column is not included)!
Controller	Controllers that support RTHMI
Example	HMI_TABLEVALUE(16, 1 ,2, 1, 500) 'in window 16, set the row 3 column 2 cell value of the first control as 500 Then: In "output" command, enter this command, then, the corresponding value will be set as 500. 

	Axis	SPEED	ACCEL	DECEL
1	0	100	10	10
2	1			
3	2			
4	3			

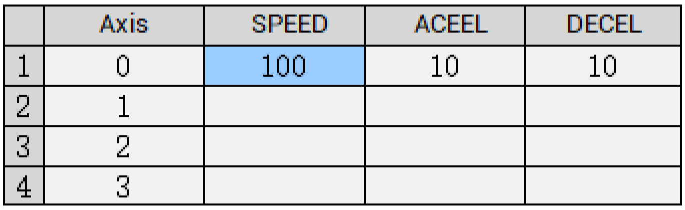
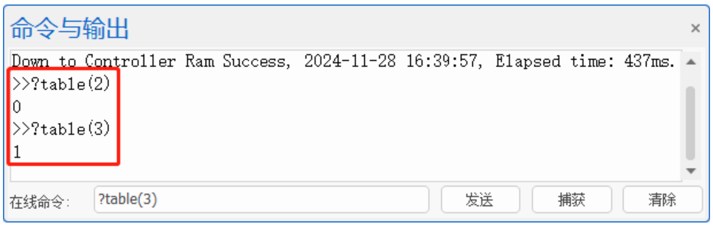
<table>
<tr><td></td><td>
<table>
<tr><th></th><th>Axis</th><th>SPEED</th><th>ACCEL</th><th>DECEL</th></tr>
<tr><td>1</td><td>0</td><td>100</td><td>10</td><td>10</td></tr>
<tr><td>2</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>2</td><td>500</td><td></td><td></td></tr>
<tr><td>4</td><td>3</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

6.6.26. HMI_TABLETEXT – Set / Get Table Content

<table>
<tr><td><b>Type</b></td><td>“report view” control operation commands</td></tr>
<tr><td><b>Description</b></td><td>Set / get assigned content of the table (character string).
When the table items are assigned as string type, it will call this command to set and get data.</td></tr>
<tr><td><b>Grammar</b></td><td>HMI_TABLETEXT (winid, controlid, row, col, “string”)
string = HMI_TABLETEXT (winid, controlid, row, col)
winid: window No. in HMI file
controlid: component No.
row: table assigned row, -1 means select all rows.
col: table assigned column, -1 means select all columns
string: table assigned item’s content (string)
When several are selected, use table delimiter “,” to segment several string, and use the line break “CHR(10)” to change the line for character string.
For Example: HMI_TABLETEXT (10,1,-1,-1, “aaa, bbb, ccc + CHR (10) + ddd, eee, fff, ggg, hhh, iii”), it means all rows and columns of the first control of window 10 are selected, then, set data are below: there are 9 cells data, the first row has 3 data, the second row has 6 data.
<table>
<tr><td>aaa</td><td>bbb</td><td>ccc</td><td></td><td></td><td></td></tr>
<tr><td>ddd</td><td>eee</td><td>fff</td><td>ggg</td><td>hhh</td><td>iii</td></tr>
</table>
Note: This command only can operate selected part. Row No. and column No. start coding from 0 (head row / head column is not included)!</td></tr>
<tr><td><b>Controller</b></td><td>Controllers that support RTHMI</td></tr>
<tr><td><b>Example</b></td><td>HMI_TABLETEXT(16, 1 ,1, 1, “x/t”)
‘in window 16, set the row 2 column 2 cell value of the first control as x/t
Then:
In “output” command, enter this command, then, the corresponding content will be x/t.

<table>
<tr><th></th><th>Axis</th><th>SPEED</th><th>ACCEL</th><th>DECEL</th></tr>
<tr><td>1</td><td>0</td><td>100</td><td>10</td><td>10</td></tr>
<tr><td>2</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>2</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>3</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

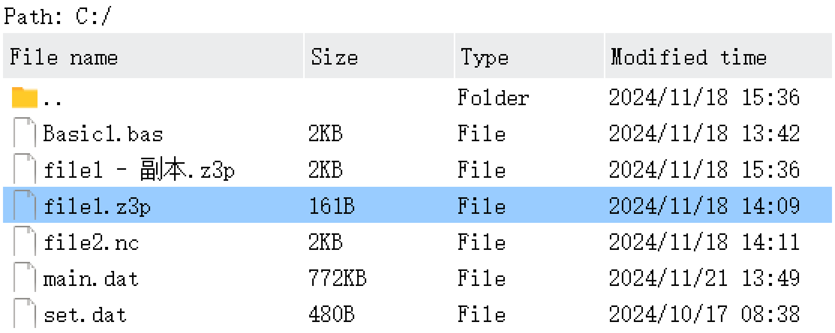
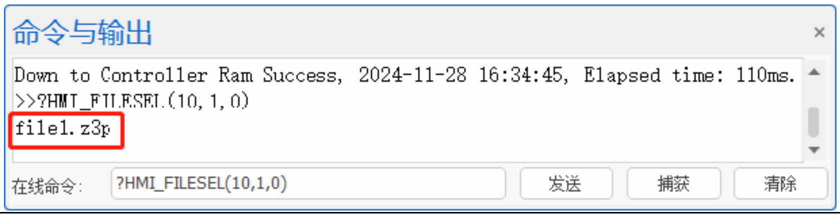
<table>
<tr><td></td><td>
<table>
<tr><th></th><th>Axis</th><th>SPEED</th><th>ACCEL</th><th>DECEL</th></tr>
<tr><td>1</td><td>0</td><td>100</td><td>10</td><td>10</td></tr>
<tr><td>2</td><td>1</td><td>x/t</td><td></td><td></td></tr>
<tr><td>3</td><td>2</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>3</td><td></td><td></td><td></td></tr>
</table></td></tr>
</table>

6.6.27. HMI_TABLECURSOR – Get Current Selected Row & Column

<table>
<tr><td><b>Type</b></td><td>“report view” control operation commands</td></tr>
<tr><td><b>Description</b></td><td>Get current row and column of assigned “report view” control, and save it into TABLE.</td></tr>
<tr><td><b>Grammar</b></td><td>HMI_TABLECURSOR (winid, controlid, num [,num2])
winid: window No. in HMI file
controlid: component No.
num: save current selected row No. and column No. into table (num), table (num+1).
➢ When row No. = -1 – select all rows.
➢ When column No. = -1 – select all columns.
➢ When row No. = -1 & column No. = -1 – no selected.
Note: for row No. and column No., they start coding from 0 (head row / head column is not included)!</td></tr>
<tr><td><b>Controller</b></td><td>Controllers that support RTHMI</td></tr>
<tr><td><b>Example</b></td><td>HMI_TABLECURSOR (16, 1, 2)
‘get the current selected row and column No. of the first component in window 16, and save them into table (2) and table (3) separately.
➢ in “output” command, enter “?table(2)” & “?table(3)” to print current selected row No. and column No.

<table>
<tr><th></th><th>Axis</th><th>SPEED</th><th>ACEEL</th><th>DECEL</th></tr>
<tr><td>1</td><td>0</td><td>100</td><td>10</td><td>10</td></tr>
<tr><td>2</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>2</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>3</td><td></td><td></td><td></td></tr>
</table>


<pre>
Down to Controller Ram Success, 2024-11-28 16:39:57, Elapsed time: 437ms.
>>?table(2)
0
>>?table(3)
1
</pre>
在线命令: ?table(3)
</td></tr>
</table>

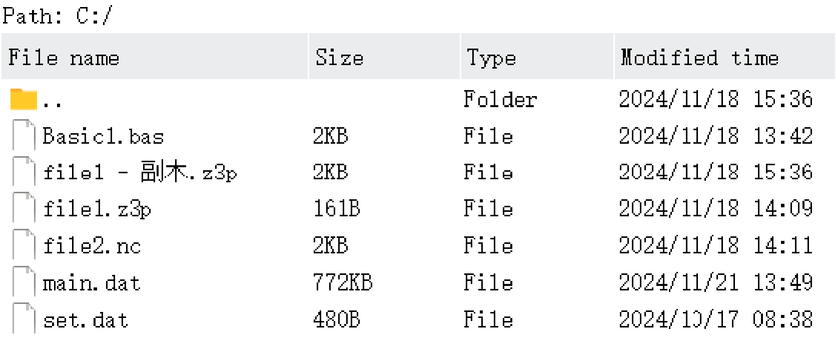
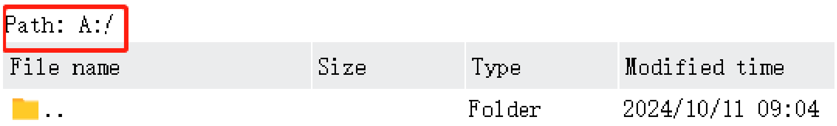
6.6.28. HMI_FILESEL – Get Selected File / Folder Name

<table>
<tr><td><b>Type</b></td><td>“file browser” control operation commands</td></tr>
<tr><td><b>Description</b></td><td>Get current selected file / folder name (without path) of “file browser” control.</td></tr>
<tr><td><b>Grammar</b></td><td>filename = HMI_FILESEL (winid, controlid, [,tabid])
winid: window No. in HMI file
controlid: component No.
tabid: table No., allow select, whether folder is selected.
table (tabid) = 0 is folder, = 1 means folder, =-1 means not select
filename: return current selected file / folder name, for “not select” state, it will return to empty.</td></tr>
<tr><td><b>Controller</b></td><td>Controllers that support RTHMI</td></tr>
<tr><td><b>Example</b></td><td>?HMI_FILESEL (10,1,0)   ‘get and print current selected file name of window 10 first control.
In “output” command, enter this command to print current selected file name.

Path: C:/
<table>
<tr><th>File name</th><th>Size</th><th>Type</th><th>Modified time</th></tr>
<tr><td>..</td><td></td><td>Folder</td><td>2024/11/18 15:36</td></tr>
<tr><td>Basic1.bas</td><td>2KB</td><td>File</td><td>2024/11/18 13:42</td></tr>
<tr><td>file1 - 副本.z3p</td><td>2KB</td><td>File</td><td>2024/11/18 15:36</td></tr>
<tr><td>file1.z3p</td><td>161B</td><td>File</td><td>2024/11/18 14:09</td></tr>
<tr><td>file2.nc</td><td>2KB</td><td>File</td><td>2024/11/18 14:11</td></tr>
<tr><td>main.dat</td><td>772KB</td><td>File</td><td>2024/11/21 13:49</td></tr>
<tr><td>set.dat</td><td>480B</td><td>File</td><td>2024/10/17 08:38</td></tr>
</table>


Down to Controller Ram Success, 2024-11-28 16:34:45, Elapsed time: 110ms.
>>?HMI_FILESEL(10,1,0)
file1.z3p
?HMI_FILESEL(10,1,0)</td></tr>
</table>

6.6.29. HMI_FILEPATH – Set / Get Current Path

Type	“file browser” control operation commands
Description	Set / get current path (use “/” to end)
Grammar	HMI_FILEPATH (winid, controlid, filepath) filepath = HMI_FILEPATH (winid, controlid)

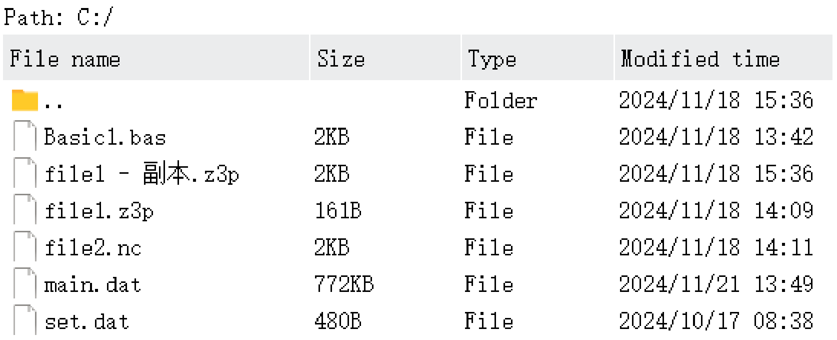
<table>
<tr><td></td><td>winid: window No. in HMI file
controlid: component No.
filepath: return / set current path
valid disks are “C:/” and “A:/”, default is C disk. C disk is flash, A disk is U disk.</td></tr>
<tr><td><b>Controller</b></td><td>Controllers that support RTHMI</td></tr>
<tr><td><b>Example</b></td><td>HMI_FILEPATH(10, 1, “A:/”) ‘set path of window 10 first control as A disk
➢ Current default path is C disk

Path: C:/
<table>
<tr><th>File name</th><th>Size</th><th>Type</th><th>Modified time</th></tr>
<tr><td>..</td><td></td><td>Folder</td><td>2024/11/18 15:36</td></tr>
<tr><td>Basic1.bas</td><td>2KB</td><td>File</td><td>2024/11/18 13:42</td></tr>
<tr><td>file1 - 副本.z3p</td><td>2KB</td><td>File</td><td>2024/11/18 15:36</td></tr>
<tr><td>file1.z3p</td><td>161B</td><td>File</td><td>2024/11/18 14:09</td></tr>
<tr><td>file2.nc</td><td>2KB</td><td>File</td><td>2024/11/18 14:11</td></tr>
<tr><td>main.dat</td><td>772KB</td><td>File</td><td>2024/11/21 13:49</td></tr>
<tr><td>set.dat</td><td>480B</td><td>File</td><td>2024/10/17 08:38</td></tr>
</table>
➢ In “output” command, enter this command, the path is set as A disk.

Path: A:/
<table>
<tr><th>File name</th><th>Size</th><th>Type</th><th>Modified time</th></tr>
<tr><td>..</td><td></td><td>Folder</td><td>2024/10/11 09:04</td></tr>
</table>
</td></tr>
</table>

6.6.30. HMI_FILEFILTER – Set File Filter

Type	“file browser” control operation commands
Description	Set file filter, filter to show current path all files. Note: file filter only filters files, folder will be not filtered.
Grammar	HMI_FILEFILTER (winid, controlid, strfilter) strfilter = HMI_FILEFILTER (winid, controlid) winid: window No. in HMI file controlid: component No. strfilter: file filter item, support symbol “*”, take “|” to space several filter items. enter “” to cancel file filter, show all files. enter “*.*” to show all files that are with suffix. enter “*.nc” | *.cnc” to show all nc and cnc files. Note: filter options are not case sensitive.
Controller	Controllers that support RTHMI
Example	HMI_FILEFILTER(10, 1, “*.bas”)

‘show all bas files of window 10 first control

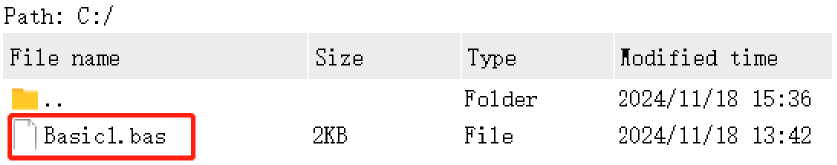
- File filter shows all C disk files by default



Path: C:/

File name	Size	Type	Modified time
..		Folder	2024/11/18 15:36
Basic1.bas	2KB	File	2024/11/18 13:42
file1 - 副本.z3p	2KB	File	2024/11/18 15:36
file1.z3p	161B	File	2024/11/18 14:09
file2.nc	2KB	File	2024/11/18 14:11
main.dat	772KB	File	2024/11/21 13:49
set.dat	480B	File	2024/10/17 08:38

- In “output” command, enter this command, it only shows C disk all files



Path: C:/

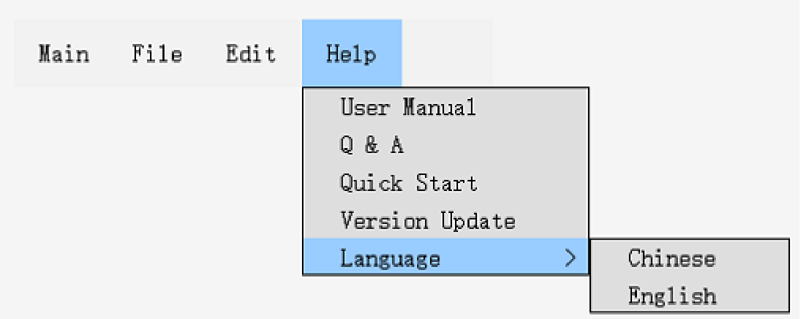
File name	Size	Type	Modified time
..		Folder	2024/11/18 15:36
Basic1.bas	2KB	File	2024/11/18 13:42

6.6.31. HMI_MENUITEM – Get / Set Menu Item State

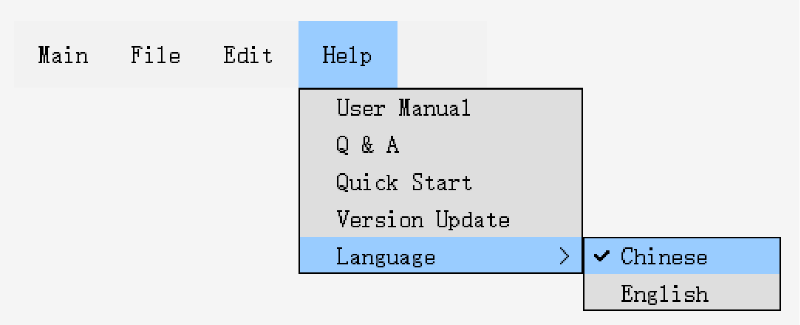
Type	“menu” control operation commands
Description	Get / modify menu item state
Grammar	HMI_MENUITEM (id, option, value [,winid, controlid]) value = HMI_MENUITEM (id, option, [,winid, controlid]) id: menu item No. (ID) option: data option, 0 – set menu item as (checked) state, 1 – set menu item as gray (disable) state. value: option value when option = 0 value = 0: set this menu item as “not select” state. value ≠ 0: set this menu item as “selected” state. when option = 1 value = 0: set this menu item as “disable” state. value ≠ 0: set this menu item as “checked” state. winid: window No. in HMI file controlid: component No.
Controller	Controllers that support RTHMI
Example	HMI_MENUITEM (1031, 0, 1, 10, 1) ‘make the menu ID 1031 of window 10 first control as selected HMI_MENUITEM (1032, 1, 0, 10, 1)

'make the menu ID 1032 of window 10 first control as gray state

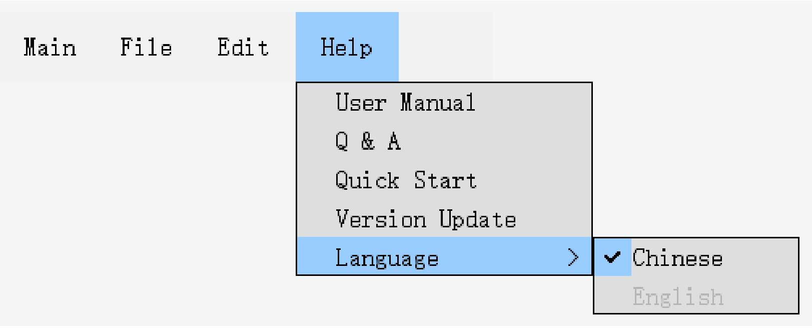
- In this menu, "Chinese" menu ID is 1031, "English" menu ID is 1032.



- Enter the first one command in "output", "Chinese" menu item will be checked.



- Enter the second one command in "output", "English" menu item will be disabled.



6.6.32. HMI_CODEPAGE – Get / Set Code Page

Type	Operation command
Description	This is used when displaying other languages (except Chinese) in HMI. After setting it, the fonts of other languages will be displayed correctly, avoiding garbled characters.
Grammar	HMI_ CODEPAGE=value value = HMI_ CODEPAGE

	value: code page No., you could check needed one in website.
Controller	Controllers that support RTHMI
Example	HMI_CODEPAGE = 936 'set as Simplified Chinese code page ? HMI_CODEPAGE 'print the currently used code page encoding

6.6.33. HMI_EVENTHIDE – Hide Event

Type	Operation command
Description	This is used when displaying other languages (except Chinese) in HMI. After setting it, the fonts of other languages will be displayed correctly, avoiding garbled characters.
Grammar	HMI_EVENTHIDE(winid, controlid, mode) mode = HMI_EVENTHIDE (winid, controlid) winid: window No. in HMI controlid: the component No. mode: hide the control mode =0: show all events =1: hide the event of “confirmed” =2: hide the event of “resumed” =3: hide the event of “confirmed” / “resumed” =3: hide the event of “confirmed” & “resumed”
Controller	Controllers that support RTHMI
Example	HMI_EVENTHIDE(10,1, 1) 'hide the “event show” component’s confirmed event.

6.6.34. HMI_EVENTDEL – Delete Event

Type	Operation command
Description	Delete current selected event of “event show” component, it only can delete resumed event.
Grammar	HMI_EVENTDEL(winid, controlid) winid: window No. in HMI controlid: the component No.
Controller	Controllers that support RTHMI
Example	HMI_EVENTDEL(10, 1) 'delete current selected event of “event show” component

6.6.35. HMI_XYChartClear – Clear XY Chart Data

Type	Operation command
Description	Clear curve data.
Grammar	HMI_XYChartClear (winid, controlid, channel) winid: window No. in HMI controlid: the component No. channel: assigned channel, start from 0 (-1 means clear all channels data)
Controller	Controllers that support RTHMI
Example	'HMI_XYChartClear(10, 1, 0) 'clear data for channel 0. 'HMI_XYChartClear(10, 1, -1) 'clear data for all channels.

6.6.36. HMI_XYChartWrite – Set XY Chart Data

Type	Operation command
Description	Write data for the chart.
Grammar	HMI_XYChartWrite (winid, controlid, channel, x, y) winid: window No. in HMI controlid: the component No. channel: assigned channel, start from 0 x, y: write the coordinates
Controller	Controllers that support RTHMI
Example	'write 3 points (25, 25) (50, 25) (50, 50) for channel 0 HMI_XYChartWrite (10, 1, 0, 25, 25) HMI_XYChartWrite (10, 1, 0, 50, 25) HMI_XYChartWrite (10, 1, 0, 50, 50)

6.6.37. HMI_XYChartRange – Set XY Chart Limit Range

Type	Operation command
Description	Set the upper and lower limit ranges of XY chart.
Grammar	HMI_XYChartRange (winid, controlid, channel [, minx, miny, maxx, maxy]) winid: window No. in HMI controlid: the component No.

	channel: assigned channel, start from 0 minx, miny: set the lower limit range maxx, maxy: set the upper limit range
Controller	Controllers that support RTHMI
Example	'set channel 0 displayed range as (0,0)->(100,100) HMI_XYChartRange (10, 1, 0, 0, 0, 100, 100)

6.6.38. HMI_CALLSUB – Call SUB Function

Type	Call type command
Description	The SUB function is called dynamically, and execution continues after the SUB function finishes executing.
Grammar	HMI_CallSub (strSubName) strSubName: called SUB name
Controller	Controllers that support RTHMI
Example	'execute the MySub function HMI_CALLSUB("MySub") 'define the SUB function global sub MySub() 'execute MySub' end sub

6.6.39. HMI_MSGOUT – Output Message

Type	HMI command
Description	The SUB function is called dynamically, and execution continues after the SUB function finishes executing.
Grammar	HMI_MsgOut(strtype, strmsg, winid, controlid [, ifInfo]) strtype: the type of output “TRACE”: track & output “WARN”: warn & output “ERROR”: error & output “PRINT”: print & output “CLEAR”: clear output strmsg: output the message (character string)

	winid: window No. in HMI controlid: the component No. ifInfo: whether to automatically output the file row information, default is 0, which means not output. Note: the message output by this command will automatically wrap to a newline.
Controller	Controllers that support RTHMI
Example	'the component "message out" ID is 3, and in window 10 HMI_MsgOut("WARN", "this is a warning.", 10, 3) warn: this is a warning. HMI_MsgOut("ERROR", "this is a error." , 10, 3, 1) error of file:xx line:xx task:xx: this is a error. error of online command: this is a error. HMI_MsgOut("PRINT", "this is a print." , 10, 3) this is a print.

6.6.40. HMI_TrChartShow – Show Trend Chart

Type	HMI command
Description	Show the trend chart.
Grammar	*Get the assigned channel displayed state* bifshow = HMI_TrChartShow (winid, controlid, channel) *Set whether to show assigned channel* HMI_TrChartShow (winid, controlid, channel, bifshow) winid: window No. in HMI controlid: the component No. channel: assign the channel, starting from 0 bifshow: whether to show
Controller	Controllers that support RTHMI

Example	'show channel 0
	HMI_TrChartShow(10, 1, 0, TRUE)
	
	'get the channel 0 displayed state
	bifshow = HMI_TrChartShow(10, 1, 0)

6.6.41. HMI_TrChartRange – Set Trend Chart Range

Type	HMI command
Description	Show the trend chart.
Grammar	HMI_TrChartRange (winid, controlid, channel, min, maxx) winid: window No. in HMI controlid: the component No. channel: assign the channel, starting from 0 min: minimal value max: maximum value
Controller	Controllers that support RTHMI
Example	'set the trend chart's channel 0 displayed range in Y as -100 ~ 100. HMI_TrChartRange(10, 1, 0, -100, 100)

6.6.42. HMI_TrChartXDis – Set Trend Chart "X" Gap

Type	HMI command
Description	Set trend chart X-axis spacing. Note: X-axis spacing represents the pixel between two sampling points.
Grammar	HMI_TrChartXDis (winid, controlid, xdis) winid: window No. in HMI controlid: the component No. xdis: the axis X spacing
Controller	Controllers that support RTHMI
Example	'set the gap between 2 sampling points as 10 pixel. HMI_TrChartXDis (10, 1, 10)

6.6.43. HMI_SampClear – Clear Trend Chart Data

Type	HMI command
Description	Clear trend chart's sampling data.
Grammar	HMI_SampClear (id) id: sampling No.
Controller	Controllers that support RTHMI

Example	'clear sampling point 0 data HMI_SampClear (0)

6.6.44. HMI_SampStart – Start Trend Chart "Sampling"

Type	HMI command
Description	Start sampling data for trend chart.
Grammar	HMI_SampStart (id) id: sampling No.
Controller	Controllers that support RTHMI
Example	'start sampling 0 HMI_SampStart (0)

6.6.45. HMI_SampPause – Pause Trend Chart "Sampling"

Type	HMI command
Description	Pause sampling data for trend chart.
Grammar	HMI_SampStart (id) id: sampling No.
Controller	Controllers that support RTHMI
Example	'pause sampling 0 HMI_SampPause (0)

6.6.46. HMI_SampSave – Pause Trend Chart "Sampling"

Type	HMI command
Description	Save the sampling data into assigned file.
Grammar	HMI_SampSave(id, savemode, filename) id: sampling No. savemode: storage mode 0 – save as new file, if the file exists, it will be covered. 1 – append to save, if the file already exists, append to it. filename: saved file name (include the path)
Controller	Controllers that support RTHMI
Example	'Save the current sampled data of sample 0 to a file.

	HMI_SampSave(0, 0, "samp0_20250818.txt")

6.6.47. HMI_GDataClear – Clear Group Data

Type	HMI command
Description	Clear the group data.
Grammar	HMI_GDataClear(winid, controlid, channel) id: sampling No. winid: window No. in HMI controlid: the component No. channel: assigned channel
Controller	Controllers that support RTHMI
Example	'clear data group channel 0 data HMI_GDataClear(10, 1, 0)

6.6.48. HMI_GDataWrite – Write into Group Data

Type	HMI command
Description	Write into group data.
Grammar	HMI_GDataWrite(winid, controlid, channel, y [,pointNum]) winid: window No. in HMI controlid: the component No. channel: assigned channel y: the table index for writing coordinates (y) / storing continuous coordinates (y). pointNum: the number of points to write, defaults to 0. When pointNum > 0, the y parameter is used as the table index for storing continuous Y positions.
Controller	Controllers that support RTHMI
Example	'write one point into data group channel 0 HMI_GDataWrite(10, 1, 0, 66) 'write 5 points into data group channel 0 table(0) = 10 table(1) = 50 table(2) = 30

	table(3) = 40 table(4) = 20 HMI_GDataWrite(10, 1, 0, 0, 5)

6.6.49. HMI_GDataRange – Set Data Group Show Range

Type	HMI command
Description	Set the data group show upper and lower limit range.
Grammar	HMI_XYChartRange(winid, controlid, channel, miny, maxy) winid: window No. in HMI controlid: the component No. channel: assigned channel miny: show the lower limit value maxy: show the upper limit value
Controller	Controllers that support RTHMI
Example	'set the data group channel 0 lower and upper limit range as -100~100 HMI_XYChartRange(10, 1, 0, -100, 100)

6.6.50. HMI_RECIPEGNAME – Get All Recipe Group Name of Recipe Library

Type	Recipe Library Command
Description	Get all recipe group name of recipe library.
Grammar	gname = Hmi_RecipeGName(strSel) strSel: select the command “FIRST”: get 1st recipe group name “NEXT”: get the next recipe group name gname: returned recipe group name, if it is empty, which means there is no recipe group behind.
Controller	Controllers that support RTHMI
Example	local strname(128) 'get 1st recipe group name strname = Hmi_RecipeGName("FIRST") 'get the next recipe group name strname = Hmi_RecipeGName("NEXT")

6.6.51. HMI_RECIPEGINFO – Get Recipe Group Recipe Information

<table>
<tr><td><b>Type</b></td><td>Recipe Library Command</td></tr>
<tr><td><b>Description</b></td><td>Get recipe group recipe information.</td></tr>
<tr><td><b>Grammar</b></td><td>value = Hmi_RecipeGInfo (gname, strsel [, para])
gname: recipe group name
strSel: select the information
“LINES”: recipe’s rows numbers
“ITEMS”: recipe’s columns numbers
“DTYPE”: assign the recipe item’s data type [para is valid, indicating which formula item.]
“DLEN”: assign the recipe item’s data length [para is valid, indicating which formula item.]
“DDOT”: assign the recipe item’s decimal numbers [para is valid, indicating which formula item.]
“EDIT”: assign whether the recipe item is editable [para is valid, indicating which formula item.]
gname: returned information

➢ <b>Data Types:</b>
<table>
<tr><td>1</td><td>Bit</td></tr>
<tr><td>-8 / 8</td><td>int8 / unit8</td></tr>
<tr><td>-16 / 16</td><td>int16 / unit16</td></tr>
<tr><td>-32 / 32</td><td>int32 / unit32</td></tr>
<tr><td>-64 / 64</td><td>int64 / unit64</td></tr>
<tr><td>2</td><td>character string</td></tr>
<tr><td>33</td><td>float</td></tr>
<tr><td>65</td><td>double</td></tr>
</table>
</td></tr>
<tr><td><b>Controller</b></td><td>Controllers that support RTHMI</td></tr>
<tr><td><b>Example</b></td><td>'print the recipe rows of assigned recipe group
?Hmi_RecipeGInfo("RecipeGroup0", "LINES")</td></tr>
</table>

6.6.52. HMI_RECIPECELLSTR – Get / Set Recipe Character String Data of Assigned Row & Column

Type	Recipe Library Command
Description	Get / set recipe character string data of assigned row & column (cell).

Grammar	Set: Hmi_RecipeCellStr(gname, row, col, strData) Get: strData = Hmi_RecipeCellStr (gname, row, col) gname: recipe group name row: assigned row col: assigned column strData: recipe item data, in character string format
Controller	Controllers that support RTHMI
Example	'print the character string data of assigned recipe group cell (1, 1) ?Hmi_RecipeCellStr ("RecipeGroup0", 1, 1)

6.6.53. HMI_RECIPECELLVAL – Get / Set Recipe Value of Assigned Row & Column

Type	Recipe Library Command
Description	Get / set recipe value data of assigned row & column (cell).
Grammar	Set: Hmi_RecipeCellVal (gname, row, col, valData) Get: valData = Hmi_RecipeCellVal (gname, row, col) gname: recipe group name row: assigned row col: assigned column valData: recipe item data, in value format
Controller	Controllers that support RTHMI
Example	'print the value data of assigned recipe group cell (1, 1) ?Hmi_RecipeCellVal ("RecipeGroup0", 1, 1)

6.6.54. HMI_RECIPEIMPORT – Import Recipe File

Type	Recipe Library Command
Description	Import the recipe file.
Grammar	Hmi_RecipeImport (gname, filename, mode) gname: recipe group name filename: recipe file name mode: import modes: ● 0: cover mode, clear old data ● 1: append mode, add at the end of old data
Controller	Controllers that support RTHMI
Example	'import recipe file “RecipeFile0.rdb” from flash content Hmi_RecipeImport ("RecipeGroup0", "RecipeFile0.rdb", 0)

6.6.55. HMI_RECIPESAVE – Save Recipe File

Type	Recipe Library Command
Description	Save & export the configuration file.
Grammar	Hmi_RecipeSave(gname, filename) gname: recipe group name filename: recipe file name
Controller	Controllers that support RTHMI
Example	'save recipe file “RecipeFile0.rdb” to flash content Hmi_RecipeSave ("RecipeGroup0", "RecipeFile0.rdb")

6.6.56. HMI_RECIPELOAD – Load / Get Recipe Group

Type	Recipe Component Command
Description	Load the recipe group into recipe component / get the current loading recipe group.
Grammar	'load recipe group Hmi_RecipeLoad (winid, controlid, gname) winid: window No. where the recipe component is located controlid: recipe component No. gname: recipe group name 'get current loading recipe group gname = Hmi_RecipeLoad (winid, controlid)
Controller	Controllers that support RTHMI
Example	'the object 1 (component 1) in window 10 is the recipe component, dynamically load the recipe group. Hmi_RecipeLoad (10, 1, "tool")

6.6.57. HMI_RECIPECLOSE – Close Current Loading Recipe Group

Type	Recipe Component Command
Description	Close current loading recipe group.
Grammar	Hmi_RecipeClose (winid, controlid) winid: window No. where the recipe component is located controlid: recipe component No.

Controller	Controllers that support RTHMI
Example	'the object 1 (component 1) in window 10 is the recipe component, close current loaded recipe group. Hmi_RecipeClose (10, 1)

6.6.58. HMI_RECIPECURDATA – Get / Set Cursor Item Recipe Data

Type	Recipe Component Command
Description	Get / set the recipe data at the current cursor position.
Grammar	Set: Hmi_RecipeCurData (winid, controlid, strData) Get: strData = Hmi_RecipeCurData (winid, controlid) winid: window No. where the recipe component is located controlid: recipe component No. strdata: recipe item data, in character string format
Controller	Controllers that support RTHMI
Example	'get the recipe cursor item data value = VAL(Hmi_RecipeCurData (10, 1))

6.6.59. HMI_RECIPEMOVE – Move Current Recipe Row

Type	Recipe Component Command
Description	Move current recipe row.
Grammar	Hmi_RecipeMove(winid, controlid, strSel [,para]) winid: window No. where the recipe component is located controlid: recipe component No. strSel: move “FIRST”: move to the head row “NEXT”: move to the next row “PREV”: move to the previous row “LAST”: move to the last row “LINE”: move to the assigned row, para is valid (= target row No.) para: special parameters
Controller	Controllers that support RTHMI
Example	'move current cursor item recipe row to the next row Hmi_RecipeMove(10, 1, "NEXT")

6.6.60. HMI_RECIPEINSERT – Insert One Row Recipe

Type	Recipe Component Command
Description	Insert one row recipe, and copy current cursor recipe data to new inserted recipe row.
Grammar	Hmi_RecipeInsert(winid, controlid, line) winid: window No. where the recipe component is located controlid: recipe component No. line: where the row is inserted ● -1: insert before the current cursor line ● > max recipe row: insert at the end
Controller	Controllers that support RTHMI
Example	'insert one row recipe data at the first row Hmi_RecipeInsert(10, 1, 1)

6.6.61. HMI_RECIPEDELETE – Delete Recipe Row

Type	Recipe Component Command
Description	Delete the recipe row.
Grammar	Hmi_RecipeDelete(winid, controlid, line) winid: window No. where the recipe component is located controlid: recipe component No. line: delete which rows ● -1: delete current cursor row ● -2: delete the end row
Controller	Controllers that support RTHMI
Example	'delete the row 3 recipe Hmi_RecipeDelete(10, 1, 3)

6.6.62. HMI_RECIPESORT – Sort the Recipe Data

Type	Recipe Component Command
Description	Sort the recipe data.
Grammar	Hmi_RecipeSort(winid, controlid, col, sort) winid: window No. where the recipe component is located controlid: recipe component No. col: the column, starting from 0, it doesn't include the sequence column. sort: sorting mode, 0 – ascending, non-0: descending

Controller	Controllers that support RTHMI
Example	'the recipe rows are sorted in descending order by the second column. Hmi_RecipeSort(10, 1, 2, 1)

6.6.63. Window Message List

ID	Message Type	Message Descriptions
0	Open window	Trigger message when the window is opened.
1	Close window	Trigger message when the window is closed.
2	Change window size	Trigger message when the window size is changed.
3	Move window position	Trigger message when window position is moved.
4	Click the mouse left button	Trigger message when the left mouse button is pressed & released consecutively within a window.
5	Press the mouse left button	Trigger message when the mouse left button is pressed within the window.
6	Release the mouse left button	Trigger message when the mouse left button is released within the window.
7	Move the mouse	Trigger message when the mouse is moved within the window.
8	Press the mouse right button	Trigger message when the mouse right button is pressed within the window.
9	Release the mouse right button	Trigger message when the mouse right button is released within the window.
10	Press the mouse middle button	Trigger message when the mouse middle button is pressed within the window.
11	Release the mouse middle button	Trigger message when the mouse middle button is released within the window.
12	Scroll the mouse	Trigger message when the mouse is scrolled within the window.

6.6.64. HMI_WINMSGCALL – Window Message ON / OFF

Type	Message configuration command.
Description	Set the call switch of window message. That is, when assigned window message is triggered, it will automatically call SUB function that is bound with the window.

Grammar	switch = Hmi_WinMsgCall(winid, msgtype) Hmi_WinMsgCall(winid, msgtype, switch) winid: window No. msgtype: message type, please refer to window message list. switch: the message switch, 0 – OFF, 1 – ON
Controller	Controllers that support RTHMI
Example	'suppose that the window 10 configures the message call function "OnWin10MsgCall" Hmi_WinMsgCall (10, 0, 1) 'switch on "window message" 'window 10 message call function global sub OnWin10MsgCall (msgtype) ?"use window 10 to trigger the message" msgtype end sub Note: when the window 10 is opened, it will trigger OnWin10MsgCall(0).

6.6.65. Component Message List

ID	Message Type	Message Descriptions
0	Start drawing	Trigger message before drawing
1	End drawing	Trigger message when drawing is completed.
2	Move the mouse into component	Trigger message when the mouse enters the component range.
3	Move the mouse out of component	Trigger message when the mouse is away from the component range.
4	Click the mouse left button	Trigger message when the left mouse button is pressed & released consecutively within a window.
5	Press the mouse left button	Trigger message when the mouse left button is pressed within the window.
6	Release the mouse left button	Trigger message when the mouse left button is released within the window.
7	Move the mouse	Trigger message when the mouse is moved within the window.
8	Press the mouse right button	Trigger message when the mouse right button is pressed within the window.
9	Release the mouse right button	Trigger message when the mouse right button is released within the window.
10	Press the mouse middle button	Trigger message when the mouse middle button is pressed within the window.
11	Release the mouse middle	Trigger message when the mouse middle button is

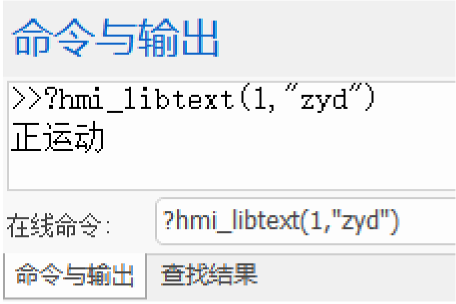
	button	released within the window.
12	Scroll the mouse	Trigger message when the mouse is scrolled within the window.

6.6.66. HMI_CTRLMSGCALL – Component Message ON / OFF

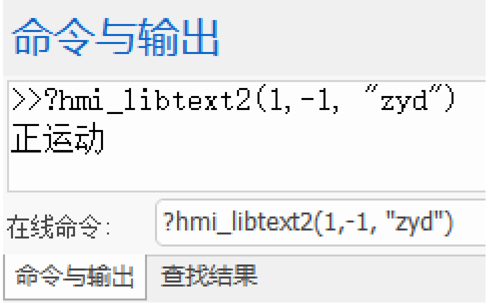
Type	Message configuration command.
Description	Set the call switch of component message. That is, when assigned component message is triggered, it will automatically call SUB function that is bound with the component.
Grammar	switch = Hmi_CtrlMsgCall(winid, controlid, msgtype) Hmi_CtrlMsgCall(winid, controlid, msgtype, switch) winid: window No. controlid: component No. msgtype: message type, please refer to component message list. switch: the message switch, 0 – OFF, 1 – ON
Controller	Controllers that support RTHMI
Example	'suppose that the window 10 configures the message call function "OnMsgCall" Hmi_CtrlMsgCall (10, 1, 0, 5) 'press the mouse left button to trigger the message. 'window 10's component 1 message call function global sub OnMsgCall (msgtype) //it must transfer the parameter msgtype (message type), then it can trigger the SUB. ?"trigger the message" msgtype end sub Note: when the mouse left button is pressed within the component 1 range, it will trigger OnMsgCall(5).

6.6.67. HMI_LIBTEXT – Get Current Language State Contents of Text Library

Type	Text library operation command.
Description	Get the current language state's contents of text library.

Grammar	“character string” = HMI_LIBTEXT (state, “text library name”) state: select the text library state, it refers to current language settings
Controller	Valid in 5XX and its firmware above 20180418.
Example	?HMI_LIBTEXT (1, “zyd”)  >>?hmi_libtext(1,”zyd”) 正运动 在线命令: ?hmi_libtext(1,"zyd")

6.6.68. HMI_LIBTEXT2 – Get Text Library State Contents

Type	Text library operation command.
Description	Get text library state contents.
Grammar	HMI_LIBTEXT2(state, lang, "text library name") state: select the text library state lang: the current language No., -1: use current language settings
Controller	Valid in 5XX and its firmware above 20180418.
Example	?HMI_LIBTEXT2 (1, -1, “zyd”)  >>?hmi_libtext2(1,-1, ”zyd”) 正运动 在线命令: ?hmi_libtext2(1,-1, "zyd")

Chapter VII DT Motion Functions

In order to support the variable number of parameters of the G code, the DT motion function is added, and the command calls the parameter motion of the TABLE table.

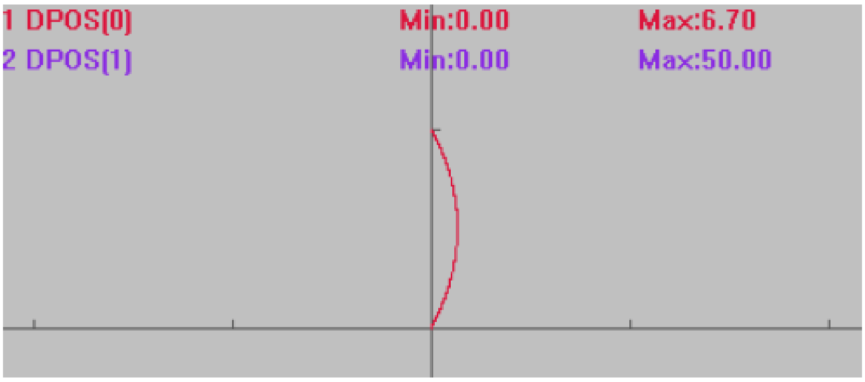
The commands without ABS suffix are relative motion commands, and those with ABS suffix are absolute motion commands.

7.1. MOVEDTSP/MOVEDTABSSP – DT Line Motion

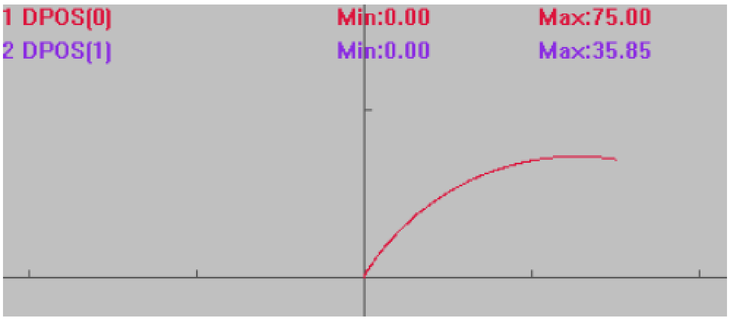
Type	DT Motion Functions
Description	Store the axis No. and movement distance in the TABLE table respectively, and perform linear movement through the TABLE list. Select the axis No. in the TABLE table using bit presence.
Grammar	MOVEDTSP(maximum number of axes, bit present, axis dt list starting number, distance dt list starting number)
Controller	General
Example	TABLE(10,4,5,6) 'TABLE Table 10 stores three axes, axis 4, 5, 6 TABLE(20,100,50,-10) 'TABLE Table 20 stores the motion distance of three axes WHILE 1 IF SCAN_EVENT(IN(0))>0 THEN MOVEDTSP(3,5,10,20) 'the distance in the TABLE table for each movement 'MOVEDTSPABS(3,5,10,20) 'moves to position in TABLE table 'bit present as 5, converted to binary 0101, only axis 4 and axis 6 motion are selected ENDIF WEND

7.2. MOVECIRCDTSP/MOVECIRCDTABSSP – DT Arc

Type	DT Motion Functions
Description	Store the axis number and movement distance in the TABLE table

	respectively, and perform circular movement through the TABLE list. Select the axis number in the TABLE table using bit presence. Draw the circle according to positions of current point (starting point) and middle point and radius. And the center of circle is calculated automatically. When the linear distance between the end point and the starting point is greater than the radius, draw a semicircle with these two points, the radius is half the distance between the lines, and the center of the circle is the midpoint of the line.
Grammar	MOVECIRCDTSP(maximum number of axes, bit present, axis dt list, end point dt list, radius, cw 0/ccw 1)
Controller	General
Example	BASE(0,1,2) ATYPE=1,1,1 DPOS=0,0,0 TABLE(10,0,1) ‘TABLE(10) stores axis 0, axis 1 TABLE(20,0,50) ‘only needs to store one end point coordinate (X, Y) TRIGGER WHILE 1 IF SCAN_EVENT(IN(0))>0 THEN ‘MOVECIRCDTABSSP(6,3,10,20,50,1) MOVECIRCDTSP(6,3,10,20,50,1) 'from the starting point, passing (0,50), with a radius of 50, draw an arc counterclockwise ENDIF WEND  1 DPOS(0) Min:0.00 Max:6.70 2 DPOS(1) Min:0.00 Max:50.00

7.3. MOVECIR2CDTSP/MOVECIR2CDTABSSP – DT Arc by 3-Point

Type	DT Motion Functions
Description	Store the axis number and movement distance in the TABLE table respectively, and perform circular movement through the TABLE list. Select the axis number in the TABLE table using bit presence. Draw an arc based on the start point, reference point, and end point.
Grammar	MOVECIRC2DTSP(max number of axes, bit exists, axis dt list, end point dt list, reference point dt list, mode) mode: <0: reference point is in the front of current point =0: reference point is in the middle >0: reference point is behind current end point
Controller	General
Example	BASE(0,1) ATYPE=1,1 DPOS=0,0 TABLE(10,0,1) 'TABLE(10) stores axis 0 and axis 1 TABLE(20,50,10) 'TABLE(20) stores end point coordinates TABLE(30,25,25) 'TABLE(30) stores reference point coordinates TRIGGER WHILE 1 IF SCAN_EVENT(IN(0))>0 THEN MOVECIR2CDTSP(3,3,10,20,30,0) 'when MODE = 0, the trajectory passes reference point, then it runs to the position of reference point + end point ENDIF WEND 

7.4. MSPHERICALDTSP/MSPHERICALDTABSSP – DT Space Arc

Type	DT Motion Functions
Description	Store the axis number and movement distance in the TABLE table respectively, and perform spherical movement through the TABLE list. Select the axis number in the TABLE table using bit presence. Draw an arc based on the start point, reference point, and end point.
Grammar	MSPHERICALDTSP (axis number, bit presence, axis dt list, end point dt list, reference point dt list, mode) mode: specify the second point position -1: it is in the front of current point, it is only for reference, it will not run to the reference point. 0: it is in the middle, it runs to reference point 1: it is behind current point and end point
Controller	General

Example	BASE(0,1,2)
	ATYPE=1,1,1
	DPOS=0,0,0
	TABLE(10,0,1,2) ‘TABLE(10) stores axis 0, axis 1 and axis 2
	TABLE(20,0,0,100) ‘TABLE(20) stores end point coordinates
	TABLE(30,30,40,50) ‘TABLE(30) stores reference point coordinates
	WHILE 1
	    IF SCAN_EVENT(IN(0))>0 THEN
	        MSPHERICALDTSP(3,7,10,20,30,0)
	        'when MODE = 0, the trajectory passes reference point, then it runs to the position of reference point + end point
	    ENDIF
	WEND

Chapter VIII Reference Routines

8.1. Single Axis Motion

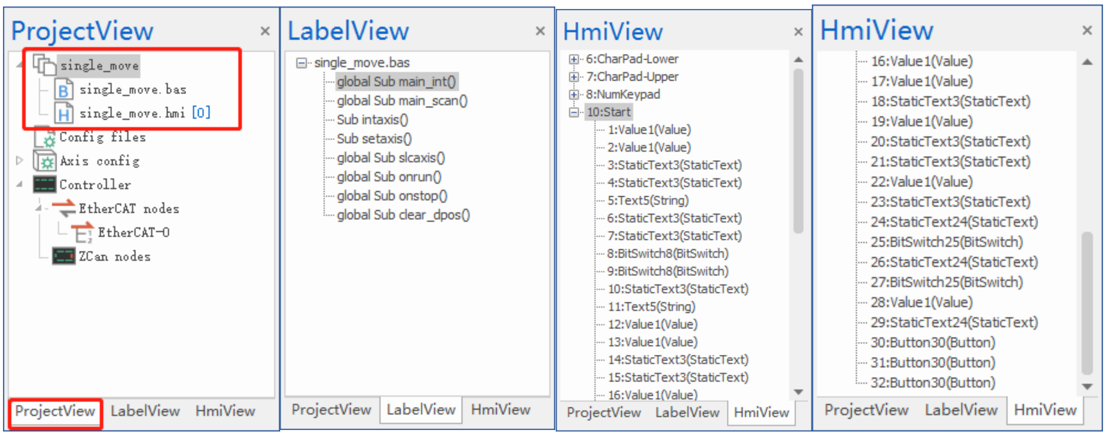
This routine is a single-axis motion routine, which contains two files, and the Basic program is called and executed by HMI. This routine shows all kinds of components usages.

- **Target:** achieve single-axis motion control and state reading through HMI component.
- **How to Do?**
  - Call BASIC function to control axis motion and stop through "Button" component.
  - Read / write data through several "value" components, like, read motion state, write axis parameter.
  - Switch motion mode, motion direction through "bitswitch" component.
- **Interface Creating:**

Create Basic file and HMI file, then define needed GLOBAL SUB function in BASIC for HMI. According to below showing, add HMI component files.

- FileView: the files contained in the project.
- LabelView: SUB sub-process contained in each file.
- HmiView: configuration windows and configuration object.

In HMI programming mode, it is often necessary to use the HmiView to switch configuration programming windows.



single_move
single_move.bas
single_move.hmi [0]
Config files
Axis config
Controller
EtherCAT nodes
EtherCAT-0
ZCan nodes

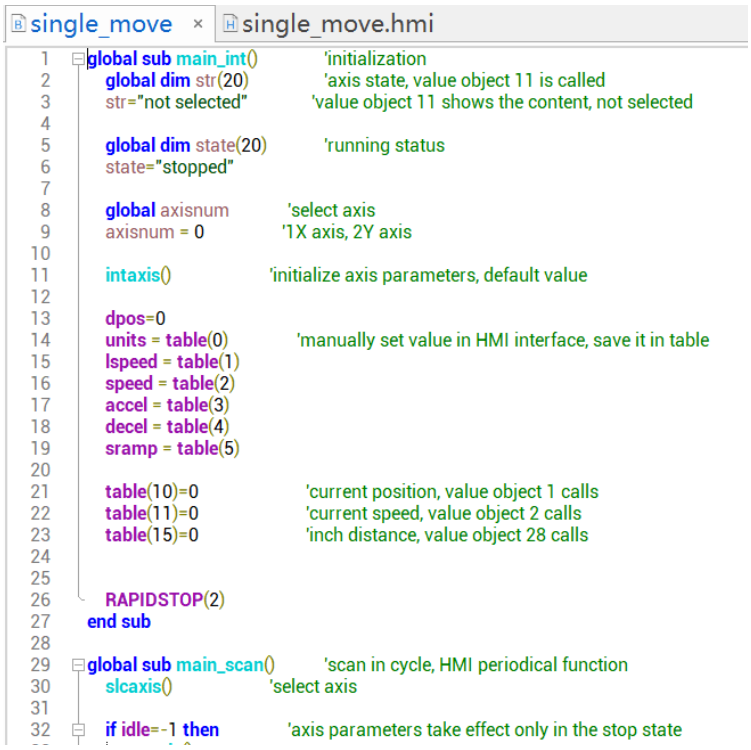
single_move.bas
global Sub main_int()
global Sub main_scan()
Sub intaxis()
Sub setaxis()
global Sub slcaxis()
global Sub onrun()
global Sub onstop()
global Sub clear_dpos()

6:CharPad-Lower
7:CharPad-Upper
8:NumKeypad
10:Start
1:Value1(Value)
2:Value1(Value)
3:StaticText3(StaticText)
4:StaticText3(StaticText)
5:Text5(String)
6:StaticText3(StaticText)
7:StaticText3(StaticText)
8:BitSwitch8(BitSwitch)
9:BitSwitch8(BitSwitch)
10:StaticText3(StaticText)
11:Text5(String)
12:Value1(Value)
13:Value1(Value)
14:StaticText3(StaticText)
15:StaticText3(StaticText)
16:Value1(Value)

16:Value1(Value)
17:Value1(Value)
18:StaticText3(StaticText)
19:Value1(Value)
20:StaticText3(StaticText)
21:StaticText3(StaticText)
22:Value1(Value)
23:StaticText3(StaticText)
24:StaticText24(StaticText)
25:BitSwitch25(BitSwitch)
26:StaticText24(StaticText)
27:BitSwitch25(BitSwitch)
28:Value1(Value)
29:StaticText24(StaticText)
30:Button30(Button)
31:Button30(Button)
32:Button30(Button)

➢ Routine Description:

● Basic Program Interface



```
global sub main_int()          'initialization
    global dim str(20)         'axis state, value object 11 is called
    str="not selected"       'value object 11 shows the content, not selected

    global dim state(20)       'running status
    state="stopped"

    global axisnum         'select axis
    axisnum = 0            '1X axis, 2Y axis

    intaxis()            'initialize axis parameters, default value

    dpos=0
    units = table(0)         'manually set value in HMI interface, save it in table
    lspeed = table(1)
    speed = table(2)
    accel = table(3)
    decel = table(4)
    sramp = table(5)

    table(10)=0             'current position, value object 1 calls
    table(11)=0             'current speed, value object 2 calls
    table(15)=0             'inch distance, value object 28 calls


    RAPIDSTOP(2)
end sub

global sub main_scan()         'scan in cycle, HMI periodical function
    slcaxis()            'select axis

    if idle=-1 then          'axis parameters take effect only in the stop state
```

--Basic Program Codes--

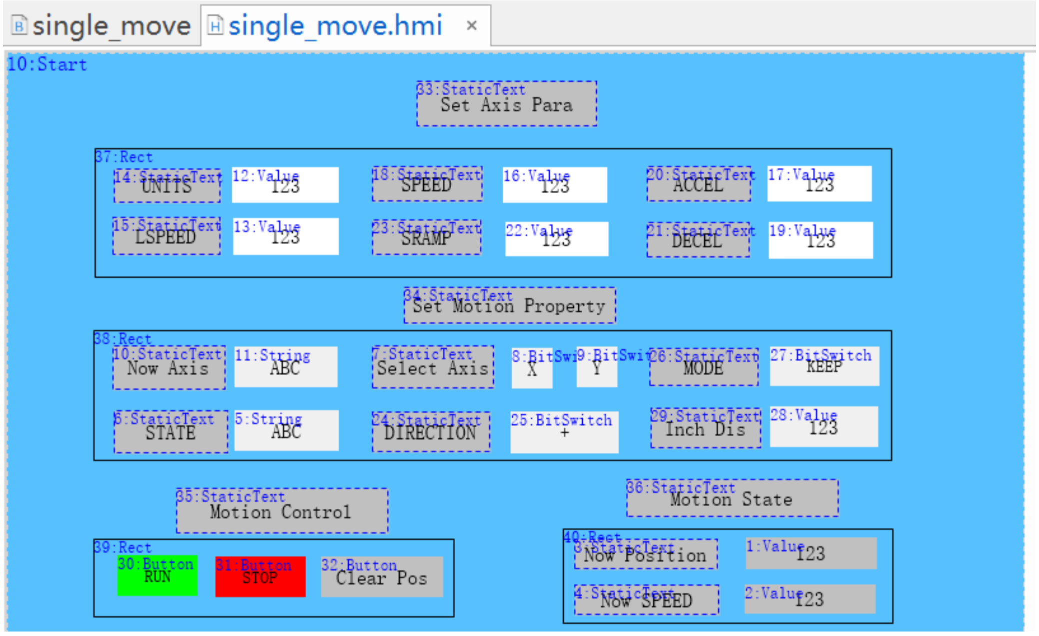
```
global sub main_int()                 'initialization
        global dim str(20)            'axis state, value object 11 is called
        str="not selected"           ‘value object 11 shows the content, not selected

        global dim state(20)          'running status
        state="stopped"

        global axisnum                'select axis
        axisnum = 0                  ‘1X axis, 2Y axis

        intaxis()                     'initialize axis parameters, default value

        dpos=0
        units = table(0)             ‘manually set value in HMI interface, save it in table
        lspeed = table(1)
        speed = table(2)
```

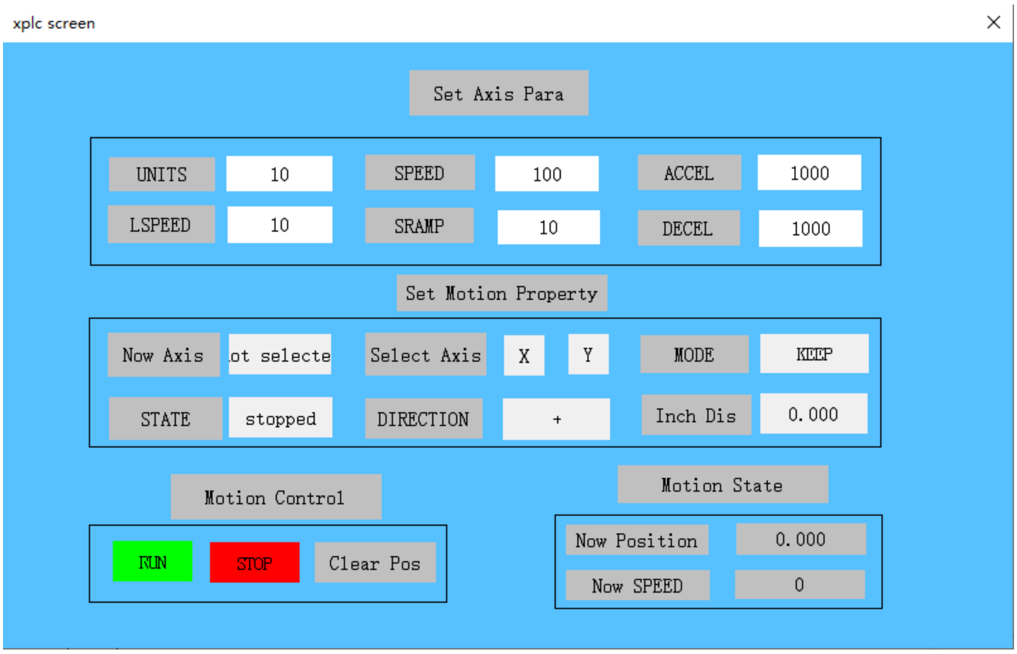
```
    accel = table(3)
    decel = table(4)
    sramp = table(5)

    table(10)=0                               'current position, value object 1 calls
    table(11)=0                               'current speed, value object 2 calls
    table(15)=0                               'inch distance, value object 28 calls

    RAPIDSTOP(2)
end sub

global sub main_scan()              'scan in cycle, HMI periodical function
    slcaxis()                            'select axis

    if idle=-1 then                 'axis parameters take effect only in the stop state
        setaxis()
    endif

    table(10)=DPOS                       ‘get showing dynamically
    table(11)=MSPEED

    if idle=-1 then
        state="stopped"
    endif
end sub

sub intaxis()                         'axis parameter initialization
    table(0)=10                       'units      pulse amount
    table(1)=10                       'lspeed     initial speed
    table(2)=100                      'speed      running speed
    table(3)=1000                'accel      acceleration
    table(4)=1000                'decel      deceleration
    table(5)=10                       'sramp      S curve time
end sub
```

```
sub setaxis()                           'axis parameters setting
    units = table(0)
    lspeed = table(1)
    speed = table(2)
    accel = table(3)
    decel = table(4)
    sramp = table(5)
end sub

global sub slcaxis()                    'select function for axis
    if MODBUS_BIT(0)=1 then             'modbus_bit(0) relates to axis X button in hmi
        cancel(2)axis(1)                ‘when axis is changed, stop axis Y axis 1 motion

        str="axis X"                    ‘display content: axis X
        axusnum=1
        base(0)
    elseif MODBUS_BIT(1)=1 then         ‘modbus_bit(1) relates to axis Y button in hmi
        cancel(2)axis(0)                'When replacing the selected axis, stop Axis X axis 0
movement. If you need to keep moving when changing the axis, please remove the cancel command

        str="axis Y"                    ‘display content: axis Y
        axisnum=2
        base(1)                         ‘select axis Y
    endif
end sub

global sub onrun()                      'running button is called
    if a=0 then
        return                          ‘axisnum = 0, no axis number is selected
    elseif MODBUS_BIT(20)=0 then        'modbus_bit(20) relates to “motion mode” button in
hmi, when it is 0, means continuous.

        if MODBUS_BIT(10)=0 then  'modbus_bit(10) relates to “motion direction” button in
hmi
            vmove(1)
```

```
          elseif MODBUS_BIT(10)=1 then
               VMOVE(-1)
          endif
     elseif MODBUS_BIT(20)=1 then        ‘motion mode, when it is 1, means inching
          move(table(15))                ‘inching distance is specified, value object 28
     endif

     if idle=0 then
          state="running"
     endif
end sub

global sub onstop()       'stopped button is called
     state="stopped"
     RAPIDSTOP(2)
end sub

global sub clear_dpos()  'clear coordinates button is called
     dpos=0
end sub
```

- **HMI Program Interface**





UNITS 10 SPEED 100 ACCEL 1000

LSPEED 10 SRAMP 10 DECEL 1000

Now Axis ot selecte MODE KEEP

STATE stopped DIRECTION + Inch Dis 0.000

Now Position 0.000

Now SPEED 0

- When using this routine, please select needed axes X & Y, otherwise, it can't move. Then select needed motion direction and mode, if you chose INCH, you need to set inch distance,
- For axis parameters, you can custom or use default value, that is, after calling keyboard window, then you can enter needed value. Next, you can click RUN to make axis move. SPEED and DPOS are read into "value" 1 & 2.
- When you clicked STOP, the motion will stop immediately, when you click "clear pos", it will clear current position DPOS.

8.2. Conversion Between Physical Key and Virtual Key

Instructions related to physical keys and virtual keys can only be used in the refresh sub of custom object.

➢ **Target:** bind ZHD400X teach pendant physical keys with customized virtual keys.

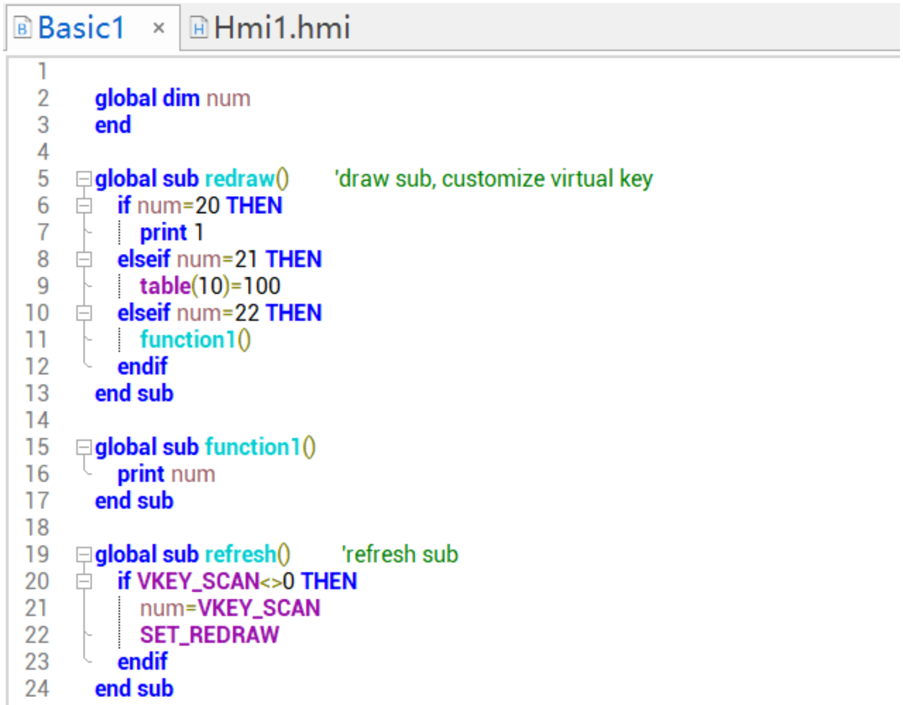
➢ **How to Do?**

- Define virtual keys in BASIC program (take "print value" as the example).
- Bind them through "key trans" function in HMI.
- Use "Custom" component to call refresh sub and draw sub in BASIC program.

- **Basic Program:**

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```

global dim num
end

global sub redraw()      'draw sub, customize virtual key
    if num=20 THEN
        print 1
    elseif num=21 THEN
        table(10)=100
    elseif num=22 THEN
        function1()
    endif
end sub

global sub function1()
    print num
end sub

global sub refresh()      'refresh sub
    if VKEY_SCAN<>0 THEN
        num=VKEY_SCAN
        SET_REDRAW
    endif
end sub
```

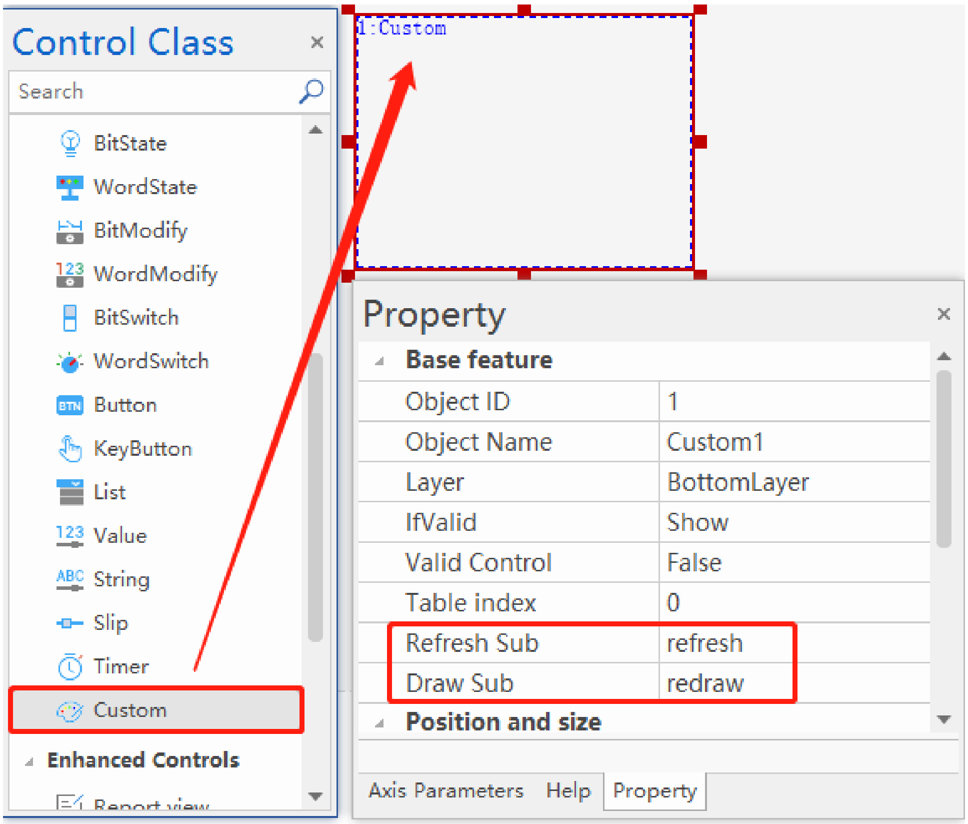
--BASIC Program Codes--

```
global dim num
end


global sub redraw()              'draw sub, customize virtual key
    if num=20 THEN
        print 1
    elseif num=21 THEN
        table(10)=100
    elseif num=22 THEN
        function1()
    endif
end sub


global sub function1()
    print num
```

```
end sub

global sub refresh()                'refresh sub
        if VKEY_SCAN<>0 THEN
                num=VKEY_SCAN
                SET_REDRAW
        endif
end sub
```

- **HMI Interface:**

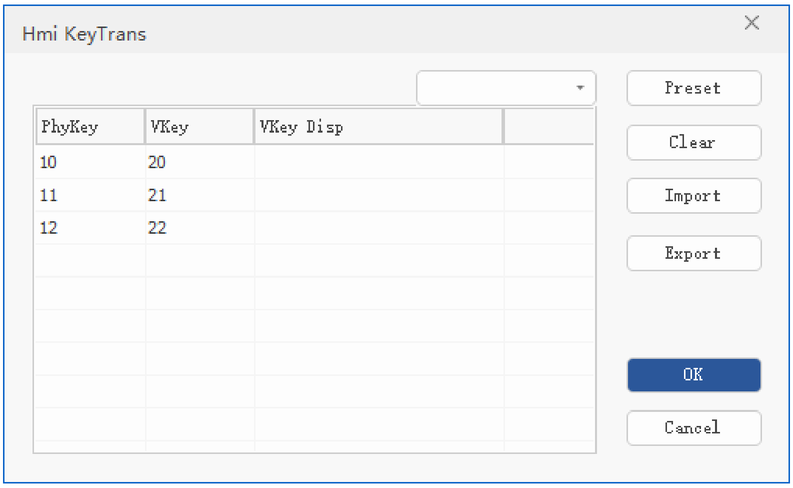
- Step 1: create one HMI file, set auto run task No., and open HMI window.
- Step 2: add one "Custom" component, adjust its size and put it in suitable position, then, in "refresh sub" and "draw sub" (refresh & redraw), call corresponding BASIC sub.



Base feature	
Object ID	1
Object Name	Custom1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Table index	0
Refresh Sub	refresh
Draw Sub	redraw
Position and size	

- Step 3: open "Key Trans" in HMI menu, bind virtual keys 20, 21, 22 with physical keys 11, 12, 13, and click OK.

Note: physical key values are obtained from ZHD400X teach pendant user manual. If you use other external devices, please see corresponding manual.



PhyKey	VKey	VKey Disp
10	20	
11	21	
12	22	

- **Effect:**

When connected controller and teach pendant, then download the program into controller and teach pendant.

- When physical key 11 is pressed, the output window will print "1".
- When physical key 12 is pressed, TABLE (10) value is 100.
- When physical key 13 is pressed, it calls sub function function1, and its function can be customized.

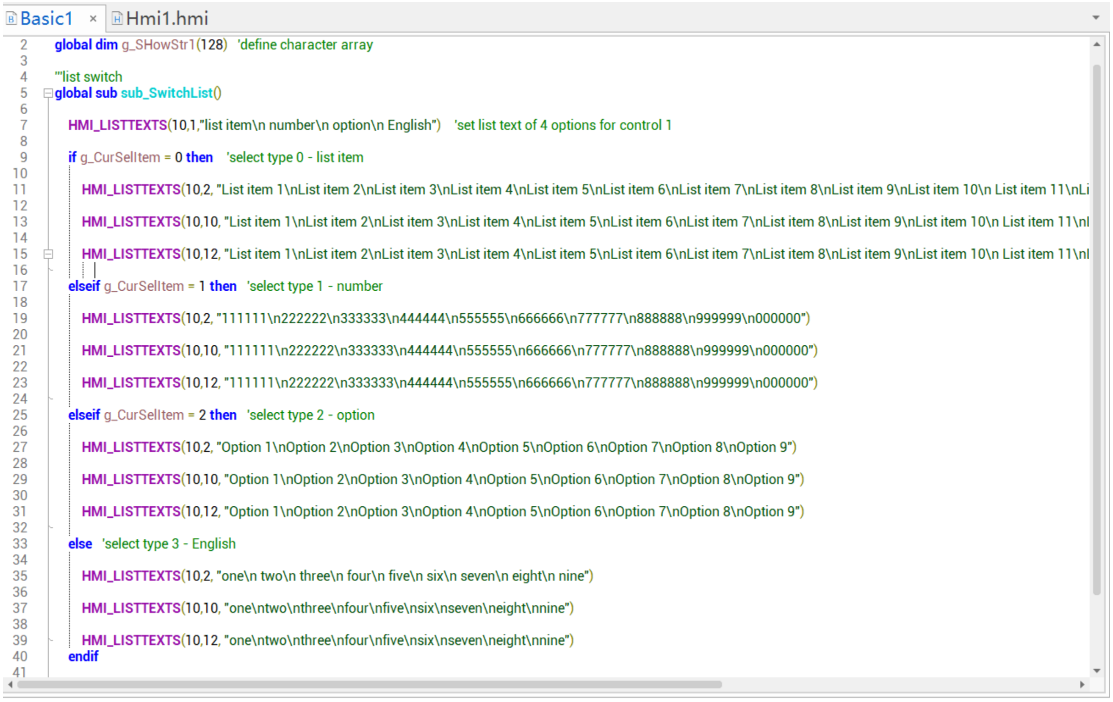
8.3. Dynamic List

This routine is an example of using the "list", and the reference routine is as follows:

- **Target:** match HMI_LISTTEXTS command with "List" component to control dynamic list. That is, the selection in one list can control another list' item, then no need to build multiple list when there are many options.
- **How to Do?**
  - In BASIC file, use HMI_LISTTEXTS command, and define GLOBAL SUB function.
  - Add several "list" component, and bind it with each registers (recommend different lists bind with different registers).
  - Call BASIC SUB function for the first "list" component, then achieve dynamic control.
- **Basic Program:**

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```
global dim g_SHowStr1(128)  'define character array

'''list switch
global sub sub_SwitchList()

  HMI_LISTTEXTS(10,1,"list item\n number\n option\n English")   'set list text of 4 options for control 1

  if g_CurSelItem = 0 then   'select type 0 - list item

    HMI_LISTTEXTS(10,2, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nLi

    HMI_LISTTEXTS(10,10, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nl

    HMI_LISTTEXTS(10,12, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nl

  elseif g_CurSelItem = 1 then  'select type 1 - number

    HMI_LISTTEXTS(10,2, "111111\n222222\n333333\n444444\n555555\n666666\n777777\n888888\n999999\n000000")

    HMI_LISTTEXTS(10,10, "111111\n222222\n333333\n444444\n555555\n666666\n777777\n888888\n999999\n000000")

    HMI_LISTTEXTS(10,12, "111111\n222222\n333333\n444444\n555555\n666666\n777777\n888888\n999999\n000000")

  elseif g_CurSelItem = 2 then  'select type 2 - option

    HMI_LISTTEXTS(10,2, "Option 1\nOption 2\nOption 3\nOption 4\nOption 5\nOption 6\nOption 7\nOption 8\nOption 9")

    HMI_LISTTEXTS(10,10, "Option 1\nOption 2\nOption 3\nOption 4\nOption 5\nOption 6\nOption 7\nOption 8\nOption 9")

    HMI_LISTTEXTS(10,12, "Option 1\nOption 2\nOption 3\nOption 4\nOption 5\nOption 6\nOption 7\nOption 8\nOption 9")

  else  'select type 3 - English

    HMI_LISTTEXTS(10,2, "one\n two\n three\n four\n five\n six\n seven\n eight\n nine")

    HMI_LISTTEXTS(10,10, "one\ntwo\nthree\nfour\nfive\nsix\nseven\neight\nnine")

    HMI_LISTTEXTS(10,12, "one\ntwo\nthree\nfour\nfive\nsix\nseven\neight\nnine")
  endif
```

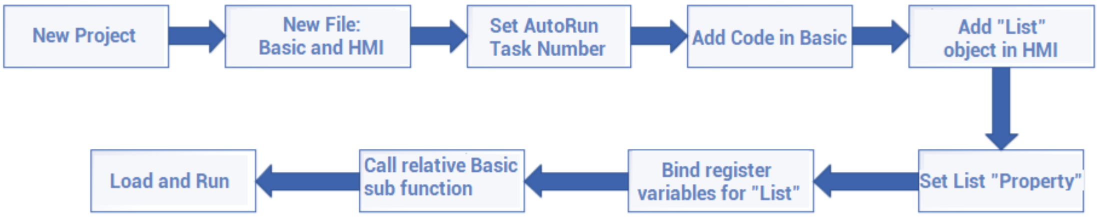
--BASIC Program Codes--

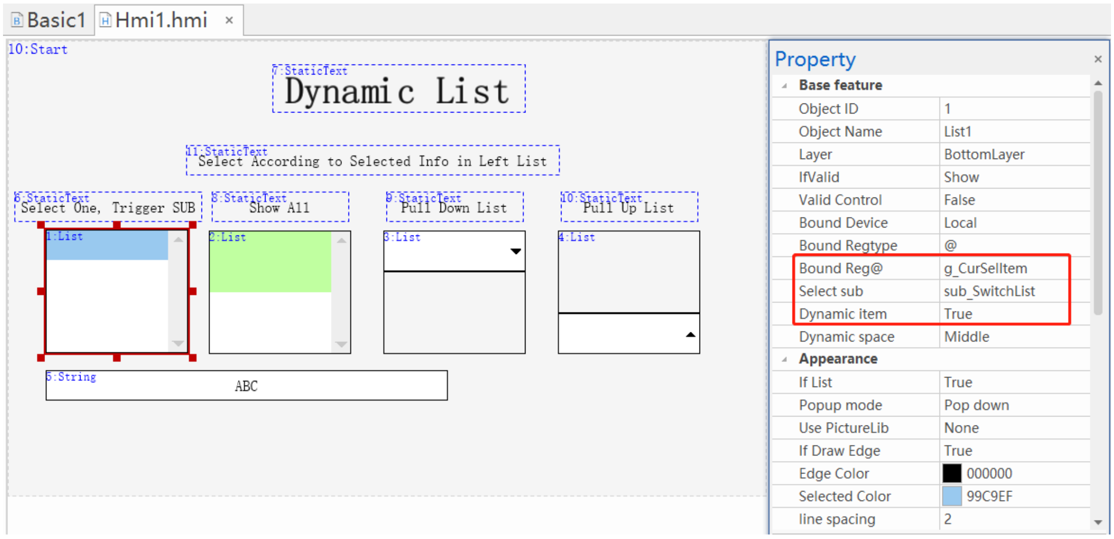
```
    global dim g_CurSelItem         'define list item types, 0-list item, 1-number, 2-option, 3-English
    global dim g_SHowStr1(128)      'define character array


    '''list switch
    global sub sub_SwitchList()


        HMI_LISTTEXTS(10,1,"list item\n number\n option\n English")      'set list text of 4 options for control 1


        if g_CurSelItem = 0 then        'select type 0 - list item


            HMI_LISTTEXTS(10,2, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nList item 12")    'set list text for control 2
            HMI_LISTTEXTS(10,10, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nList item 12")    'set list text for control 10

            HMI_LISTTEXTS(10,12, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nList item 12")    'set list text for control 12

        elseif g_CurSelItem = 1 then      'select type 1 - number

            HMI_LISTTEXTS(10,2, "111111\n222222\n333333\n444444\n555555\n666666\n777777\n888888\n999999\n000000")

            HMI_LISTTEXTS(10,10, "111111\n222222\n333333\n444444\n555555\n666666\n777777\n888888\n999999\n000000")

            HMI_LISTTEXTS(10,12, "111111\n222222\n333333\n444444\n555555\n666666\n777777\n888888\n999999\n000000")

        elseif g_CurSelItem = 2 then      'select type 2 - option

            HMI_LISTTEXTS(10,2, "Option 1\nOption 2\nOption 3\nOption 4\nOption 5\nOption 6\nOption 7\nOption 8\nOption 9")

            HMI_LISTTEXTS(10,10, "Option 1\nOption 2\nOption 3\nOption 4\nOption 5\nOption 6\nOption 7\nOption 8\nOption 9")

            HMI_LISTTEXTS(10,12, "Option 1\nOption 2\nOption 3\nOption 4\nOption 5\nOption 6\nOption 7\nOption 8\nOption 9")

        else      'select type 3 - English

            HMI_LISTTEXTS(10,2, "one\n two\n three\n four\n five\n six\n seven\n eight\n nine")
```

```
                HMI_LISTTEXTS(10,10,
"one\ntwo\nthree\nfour\nfive\nsix\nseven\neight\nnine")

                HMI_LISTTEXTS(10,12,
"one\ntwo\nthree\nfour\nfive\nsix\nseven\neight\nnine")
          endif

          g_SHowStr1 = "trigger selected call sub, select list item"g_CurSelItem+1      'control
3 shows the character string saved in register

     endsub
```

- **HMI Interface:**

  - Create a new HMI file and set the auto run task No.
  - Open the HMI window, select the "list" object, place 4 lists (the number of lists depends on specific needs), and set the properties of each list, such as adjusting the "Selected Color" (the default black will cover the font) , adjust the "If List ": True - display all (pat attention to adjust the height of object), False - pull down/pull up the list. You can adjust the pop-up or pop-down through the "Popup mode".
  - Select the bound register variable for the first list, and call the Basic subfunction to realize the dynamic control of the list, and ensure that the register variables numbers bound to the other two lists are different.
  - Finally, you can create a "text" object to add instructions according to your needs.
  - If you need to use HMI_LISTTEXTS command, please select “true” for “Dynamic Item” in “list” property.

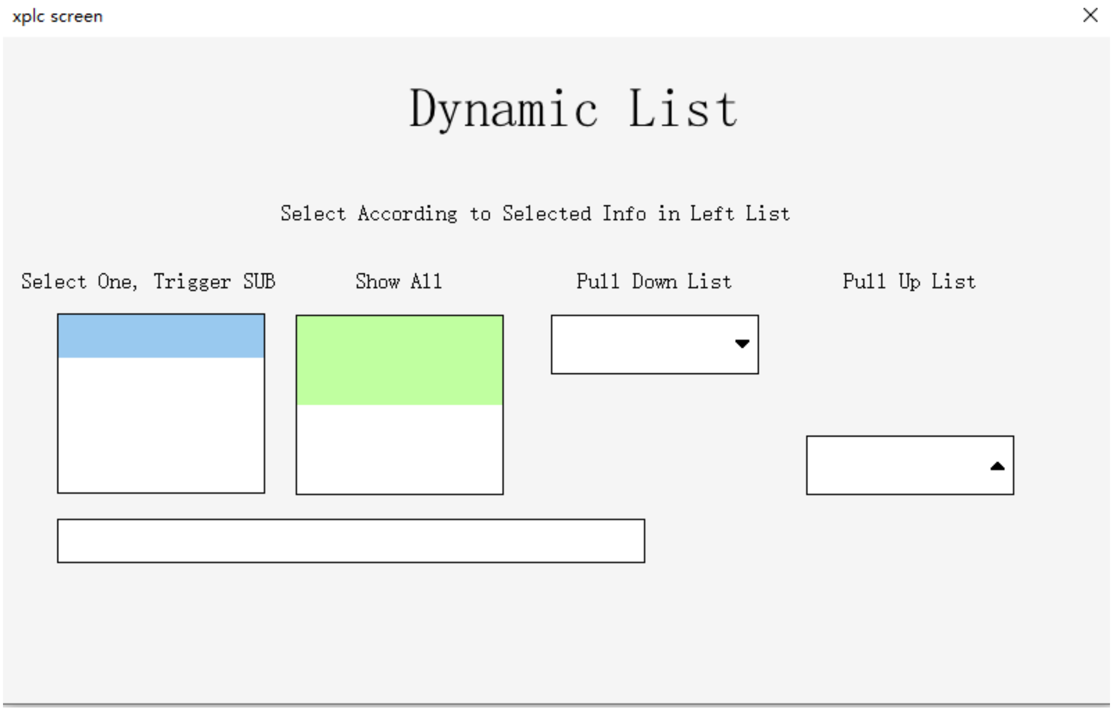




Property

Base feature	
Object ID	1
Object Name	List1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	@
Bound Reg@	g_CurSelItem
Select sub	sub_SwitchList
Dynamic item	True
Dynamic space	Middle
Appearance	
If List	True
Popup mode	Pop down
Use PictureLib	None
If Draw Edge	True
Edge Color	■ 000000
Selected Color	🟦 99C9EF
line spacing	2

- **Effect:**



8.4. View Zoom

This routine is an example of using the “Slip”, and the reference routine is as follows:

- **Target:** zoom in and out the graphic in “custom” component by “slide” object (take rectangle as the example).

- **How to Do?**

- In BASIC file, define rectangle variables, and refresh sub and draw sub.

A. Define Parameter Variables, Set Initial Value

*define slip global zoom variables (g_zoom), set initial value.

*define graphic coordinate array, and assign coordinate initial values for each group of XY.

B. Define Global Refresh Sub

*call refresh sub (reflesh) for “custom” component.

C. Define Global Draw Sub

This routine mainly realizes the reduction and enlargement of the graph with the relationship of multiples.

*Set the local variable parameter (zoom) and g_zoom to get its multiple relationship.

*Set the local variable XY coordinates for drawing graphics, and get the final XY coordinates according to the relationship between the scaling amount and the XY coordinate array.

*Write the graphics commands that need to be drawn.

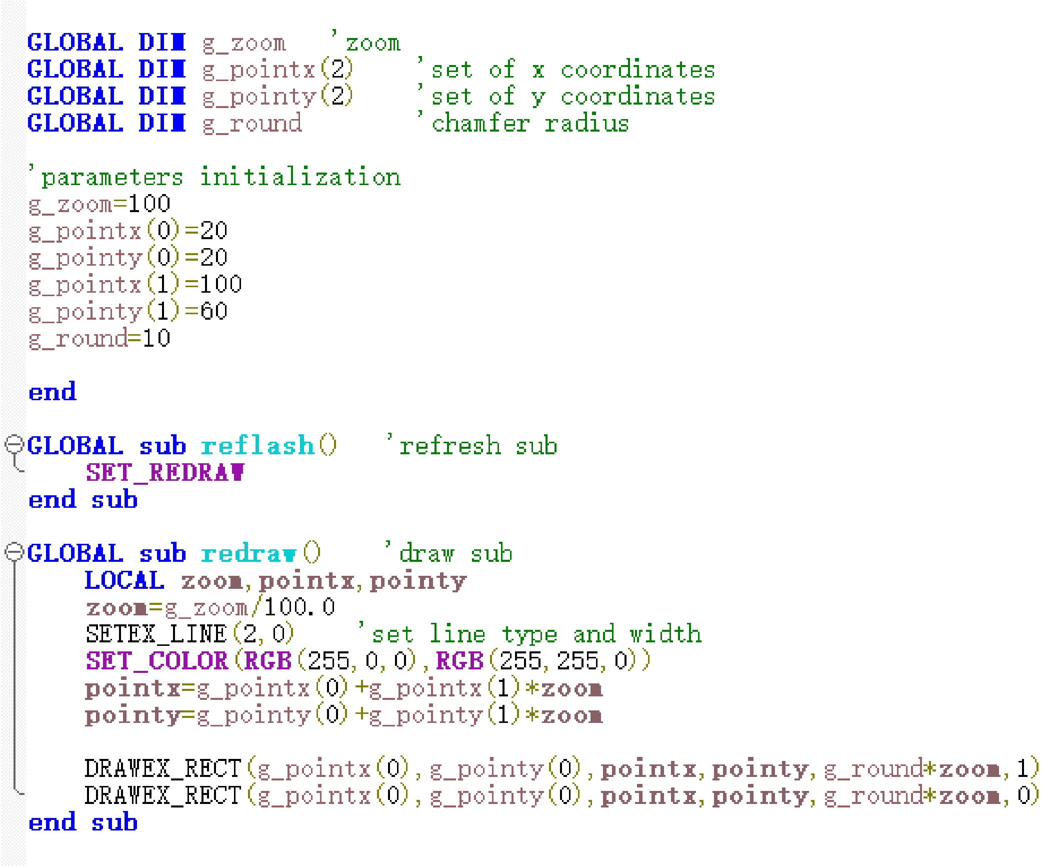
- Add “custom” component, and call BASIC defined refresh sub and draw sub.

- Add “slip” component, and bind with register (g_zoom).

- **Basic Program:**

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```
GLOBAL DIM g_zoom   'zoom
GLOBAL DIM g_pointx(2)    'set of x coordinates
GLOBAL DIM g_pointy(2)    'set of y coordinates
GLOBAL DIM g_round        'chamfer radius

'parameters initialization
g_zoom=100
g_pointx(0)=20
g_pointy(0)=20
g_pointx(1)=100
g_pointy(1)=60
g_round=10

end

GLOBAL sub reflash()    'refresh sub
    SET_REDRAW
end sub

GLOBAL sub redraw()     'draw sub
    LOCAL zoom,pointx,pointy
    zoom=g_zoom/100.0
    SETEX_LINE(2,0)     'set line type and width
    SET_COLOR(RGB(255,0,0),RGB(255,255,0))
    pointx=g_pointx(0)+g_pointx(1)*zoom
    pointy=g_pointy(0)+g_pointy(1)*zoom

    DRAWEX_RECT(g_pointx(0),g_pointy(0),pointx,pointy,g_round*zoom,1)
    DRAWEX_RECT(g_pointx(0),g_pointy(0),pointx,pointy,g_round*zoom,0)
end sub
```

--BASIC Program Codes--

```
GLOBAL DIM g_zoom      'zoom

GLOBAL DIM g_pointx(2)      'set of x coordinates

GLOBAL DIM g_pointy(2)      'set of y coordinates

GLOBAL DIM g_round          'chamfer radius


'parameters initialization

g_zoom=100

g_pointx(0)=20

g_pointy(0)=20

g_pointx(1)=100

g_pointy(1)=60

g_round=10


end


GLOBAL sub reflash()     'refresh sub

     SET_REDRAW
```

```
end sub

GLOBAL sub redraw()       'draw sub
        LOCAL zoom,pointx,pointy
        zoom=g_zoom/100.0
        SETEX_LINE(2,0)       'set line type and width
        SET_COLOR(RGB(255,0,0),RGB(255,255,0))
        pointx=g_pointx(0)+g_pointx(1)*zoom
        pointy=g_pointy(0)+g_pointy(1)*zoom

        DRAWEX_RECT(g_pointx(0),g_pointy(0),pointx,pointy,g_round*zoom,1)
        DRAWEX_RECT(g_pointx(0),g_pointy(0),pointx,pointy,g_round*zoom,0)
end sub
```

- **HMI Interface:**

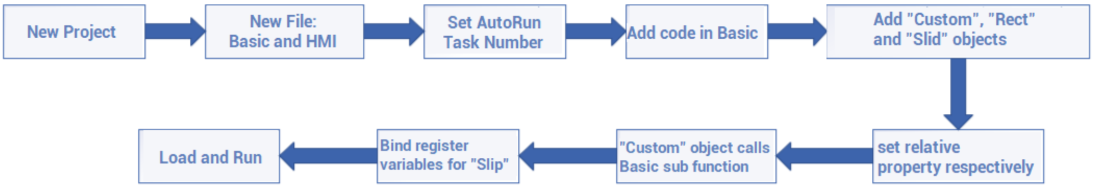
In “custom” object, use the command to draw one rectangle, then you can use the slider to zoom in and out rectangle.

Step 1: create a new HMI file, set the auto-run task number, and open the HMI window.

Step 2: add a "custom” object, adjust its size and place it in an appropriate position, and call the corresponding Basic SUB the refresh sub and draw sub in the "Property".

Step 3: draw a "rect" object with the same size as the "custom” object, and move it to coincide with the position of the "custom” object.

Step 4: draw the "slip" object, adjust the style, size, display value range, etc. Scales can be set through “Main scale number” and “Deputy scale number” in the "Property" (below: main scale is 5, and deputy scale is 1), and put in suitable position (it is recommended to put outside the “custom” object), at the same time, bind register variables.


New Project

New File: Basic and HMI

Set AutoRun Task Number

Add code in Basic

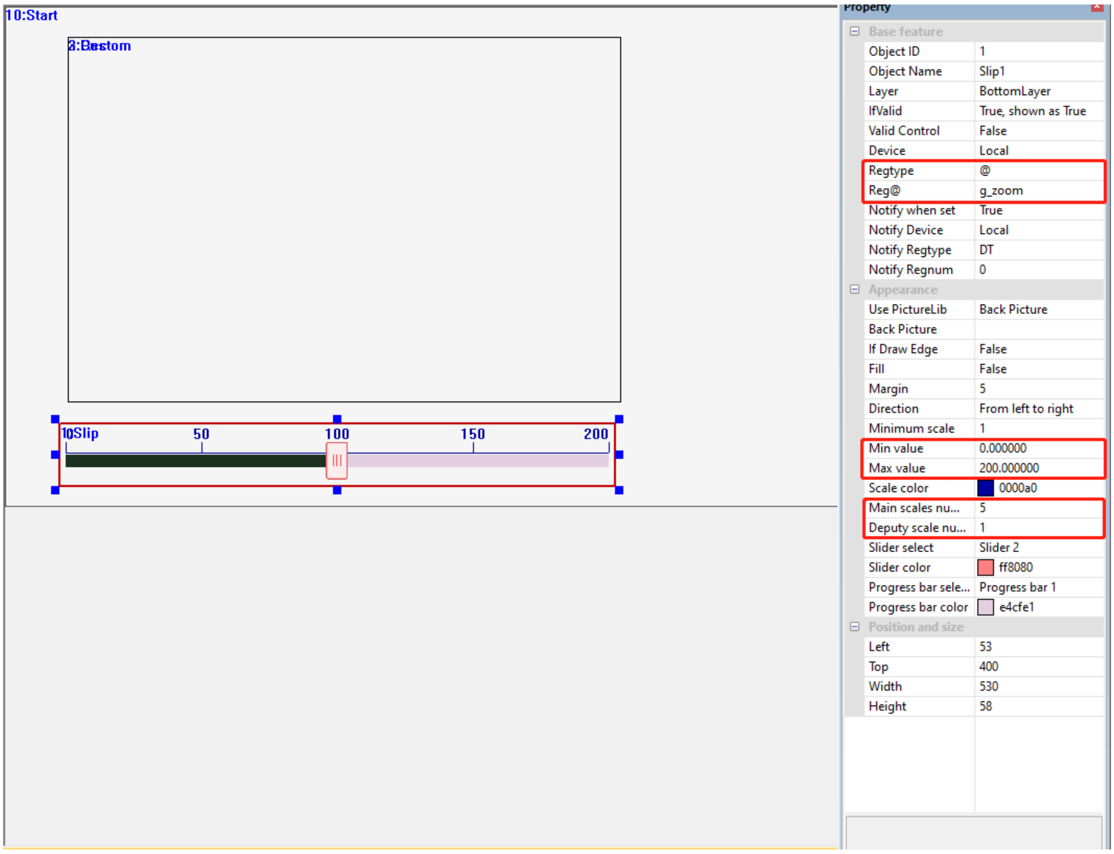
Add "Custom", "Rect" and "Slid" objects

set relative property respectively

"Custom" object calls Basic sub function

Bind register variables for "Slip"

Load and Run

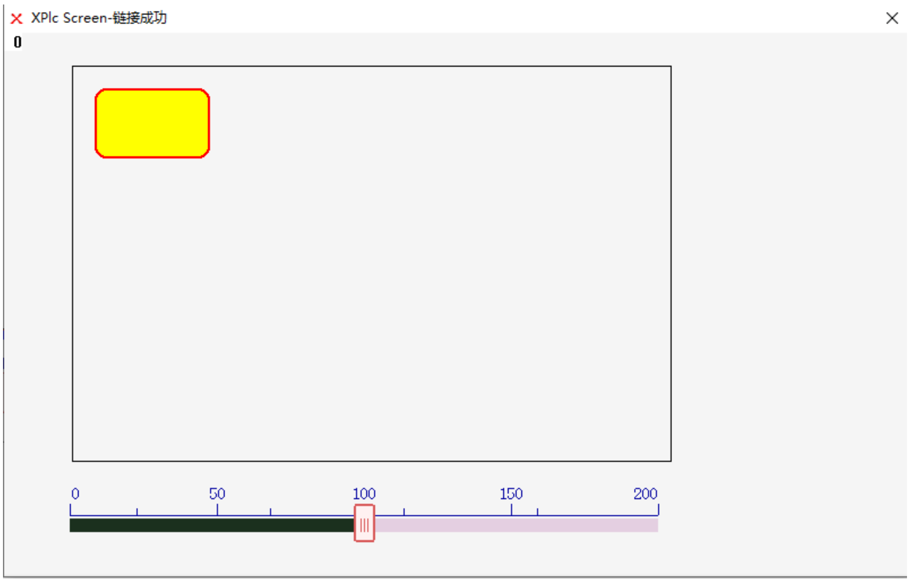


Property

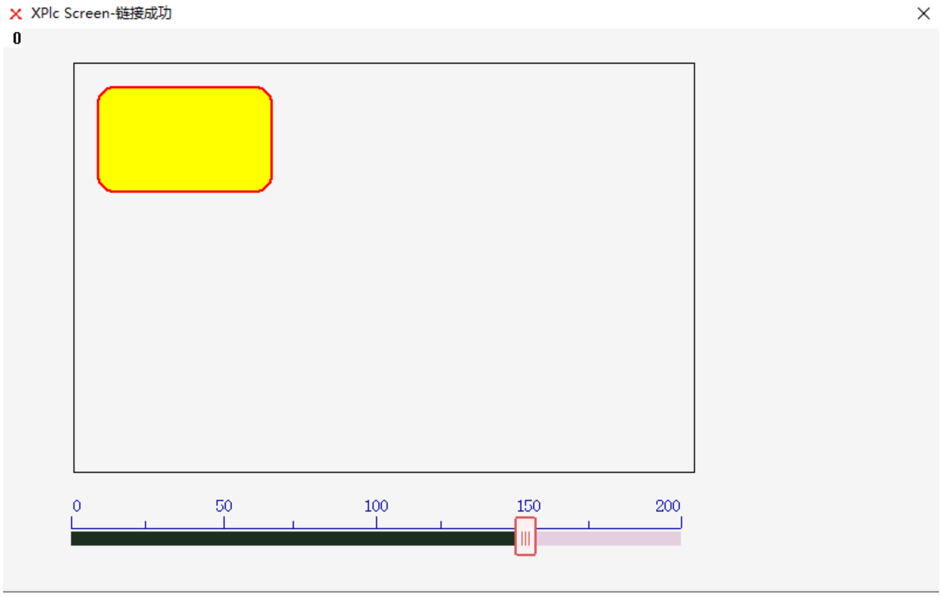
⊟ Base feature	
Object ID	1
Object Name	Slip1
Layer	BottomLayer
IfValid	True, shown as True
Valid Control	False
Device	Local
Regtype	@
Reg@	g_zoom
Notify when set	True
Notify Device	Local
Notify Regtype	DT
Notify Regnum	0
⊟ Appearance	
Use PictureLib	Back Picture
Back Picture	
If Draw Edge	False
Fill	False
Margin	5
Direction	From left to right
Minimum scale	1
Min value	0.000000
Max value	200.000000
Scale color	■ 0000a0
Main scales nu...	5
Deputy scale nu...	1
Slider select	Slider 2
Slider color	🟥 ff8080
Progress bar sele...	Progress bar 1
Progress bar color	□ e4cfe1
⊟ Position and size	
Left	53
Top	400
Width	530
Height	58

- **Effects:**

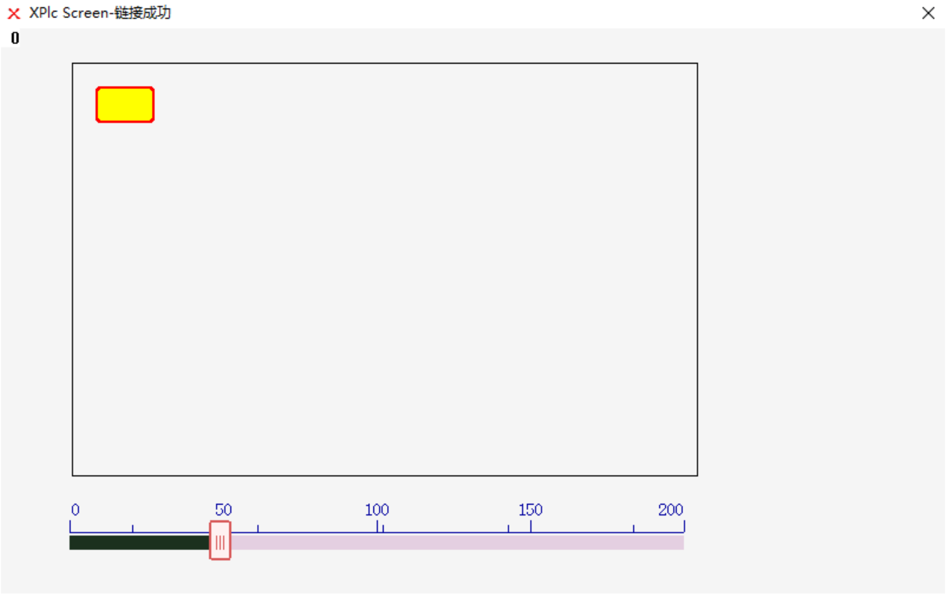
After running, since the initial value of the setting is 100, the slider is in the middle position by default, that is, at the scale of 100, and the initial size of the graph is as shown in the figure.



After dragging the slider to 150, the magnification effect achieved is as shown in the figure.



After dragging the slider to 50, the zoom out effect is as shown in the figure.



8.5. Scroll Bar

This routine is an example of using the “scroll bar”, and the reference routine is as follows:

- **Target:** match scroll bar command with components “custom” & “value” to achieve custom scroll bar and line No. refresh.
- **How to Do?**

- In BASIC file, use scroll bar command to edit program and define global sub function.
- Set Scroll Bar:

  Scroll bar initialization: Use the **SCROLLBAR_INIT** command to determine the No. and position of the scroll bar, set the width, height, color, and total and display lines of the scroll bar.

  Refresh the scroll bar: Use the **SCROLLBAR_RELASH** command to refresh the scroll bar with the corresponding No., and refresh and redraw the scroll bar when changing its position.

  Draw the scroll bar: Use the **SCROLLBAR_DRAW** command to draw the scroll bar.

- Add “custom” component, and call refresh sub and draw sub separately.
- Add several “value” components, bind with register variables, each bound register variable value should plus 1.
- Add “timer” component to regularly refresh.

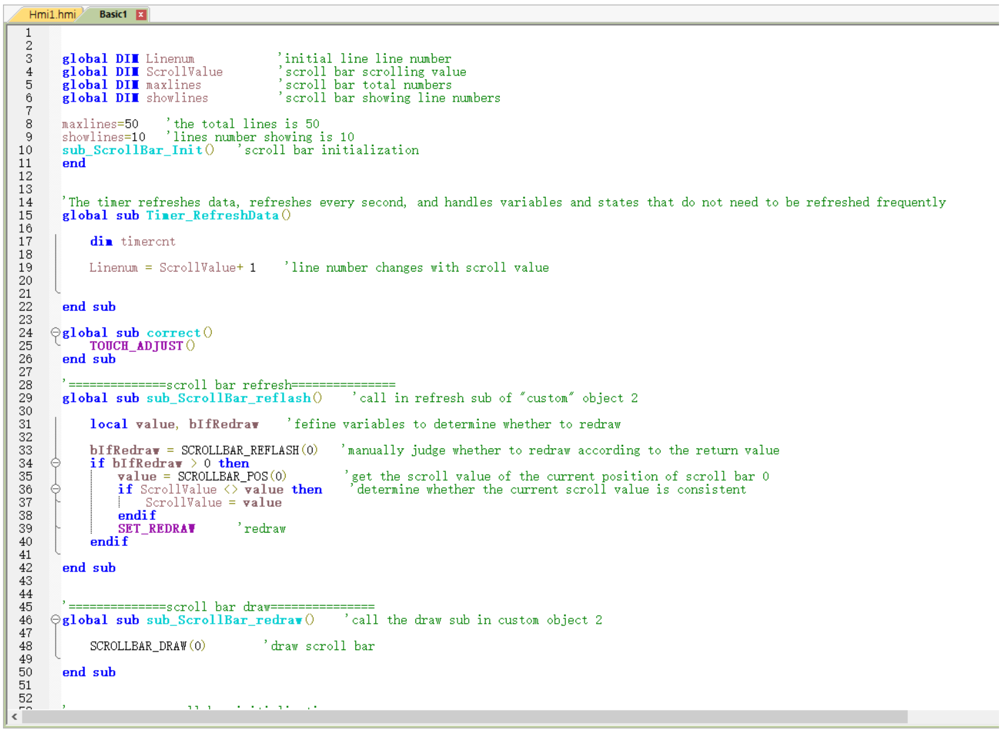
➢ **Applications**

In scenarios such as file management, writing multiple lines of text, or customizing drawing view scaling, you can add scroll bars to allow for more management space. (This example mainly demonstrates the scroll bar in a scenario with multiple lines of text. You can observe changes through line numbers.)

➢ **Basic Program:**

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```


global DIM Linenum           'initial line line number
global DIM ScrollValue       'scroll bar scrolling value
global DIM maxlines          'scroll bar total numbers
global DIM showlines         'scroll bar showing line numbers

maxlines=50    'the total lines is 50
showlines=10   'lines number showing is 10
sub_ScrollBar_Init()    'scroll bar initialization
end


'The timer refreshes data, refreshes every second, and handles variables and states that do not need to be refreshed frequently
global sub Timer_RefreshData()

    dim timercnt

    Linenum = ScrollValue+ 1    'line number changes with scroll value


end sub

global sub correct()
    TOUCH_ADJUST()
end sub

'==============scroll bar refresh==============
global sub sub_ScrollBar_reflash()    'call in refresh sub of "custom" object 2

    local value, bIfRedraw    'fefine variables to determine whether to redraw

    bIfRedraw = SCROLLBAR_REFLASH(0)   'manually judge whether to redraw according to the return value
    if bIfRedraw > 0 then
        value = SCROLLBAR_POS(0)         'get the scroll value of the current position of scroll bar 0
        if ScrollValue <> value then     'determine whether the current scroll value is consistent
            ScrollValue = value
        endif
        SET_REDRAW      'redraw
    endif

end sub


'==============scroll bar draw==============
global sub sub_ScrollBar_redraw()    'call the draw sub in custom object 2

    SCROLLBAR_DRAW(0)        'draw scroll bar

end sub

```

- **--BASIC Program Codes--**

```
global DIM Linenum                    'initial line line number

global DIM ScrollValue            'scroll bar scrolling value

global DIM maxlines                 'scroll bar total numbers

global DIM showlines                'scroll bar showing line numbers


maxlines=50        'the total lines is 50

showlines=10      'lines number showing is 10

sub_ScrollBar_Init()      'scroll bar initialization

end



'The timer refreshes data, refreshes every second, and handles variables and states that do not need to be refreshed frequently

global sub Timer_RefreshData()


        dim timercnt


        Linenum = ScrollValue+ 1        'line number changes with scroll value

```

```

end sub


global sub correct()
    TOUCH_ADJUST()
end sub


'=============scroll bar refresh==============
global sub sub_ScrollBar_reflash()      'call in refresh sub of "custom" object 2


    local value, bIfRedraw      'fefine variables to determine whether to redraw


    bIfRedraw = SCROLLBAR_REFLASH(0)      'manually judge whether to redraw according to the return value
    if bIfRedraw > 0 then
        value = SCROLLBAR_POS(0)      'get the scroll value of the current position of scroll bar 0
        if ScrollValue <> value then      'determine whether the current scroll value is consistent
            ScrollValue = value
        endif
        SET_REDRAW      'redraw
    endif


end sub



'=============scroll bar draw==============
global sub sub_ScrollBar_redraw()      'call the draw sub in custom object 2


    SCROLLBAR_DRAW(0)      'draw scroll bar


end sub


'============scroll bar initialization=============
global sub sub_ScrollBar_Init()

    SCROLLBAR_INIT(0, 378, 0, 22, 350, maxlines, showlines)     'set the width of scroll bar
is 22, the height is 350, total line numbers is 50, and showing line numbers is 10
End sub
```

- **HMI Interface:**

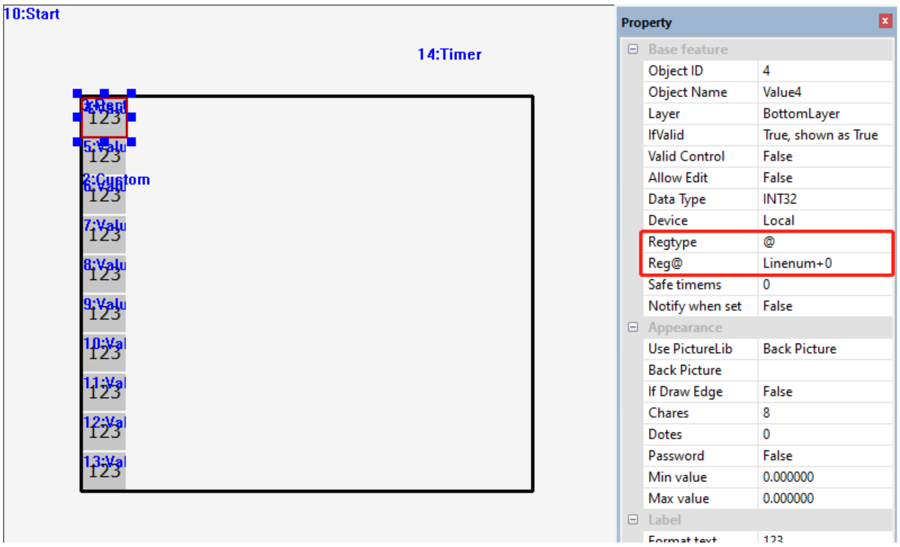
In “custom” component, draw the scroll bar, and use “value” component to show the line No., also use “timer” to refresh the line No.

Step 1: create a new HMI file, set the auto run task No., and open the HMI window.

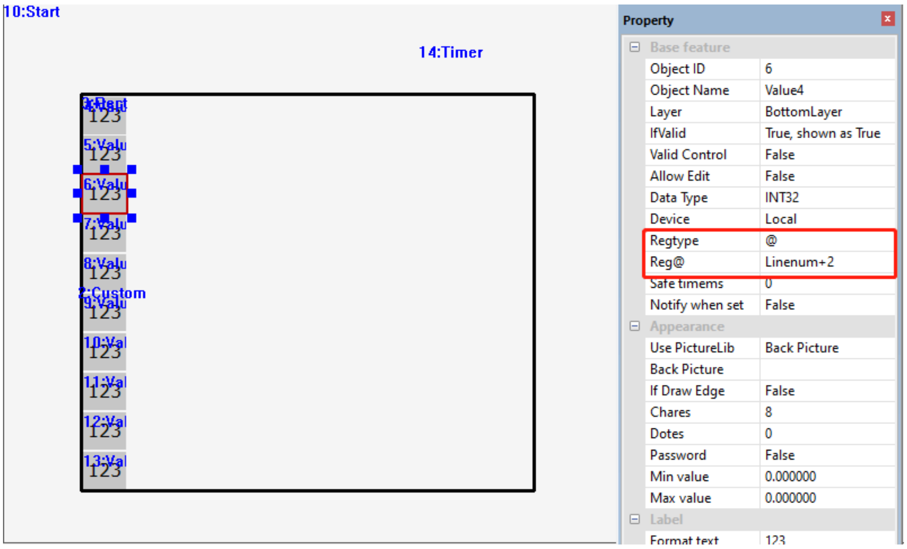
Step 2: add one "custom” object as a background, adjust its size (here, 400*350) to a suitable position, and call corresponding Basic sub function respectively in refresh sub (sub_ScrollBar_reflash()) and draw sub (sub_ScrollBar_redraw()) in “Property” according to needs.

Step 3: add a "Rect" object as a frame, adjust its size to be the same as that of the "custom” object, and place it at the same position as the "custom” object. (to determine the position of the borderless custom object).

Step 4: add multiple "value" components, the number depends on the actual situation (this example adds 10), adjust their sizes and place them in appropriate positions. In "Property", bind each "value" component to the corresponding register variable, and add 1 to the register variables bound to each "value display" component one by one (such as the register variable of the first component +0, the register variable of the second component +1, the register variable of the third component +2, and so on), and then adjust other property contents according to needs. As shown in the figure below.



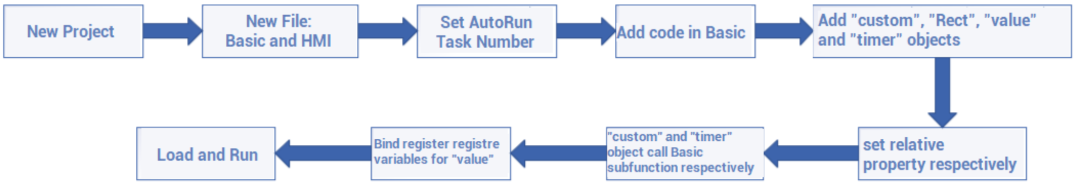
Base feature	
Object ID	4
Object Name	Value4
Layer	BottomLayer
IfValid	True, shown as True
Valid Control	False
Allow Edit	False
Data Type	INT32
Device	Local
Regtype	@
Reg@	Linenum+0
Safe timems	0
Notify when set	False
Appearance	
Use PictureLib	Back Picture
Back Picture	
If Draw Edge	False
Chares	8
Dotes	0
Password	False
Min value	0.000000
Max value	0.000000
Label	
Format text	123



Property

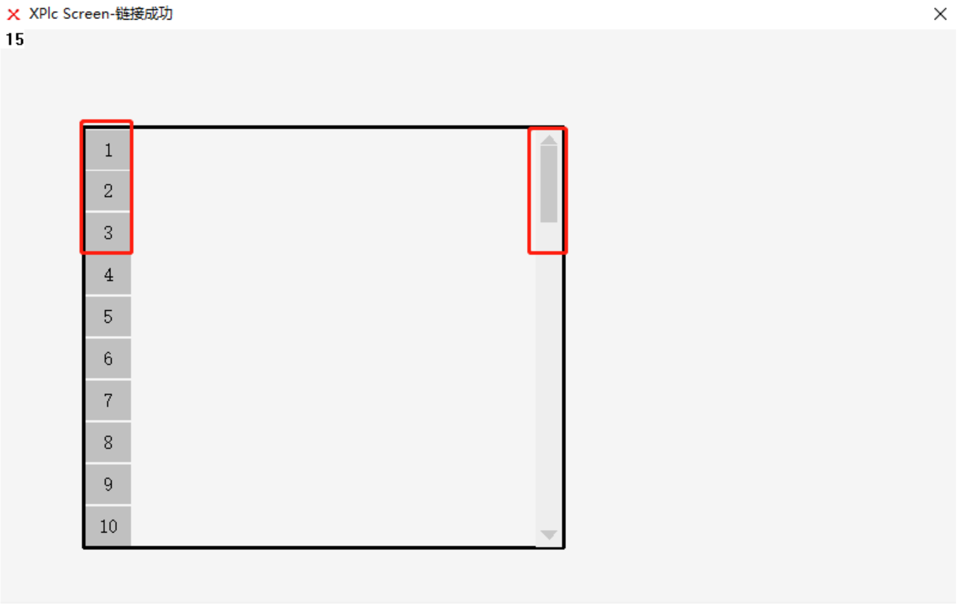
Base feature	
Object ID	6
Object Name	Value4
Layer	BottomLayer
IfValid	True, shown as True
Valid Control	False
Allow Edit	False
Data Type	INT32
Device	Local
Regtype	@
Reg@	Linenum+2
Safe timems	0
Notify when set	False
Appearance	
Use PictureLib	Back Picture
Back Picture	
If Draw Edge	False
Chares	8
Dotes	0
Password	False
Min value	0.000000
Max value	0.000000
Label	
Format text	123

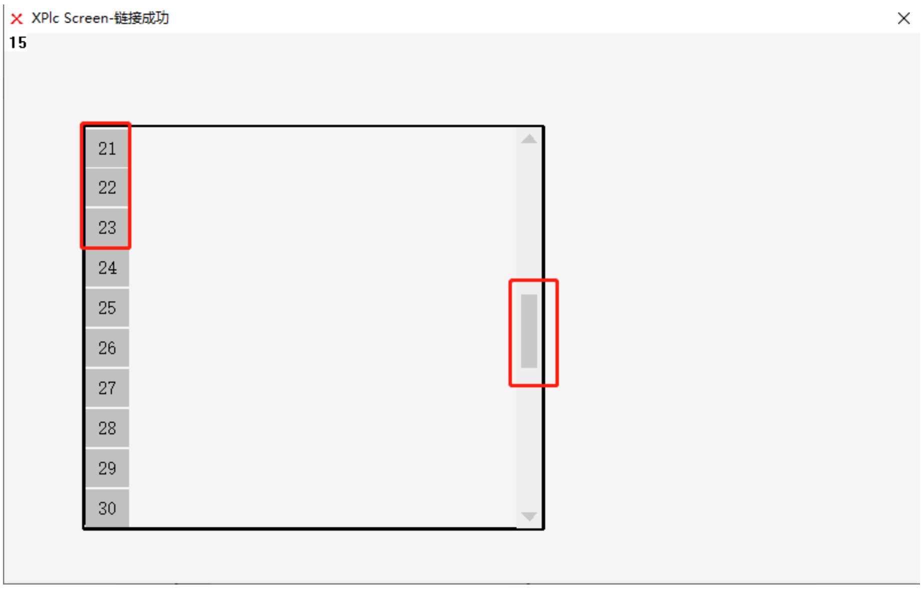
Step 5: add a "timer" object, place it in a suitable position, and adjust the "times ms" in the "property" appropriately. The lower the value, the shorter the trigger period of the timer. Select call sub in "Action" of "Property", and call the Basic subfunction at "Action Function".


New Project
New File: Basic and HMI
Set AutoRun Task Number
Add code in Basic
Add "custom", "Rect", "value" and "timer" objects
set relative property respectively
"custom" and "timer" object call Basic subfunction respectively
Bind register registre variables for "value"
Load and Run

➢ **Effect:**

As the scroll bar scrolls, the line No. will change.





8.6. CAD Function

This only introduces component simple functions, if for full function DEMO, please contact us.

8.6.1. Import Vector Graphic in CAD Component

- **Target:** use CAD command to match "CAD" component with "Button" component to insert and close vector graphics.
- **How to Do?**
  - In BASIC file, use CAD_LOADFILE and CAD_CLOSE commands to edit program and define GLOBAL sub function.

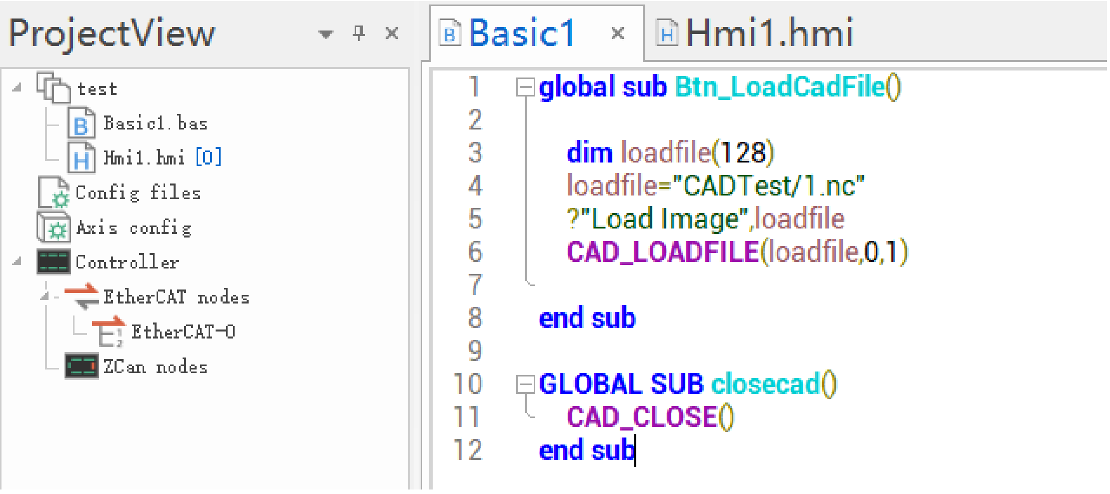
    define "import CAD graphic" function

    "button" component 1 calls action sub, which uses CAD_LOADFILE command to import CAD image.

    "button" component 2 calls action sub, which uses CAD_CLOSE command to close CAD image.
  - Add "CAD" component, set channel No., layer color, etc.
  - Add "Button" component, and in its property, call corresponding sub in "action'.
- **Basic Program:**

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```
global sub Btn_LoadCadFile()

    dim loadfile(128)
    loadfile="CADTest/1.nc"
    ?"Load Image",loadfile
    CAD_LOADFILE(loadfile,0,1)

end sub

GLOBAL SUB closecad()
    CAD_CLOSE()
end sub
```

- **--BASIC Program Codes--**

```
global sub Btn_LoadCadFile()

        dim loadfile(128)
        loadfile="CADTest/1.nc"
        ?"Load Image",loadfile
        CAD_LOADFILE(loadfile,0,1)

end sub

GLOBAL SUB closecad()
        CAD_CLOSE()
end sub
```

- **HMI Interface:**

Step 1: new build one HMI file, set one auto run task No. for it, open HMI window.

Step 2: under controller / simulator flash content, insert one folder that is with vector graphic.

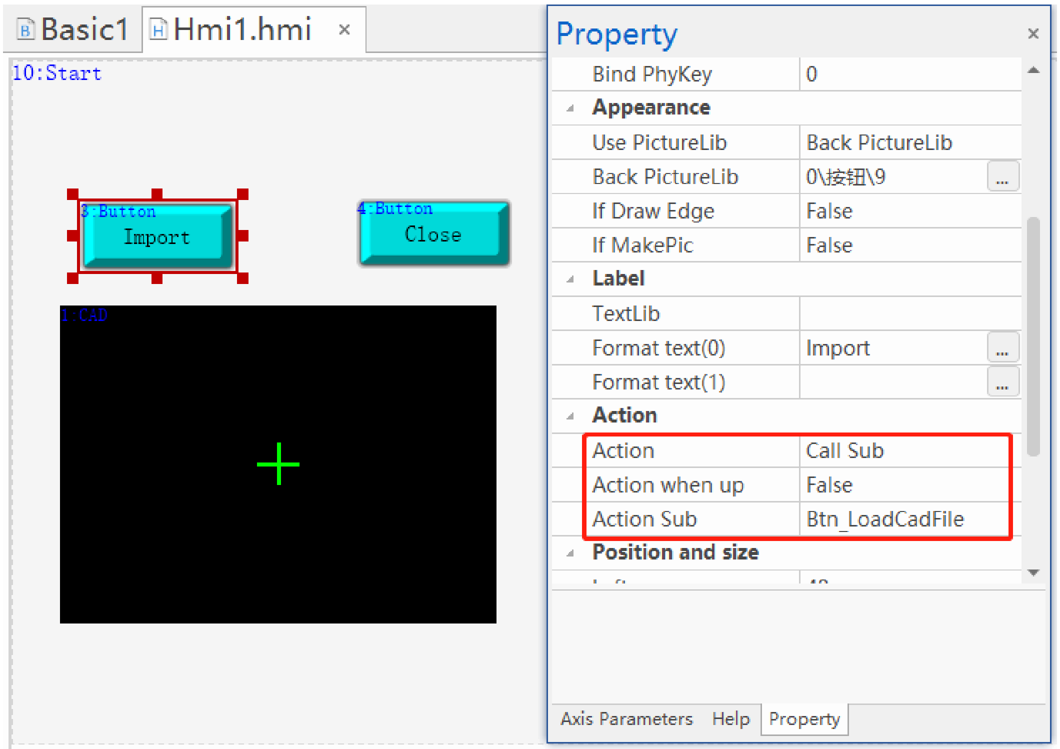
Step 3: add one CAD file, and adjust its size, put it at suitable position.

Step 4: in BASIC file, use CAD_LOADFILE to edit “insert vector graphic” function, set the graphic’s file name with path, import complex curve segment precision, select load layer.

Step 5: in HMI file, add one “Button” file, adjust size, put suitable position, set the format text as “import”, and set CALL SUB for it, the action sub selects the SUB you edited in step 4.

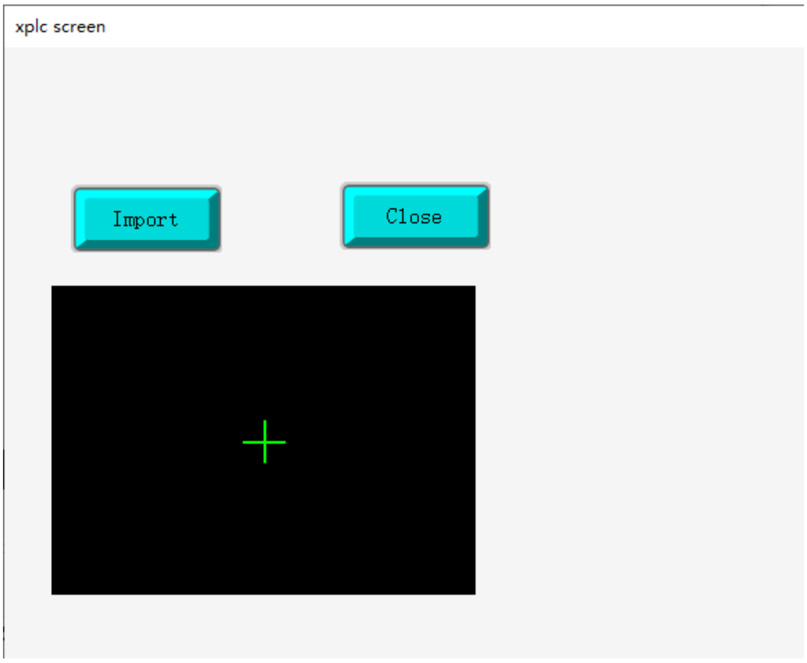
Step 6: in BASIC file, use CAD_CLOSE to edit “close the graphic” function.

Step 7: in HMI, add one “Button” component, adjust size, put into suitable position, set the format text as “close”, and set CALL SUB for it, the action sub selects the SUB you edited in step 6.

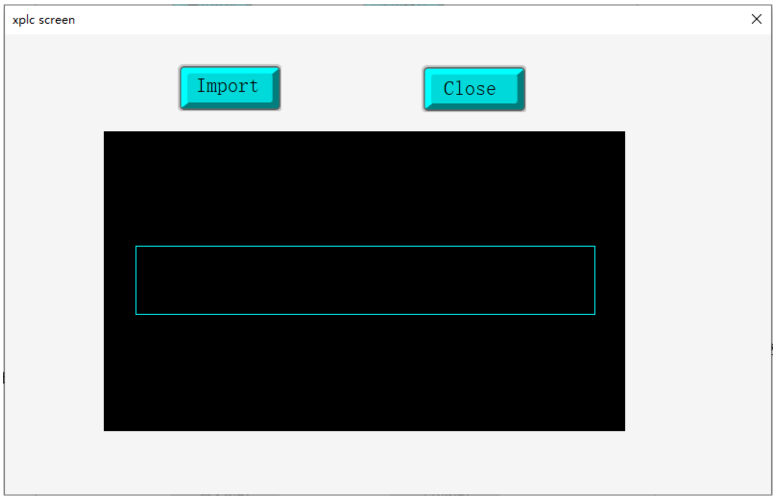


Bind PhyKey	0
Appearance	
Use PictureLib	Back PictureLib
Back PictureLib	0\按钮\9
If Draw Edge	False
If MakePic	False
Label	
TextLib	
Format text(0)	Import
Format text(1)	
Action	
Action	Call Sub
Action when up	False
Action Sub	Btn_LoadCadFile
Position and size	

- **Effect:**



After clicked “import”, it will show imported CAD vector graphic in CAD component, at this time, make the mouse stop at CAD component area, CAD graphic can be zoomed in and out by scroll, also the position can be changed by dragging CAD area.



8.6.2. CAD & File 3 Editor Components

This routine shows how to match CAD component with file 3 file.

- **Target:** based on above routine 8.6.1, import CAD graphic file BASIC codes and G codes into file 3 editor component.

- **How to Do?**

  - In BASIC file, use CAD_TOCODE and FILE3_BUFLOAD commands to edit program and define GLOBAL sub function.

    define "Export CAD Graphic BASIC Code" SUB function

    (1) define array, for saving image path.

    (2) in SUB, use CAD_TOCODE command, and in the command, type=0 means export basic codes.

    (3) use FILE3_BUFLOAD command to load file 3.

    define "Export CAD Graphic G Code" SUB function

    (1) define array, for saving image path.

    (2) in SUB, use CAD_TOCODE command, and in the command, type=1 means export G codes.

    (3) use FILE3_BUFLOAD command to load file 3.

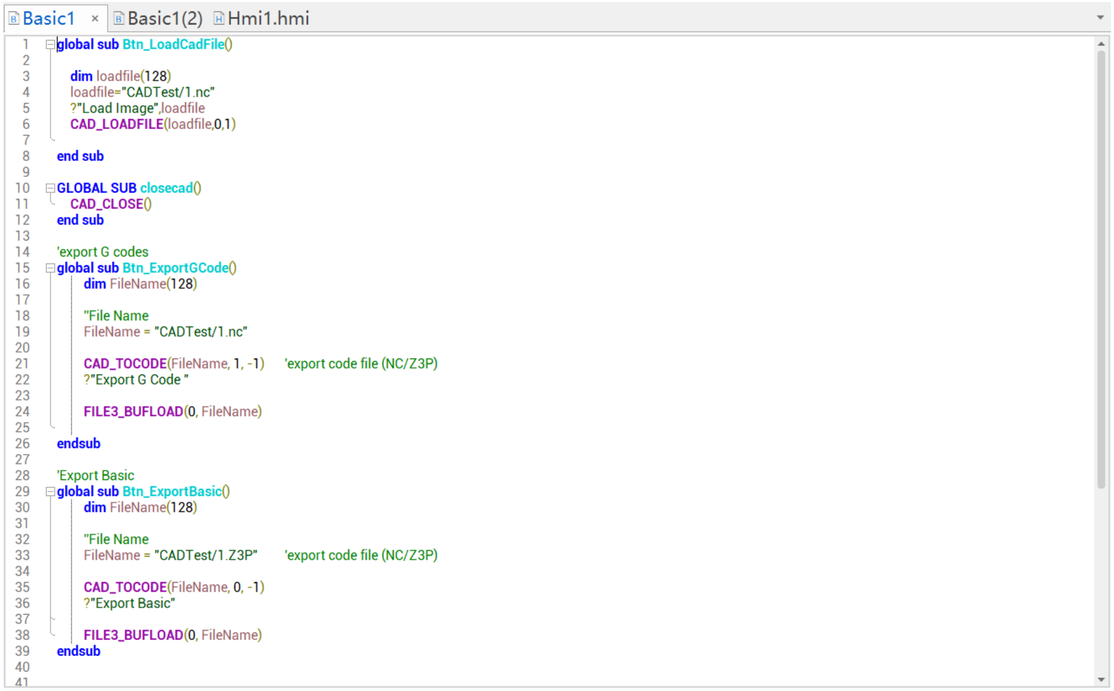
- Add “File3 Editor” component, set channel No., allow edit, etc.

- Add “Button” component, and in its property, call corresponding sub in “action’.

➢ Basic Program:

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```
global sub Btn_LoadCadFile()

    dim loadfile(128)
    loadfile="CADTest/1.nc"
    ?"Load Image",loadfile
    CAD_LOADFILE(loadfile,0,1)

end sub

GLOBAL SUB closecad()
    CAD_CLOSE()
end sub

'export G codes
global sub Btn_ExportGCode()
        dim FileName(128)

        "File Name
        FileName = "CADTest/1.nc"

        CAD_TOCODE(FileName, 1, -1)     'export code file (NC/Z3P)
        ?"Export G Code "

        FILE3_BUFLOAD(0, FileName)

endsub

'Export Basic
global sub Btn_ExportBasic()
        dim FileName(128)

        "File Name
        FileName = "CADTest/1.Z3P"      'export code file (NC/Z3P)

        CAD_TOCODE(FileName, 0, -1)
        ?"Export Basic"

        FILE3_BUFLOAD(0, FileName)
endsub
```

➢ --BASIC Program Codes--

```
global sub Btn_LoadCadFile()

        dim loadfile(128)
        loadfile="CADTest/1.nc"
        ?"Load Image",loadfile
        CAD_LOADFILE(loadfile,0,1)

end sub

GLOBAL SUB closecad()
        CAD_CLOSE()
end sub

'export G codes
global sub Btn_ExportGCode()
```

```
        dim FileName(128)

        ''File Name
        FileName = "CADTest/1.nc"

        CAD_TOCODE(FileName, 1, -1)          'export code file (NC/Z3P)
        ?"Export G Code "

        FILE3_BUFLOAD(0, FileName)

endsub

'Export Basic
global sub Btn_ExportBasic()
        dim FileName(128)

        ''File Name
        FileName = "CADTest/1.Z3P"           'export code file (NC/Z3P)

        CAD_TOCODE(FileName, 0, -1)
        ?"Export Basic"

        FILE3_BUFLOAD(0, FileName)
endsub
```

- **HMI Interface:**

Step 1: after 8.6.1 routine, add one FILE3 EDITOR component, adjust size and put suitable position. and in the editor property, vertical scroll and horizontal scroll are set as TRUE, then it can show full BASIC code and G code.

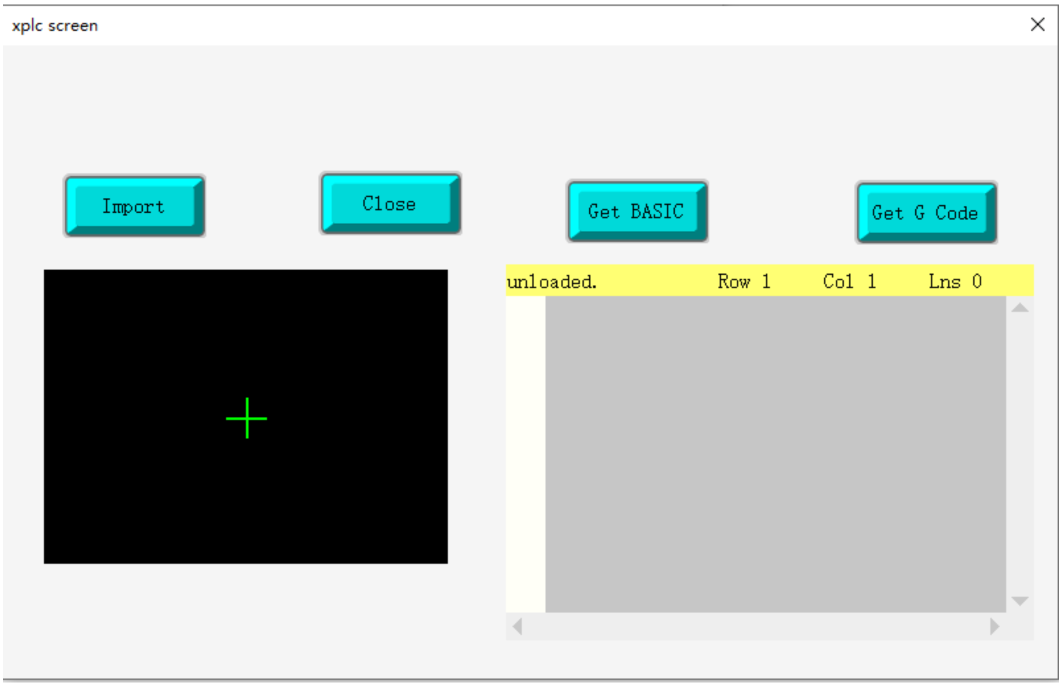
Step 2: In BASIC file, use CAD_TOCODE command to edit “export / get BASIC code” function, also set its code file name, type (type=0), and export layer.

Step 3: add one BUTTON component, adjust size and put suitable position. Set its format text as “get Basic”, sets action as CALL SUB, the action sub should select the SUB name edited in step 2 (here, Btn_ExportBasic).

Step 4: In BASIC file, use CAD_TOCODE command to edit “export / get g code” function, also set its code file name, type (type=1), and export layer.

Step 5: add one BUTTON component, adjust size and put suitable position. Set its format text as “get G code”, sets action as CALL SUB, the action sub should select the SUB name edited in step 4 (here, Btn_ExportBasic).

- **Effect:**



unloaded. Row 1 Col 1 Lns 0

After clicked “import”, it will show imported CAD vector graphic in CAD component, at this time, click “get basic” or “get G code”, file3 editor will show corresponding Basic codes and G codes.

8.6.3. CAD & Custom Components

This routine shows how to match CAD component with custom component.

- **Target:** use CAD_TOTABLE, CAD_LOADTABLE, CAD_DRAW commands with CAD component in custom component to draw CAD graphic.

- **How to Do?**

  - In BASIC file, use CAD_LOADTABLE and CAD_DRAW commands to edit program and define GLOBAL sub function. (CAD_LOADTABLE: load CAD graphic, CAD_TOTABLE: export to TABLE and close CAD, CAD_LOADTABLE: load graphic from TABLE)

    define “draw sub”

    Call Draw sub in “custom” component, use CAD_DRAW command to draw the CAD graphic.

define "refresh sub

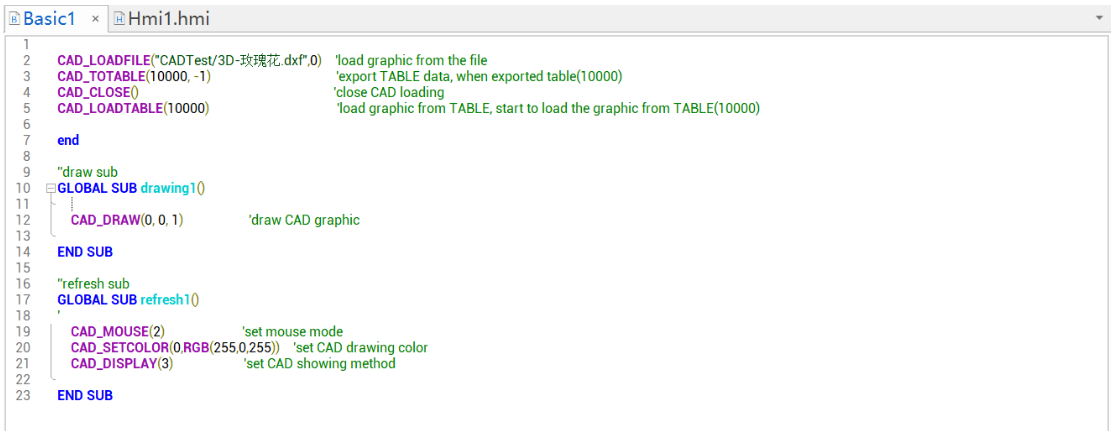
Call refresh sub in "custom" component, use CAD_SETCOLOR and other commands to refresh CAD graphic color, showing method, etc.

- Add "Custom" component, in its property, select corresponding SUB function for refresh sub and draw sub.
- Add "CAD" component, and set channel No., layer color, etc.

- **Basic Program:**

Define variable and build SUB function in BASIC.

After created new project .zpj, new build one BASIC file, set auto run task No., and add below code in BASIC program.



```

CAD_LOADFILE("CADTest/3D-玫瑰花.dxf",0)   'load graphic from the file
CAD_TOTABLE(10000, -1)                   'export TABLE data, when exported table(10000)
CAD_CLOSE()                              'close CAD loading
CAD_LOADTABLE(10000)                     'load graphic from TABLE, start to load the graphic from TABLE(10000)

end

"draw sub
GLOBAL SUB drawing1()

  CAD_DRAW(0, 0, 1)             'draw CAD graphic

END SUB

"refresh sub
GLOBAL SUB refresh1()
'
  CAD_MOUSE(2)                  'set mouse mode
  CAD_SETCOLOR(0,RGB(255,0,255))   'set CAD drawing color
  CAD_DISPLAY(3)                'set CAD showing method

END SUB
```

- **--BASIC Program Codes--**

```
CAD_LOADFILE("CADTest/3D-玫瑰花.dxf",0)      'load graphic from the file
CAD_TOTABLE(10000, -1)                                  'export  TABLE  data,
when exported table(10000)
CAD_CLOSE()                                                        'close
CAD loading
CAD_LOADTABLE(10000)                                 'load  graphic  from
TABLE, start to load the graphic from TABLE(10000)

end

"draw sub
GLOBAL SUB drawing1()

    CAD_DRAW(0, 0, 1)                       'draw CAD graphic

```

```
END SUB

"refresh sub
GLOBAL SUB refresh1()
'
        CAD_MOUSE(2)                              'set mouse mode
        CAD_SETCOLOR(0,RGB(255,0,255))       'set CAD drawing color
        CAD_DISPLAY(3)                             'set CAD showing method

END SUB
```

- **HMI Interface:**

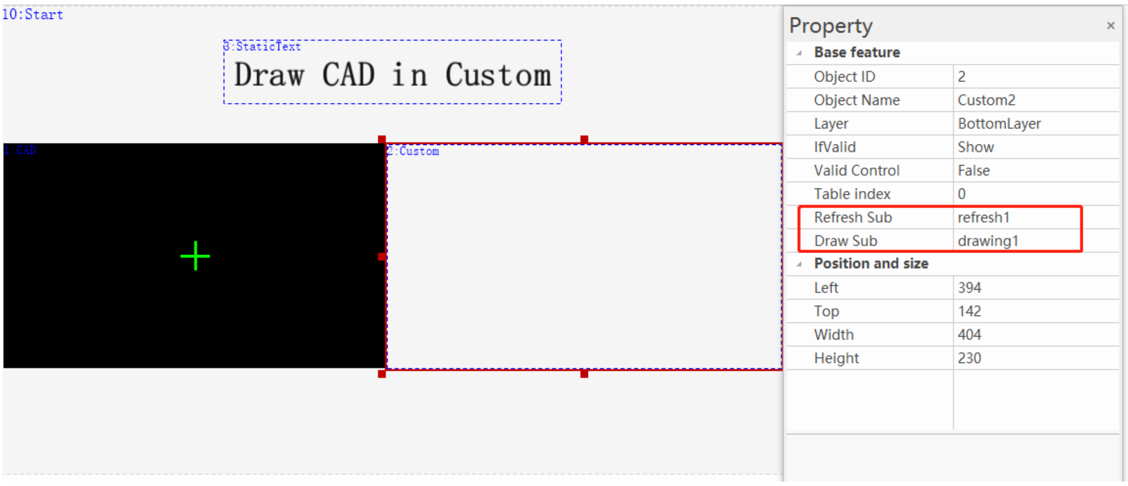
Step 1: new build one HMI file, set one auto run task No. for it, open HMI window.

Step 2: under controller / simulator flash content, insert one folder that is with vector graphic.

Step 3: add one CAD component and one custom component, and adjust size, put at suitable position.

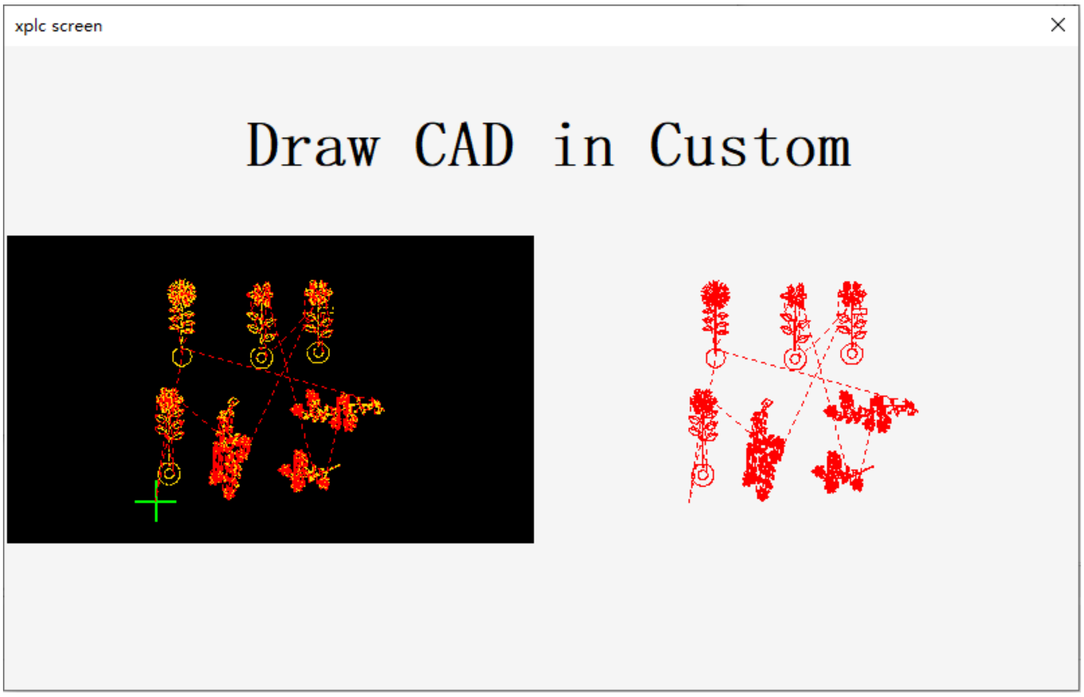
Step 4: in BASIC file, edit refresh sub and draw sub. In this routine, draw sub is drawing1, refresh sub is refresh1.

Step 5: in custom component, call sub functions edited in step 4.



Base feature	
Object ID	2
Object Name	Custom2
Layer	BottomLayer
IfValid	Show
Valid Control	False
Table index	0
Refresh Sub	refresh1
Draw Sub	drawing1
Position and size	
Left	394
Top	142
Width	404
Height	230

- Effect:



8.7. Usage of File Browser

Exact Path: Zmotion – Download – Development Examples – Quick Start – HMI.

This routine simply shows how to use file browser, which includes 2 files, one HMI file and one Basic file, and the HMI file is opened by Basic file, other functions of Basic file is called and executed by HMI. There are several components here.

[Target]

Use HMI_FILEFILTER, HMI_FILESEL, FILE_FILEPATH, HMI_CONTROLATTR commands together with “file browser”, “list”, “string”, “button”, “file browser”, “tree” components,

[What to Do?]

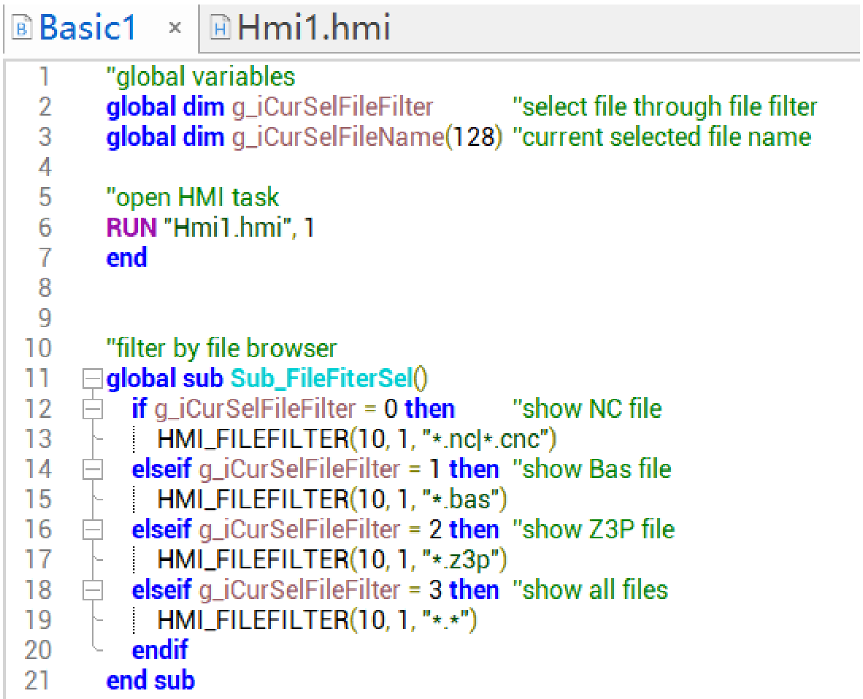
- In Basic file, use HMI_FILEFILTER, FILE_FILESEL, HMI_FILTERPATH commands to edit the program and define global SUB function.
- Use “file browser” component to call Basic SUB function to filter file type and select file.
- Use “list” component to bind with the register that corresponds to file type. That, set “select sub” and needed “state num”, enter corresponding list text.
- Use “string” component to bind with the register that corresponds to file name showing,

- Use “button” component to call BASIC SUB function to open the file.
- Use FILE3_BUFLOAD command to load selected file 3.
- Add “file 3 editor”, set channel No., allow edit properties.
- Add “tree” component, set bound register type and No., then, in “list text”, design tree nodes structure and edit GLOBAL SUB function. According to tree nodes ID, use HMI_CONTROLARRT command to edit global SUB function, then call it in tree property, achieving “select multi-row” in file 3 editor.

[How to Do?]

Step 1: in Basic program, open HMI program, define variables, and build SUB function.

After building the new project .zpj, new build the Basic file, set auto run task No., and add below program in program editing interface.



```
"global variables
global dim g_iCurSelFileFilter          "select file through file filter
global dim g_iCurSelFileName(128)  "current selected file name

"open HMI task
RUN "Hmi1.hmi", 1
end


"filter by file browser
global sub Sub_FileFiterSel()
  if g_iCurSelFileFilter = 0 then        "show NC file
    HMI_FILEFILTER(10, 1, "*.nc|*.cnc")
  elseif g_iCurSelFileFilter = 1 then  "show Bas file
    HMI_FILEFILTER(10, 1, "*.bas")
  elseif g_iCurSelFileFilter = 2 then  "show Z3P file
    HMI_FILEFILTER(10, 1, "*.z3p")
  elseif g_iCurSelFileFilter = 3 then  "show all files
    HMI_FILEFILTER(10, 1, "*.*")
  endif
end sub
```

--Basic Program--

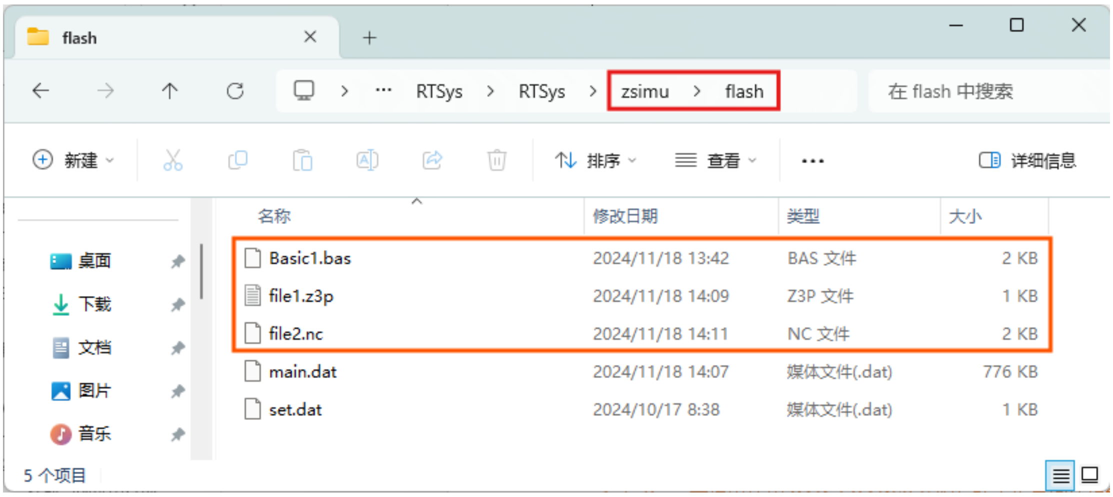
```
"global variables
global dim g_iCurSelFileFilter                    "select file through file filter
global dim g_iCurSelFileName(128)     "current selected file name


"open HMI task
RUN "Hmi1.hmi", 1
```

```
end



"filter by file browser
global sub Sub_FileFiterSel()
    if g_iCurSelFileFilter = 0 then          "show NC file
        HMI_FILEFILTER(10, 1, "*.nc|*.cnc")
    elseif g_iCurSelFileFilter = 1 then      "show Bas file
        HMI_FILEFILTER(10, 1, "*.bas")
    elseif g_iCurSelFileFilter = 2 then      "show Z3P file
        HMI_FILEFILTER(10, 1, "*.z3p")
    elseif g_iCurSelFileFilter = 3 then      "show all files
        HMI_FILEFILTER(10, 1, "*.*")
    endif
end sub

"single-click to select file in file browser
global sub Sub_FileSelect()
    local strCurFileName(128)
    strCurFileName = HMI_FILESEL(10, 1, 0)
    if table(0) = 0 then                                  "select the file
        g_iCurSelFileName = strCurFileName
    endif
end sub

"double click the file in file browser
global sub Sub_FileDlbClick()
    local strCurFileName(128)
    strCurFileName = HMI_FILESEL(10, 1, 0)
    if table(0) = 0 then                                 "select the file, directly open
        g_iCurSelFileName = strCurFileName
        Sub_FileOpen()
    endif
end sub
```

```
"open the file in file browser
global sub Sub_FileOpen()
    HMI_SHOWWINDOW(11,6)
    local strFilePathName(512)
    "when the root directory is "C:/", no need "/"
    if strFilePathName(2) = 47 then
        strFilePathName = HMI_FILEPATH(10, 1) + g_iCurSelFileName
    else
        strFilePathName = HMI_FILEPATH(10, 1) + "/" + g_iCurSelFileName
    endif
    "open file 2 file
    FILE3_BUFLOAD(0, strFilePathName)
    ?"Open File3 File", strFilePathName
end sub
```

Step 2: HMI Interface

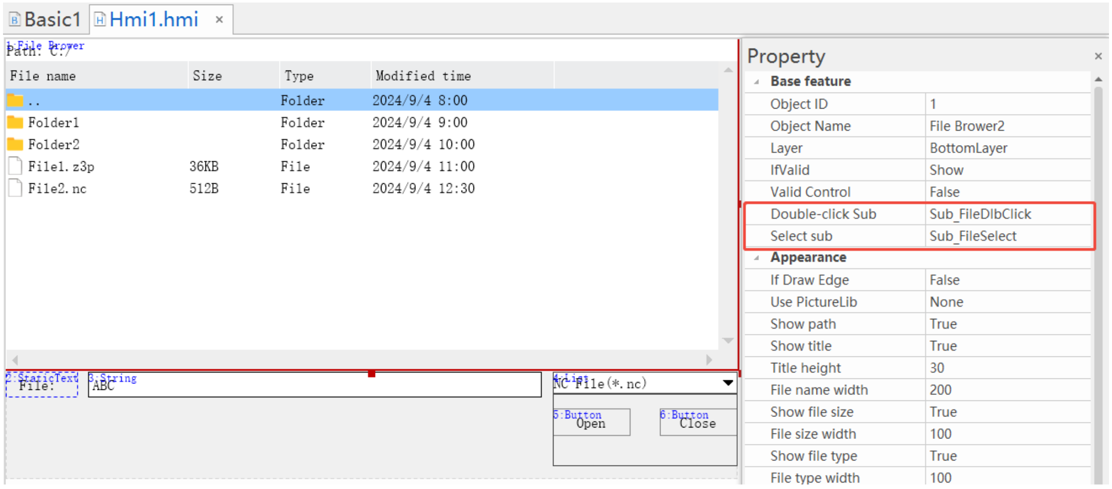
(1) Under controller / simulator flash, insert some files with types of basic, z3p, nc.



名称	修改日期	类型	大小
Basic1.bas	2024/11/18 13:42	BAS 文件	2 KB
file1.z3p	2024/11/18 14:09	Z3P 文件	1 KB
file2.nc	2024/11/18 14:11	NC 文件	2 KB
main.dat	2024/11/18 14:07	媒体文件(.dat)	776 KB
set.dat	2024/10/17 8:38	媒体文件(.dat)	1 KB

(2) Create one HMI file, open HMI base window 10.

(3) In base window 10, add one “file browser” component, select “Double-click sub” and “select sub”, then set other properties as needed.



File name	Size	Type	Modified time
..		Folder	2024/9/4 8:00
Folder1		Folder	2024/9/4 9:00
Folder2		Folder	2024/9/4 10:00
File1.z3p	36KB	File	2024/9/4 11:00
File2.nc	512B	File	2024/9/4 12:30

Property

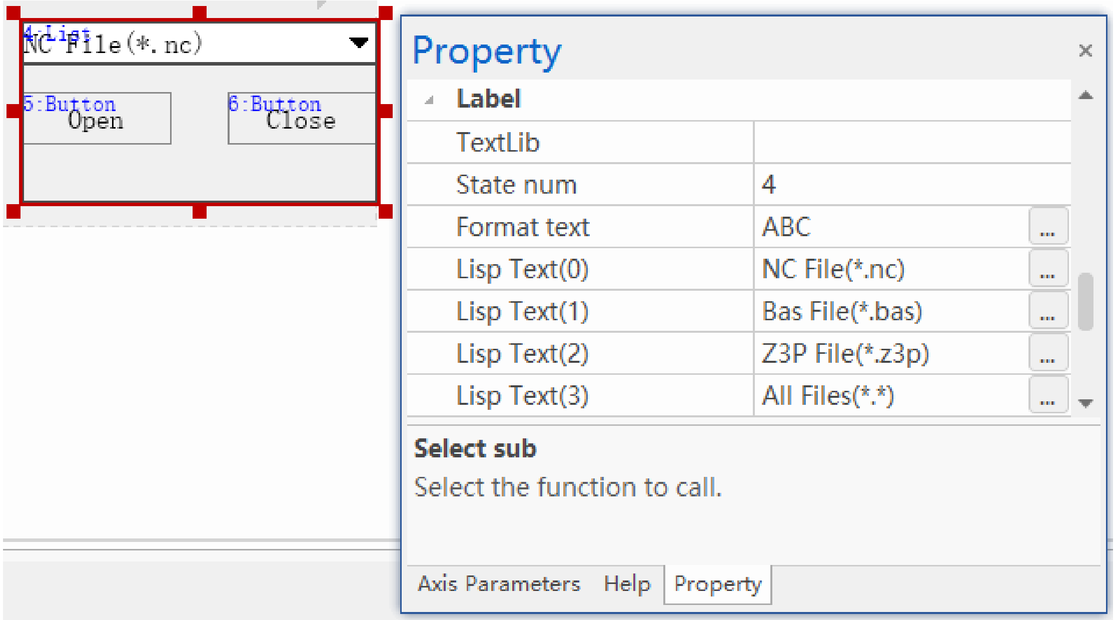
Base feature	
Object ID	1
Object Name	File Brower2
Layer	BottomLayer
IfValid	Show
Valid Control	False
Double-click Sub	Sub_FileDlbClick
Select sub	Sub_FileSelect
Appearance	
If Draw Edge	False
Use PictureLib	None
Show path	True
Show title	True
Title height	30
File name width	200
Show file size	True
File size width	100
Show file type	True
File type width	100

(4) In “file browser” component, add one “static text” component, one “string” component, one “list” component and two “button” components, then, adjust the size, put them at suitable position.

a. Set “static text” component format text as “file: ”.

b. Bind “g_iCurSelFileName” register with “string” component to show selected file name.

c. In “list” component, select “Sub_FileFilterSel” for select sub. And set it as “pop down” for popup mode. Bind “g_iCurSelFileFilter” register, and set the state numbers as 4, each state corresponding list text are set as below:



Property

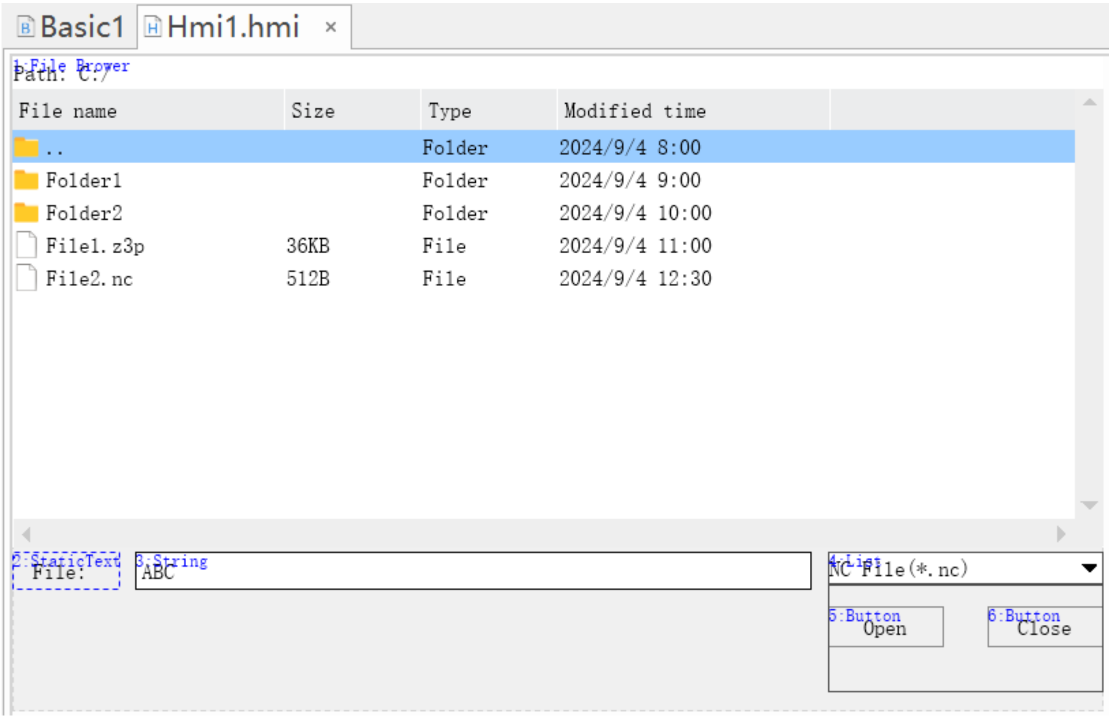
Label	
TextLib	
State num	4
Format text	ABC
Lisp Text(0)	NC File(*.nc)
Lisp Text(1)	Bas File(*.bas)
Lisp Text(2)	Z3P File(*.z3p)
Lisp Text(3)	All Files(*.*)

Select sub

Select the function to call.

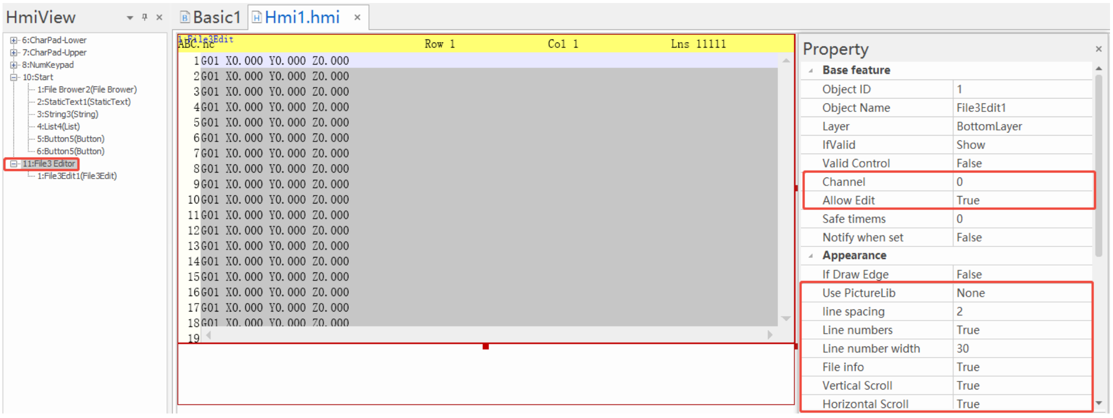
d. Call “Sub_FileOpen” function in “button” component to open file 3, select “close assigned window” action for another “button” to close file 3.

(5) The whole interface of base window 10 is shown as below:



File name	Size	Type	Modified time
..		Folder	2024/9/4 8:00
Folder1		Folder	2024/9/4 9:00
Folder2		Folder	2024/9/4 10:00
File1.z3p	36KB	File	2024/9/4 11:00
File2.nc	512B	File	2024/9/4 12:30

(6) Create one window 11 “popup window”, and add one “file3 editor” component, set the channel No., allow edit, and other properties. Use FILE3_BUFLOAD command to load assigned file3 into disk.

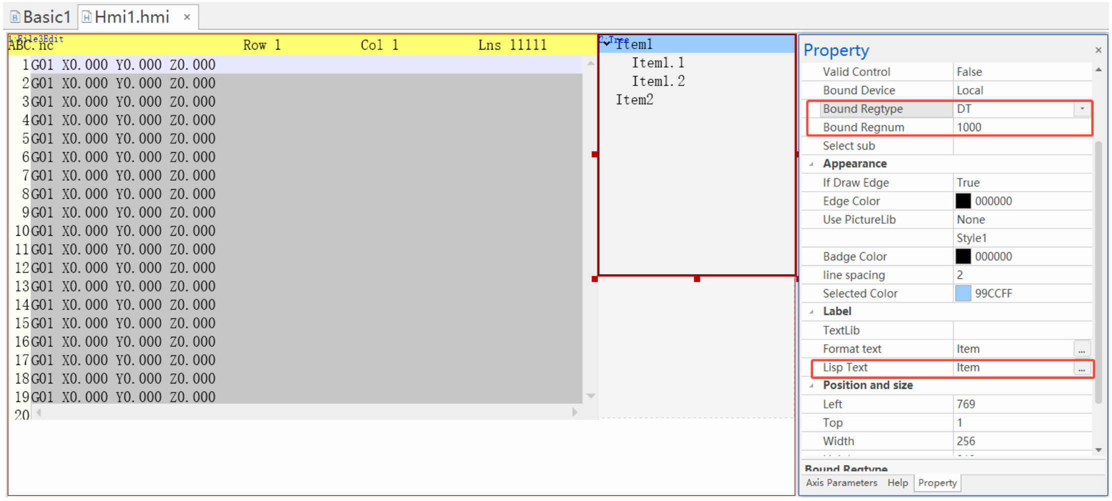


```
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
G01 X0.000 Y0.000 Z0.000
```

Property

Base feature	
Object ID	1
Object Name	File3Edit1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Channel	0
Allow Edit	True
Safe timems	0
Notify when set	False
Appearance	
If Draw Edge	False
Use PictureLib	None
line spacing	2
Line numbers	True
Line number width	30
File info	True
Vertical Scroll	True
Horizontal Scroll	True

(7) In window 11, add one “tree” component, set bound register type and No., then in list text, design tree nodes and edit the content.

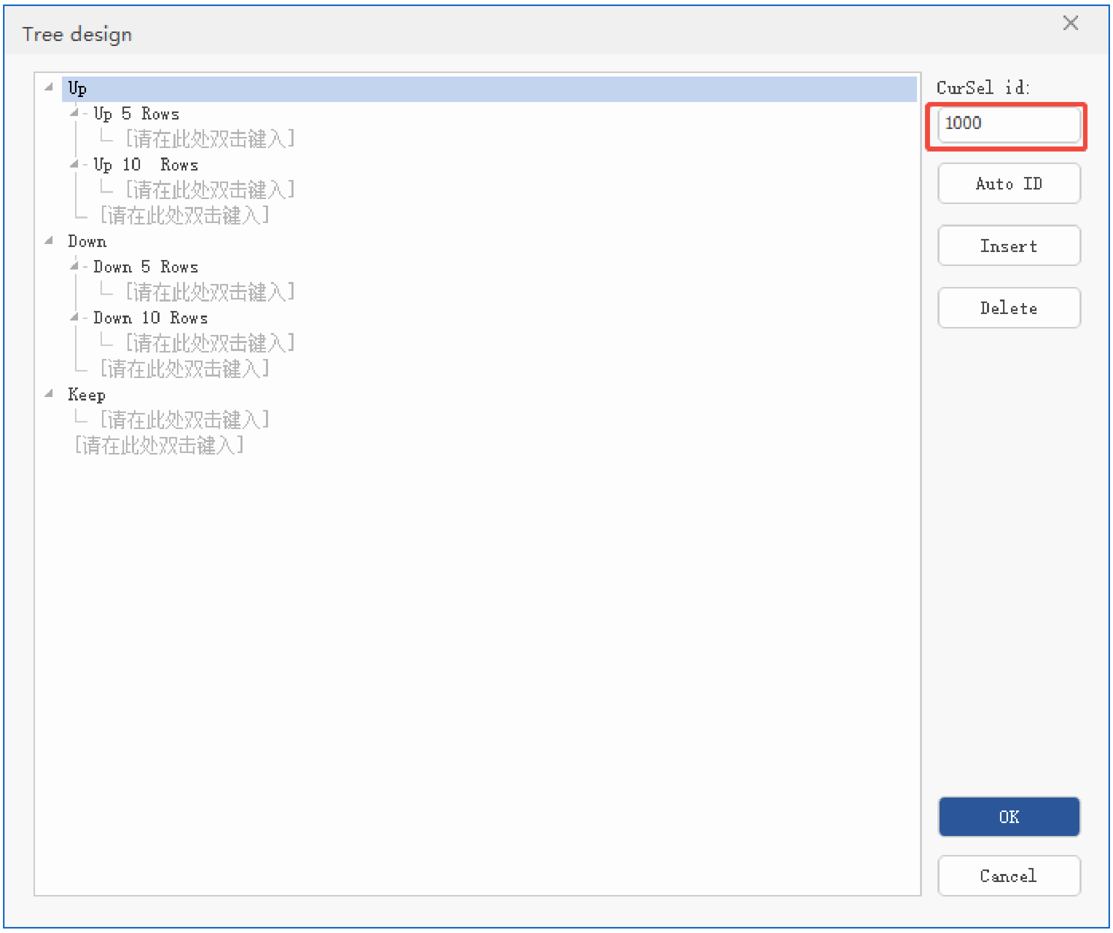


```
 1G01 X0.000 Y0.000 Z0.000
 2G01 X0.000 Y0.000 Z0.000
 3G01 X0.000 Y0.000 Z0.000
 4G01 X0.000 Y0.000 Z0.000
 5G01 X0.000 Y0.000 Z0.000
 6G01 X0.000 Y0.000 Z0.000
 7G01 X0.000 Y0.000 Z0.000
 8G01 X0.000 Y0.000 Z0.000
 9G01 X0.000 Y0.000 Z0.000
10G01 X0.000 Y0.000 Z0.000
11G01 X0.000 Y0.000 Z0.000
12G01 X0.000 Y0.000 Z0.000
13G01 X0.000 Y0.000 Z0.000
14G01 X0.000 Y0.000 Z0.000
15G01 X0.000 Y0.000 Z0.000
16G01 X0.000 Y0.000 Z0.000
17G01 X0.000 Y0.000 Z0.000
18G01 X0.000 Y0.000 Z0.000
19G01 X0.000 Y0.000 Z0.000
20
```

Property

Valid Control	False
Bound Device	Local
Bound Regtype	DT
Bound Regnum	1000
Select sub	
Appearance	
If Draw Edge	True
Edge Color	■ 000000
Use PictureLib	None
	Style1
Badge Color	■ 000000
line spacing	2
Selected Color	■ 99CCFF
Label	
TextLib	
Format text	Item
Lisp Text	Item
Position and size	
Left	769
Top	1
Width	256

According to tree node ID, in Basic, use HMI_CONTROLATTR command to edit SUB_TREE function, then in tree component property, call this function in select sub to select several lines for file3 editor.



CurSel id: 1000

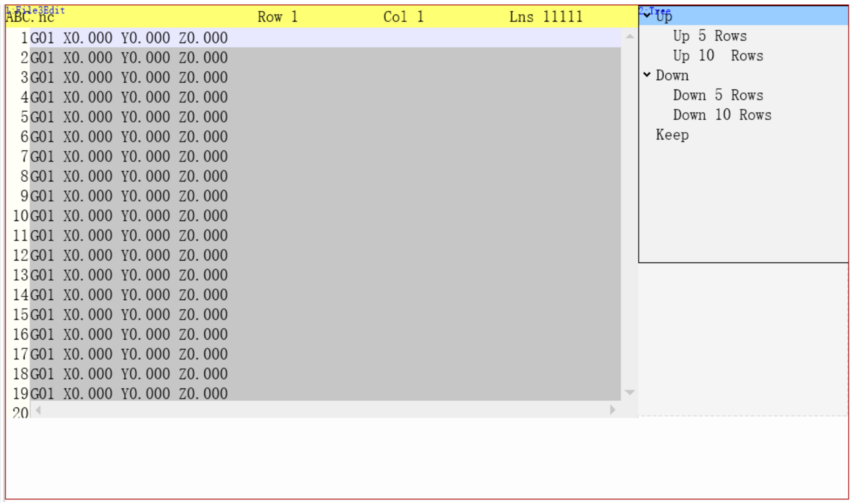
```
Up
  Up 5 Rows
    [请在此处双击键入]
  Up 10  Rows
    [请在此处双击键入]
  [请在此处双击键入]
Down
  Down 5 Rows
    [请在此处双击键入]
  Down 10 Rows
    [请在此处双击键入]
  [请在此处双击键入]
Keep
  [请在此处双击键入]
[请在此处双击键入]
```

Corresponding BASIC Code:

```
"tree & HMI_CONTROLATTR -- achieve "select multi-row" of file3 editor
global sub sub_tree ()
```

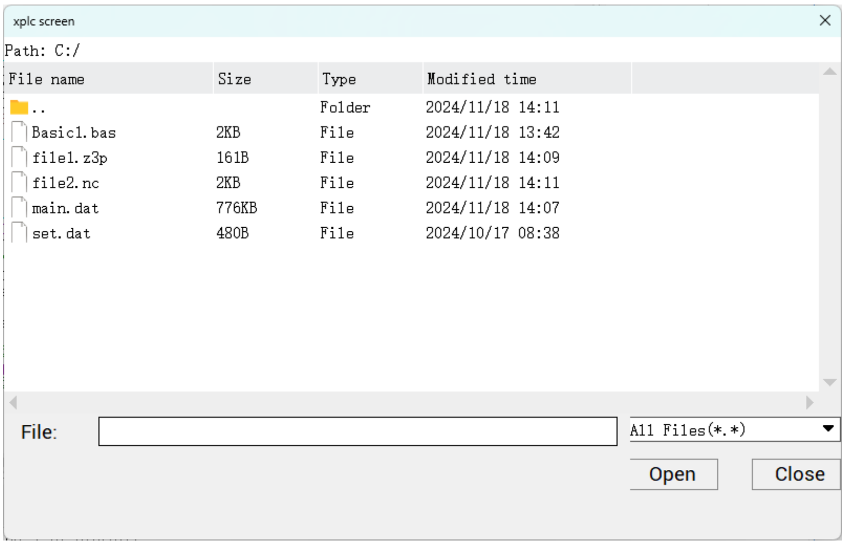
```
    if TABLE(1000) = 1001 THEN
        HMI_CONTROLATTR ("MSELROWS", -5, 11, 1)        'up to select 5 rows
    elseif TABLE (1000) = 1002 THEN
        HMI_CONTROLATTR ("MSELROWS", -10, 11, 1)      'up to select 10 rows
    elseif TABLE(1000) = 1004 THEN
        HMI_CONTROLATTR ("MSELROWS", 5, 11, 1)         'down to select 5 rows
    elseif TABLE (1000) = 1005 THEN
        HMI_CONTROLATTR ("MSELROWS", 10, 11, 1)       'down to select 10 rows
    elseif TABLE (1000) = 1006 THEN
        HMI_CONTROLATTR ("MSELROWS", 0, 11, 1)         'keep now row
    endif
end sub
```

(8) The whole interface of base window 11:



```
ABC.nc                    Row 1        Col 1        Lns 11111
 1G01 X0.000 Y0.000 Z0.000
 2G01 X0.000 Y0.000 Z0.000
 3G01 X0.000 Y0.000 Z0.000
 4G01 X0.000 Y0.000 Z0.000
 5G01 X0.000 Y0.000 Z0.000
 6G01 X0.000 Y0.000 Z0.000
 7G01 X0.000 Y0.000 Z0.000
 8G01 X0.000 Y0.000 Z0.000
 9G01 X0.000 Y0.000 Z0.000
10G01 X0.000 Y0.000 Z0.000
11G01 X0.000 Y0.000 Z0.000
12G01 X0.000 Y0.000 Z0.000
13G01 X0.000 Y0.000 Z0.000
14G01 X0.000 Y0.000 Z0.000
15G01 X0.000 Y0.000 Z0.000
16G01 X0.000 Y0.000 Z0.000
17G01 X0.000 Y0.000 Z0.000
18G01 X0.000 Y0.000 Z0.000
19G01 X0.000 Y0.000 Z0.000
20
```

➢ Running effect:

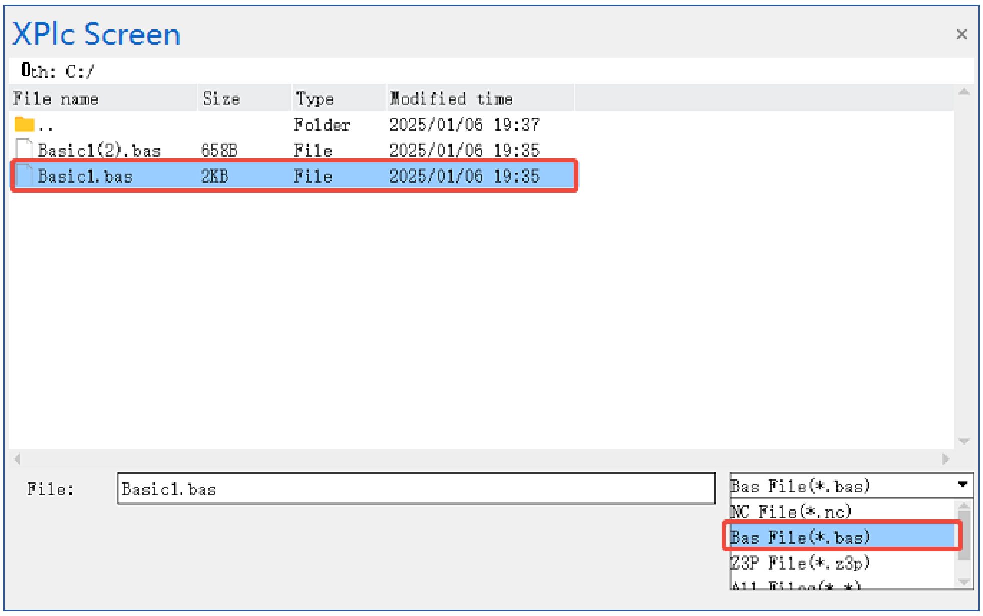


xplc screen

Path: C:/

File name	Size	Type	Modified time
..		Folder	2024/11/18 14:11
Basic1.bas	2KB	File	2024/11/18 13:42
file1.z3p	161B	File	2024/11/18 14:09
file2.nc	2KB	File	2024/11/18 14:11
main.dat	776KB	File	2024/11/18 14:07
set.dat	480B	File	2024/10/17 08:38

- Click "list" component, select needed file type, then in "file browser", it will show corresponding file types' files.



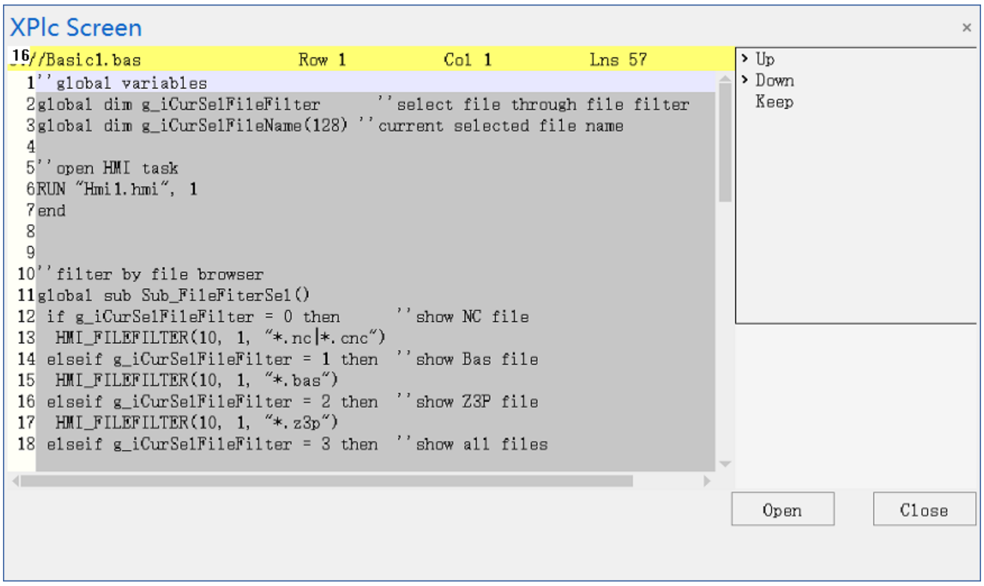
0th: C:/

File name	Size	Type	Modified time
..		Folder	2025/01/06 19:37
Basic1(2).bas	658B	File	2025/01/06 19:35
Basic1.bas	2KB	File	2025/01/06 19:35

File: Basic1.bas

Bas File(*.bas)

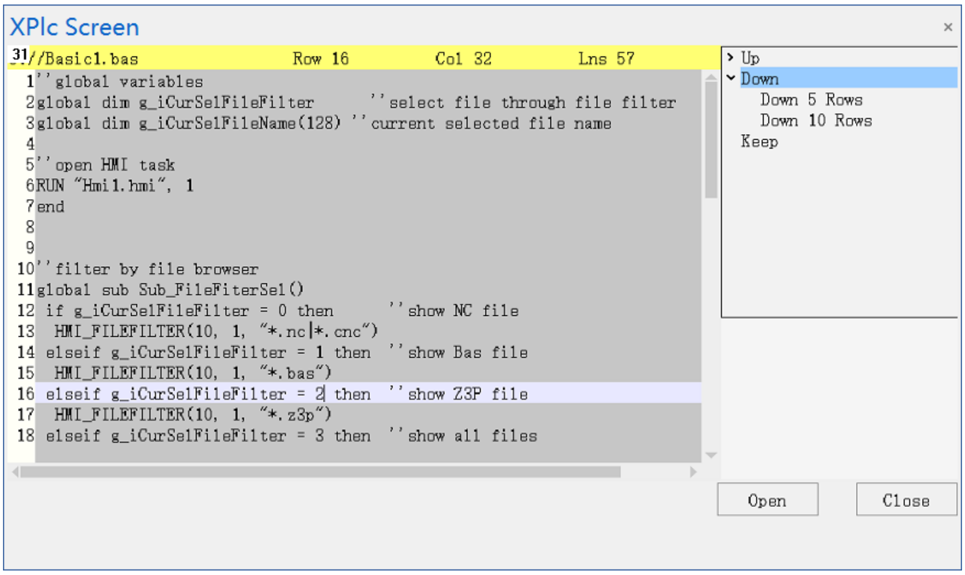
- Then, corresponding file3 editor interface appears, showing file content, you can edit it.



16//Basic1.bas   Row 1   Col 1   Lns 57

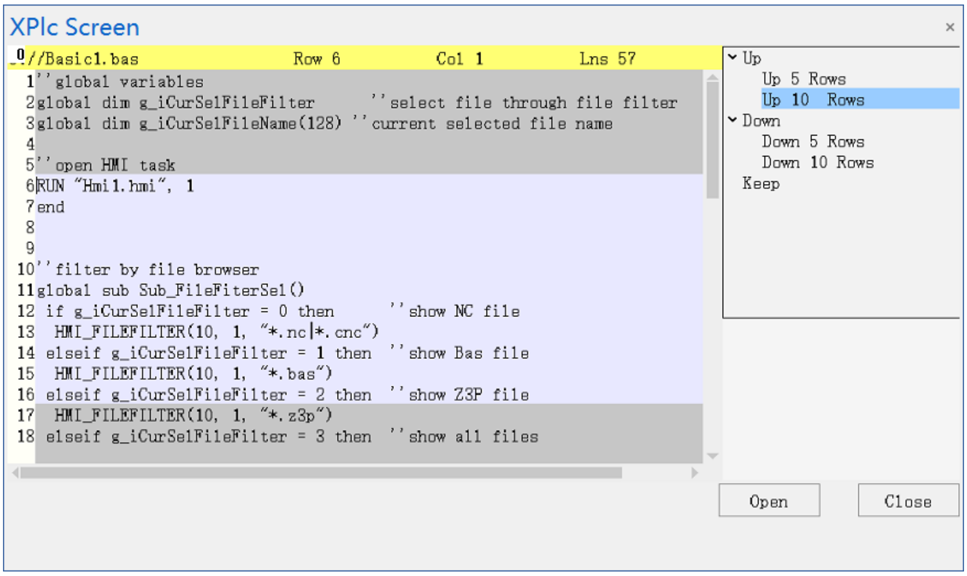
```
''global variables
global dim g_iCurSelFileFilter       ''select file through file filter
global dim g_iCurSelFileName(128) ''current selected file name

''open HMI task
RUN "Hmi1.hmi", 1
end


''filter by file browser
global sub Sub_FileFiterSel()
 if g_iCurSelFileFilter = 0 then      ''show NC file
  HMI_FILEFILTER(10, 1, "*.nc|*.cnc")
 elseif g_iCurSelFileFilter = 1 then  ''show Bas file
  HMI_FILEFILTER(10, 1, "*.bas")
 elseif g_iCurSelFileFilter = 2 then  ''show Z3P file
  HMI_FILEFILTER(10, 1, "*.z3p")
 elseif g_iCurSelFileFilter = 3 then  ''show all files
```

- Select any row, it will show tree all nodes. You can click to achieve corresponding actions


XPlc Screen

31//Basic1.bas   Row 16   Col 32   Lns 57

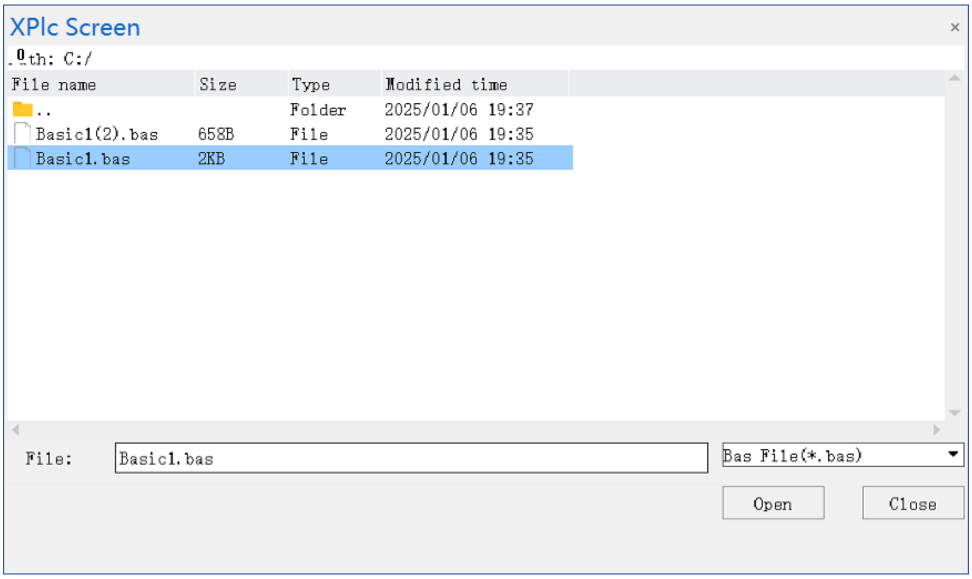
```
''global variables
global dim g_iCurSelFileFilter    ''select file through file filter
global dim g_iCurSelFileName(128) ''current selected file name

''open HMI task
RUN “Hmi1.hmi”, 1
end


''filter by file browser
global sub Sub_FileFiterSel()
 if g_iCurSelFileFilter = 0 then     ''show NC file
  HMI_FILEFILTER(10, 1, “*.nc|*.cnc”)
 elseif g_iCurSelFileFilter = 1 then ''show Bas file
  HMI_FILEFILTER(10, 1, “*.bas”)
 elseif g_iCurSelFileFilter = 2 then ''show Z3P file
  HMI_FILEFILTER(10, 1, “*.z3p”)
 elseif g_iCurSelFileFilter = 3 then ''show all files
```


XPlc Screen

0//Basic1.bas   Row 6   Col 1   Lns 57

```
''global variables
global dim g_iCurSelFileFilter    ''select file through file filter
global dim g_iCurSelFileName(128) ''current selected file name

''open HMI task
RUN “Hmi1.hmi”, 1
end


''filter by file browser
global sub Sub_FileFiterSel()
 if g_iCurSelFileFilter = 0 then     ''show NC file
  HMI_FILEFILTER(10, 1, “*.nc|*.cnc”)
 elseif g_iCurSelFileFilter = 1 then ''show Bas file
  HMI_FILEFILTER(10, 1, “*.bas”)
 elseif g_iCurSelFileFilter = 2 then ''show Z3P file
  HMI_FILEFILTER(10, 1, “*.z3p”)
 elseif g_iCurSelFileFilter = 3 then ''show all files
```

- Click “close”, file content will be closed, the interface will back to main file selection interface.


XPlc Screen

0th: C:/

File name	Size	Type	Modified time
..		Folder	2025/01/06 19:37
Basic1(2).bas	658B	File	2025/01/06 19:35
Basic1.bas	2KB	File	2025/01/06 19:35

File: Basic1.bas

Bas File(*.bas)

8.8. Usage of Trend Chart

This routine simply shows how to use trend chart.

[Target]

- Use "Trend Chart" component to periodically acquire data from the register and display the waveform.
- Detailed Functions: graphic display delay, pause, start, clear, and export the sampled data to a file.
- Use "view check" to acquire data from each channel at any given time.

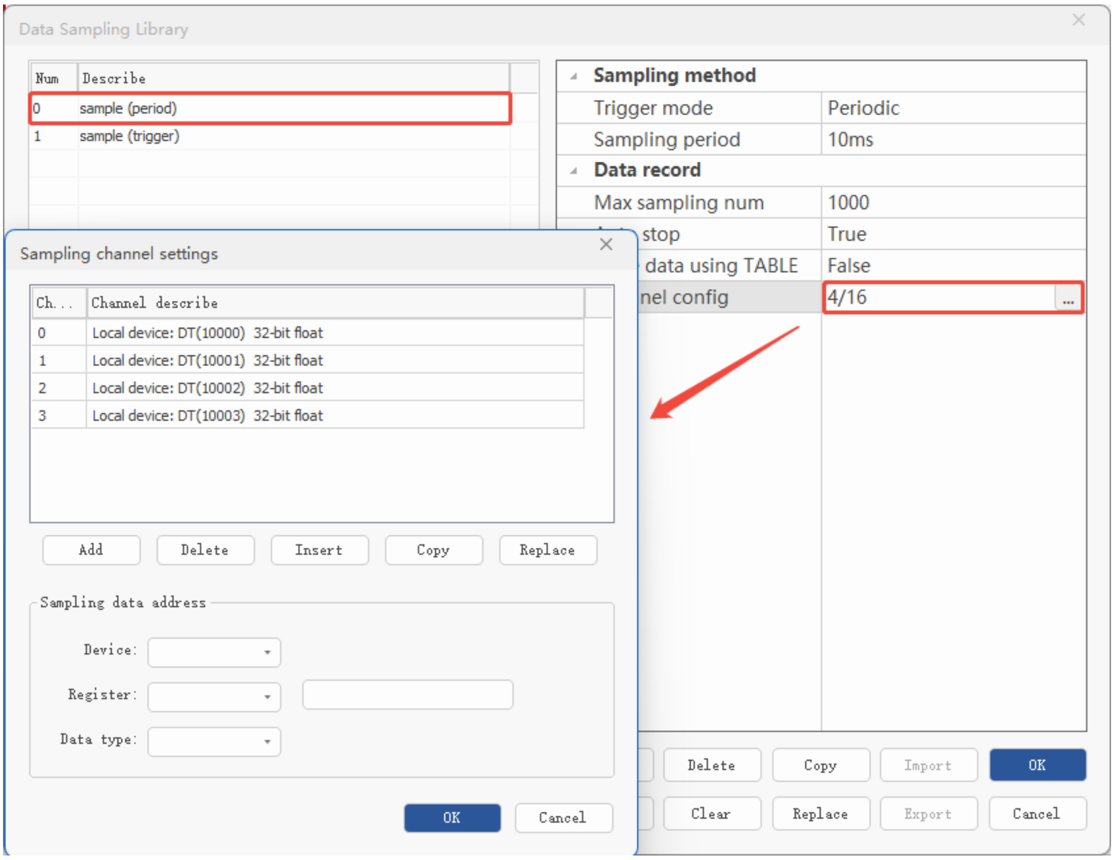
[What to Do?]

- In "resource sampling library", preset the sampling method.
- Edit Basic, defining needed functions:
  a. Change data in TABLE (10000) - TABLE(10003) in cycle according to the pattern of different function curves.
  b. Define SUB: define 5 SUB to achieve 5 functions – delay, clear sampling, pause sampling, start sampling, save file.
- In HMI, add "trend chart" component, set corresponding component's appearance. Then, select the resource sampling index from the sample list, and whether to pause "control display" & "check view display", then select corresponding register.
- In HMI, add "button" components to call corresponding SUB functions, and add "value" to show the register value, then add "bit switch" to bind with the register to pause "data sampling".

[How to Do?]

Step 1: set the sample library

Add "sample library", set corresponding descriptions, sampling method, channels. There are 4 channels, they are bound with TABLE (10000) – TABLE (10003) separately. For specific steps, please check "resource sample library".



Data Sampling Library

Num	Describe
0	sample (period)
1	sample (trigger)

Sampling method	
Trigger mode	Periodic
Sampling period	10ms
Data record	
Max sampling num	1000
stop	True
data using TABLE	False
nel config	4/16

Sampling channel settings

Ch...	Channel describe
0	Local device: DT(10000) 32-bit float
1	Local device: DT(10001) 32-bit float
2	Local device: DT(10002) 32-bit float
3	Local device: DT(10003) 32-bit float

Step 2: edit Basic program, define the variables and build SUB functions

After new build the project (.zpj), new build the Basic file, and set auto run task No., then add below codes in Basic file

--Basic Program--

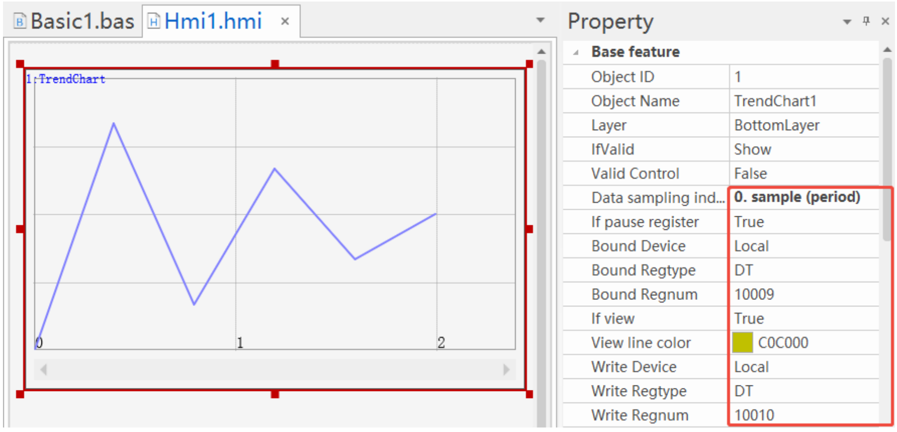
```
dim count
count = 0
while 1          'modify TABLE data in cycle according to different SUB curve relation.
      TABLE(10000) = SIN(2*PI*(count/100)) * 100
      TABLE(10001) = COS(2*PI*(count/100)) * 100
      TABLE(10002) = SIN(2*PI*(count/100) + PI/4) * 100
      TABLE(10003) = COS(2*PI*(count/100) + PI/4) * 100
      count = (count+1) MOD 100
      DELAY(9)
wend
end

'''test "delay"
GLOBAL SUB sub_xxx()
      DELAY(100)
end SUB
```

```
'''clear sampling
GLOBAL SUB sub_SampClear()
        HMI_SampClear(0)
end SUB

'''pause sampling
GLOBAL SUB sub_SampPause()
        HMI_SampPause(0)
end SUB

'''start pausing
GLOBAL SUB sub_SampStart()
        ''set 4 channels of Y value range
        HMI_TrChartRange(10, 1, 0, -200, 200)
        HMI_TrChartRange(10, 1, 1, -200, 200)
        HMI_TrChartRange(10, 1, 2, -200, 200)
        HMI_TrChartRange(10, 1, 3, -200, 200)
        ''sample library 0 starts sampling
        HMI_SampStart(0)
end SUB

'''save the file
GLOBAL SUB sub_SampSave()
        local f1
        ''check whether the file exists
        f1 = FILE_ZOPEN("sampling0.txt", "r")
        if (f1 >= 0) then
                ''file be here, append file
                FILE_ZCLOSE(f1)
                HMI_SAMPSAVE(0, 1, "sampling0.txt")
        else
                ''no file, create file
                HMI_SAMPSAVE(0, 0, "sampling0.txt")
        endif
end SUB
```

Step 3: edit HMI Interface

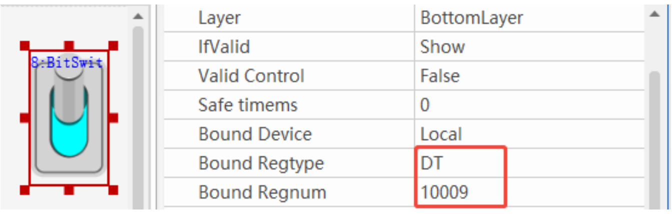
(1) Create one HMI file, set auto run task No., and open HMI window.

(2) Add one “trend chart” component, elect the resource sampling index from the sample list, and enable “pause (display control, view & check)” function.



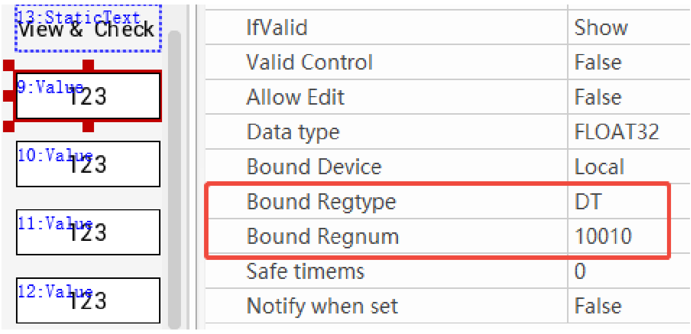
Base feature	
Object ID	1
Object Name	TrendChart1
Layer	BottomLayer
IfValid	Show
Valid Control	False
Data sampling ind...	0. sample (period)
If pause register	True
Bound Device	Local
Bound Regtype	DT
Bound Regnum	10009
If view	True
View line color	🟨 C0C000
Write Device	Local
Write Regtype	DT
Write Regnum	10010

(3) Add one “bitswitch” component, binding with “trend chart” to pause displaying bound register TABLE (10009), then sampling can be paused.



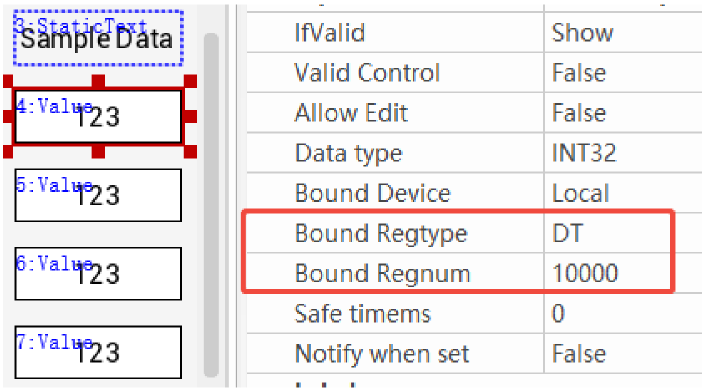
Layer	BottomLayer
IfValid	Show
Valid Control	False
Safe timems	0
Bound Device	Local
Bound Regtype	DT
Bound Regnum	10009

(4) Add 4 “value” components to bind them with “trend chart”, writing into bound register TABLE (10010-10013). To show 4 channels’ viewing & checking values.



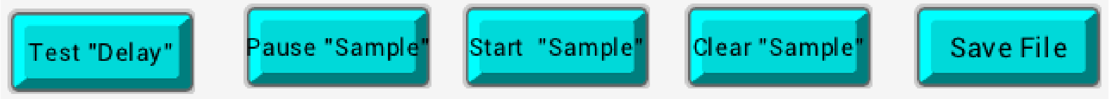
IfValid	Show
Valid Control	False
Allow Edit	False
Data type	FLOAT32
Bound Device	Local
Bound Regtype	DT
Bound Regnum	10010
Safe timems	0
Notify when set	False

(5) Add 4 “value” components to bind them with TABLE (10000) – TABLE (10003) where saves 4 channels’ data of “sample library 0”, and real-time show the current data of these 4 channels.



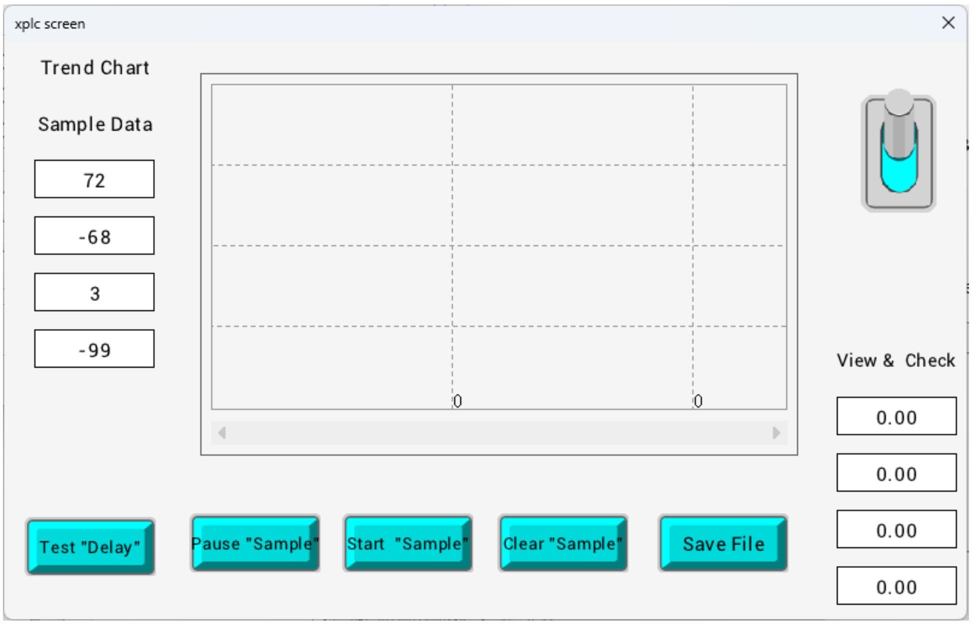
IfValid	Show
Valid Control	False
Allow Edit	False
Data type	INT32
Bound Device	Local
Bound Regtype	DT
Bound Regnum	10000
Safe timems	0
Notify when set	False

(6) Add 5 "button" functions, and bind them with corresponding Basic SUB functions separately.



[Running Effects]

- After downloading the interface: the left will show data of TABLE (10000) – TABLE (10003) in real-time.



72

-68

3

-99

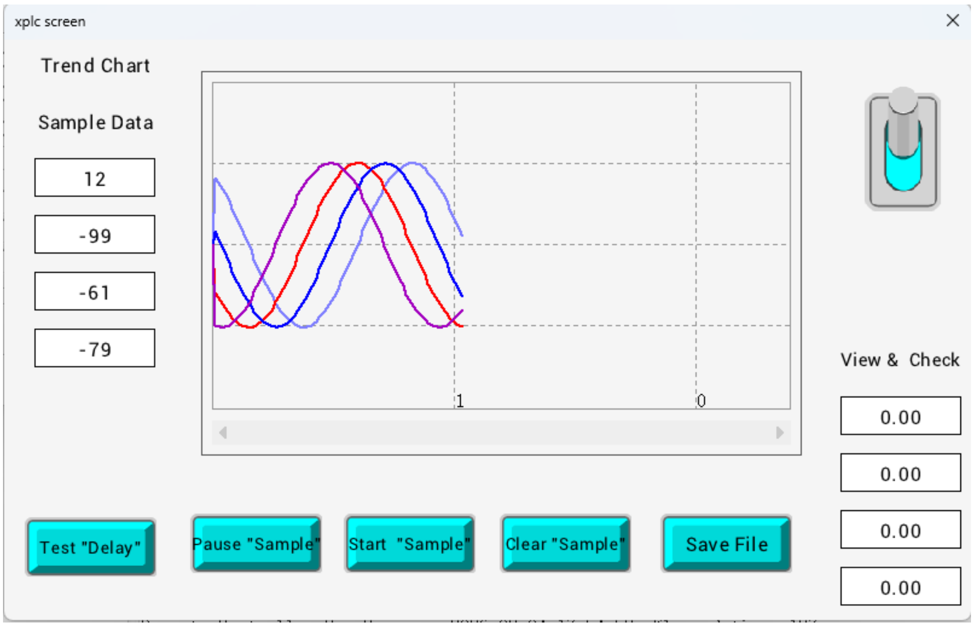
0.00

0.00

0.00

0.00

- After clicking "start sample", it will show a waveform that includes 4 channels' captured points.



12

-99

-61

-79

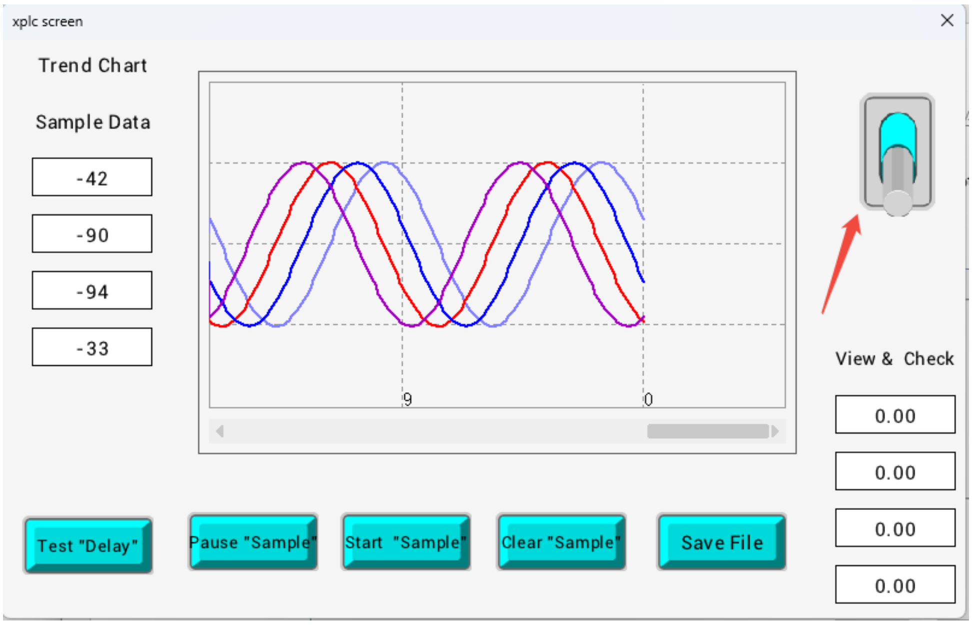
0.00

0.00

0.00

0.00

- After clicking “bitswitch”, it will pause. And it can be consumed after “clicked” again.



xplc screen

Trend Chart

Sample Data

-42

-90

-94

-33

View & Check

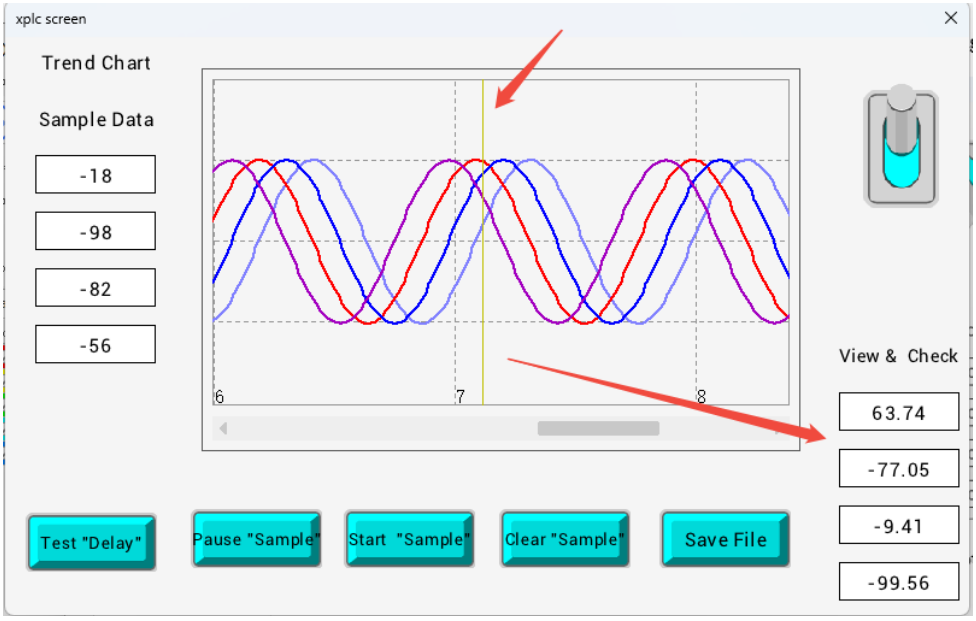
0.00

0.00

0.00

0.00

- When you click the trend chart, there will appear one yellow line, and the intersection data of the yellow line and the graphic is saved into register TABLE (10010 - 10013).



xplc screen

Trend Chart

Sample Data

-18

-98

-82

-56

View & Check

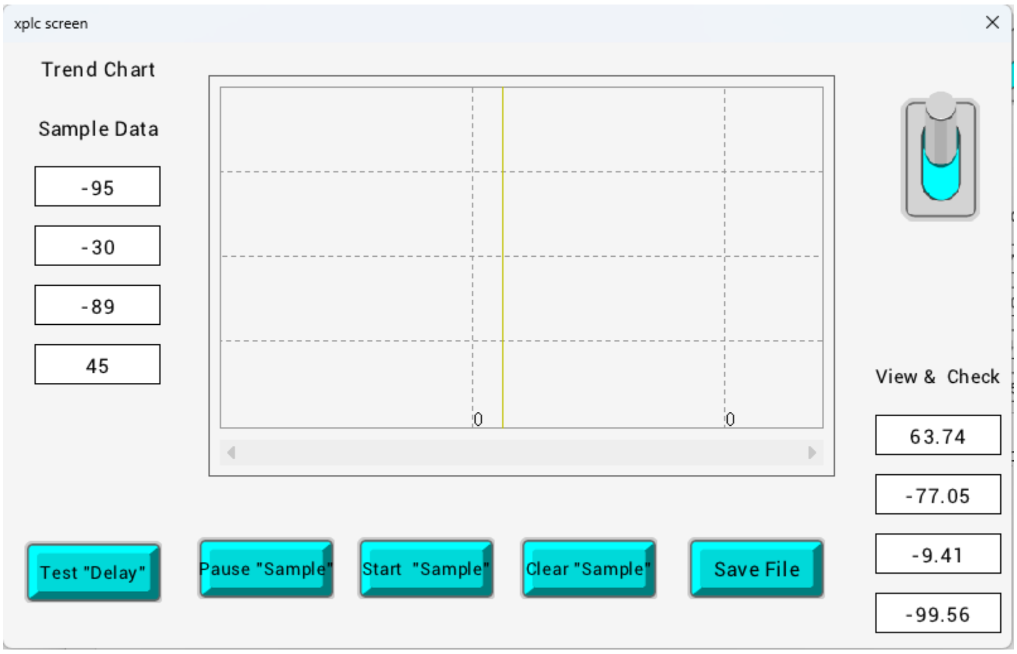
63.74

-77.05

-9.41

-99.56

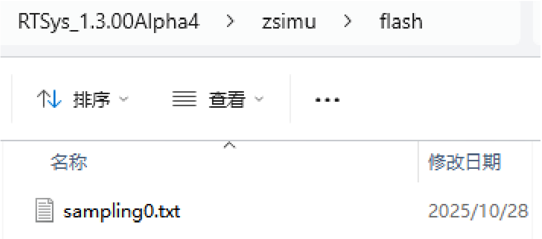
- After clicking “delay test”, the waveform will be delayed. And click “clear sample” to clear the sampling data.



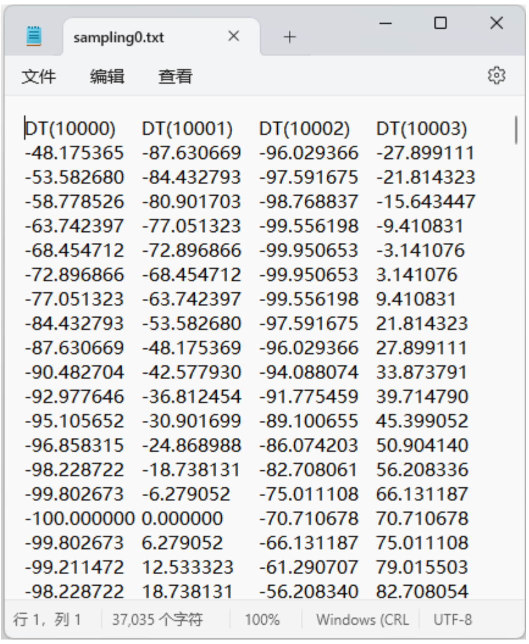
Sample Data: -95, -30, -89, 45

View & Check: 63.74, -77.05, -9.41, -99.56

- Click “save file” to export one TXT file that includes sample data.



sampling0.txt 2025/10/28



```
DT(10000)   DT(10001)   DT(10002)   DT(10003)
-48.175365  -87.630669  -96.029366  -27.899111
-53.582680  -84.432793  -97.591675  -21.814323
-58.778526  -80.901703  -98.768837  -15.643447
-63.742397  -77.051323  -99.556198  -9.410831
-68.454712  -72.896866  -99.950653  -3.141076
-72.896866  -68.454712  -99.950653  3.141076
-77.051323  -63.742397  -99.556198  9.410831
-84.432793  -53.582680  -97.591675  21.814323
-87.630669  -48.175369  -96.029366  27.899111
-90.482704  -42.577930  -94.088074  33.873791
-92.977646  -36.812454  -91.775459  39.714790
-95.105652  -30.901699  -89.100655  45.399052
-96.858315  -24.868988  -86.074203  50.904140
-98.228722  -18.738131  -82.708061  56.208336
-99.802673  -6.279052   -75.011108  66.131187
-100.000000 0.000000    -70.710678  70.710678
-99.802673  6.279052    -66.131187  75.011108
-99.211472  12.533323   -61.290707  79.015503
-98.228722  18.738131   -56.208340  82.708054
```

[and you can import this txt file in RTSys SCOPE]

8.9. Usage of Basic Extend (Manual Draw)

This routine simply shows how to use “Basic Extend”.

[Target]

- Use "Basic Extend" component to draw by clicking the mouse twice.
- Use "Basic Extend" component to clear the canvas.

That is: left-click the canvas to create a point; click again to create another point, and a line segment connects the two points; this process continues. Right-click to clear the canvas, “clear” button also can achieve same effect.

[What to Do?]

- Edit Basic to define initial SUB (init SUB), refresh SUB, and draw SUB.

1) Define Init SUB

Define a drawing initialization function, and define the first table of the drawing element to store the number of valid points.

2) Define Global Refresh SUB

a. “Basic Extend” component: table(0) stores the mouse's horizontal coordinate, and table(1) stores the mouse's vertical coordinate.

b. “Clear” component: when the mouse button is pressed, the state is set to 1, and the clear SUB (MyPalette_Clear) is executed. When it is released, the state is set to 0.

c. “Draw” component: when the left mouse button is pressed, the valid points are incremented by one, and the current mouse position is saved to the component array. When the right mouse button is pressed, the canvas is cleared.

3) Define Global Draw SUB

a. “Clear” component: set the button colors for different states, draw the button background, and display the text.

b. “Draw” component: at first, determine if there is enough remaining space in the component array. Then, set the canvas and pen colors. If the number of valid points is not 0, obtain the coordinates of the points stored in the component array space and draw the dots and the line segments between the two points.

- In HMI, add “Basic Extend” component, set corresponding component’s appearance. Then, call init SUB, refresh SUB, and draw SUB.

[How to Do?]

Step 1: edit Basic program, to define the variables and build SUB functions

After new build the project (.zpj), new build the Basic file, and set auto run task No., then add below codes in Basic file

--Basic Program--

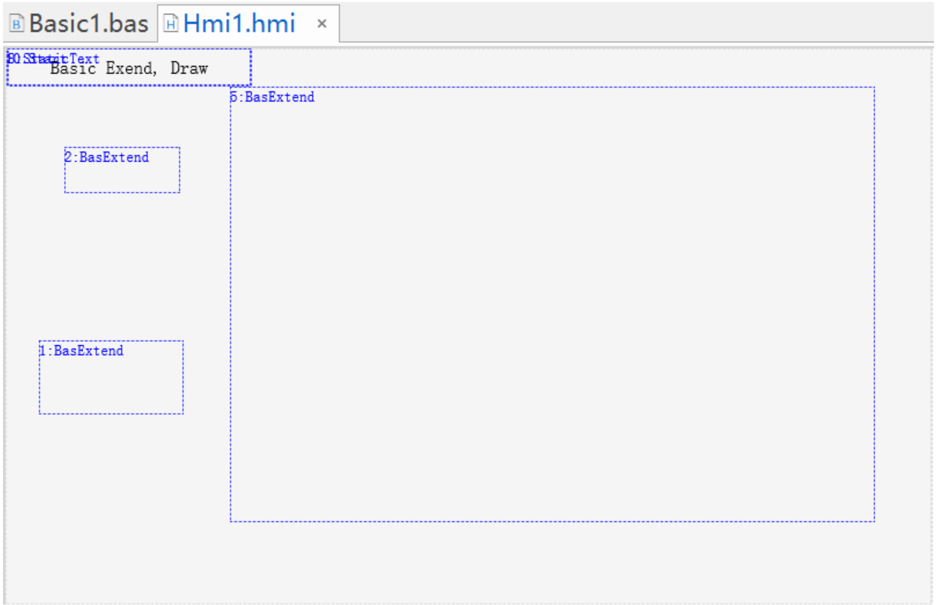
```
""""""""""""""""""""""
"Example of "Basic Extend"
"
""""""""""""""""""""""
"1. Each "Basic Extend" control has its own extended properties, which can be accessed in the
initialization, refresh, and drawing SUB functions.
"2. the basic extend control that has been defined can be reused at any window.

end

"    "custom" button -- init SUB
global sub MyButton_InitSub()
        ?"=>MyButton_InitSub"
end sub

" "custom" button -- refresh SUB
global sub MyButton_ReflashSub()
        local iKeyEvent
        local strSubName(128)
        " check "mouse" event
        iKeyEvent = MOUSE_SCAN(0)
        if iKeyEvent > 0 then
'               ?"mouse pressed"
                HMI_CONTROLATTR("STATE", 1)
                SET_REDRAW

                "whether there is SUB function, and executes it?
                strSubName = HMI_CONTROLATTRS("EXTSTR%2")
                if STRLEN(strSubName) > 0 then
```

```
                        "Call SUB
                        ?strSubName,"start executing"
                        HMI_CALLSUB(strSubName)
                        ?strSubName,"execute ends"
                endif
        elseif iKeyEvent < 0 then
'               ?"mouse released"
                HMI_CONTROLATTR("STATE", 0)
                SET_REDRAW
        endif
end sub

" "custom" button -- draw SUB
global sub MyButton_DrawSub()
        local oldDrawMode
        local corBack
        local showText(128)
        local iw, ih
        "save current drawing mode
        oldDrawMode = HMI_DEFAULTATTR("DRAW_M")
        "enable anti-aliased drawing
        HMI_DEFAULTATTR("DRAW_M", 1)
        "get component range
        iw = HMI_CONTROLATTR("SIZEX")
        ih = HMI_CONTROLATTR("SIZEY")
        "there are 2 states
        if HMI_CONTROLATTR("STATE") = 0 then
                corBack = RGB(33, 149, 246)
                showText = HMI_CONTROLATTRS("EXTSTR%0")
        else
                corBack = RGB(107, 184, 249)
                showText = HMI_CONTROLATTRS("EXTSTR%1")
        endif
        "draw the button's background
        SET_COLOR(RGB(0,0,0), corBack)
        DRAWEX_RECT(0, 0, iw-1, ih-1, 10, 1)
        "draw the button's text
        SET_COLOR(RGB(0,0,0), -1)       "not draw the background
        DRAWTEXT2(0, 0, iw-1, ih-1, 0, showText)
        "drawing ends,    restore the drawing mode
        HMI_DEFAULTATTR("DRAW_M", oldDrawMode)
end sub

'''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
"      customize "drawing"
""""""""""""""""""""""""""""

" "drawing" -- init SUB
global sub MyPalette_InitSub()
        ?"=>MyPalette_InitSub"
        "initialize "customized TABLE" component
        local iTableNum
        iTableNum = HMI_CONTROLATTR("TABLENUM")
        if iTableNum > 0 then
                "the first TABLE saves "valid point numbers"
                HMI_CONTROLATTR("TABLEVAL%0", 0)
        endif
end sub

" "drawing" -- refresh SUB
global sub MyPalette_ReflashSub()
        local iKeyEvent
        local iTableNum, iCurPointNum
        local strAttr(32)
        "watch "mouse" event (left-button)
        iKeyEvent = MOUSE_SCAN(0)
        if iKeyEvent > 0 then
                "mouse pressed
                iTableNum = HMI_CONTROLATTR("TABLENUM")
                iCurPointNum = HMI_CONTROLATTR("TABLEVAL%0")
                " add one point
                if((iCurPointNum+1)*2 + 1 <= iTableNum) then
                        strAttr = "TABLEVAL%" + TOSTR(iCurPointNum*2+1, 1, 0)
                        HMI_CONTROLATTR(strAttr, table(0))
                        strAttr = "TABLEVAL%" + TOSTR(iCurPointNum*2+1+1, 1, 0)
                        HMI_CONTROLATTR(strAttr, table(1))
                        "update valid point numbers
                        HMI_CONTROLATTR("TABLEVAL%0", iCurPointNum+1)
'                       ?"Add point = "HMI_CONTROLATTR("TABLEVAL%0")
                endif
                "refresh the drawing
                SET_REDRAW
        elseif iKeyEvent < 0 then
'               ?"mouse released"
        endif

        "check mouse (right-button)
        iKeyEvent = MOUSE_SCAN(0, 1)
```

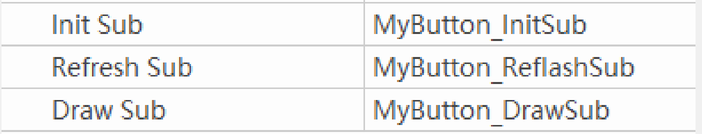
```
    if iKeyEvent < 0 then
        "right button released
        ?"clear canvas"
        HMI_CONTROLATTR("TABLEVAL%0", 0)
        SET_REDRAW
    endif
end sub

""drawing" -- draw SUB
global sub MyPalette_DrawSub()
    local oldDrawMode
    local iw, ih
    local iTableNum, iCurPointNum, index
    local lastx, lasty, curx, cury
    local strAttr(32)
    "the number of TABLE
    iTableNum = HMI_CONTROLATTR("TABLENUM")
    if iTableNum < 5 then
        "if it is not enough to save 2 points, end.
        return
    endif

    "save current drawing mode
    oldDrawMode = HMI_DEFAULTATTR("DRAW_M")
    "enable anti-aliased drawing
    HMI_DEFAULTATTR("DRAW_M", 1)

    "get component range
    iw = HMI_CONTROLATTR("SIZEX")
    ih = HMI_CONTROLATTR("SIZEY")

    "set canvas' background
    SET_COLOR(RGB(255,255,255), RGB(255,255,255))
    DRAWEX_RECT(0, 0, iw-1, ih-1, 10, 1)

    "set the brush
    SET_COLOR(RGB(255,0,0), RGB(0,0,0))

    "valid point
    iCurPointNum = HMI_CONTROLATTR("TABLEVAL%0")
    if iCurPointNum > 0 then
        "the starting point
        strAttr = "TABLEVAL%" + TOSTR(1+0, 1, 0)
        lastx = HMI_CONTROLATTR(strAttr)
        strAttr = "TABLEVAL%" + TOSTR(1+1, 1, 0)
        lasty = HMI_CONTROLATTR(strAttr)
        DRAWEX_ELLIPSE(lastx, lasty, 3, 3, 1)
        "drawing by taking points step by step
        for index=1 to iCurPointNum-1
            "take one point
            strAttr = "TABLEVAL%" + TOSTR(index*2+1+0, 1, 0)
            curx = HMI_CONTROLATTR(strAttr)
            strAttr = "TABLEVAL%" + TOSTR(index*2+1+1, 1, 0)
            cury = HMI_CONTROLATTR(strAttr)
            DRAWEX_ELLIPSE(curx, cury, 3, 3, 1)
            "draw one line
            DRAWEX_LINE(lastx, lasty, curx, cury)
            "record the last point
            lastx = curx
            lasty = cury
        next
    endif
    "drawing ends, restore the drawing mode
    HMI_DEFAULTATTR("DRAW_M", oldDrawMode)
end sub

"clear
global sub MyPalette_Clear(testnum)
    ?"clear canvas",testnum
    HMI_CONTROLATTR("TABLEVAL%0", 0, 10, 5)
end sub

""""""""""""""""""""""""""""""""""""""""""""""""
"                   custom "slide" component
""""""""""""""""""""""""""""""""""""""""""""""""
" "slide" -- init SUB
global sub MySlider_InitSub()
    ?"=>MySlider_InitSub"
    HMI_CONTROLATTR("TABLEVAL%0", 0)     "record the slide position
    HMI_CONTROLATTR("TABLEVAL%1", 0)     "whether the mouse is pressed
end sub

" "slide" -- refresh SUB
global sub MySlider_ReflashSub()
    local iKeyEvent
    local bPressed
    local curpos, minpos, maxpos, fRatio
    "slide range
    minpos = HMI_CONTROLATTR("EXTVAL%0")
    maxpos = HMI_CONTROLATTR("EXTVAL%1")
    ''get current mouse state
    bPressed = HMI_CONTROLATTR("TABLEVAL%1")
    ''check "mouse" event
    iKeyEvent = MOUSE_SCAN(0)
    if iKeyEvent > 0 then
        ''mouse pressed
        bPressed = true
    elseif iKeyEvent < 0 then
'       ?"mouse released"
        bPressed = false
    endif
    ''slide when mouse is pressed
    if bPressed then
        ''alculate the slide position
        curpos = (table(0) / HMI_CONTROLATTR("SIZEX")) * (maxpos - minpos)    +
minpos
        if curpos < minpos then
            curpos = minpos
        elseif curpos >maxpos then
            curpos = maxpos
        endif
        ''refresh
        HMI_CONTROLATTR("TABLEVAL%0", curpos)
        SET_REDRAW
    endif
    ''update mouse state
    HMI_CONTROLATTR("TABLEVAL%1", bPressed)
end sub

'''slide" --    draw SUB
global sub MySlider_DrawSub()
    local oldDrawMode
    local corBack
    local iw, ih, curx
    local curpos, minpos, maxpos, fRatio
    CONST Slider_Width = 8      'set the slide's width
    ''save current drawing mode
    oldDrawMode = HMI_DEFAULTATTR("DRAW_M")
    ''enable anti-aliased drawing
    HMI_DEFAULTATTR("DRAW_M", 1)
    ''get component range
    iw = HMI_CONTROLATTR("SIZEX")
    ih = HMI_CONTROLATTR("SIZEY")
    "draw slide's background
    SET_COLOR(RGB(0,0,0), RGB(172,172,172))
    DRAWEX_RECT(0, ih/4, iw-1, ih*3/4, 8, 1)
    "draw the slide
    SET_COLOR(RGB(0,0,0), RGB(33, 149, 246))
    curpos = HMI_CONTROLATTR("TABLEVAL%0")
    minpos = HMI_CONTROLATTR("EXTVAL%0")
    maxpos = HMI_CONTROLATTR("EXTVAL%1")
    fRatio = (curpos - minpos) / (maxpos - minpos)
    curx = Slider_Width + (iw-Slider_Width*2)*fRatio
    DRAWEX_RECT(curx-Slider_Width, 0, curx+Slider_Width-1, ih-1, Slider_Width-1, 1)
    "drawing ends, restore the drawing mode
    HMI_DEFAULTATTR("DRAW_M", oldDrawMode)
end sub
```

Step 2: edit HMI Interface



(1) Create one HMI file, set auto run task No., and open HMI window.

(2) Add one “Basic Extend”, make it as “clear” button, and adjust the size at suitable position. Then, in property window, call Basic SUB “MyButton_InitSub, MyButton_ReflashSub, MyButton_DrawSub” separately for init SUB, refresh SUB, and draw SUB.

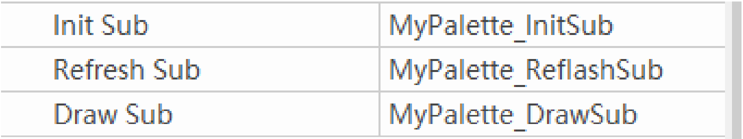


Init Sub	MyButton_InitSub
Refresh Sub	MyButton_ReflashSub
Draw Sub	MyButton_DrawSub

(3) Set “clear” button to extend resources, and three strings are defined here. Strings 0 and 1

represent the display text for the two states, and string 0 represents the name of the function called when the button is pressed.

(4) Add another “Basic Extend”, make it as “draw” component, and adjust the size at suitable position. Then, in property window, call Basic SUB “MyPalette_InitSub, MyPalette_ReflashSub, MyPalette_DrawSub” separately for init SUB, refresh SUB, draw SUB.

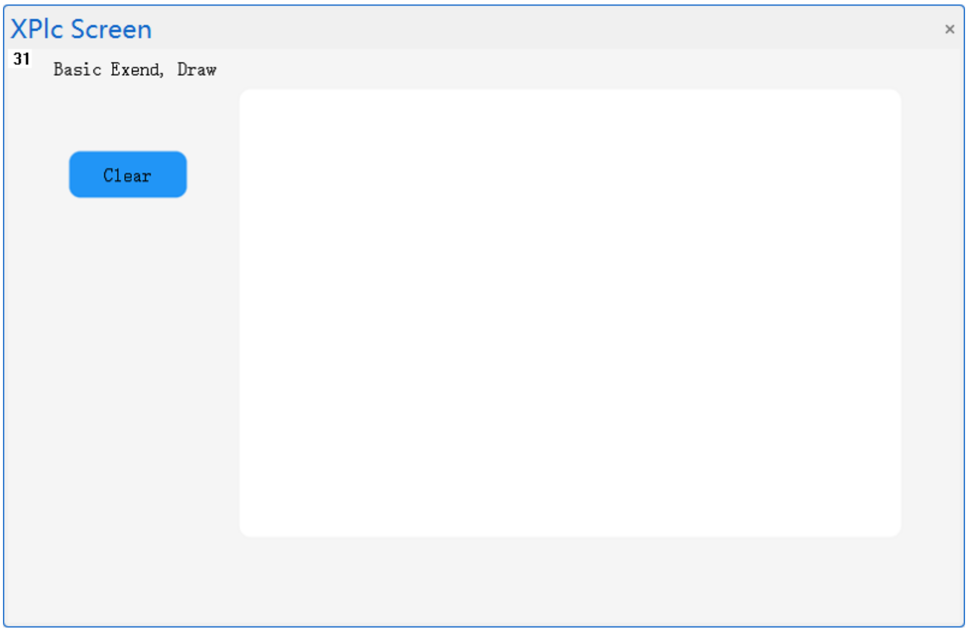


Init Sub	MyPalette_InitSub
Refresh Sub	MyPalette_ReflashSub
Draw Sub	MyPalette_DrawSub

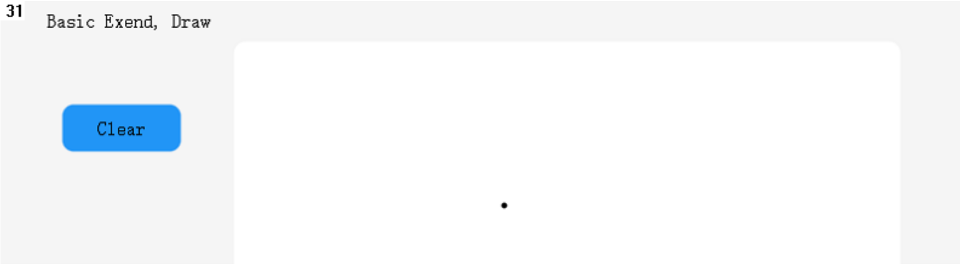
(5) Set “draw” button to extend resources, here defines valid array space for this component.

[Running Effects]

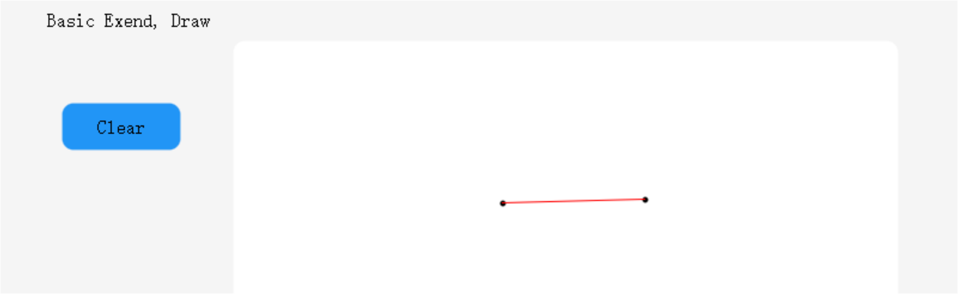
- ➢ After downloading, the initial interface:



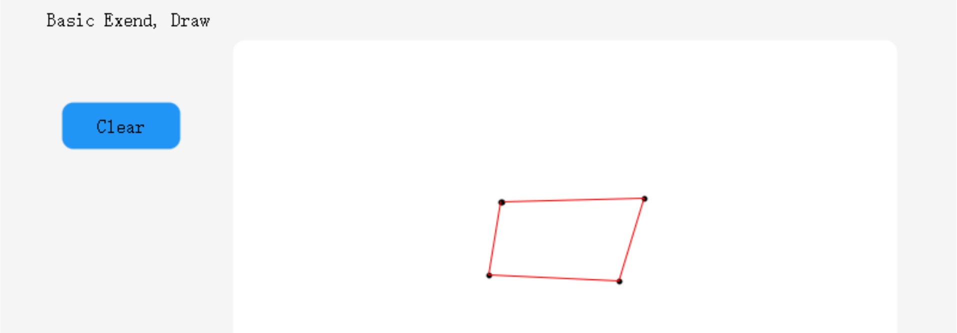
- ➢ Clicking the canvas will create a dot.



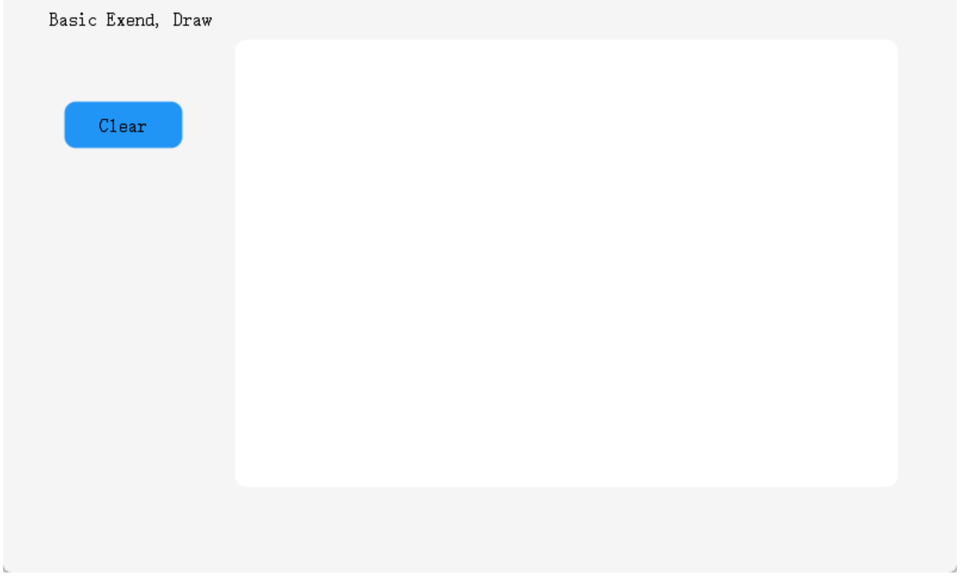
- Clicking the canvas again will create another dot, and one line.



- Similarly, you can draw any shape you want on the canvas.



- Clicking "clear" / right clicking the mouse will clear the canvas.



8.10. Usage of Recipe View

This routine simply shows how to use "Recipe View".

[Target]

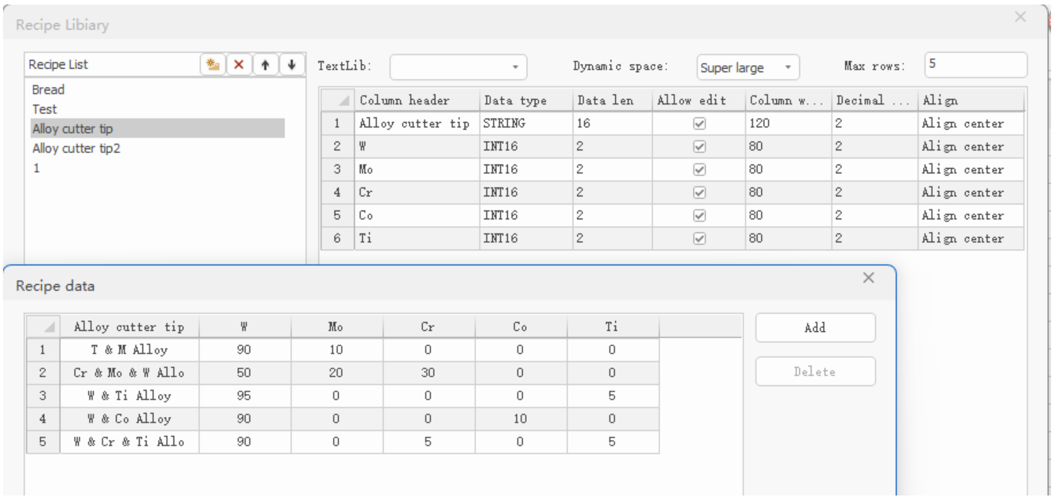
Use "Recipe View" component to show the recipe group, including related operations, like, insert, delete, move up & down, sort, save, import, etc.

[What to Do?]

- Add new recipe group into recipe library.
- In Basic file, use recipe view related commands and define global SUB functions:
  a. insert row: use HMI_RECIPEINSERT command to inert one row in recipe view.
  b. delete row: use HMI_RECIPEDELETE command to delete one row in recipe view.
  c. move up & down: use HMI_RECIPEMOVE command to move current recipe row.
  d. sort: use HMI_RECIPESORT command to sort all recipe rows.
  e. save: use HMI_RECIPESAVE command to save and export the recipe group file.
  f. import: use HMI_RECIPEIMPORT command to import the recipe group.
- In HMI, add several "button" components, and bind corresponding action SUB. Then, add "list" component to select the sorting method.

[How to Do?]

Step 1: in recipe library, preset the recipe group.



Bread
Test
Alloy cutter tip
Alloy cutter tip2
1

TextLib:

Dynamic space: Super large

Max rows: 5

	Column header	Data type	Data len	Allow edit	Column w...	Decimal ...	Align
1	Alloy cutter tip	STRING	16	☑	120	2	Align center
2	W	INT16	2	☑	80	2	Align center
3	Mo	INT16	2	☑	80	2	Align center
4	Cr	INT16	2	☑	80	2	Align center
5	Co	INT16	2	☑	80	2	Align center
6	Ti	INT16	2	☑	80	2	Align center

	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	T & M Alloy	90	10	0	0	0
2	Cr & Mo & W Allo	50	20	30	0	0
3	W & Ti Alloy	95	0	0	0	5
4	W & Co Alloy	90	0	0	10	0
5	W & Cr & Ti Allo	90	0	5	0	5

For specific steps, please check recipe library.

Step 2: edit Basic program, to define the variables and build SUB functions

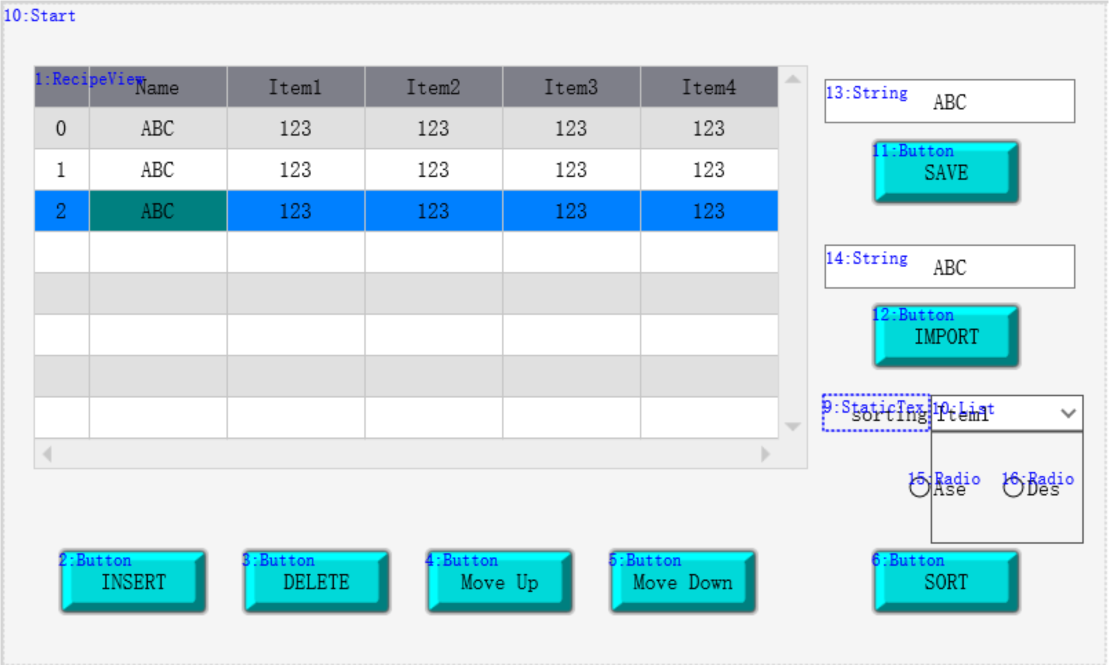
- After new build the project (.zpj), new build the Basic file, and set auto run task No.
- In Basic program, set 4 global variables to define the sorting method, column sorting, file name (for saving), imported file name, recipe view window No., and the recipe view component No.
- Define global SUB function.

--Basic Program--

```
GLOBAL DIM g_SortType
GLOBAL DIM g_SortItem
GLOBAL DIM g_SaveFileName(128)
GLOBAL DIM g_ImportFileName(128)

global CONST g_RecipeWinId = 10       'the window No. where the recipe view component is located
global CONST g_RecipeCtrlId = 1            'recipe view component No.

g_SaveFileName = "RecipeFile0.rdb"         'set the file name to save.
g_ImportFileName = "RecipeFile0.rdb"    'set the name of the imported file.
g_SortType = 0                'sorting method, 0:    ascending, non-0: descending
g_SortItem = 1                  'sort the column, column 0 is usually the recipe name and cannot be sorted.
end

'insert the row
GLOBAL sub sub_InsertLine()
    Hmi_RecipeInsert(g_RecipeWinId, g_RecipeCtrlId, -1)
ENDSUB

'delete row
GLOBAL sub sub_DeleteLine()
    Hmi_RecipeDelete(g_RecipeWinId, g_RecipeCtrlId, -1)
ENDSUB

'move up the row
GLOBAL sub sub_MoveLinePrev()
    Hmi_RecipeMove(g_RecipeWinId, g_RecipeCtrlId, "PREV")
ENDSUB

'move down the row
GLOBAL sub sub_MoveLineNext()
    Hmi_RecipeMove(g_RecipeWinId, g_RecipeCtrlId, "NEXT")
```

```
ENDSUB

'sorting
GLOBAL sub sub_SortLine()
        Hmi_RecipeSort(g_RecipeWinId, g_RecipeCtrlId, g_SortItem, g_SortType)
ENDSUB

'save the file
GLOBAL sub sub_SaveFile()
        Hmi_RecipeSave(Hmi_RecipeLoad(g_RecipeWinId, g_RecipeCtrlId), g_SaveFileName)
ENDSUB

'import the file
GLOBAL sub sub_ImportFile()
        local recipeGroupName(32)
        recipeGroupName = Hmi_RecipeLoad(g_RecipeWinId, g_RecipeCtrlId)   'get current
loading recipe group
        Hmi_RecipeImport(recipeGroupName, g_ImportFileName, 0)
ENDSUB
```

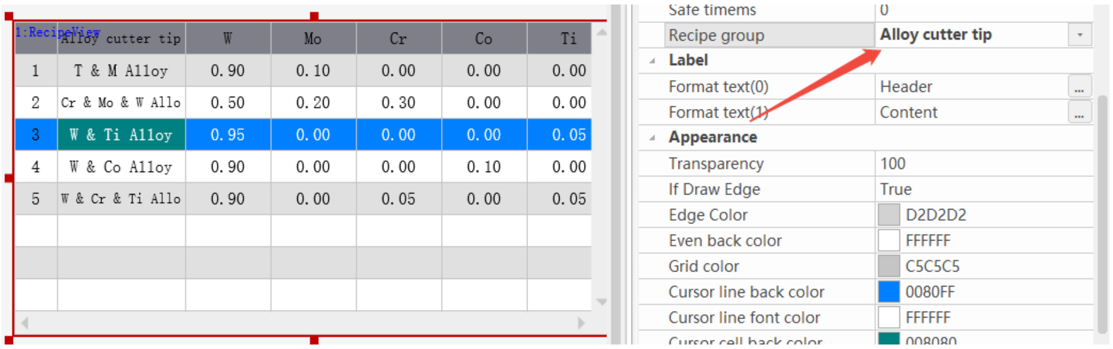
Step 3: edit HMI Interface



	Name	Item1	Item2	Item3	Item4
0	ABC	123	123	123	123
1	ABC	123	123	123	123
2	ABC	123	123	123	123

(1) Create one HMI file, set auto run task No., and open HMI window.

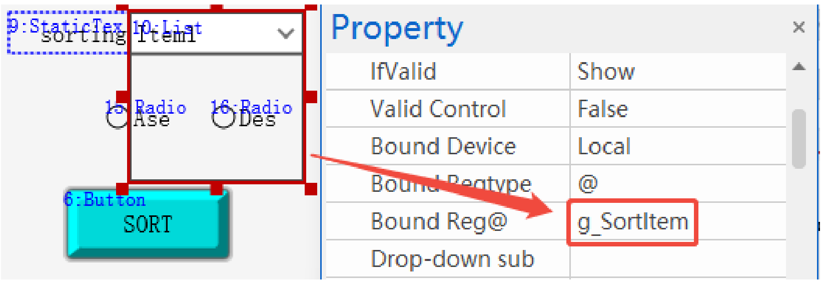
(2) Add one “Recipe View”, set its properties, and select, bind corresponding recipe group.



	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	T & M Alloy	0.90	0.10	0.00	0.00	0.00
2	Cr & Mo & W Allo	0.50	0.20	0.30	0.00	0.00
3	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05
4	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
5	W & Cr & Ti Allo	0.90	0.00	0.05	0.00	0.05

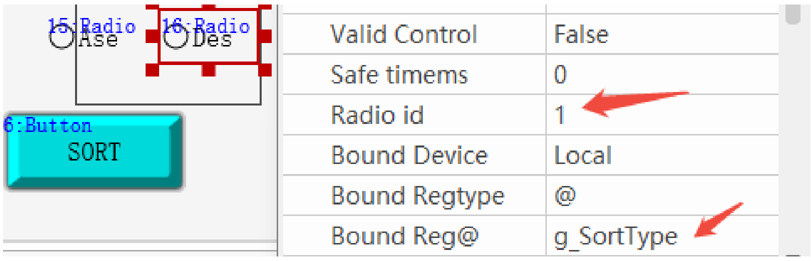
Safe timems	0
Recipe group	**Alloy cutter tip**
Label	
Format text(0)	Header
Format text(1)	Content
Appearance	
Transparency	100
If Draw Edge	True
Edge Color	D2D2D2
Even back color	FFFFFF
Grid color	C5C5C5
Cursor line back color	0080FF
Cursor line font color	FFFFFF
Cursor cell back color	008080

(3) Add “list” component, bind it with Basic column sorting variables (g_SortItem).



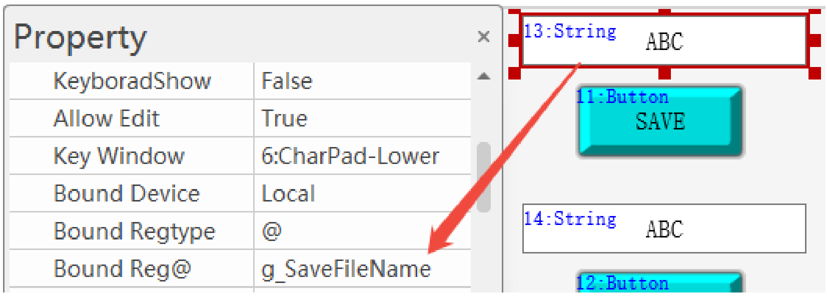
IfValid	Show
Valid Control	False
Bound Device	Local
Bound Regtype	@
Bound Reg@	g_SortItem
Drop-down sub	

(4) Add 2 “radio button” components, bind them with Basic sorting method (g_SortType), and set the button No., when “Ase” (ascending) button is pressed, g_SortType value becomes 0 to sort in ascending way; when “Des” (descending) button is pressed, g_SortType value becomes 1 to sort in descending way.

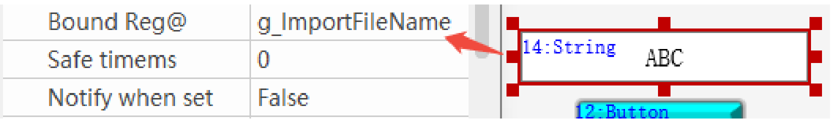


Valid Control	False
Safe timems	0
Radio id	1
Bound Device	Local
Bound Regtype	@
Bound Reg@	g_SortType

(5) Add 2 “string” components to bind them with Basic SUB of file name (for saving) (g_SaveFileName) and Basic SUB of imported file name (g_ImportFileName)

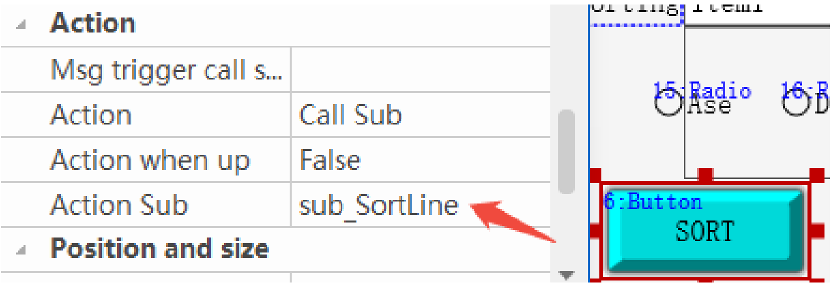


KeyboradShow	False
Allow Edit	True
Key Window	6:CharPad-Lower
Bound Device	Local
Bound Regtype	@
Bound Reg@	g_SaveFileName



Bound Reg@	g_ImportFileName
Safe timems	0
Notify when set	False

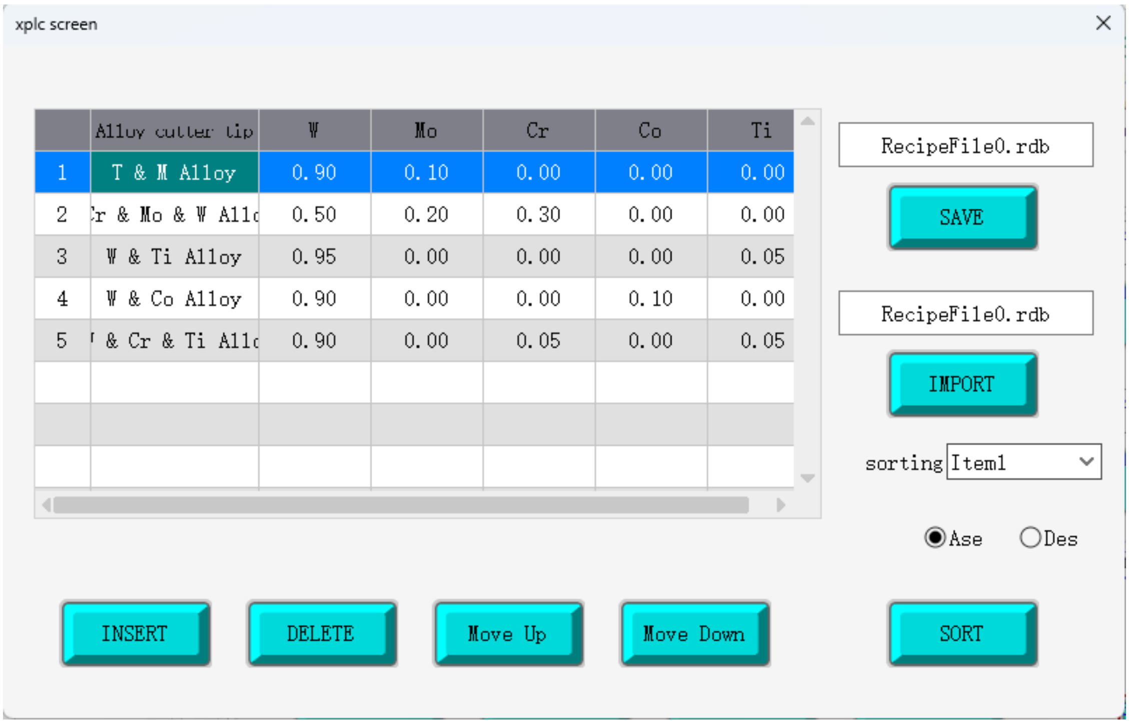
(6) Add “button” component, and bind it with corresponding action SUB.



Action	
Msg trigger call s...	
Action	Call Sub
Action when up	False
Action Sub	sub_SortLine
Position and size	

[Running Effects]

➢ After downloading, it will show the recipe group data.



	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	T & M Alloy	0.90	0.10	0.00	0.00	0.00
2	r & Mo & W All	0.50	0.20	0.30	0.00	0.00
3	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05
4	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
5	& Cr & Ti All	0.90	0.00	0.05	0.00	0.05

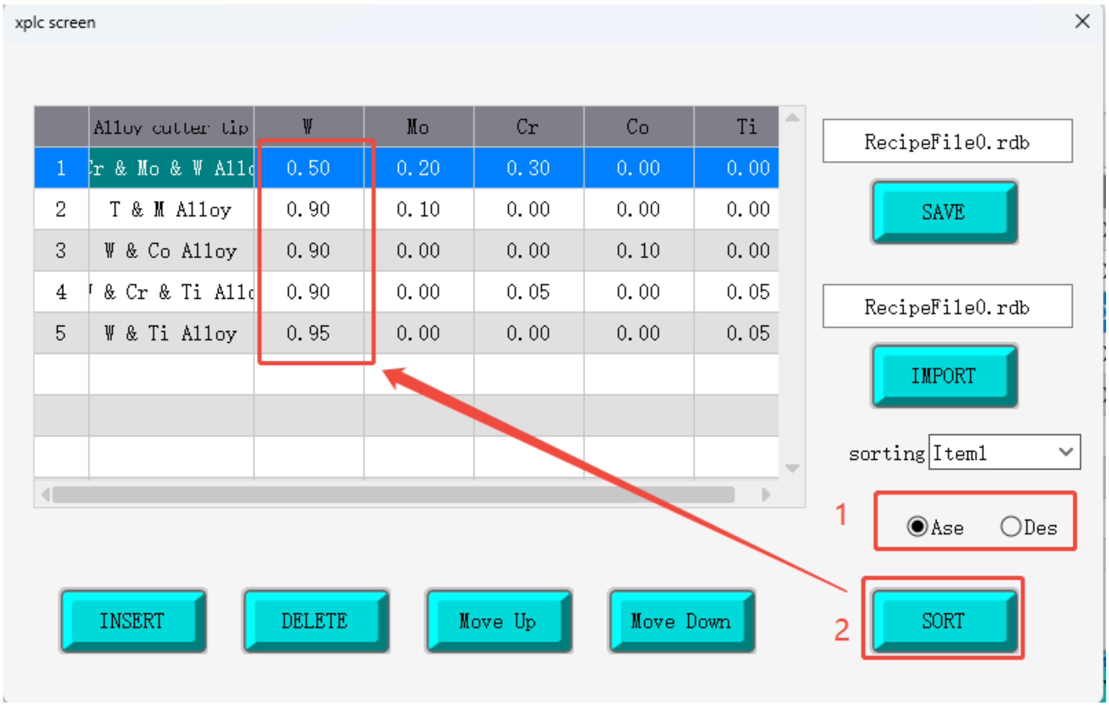
RecipeFile0.rdb

RecipeFile0.rdb

sorting Item1

● Ase ○ Des

➢ Select the first column (Item1) as the sorting option, and then select ascending. The recipe view will sort the data in the first column in ascending order.



xplc screen

	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	r & Mo & W All	0.50	0.20	0.30	0.00	0.00
2	T & M Alloy	0.90	0.10	0.00	0.00	0.00
3	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
4	& Cr & Ti All	0.90	0.00	0.05	0.00	0.05
5	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05

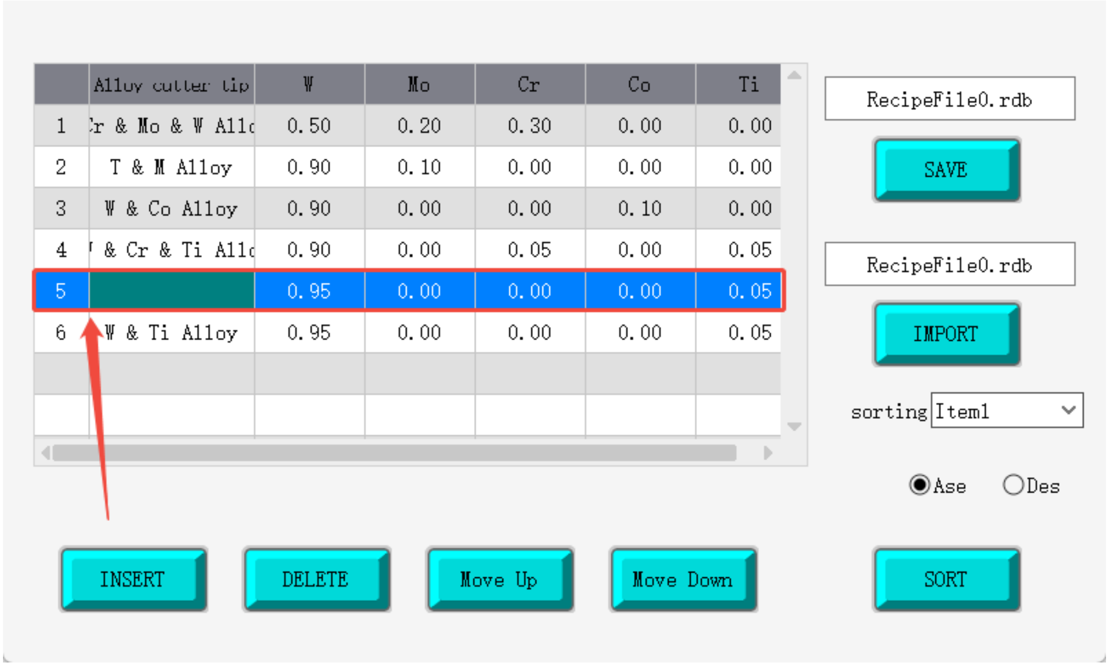
RecipeFile0.rdb

RecipeFile0.rdb

sorting Item1

● Ase ○ Des

➢ Click “insert” to insert one row data before the cursor.



	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	r & Mo & W All	0.50	0.20	0.30	0.00	0.00
2	T & M Alloy	0.90	0.10	0.00	0.00	0.00
3	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
4	& Cr & Ti All	0.90	0.00	0.05	0.00	0.05
5		0.95	0.00	0.00	0.00	0.05
6	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05

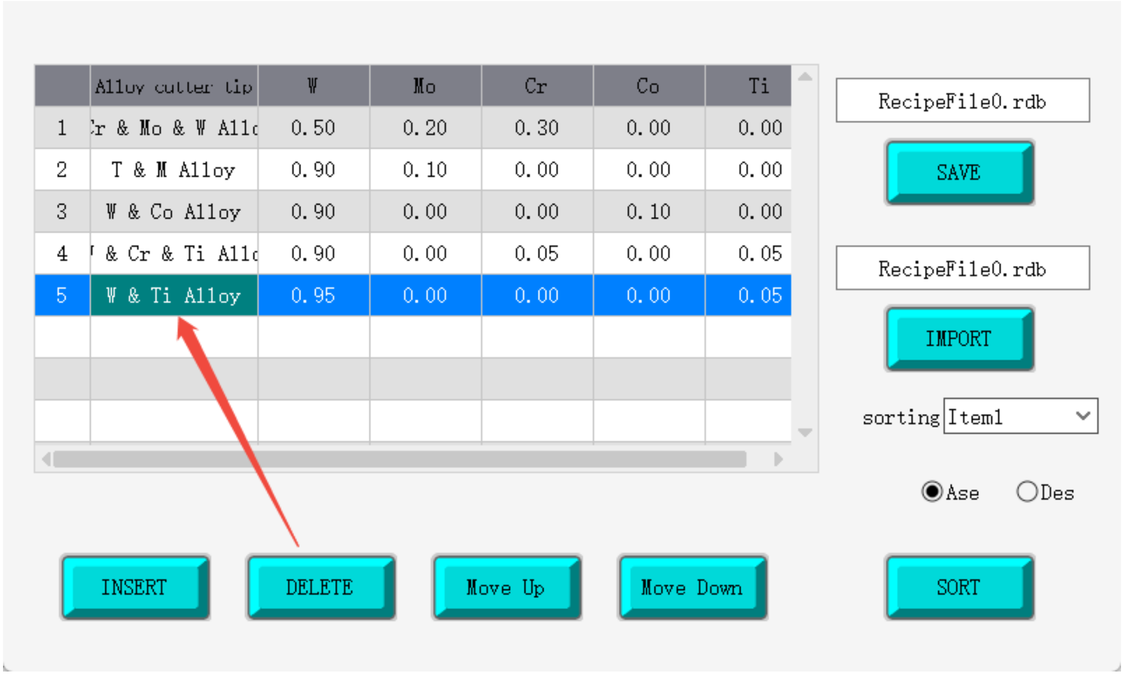
RecipeFile0.rdb

RecipeFile0.rdb

sorting Item1

● Ase ○ Des

➢ Click “delete” to delete the cursor row.



	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	Cr & Mo & W All	0.50	0.20	0.30	0.00	0.00
2	T & M Alloy	0.90	0.10	0.00	0.00	0.00
3	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
4	W & Cr & Ti All	0.90	0.00	0.05	0.00	0.05
5	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05

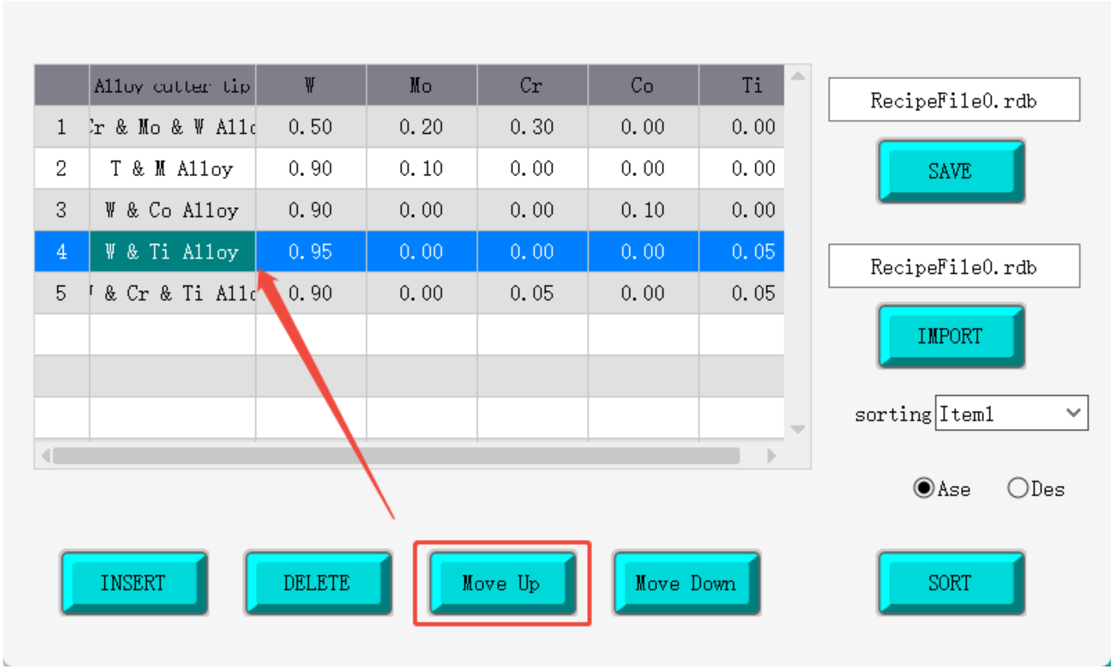
RecipeFile0.rdb

RecipeFile0.rdb

sorting Item1

● Ase ○ Des

- Click “move up” to move one row up.



	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	Cr & Mo & W All	0.50	0.20	0.30	0.00	0.00
2	T & M Alloy	0.90	0.10	0.00	0.00	0.00
3	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
4	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05
5	W & Cr & Ti All	0.90	0.00	0.05	0.00	0.05

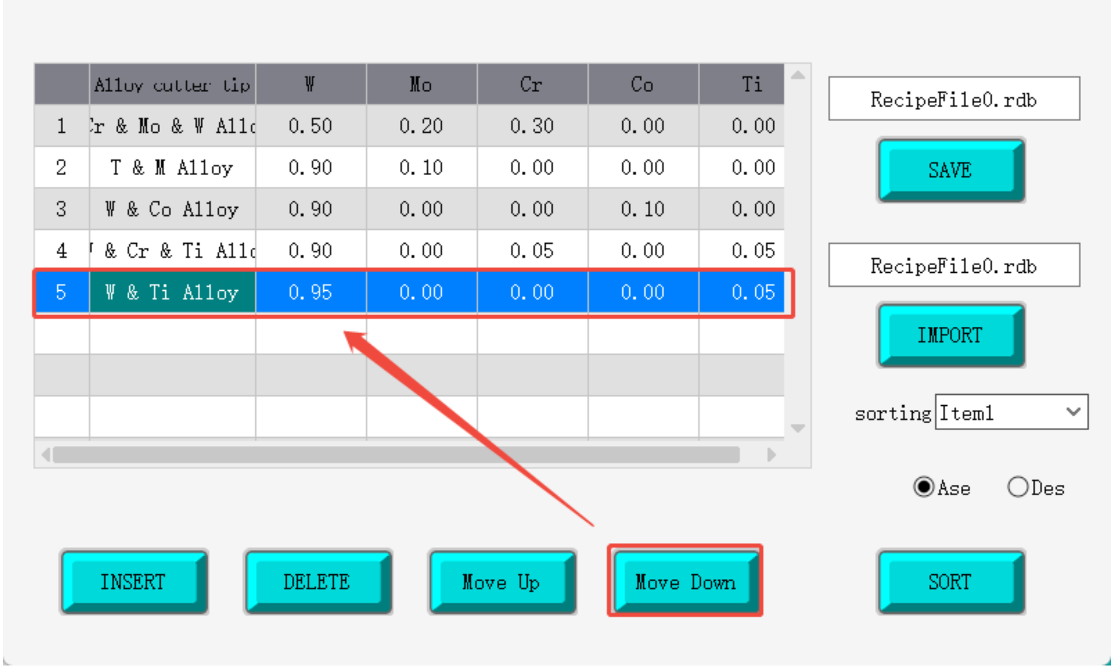
RecipeFile0.rdb

RecipeFile0.rdb

sorting Item1

● Ase ○ Des

- Click “move down” to move one row down.



	Alloy cutter tip	W	Mo	Cr	Co	Ti
1	:r & Mo & W Allc	0.50	0.20	0.30	0.00	0.00
2	T & M Alloy	0.90	0.10	0.00	0.00	0.00
3	W & Co Alloy	0.90	0.00	0.00	0.10	0.00
4	' & Cr & Ti Allc	0.90	0.00	0.05	0.00	0.05
5	W & Ti Alloy	0.95	0.00	0.00	0.00	0.05

RecipeFile0.rdb

RecipeFile0.rdb

sorting Item1

● Ase ○ Des

- Click “save” to save current recipe group into FLASH.

- Click "import" to import the recipe file.

8.11. Get Routines

More HMI routines, please refer to Zmotion Website: www.zmotionglobal.com.

Exact Path: Zmotion – Dosswnload – Development Examples – Quick Start – HMI.

Appendix

Virtual Key Description

Virtual Key	Key Value	Function Description
VKEY_0 ~VKEY9	48-57	Input digit / character 0-9
VKEY_PLUS	43	Input character +
VKEY_POINT	46	Input character . or decimal part
VKEY_MINUS	45	Input character – or invert current value
VKEY_ENTER	10	Input OK, and switch editing focus
VKEY_CLR	13	Clear all input info
VKEY_SPACE	32	Input space character “”
VKEY_TAB	9	Switch current edit focus
VKEY_BackSpace	8	Back space, delete one character from cursor place
VKEY_DEL	127	Delete
VKEY_ESC	27	Exit, and close virtual key mode
VKEY_SHIFT	171	Shift
VKEY_MENU	172	Reserved
VKEY_CONTROL	173	Reserved
VKEY_PAGE	174	Change page, reserved
VKEY_SWITCH	175	Reserved
VKEY_INS	176	Insert
VKEY_CAPS	177	Switch between uppercase and lowercase, effective when inputting compound keys
VKEY_EN_CH	178	Switch Chinese and English input method
VKEY_SYMBOL	179	Symbol (reserved)
ZKEY_F1	128	Quick button F1
ZKEY_F2	129	Quick button F2
ZKEY_F3	130	Quick button F3
ZKEY_F4	131	Quick button F4
ZKEY_F5	132	Quick button F5
ZKEY_F6	133	Quick button F6
ZKEY_F7	134	Quick button F7

ZKEY_F8	135	Quick button F8
ZKEY_START	140	ON button
ZKEY_STOP	141	STOP button
VKEY_LEFT	145	Direction Button, Move Left
VKEY_RIGHT	146	Direction Button, Move Right
VKEY_UP	147	Direction Button, Move Up
VKEY_DOWN	148	Direction Button, Move Down
ZKEY_1LEFT	150	JOG Button
ZKEY_1RIGHT	151	JOG Button
ZKEY_2LEFT	152	JOG Button
ZKEY_2RIGHT	153	JOG Button
ZKEY_3LEFT	154	JOG Button
ZKEY_3RIGHT	155	JOG Button
ZKEY_4LEFT	156	JOG Button
ZKEY_4RIGHT	157	JOG Button
ZKEY_5LEFT	158	JOG Button
ZKEY_5RIGHT	159	JOG Button
ZKEY_6LEFT	160	JOG Button
ZKEY_6RIGHT	161	JOG Button
VKEY_1_STAR	201	Composed button, 1*
VKEY_2_ABC	202	Composed button, 2abc / 2ABC, switch by VKEY_CAPS
VKEY_3_DEF	203	Composed button, 3def / 3DEF
VKEY_4_GHI	204	Composed button, 4ghi / 4GHI
VKEY_5_JKL	205	Composed button, 5jkl / 5JKL
VKEY_6_MNO	206	Composed button, 6mno /6MNO
VKEY_7_PQRS	207	Composed button, 7pqrs / 7PQRS
VKEY_8_TUV	208	Composed button, 8tuv / 8TUV
VKEY_9_WXYZ	209	Composed button, 9wxyz / 9WXYZ
VKEY_POINT_HASH	210	Composed button, .#
VKEY_A~VKEY_Z	65-90	Input “A-Z”
VKEY_a~VKEY_z	97-122	Input “a-z”
Others	0-127	Input as ASCII code

Error Code List

Error Code	Reason & Possible Solution		
5000	LCD No. Error	HMI running tasks > controller allowed.	See if it is more than allowed (allowed can be known from ?*max – max_hmi), more, please select other controllers.
5001	HMI File Error	Inside error	Please contact with us.
5002	LCD No. Conflict	Multiple HMI file use same LCD No.	See if there ae same LCD No.
5003	Unsupported Object	Inside error	Please contact with us.
5004	Insufficient Memory	Too small memory setting for VPLC7 or other controllers don't support.	1. For VPLC7XX, adjust "config --hmisize". 2. Contact with us.
5005	"Control" Error	One abnormal "layer" value set by PC software is transferred.	Contact with us.
5006	Window No. Exceed	You set too large window No.	1. Set it as a small one. 2. If it is full, contact us.
5007	Invalid Window No.	1. In base window, you opened one window that doesn't exist. 2. One invalid window is opened by the HMI_SHOW - WINDOW.	Check if you opened the base window that had been opened by the command already.
5008	HMI Content Error	Inside error	Contact with us.
5009	Same Window No.	Two HMI files or several windows use	See if they are same.

		same window No.	
5010	Object Property Lost	Inside error	Contact with us.
5011	>1 KeyboardShow in Keyboard		
5012	ACTION Type Error	Because action value is abnormal in PC configuration.	Contact with us.
5013	Too Many Events		
5014	Back to Last Window Failed		
5015	Can’t OFF Base Window		Check HMI file’s base window, and check script “close” logic.
5016	No Related Character in Font	This will not alarm, but the character that can’t be known will not be shown.	
5017	Must Use in HMI Task		
5018	Wrong Control Type	Because the control is operated by the command but it doesn’t support.	Check parameter configuration and related HMI window, see if they are consistent.
5019	Control ID Not Exist		
5020	Control ID Conflict	Different controls are set same component No.	Correct it.
5021	LCD No. Error	PC host computer error	PC host computer error
5022	No Valid LCD Found		
5023	LCD No Opened		
5024	LCD No Data		
5025	Program Reset		
5026	LCD Opened		
5027	Not Network LCD	PC host computer error (300 uses internal LCD No., the HMI with x uses network LCD No.)	
5028	Unsupported Compress	Reserved	Reserved
5029	Unsupported Color Depth	Controller doesn’t support that.	Contact with us.
5030	Unsupported Data Type	Inside error	Contact with us.

5031	Device No. Error		
5032	LCD_SET Can't Use	Reserved	
5033	Don't Set REDRAW in DRAW	In draw function, you used set_redraw command.	"set_redraw" is one refresh function that must be used in refresh function.
5034	DRAW Function Only Can Be DRAW	"draw" command is used in refresh function.	Draw command (usually the beginning of draw) must use in draw function.
5035	Can't Call in DRAW	The command that operates control is used in draw function.	Commands that operate control to show, control state can't be used in draw function.
5036	Fixed Inner LCD Resolution		
5037	LCD Resolution Beyond	Set resolution > controller allowed	You can check x and y parameter (?*max – max_hmi), that is, the resolution size.
5038	Library File Name Error	Called library file name is wrong while using text library.	Check the control "text library" or the command, correct the name.
5039	Too Many Characters		
5040	Object Property Lost	Inside error	Contact with us
5041	No KeyboardShow in keyboard		
5042	Too Many States		
5043	Unsupported Draw Property		
5044	Remote Communication Device Name Error		
5045	Remote Communication Data No Update		
5101	Invalid Date Format	You used invalid format while using SYSTEM command (like, not the format of % + letter).	Use correct data format: % + letter

5102	Control Not Exist	You use the command to operate the control that doesn't exist (such as, online change control text).	Use correct control ID.
5103	Too Many / Less Polygon Points	The polygon points is <2 / >32.	Check the point numbers, and better to use DRAW_POLYGONS.
5104	No Free Scroll Bar	You used auto-allocate ID syntax while initializing the scroll bar.	Note to release ID No. that is not used, if you need more, please contact with us.
5105	Invalid Scroll Bar ID	You called invalid scroll bar ID when using scroll bar command (such as, you used ID (>31)).	Use correct initialization scroll ID.
5106	Unsupported Function	Controller used unsupported HMI control / command.	Contact with us to see if there is new firmware.
5107	Not Load Image	You don't import the image while CAD command is executing.	Please use CAD control to import corresponding graphics, then do other CAD operations.
5108	File Broken	Usually appears when importing broken format of strong formats (bin file).	Reexport broken bin file.
5019	Menu Para Error		
5110	Not Enough table Space when Exporting, then Overflow		Make table space large through the command or change one controller if now it is the max space.

5111	Unsupported Data Type		Change as correct data type.
5112	Unsupported Control Type		Change control ID, and use correct control.
5113	Array Overflow	Exceed max value	Check array size, and see whether transferred array max value exceeds or not.
5114	Inside Error	Error in HMI inside	Contact with us.
5115	Channel Overflow		

Common Problems

Problems	Solutions
HMI interface cannot be displayed correctly.	Incorrect resolution is set. Please set the resolution according to the hardware requirements.
The page is not shown correctly after changing the HMI resolution.	After modifying the HMI resolution, you selected "No," then selected "Yes" again for “value resuming”, which caused an abnormal page resolution. You can undo this by pressing "Ctrl+Z," or by selecting "Don't Save" when closing RTSys.
“custom” can’t be refreshed & shown.	In refresh SUB, SET_REDRAW command is not used.
Fail to call SUB.	The SUB must be GLOBAL type.
Fail to operate window.	Operation type and window property type are inconsistent.
The screen is not bright, / brightness is insufficient.	Check the controller's power supply: for USB power, a high-quality cable must be used to ensure the computer's USB port has sufficient power. Otherwise, please use 24V power supply from the serial port.
The HMI displays garbled characters when using other languages (before downloading program).	➢ **Reason:** the text library doesn’t use corresponding font file and not call the code page. ➢ **Solution:** please import the font file and enter the code page for the language.

The HMI project displays garbled characters after downloading in a different language.	➢ **Reason:** the HMI encoding format wasn’t converted correctly. ➢ **Solution:** when the system is a single language, the HMI_CODEPAGE is used in the initialization SUB function to call the corresponding code page; when the system is multi-language, please use HMI_CODEPAGE to call the corresponding code page to refresh the entire system; then call corresponding font file in the text library.
Components cannot be displayed after download	This may be due to an outdated HMI version of the controller firmware, which does not support the latest component version. Higher-series models can be upgraded with firmware to support this, while lower-series models do not.