Zmotion Technology Co., ltd

RTHmi Programming Manual

V1.2.0

Content

Chapter I What is RTHmi Programming

RTHmi is the configuration design for Zmotion motion controller. Before, please confirm whether the controller supports RTHmi function or not, valid controller models will be shown later, you can check it or contact us directly.

Please download RTSys (how to use it, refer to RTSys user manual), then edit and debug RTHmi program in RTSys, controller should support RTHmi function, for simulator, its firmware version should above V5.20-20230706.

RTSys is upgraded based on ZDevelop software, which also supports Basic program, PLC program, Hmi comfiguration. You also can draw dynamically in touch screen. Please use the latest RTSys version, you can download from website (for the project that was created by higher version RTSys, not open in lower RTSys or ZDevelop).

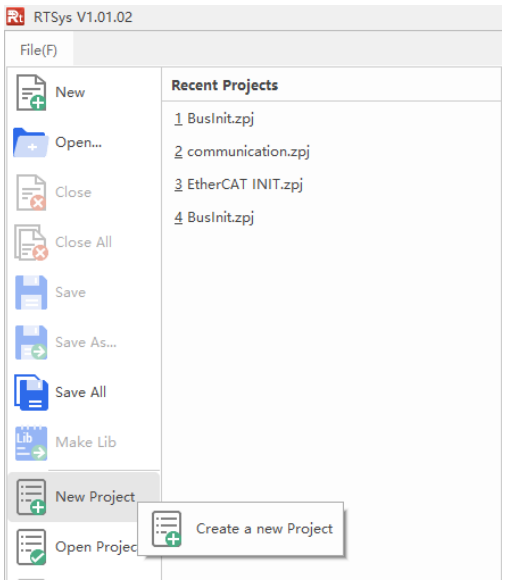
1.1. How RTHMI Programming is in RTSys

1.1.1. Hmi Development

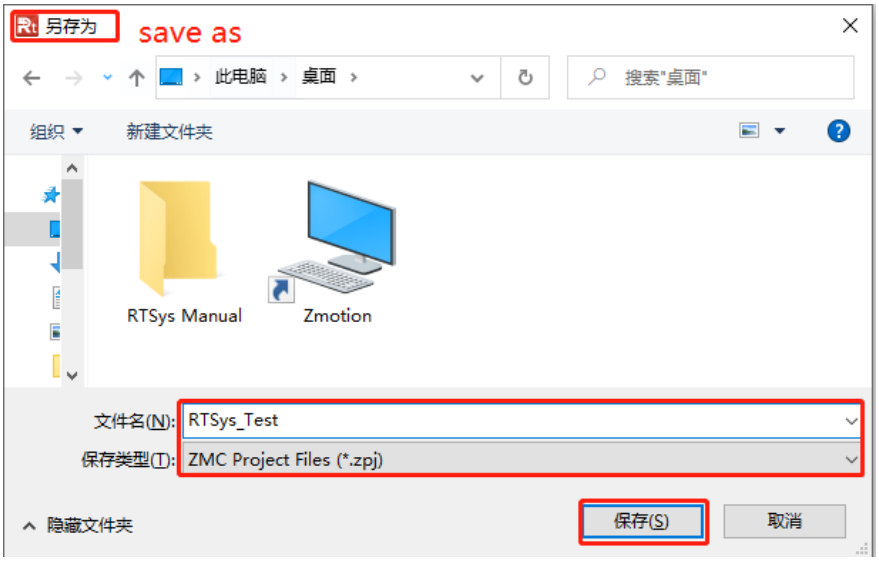
At first, build a new folder in your PC to save the project that you will create. Here take RTSys V1.01.02 as example, if you need to update, please refer to "RTSys User Manual". And there are many HMI routines in Zmotion website, you can download them to check and learm.

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.

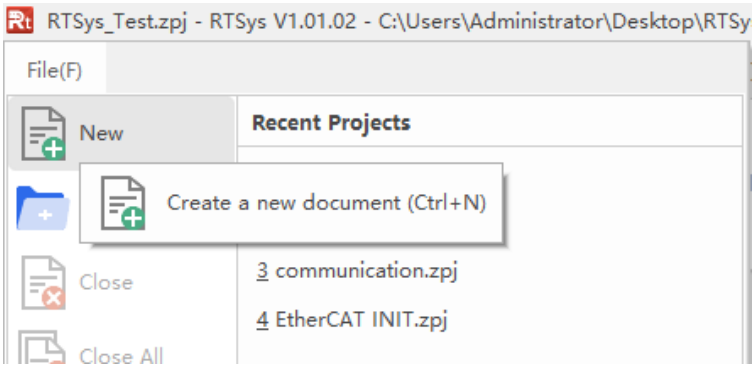
1. Build one New Project: "File" in "Menu" → "New Project".



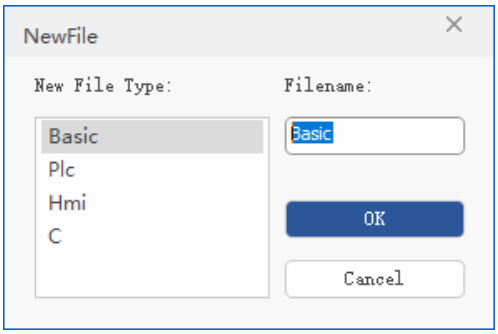
Click “New Project”, then “Save as…” will be jumped, select the folder (that was created just now), and open it, then enter folder’s name and save the project, pay attention to the suffix should be “.zpj”.



2. Build New File: File – New



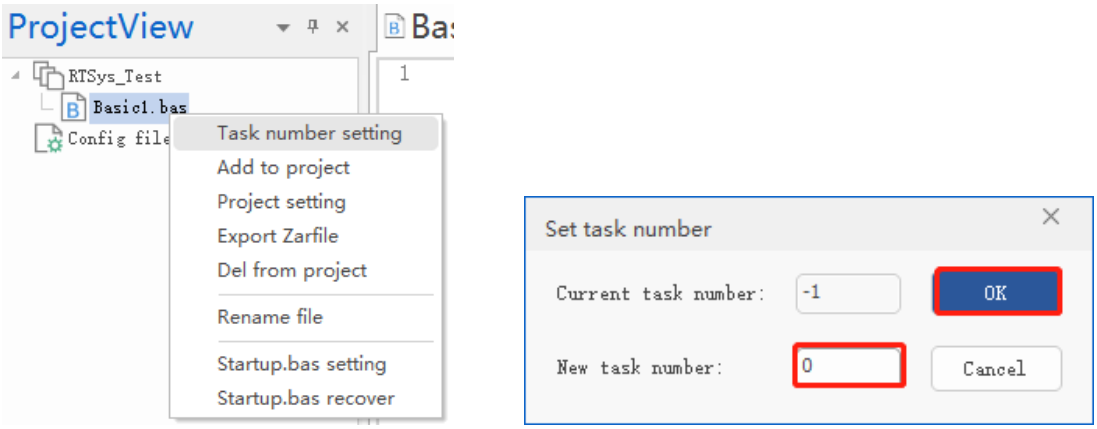
After clicking “New File”, below popping window will appear, which supports Basic/PLC/Hmi hybrid programming. Here selects the “Basic” file type and click “OK”. The name can be modified.



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, 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.

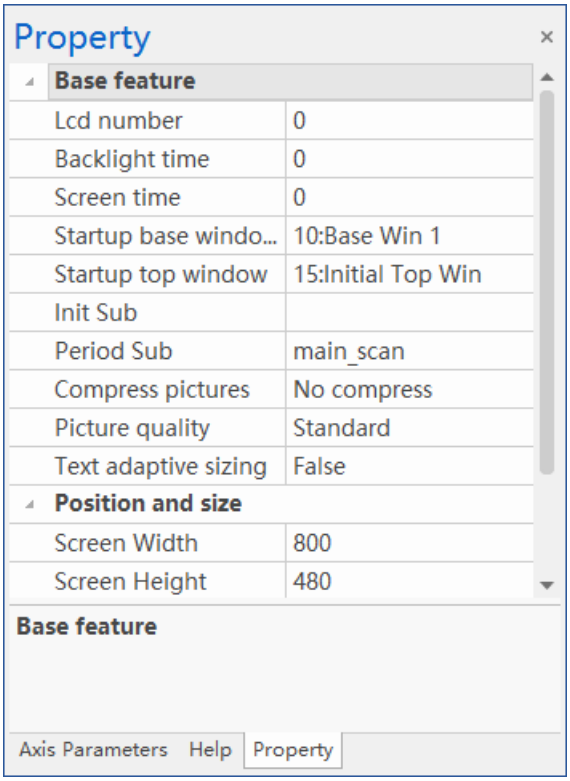
- ✧ Support setting multi-file task No., and one file must be set, otherwise, the program can’t be run.
- ✧ The files that set the auto run No. will run at the same time, and the value can be any one, no priority, but the max one can’t exceed controller’s maximum task value, you can check in controller state or send “?*max” or refer to user manual.

4. Edit HMI Program

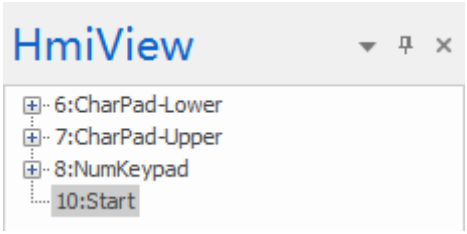
Before that, open “Hmi Setting” window at first: switch to HMI programming window, then find “HMI” in menu, then find “Hmi Setting” (or open HMI file, click empty place outside the window), according to needed teach pendant’s size, set horizontal & vertical resolution (this must be confirmed in advance!).

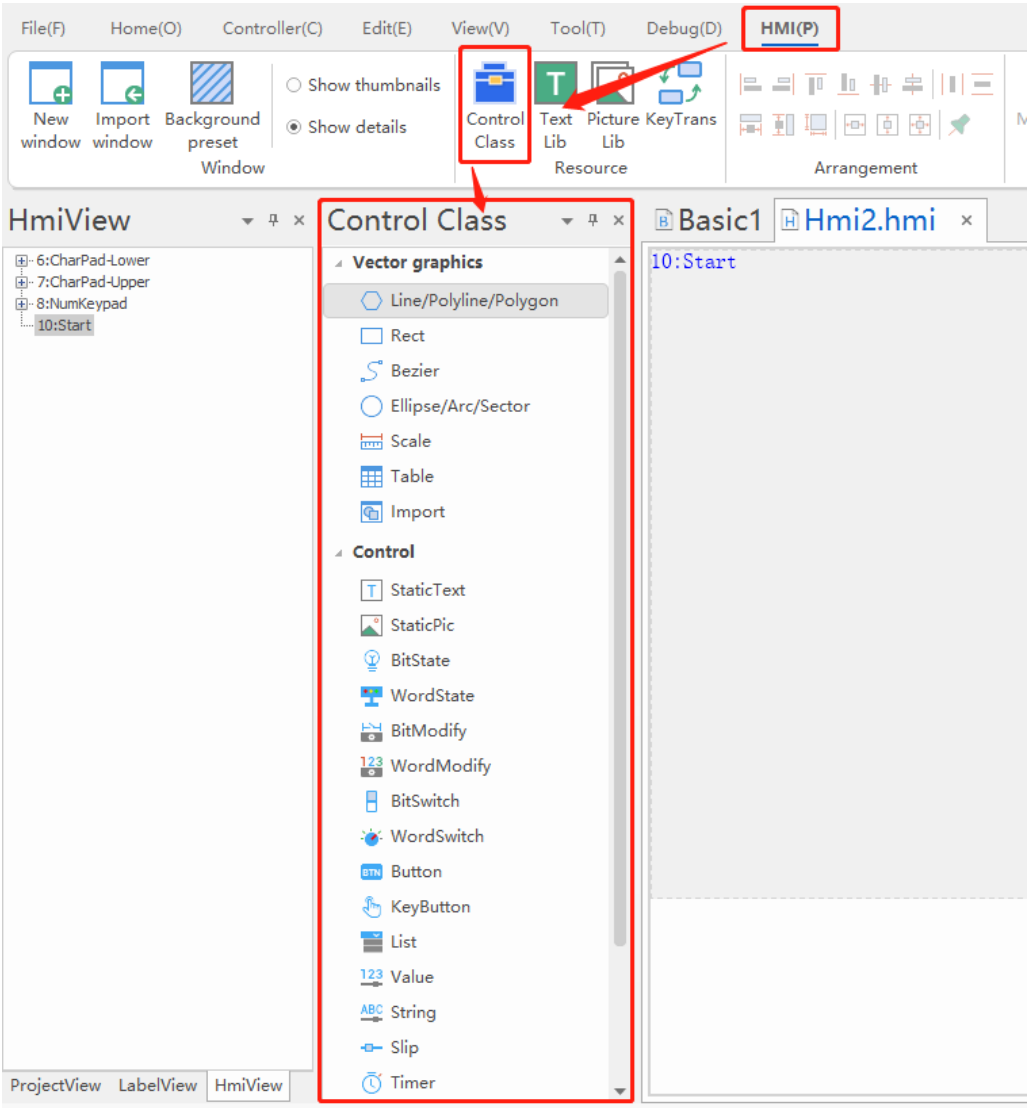
And see whether you need to set initial function and period function, select SUB subfunction that is global defined sub edited in BASIC.

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

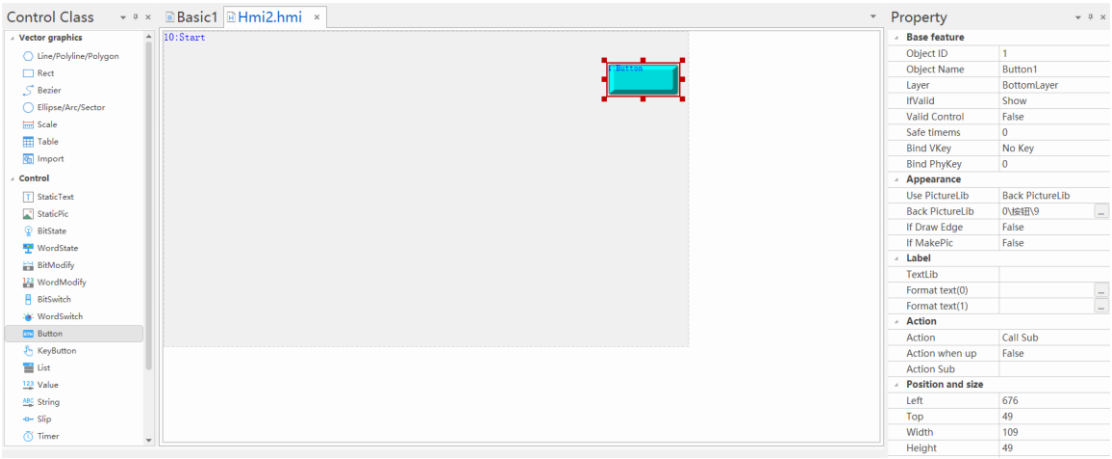


After basic setting in above window, you can add HMI components, added components will be shown in HMI window: in HMI file menu, find “control class”, there are many optional components (generally, after creating one HMI file, 3 soft keyboard windows and one initial window “10:Start” will be created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



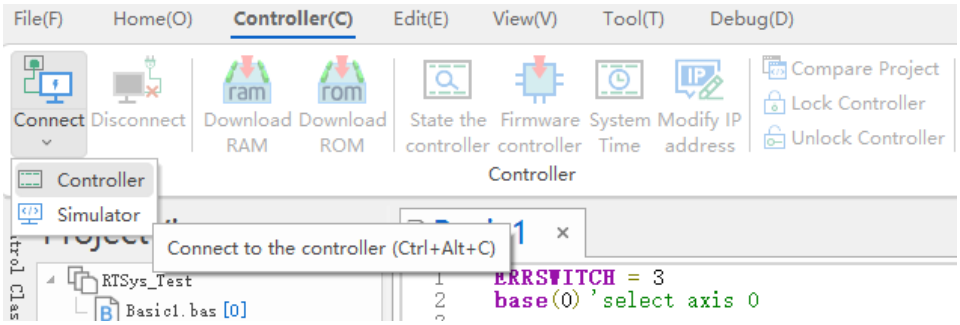


In “Control Class”, you can select the object, drag one in the HMI size range directly, and each one object’s property can be opened to set needed parameters.



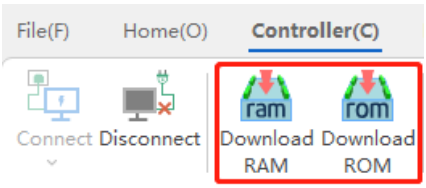
5. Connect to Controller / Simulator

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.

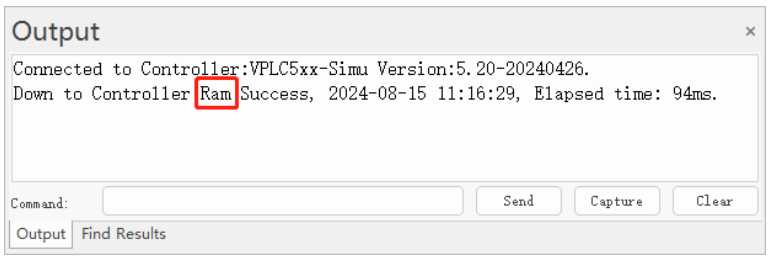


6. Download Program

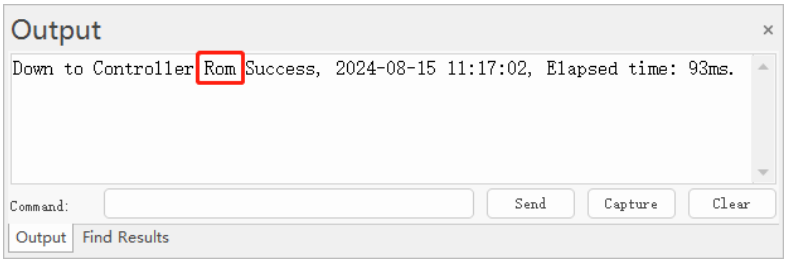
In menu bar, there are two downloading methods, **Download RAM / Download ROM**. After downloaded, also, there will tell you corresponding information in “output” window, then, program will run automatically.



--succeed in downloading RAM--



--succeed in downloading ROM--



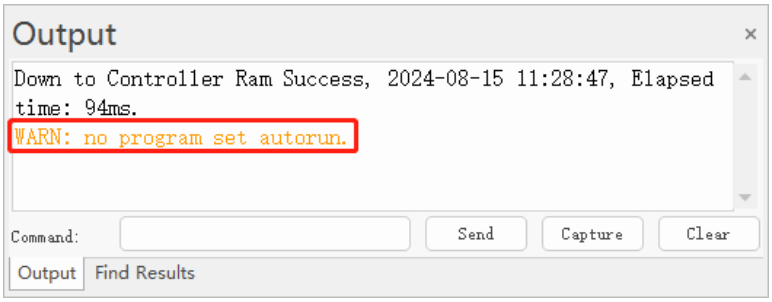
--download RAM VS download ROM--

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.

NOTES:

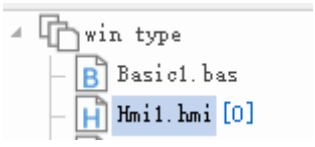
- When open the project, please select the zpj file. If only the Bas file is opened, program can't be downloaded into controller.
- ZMC00x series controllers don't support Download RAM.
- When project is not built, that is, you only open the file, it can't be downloaded into controller.
- AutoRun No., 0 means the task No., that is, it runs as task 0. Task No. doesn't have priority.
- If there is only one file, AutoRun No. must be set. If there are several files, one No. must be set. If all files of whole project are not set the task No., when downloading into controller, system will give the indication: 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 (for basic file, it can be set or not), then you can run it. And only one HMI file can be run at most, if two files run at the same time, it will report an error.



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.

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

1.1.3. Network Display Screen

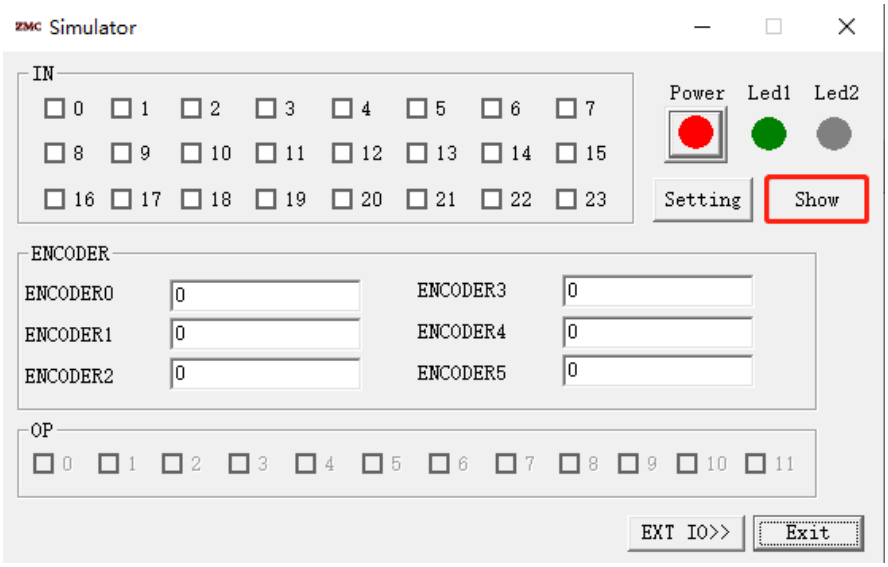
For RTHmi, you can use your computer or other touch screens as display screen through ethernet.

When the controller supports multiple screens, please select which LCD No. you use in “Hmi Setting”. If you want to 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 (this window is Chinese by default, please switch it by clicking “设置” – “语言”, then select English, restart it, no need to set it again), you can click "show" to run the HMI, and corresponding interface will be shown in “xplc screen”.

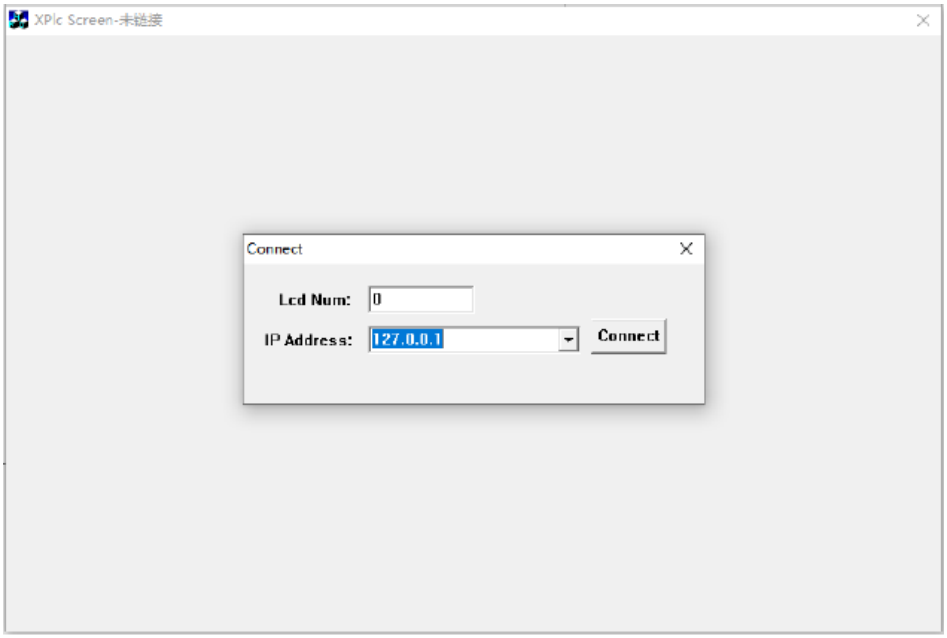
When the controller and teaching box are not connected, it can be debugged after connecting to the simulator.



1.1.5. Xplcterm Running

When there is no real touch screen, use Zmotion small simulation tool “xplc screen” to simulate the operation, and the PC can be used as a man-machine interface to connect to controller or simulator.

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.



1.1.6. Controllers with HMI function

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

Controllers that support RTHmi programming: ZMC4XX and above (below ZMC4XX controller, don’t support RTHmi, except the controller that is with H, for example, XPLC120H). For some controllers, it needs to upgrade the firmware, please contact us. (for low series controllers, you can try to use ZDevelop).

1.1.7. Valid 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

1.2. Common Problems

When a program movement error occurs, “Command and Output” window will display error information. If the error information is not seen, you can enter “?*task” command in “output” window, or in the “Debug” – “Troubleshooting” window to check the error information again. Double-click the error information, it will automatically switch to the error location of the program.

Problem	Possible Reason
HMI interface can’t be shown normally.	Incorrect resolution, please set it according to hardware requirements.
After changing the HMI resolution, the page displays abnormally.	After modifying the HMI resolution, you select "No", and then select "Yes" to restore the value, resulting in abnormal page resolution. At this time, you can use "ctrl + z" to withdraw, or select "Not Save" when closing RTSys.
“Custom” object can’t refresh.	In refresh sub, SET_REDRAW command isn’t used.
Object fails to call function.	Sub function must be GLOBAL type.
Fail to operate the window.	Window operated type and window property type is

	inconsistent.
The display screen is not ON, or the brightness is not enough.	Check the power supply of the controller: the USB power supply must use a high-quality cable and ensure that there is enough power for PC's USB, otherwise, please use the 24V power supply of the serial port.
Fail to communicate	Check the cable.

HMI Menu Bar

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

2.1. Windows

- **New window:** new build or several windows in current project, for details, please refer to Chapter III "How to Create Window".
- **Import window:** in current project, import other project or created HMI window, for details, please refer to Chapter III "How to Import Window".
- **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".

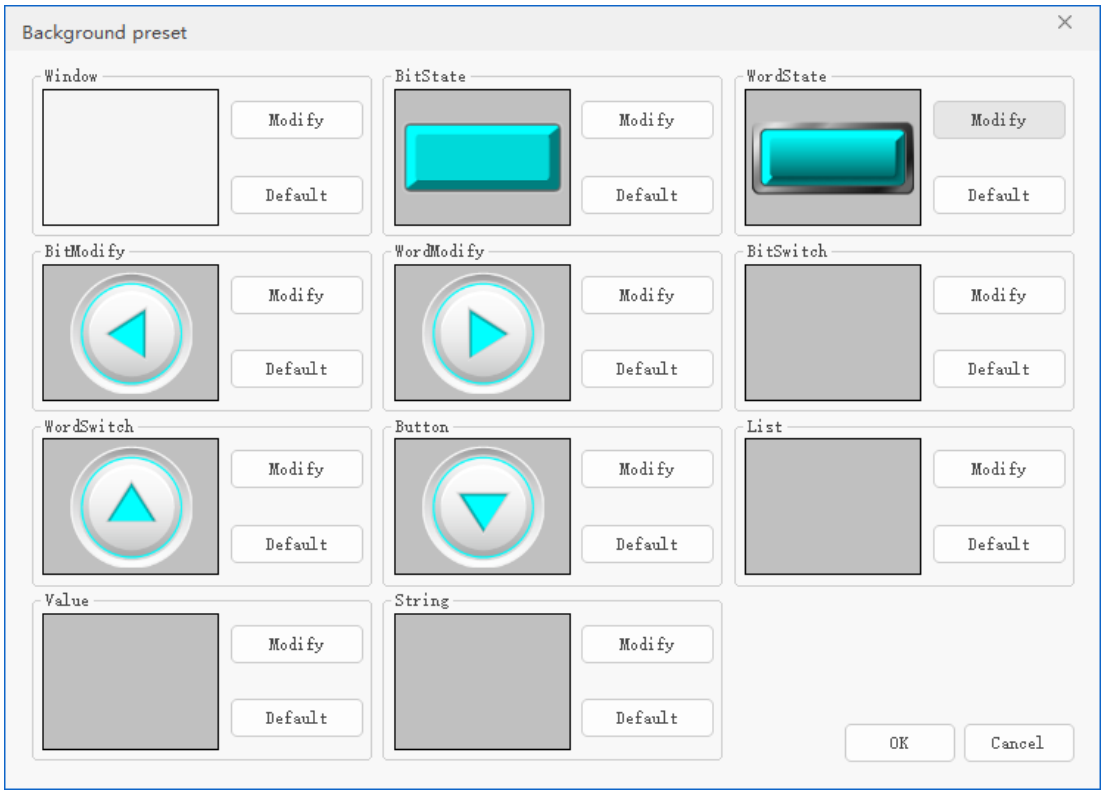
  --How to Operate: Select needed object / window, then click "modify", there are two style selections.

  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.

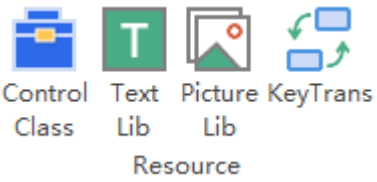
  For window background preset, use method 2, and set state 0. It takes effect after building new window.



- **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].


HmiView
6:CharPad-L
7:CharPad-U
8:NumKeypa
10:Start
11:
ProjectVi...
LabelView
HmiView
HmiView
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:
ProjectVi...
LabelView
HmiView

2.2. Resource


Control Class
Text Lib
Picture Lib
KeyTrans
Resource

2.2.1. Control Class

"Control Class" is mainly used to save all kinds of HMI objects. When developing HMI, you can add components in this window. It is opened and closed by "View" – "Control Class". More information, please refer to Chapter IV "HMI Component".

2.2.2. 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)

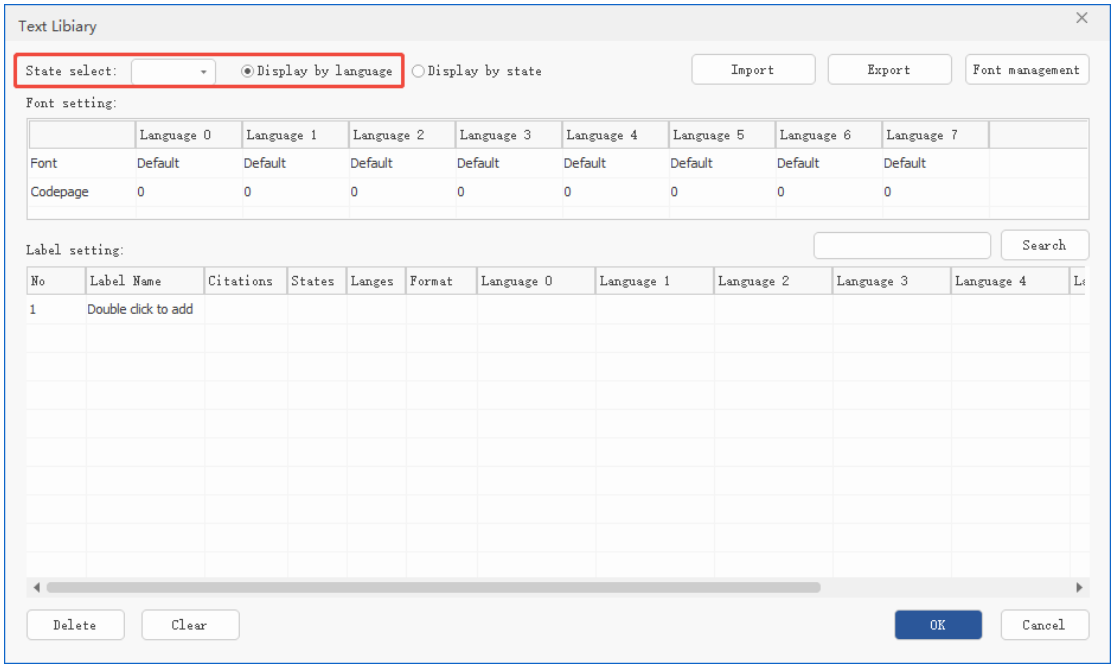
Function	**Description**	
State select	Display when "display by language" is selected, to switch the state, different text contents under different states.	
Lang select	Display when "display by state" is selected, to switch the state, different text contents under different languages.	
Display by language	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 setting	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).

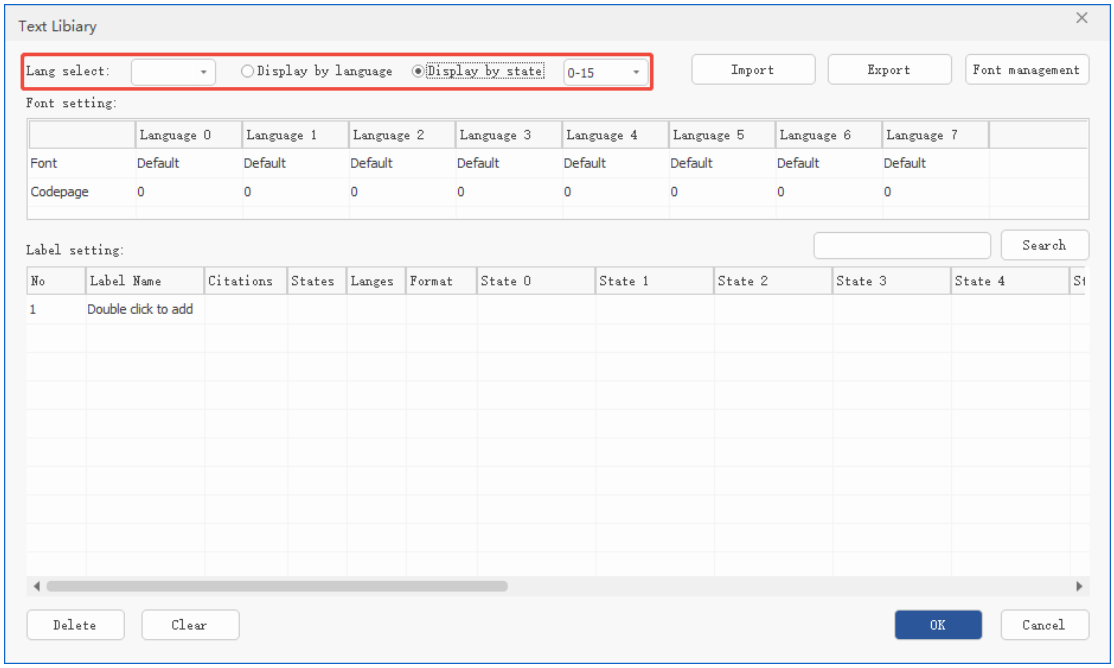
	States	Set text library's states.
	Langes	Set how many languages you will use.
	Format	Set language text format.
	Language 0-7	For "display by language", write text content
	State 0-15	For "display by state", write text content, state can be 0-15 / 16-31 / … / 240-255.

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] if the font is ttf (True Type Font) format, you can fill 65001 (UTF-8 code) in "codepage".

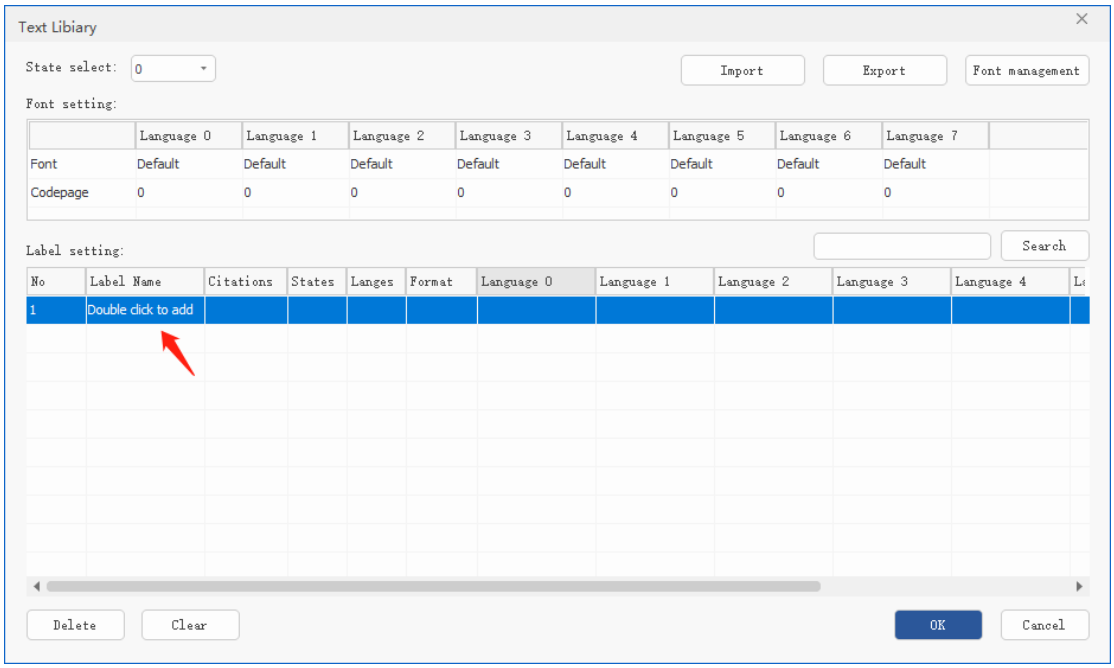




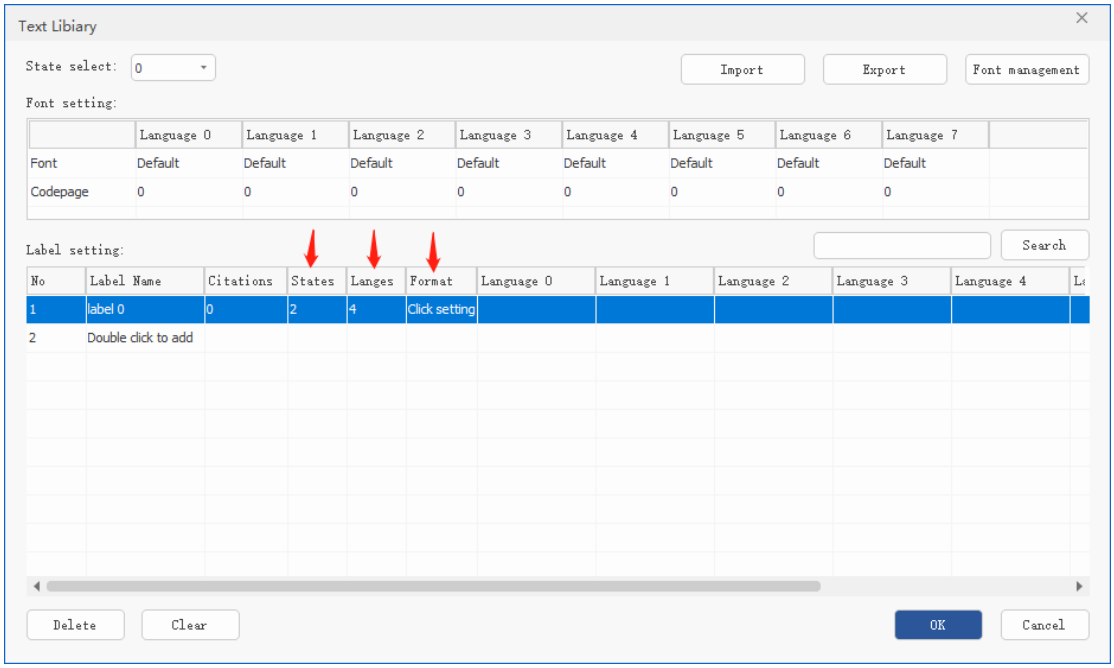
(2) How to Operate

1) Text Lib Setting

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

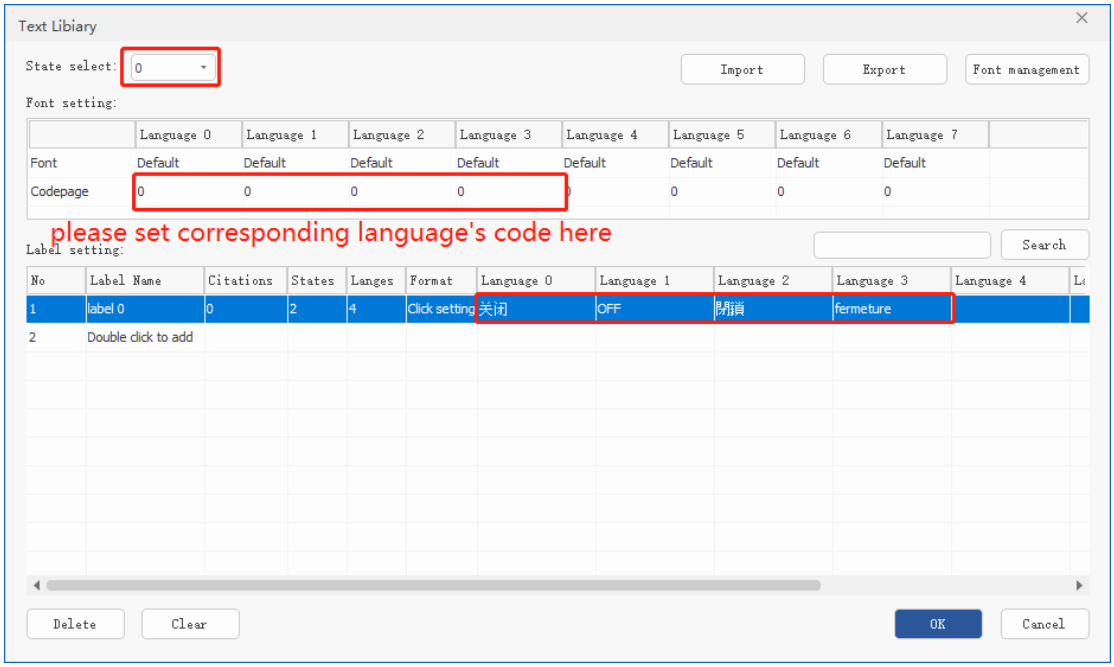


B. Set the required number of states and languages, and the format of the text.



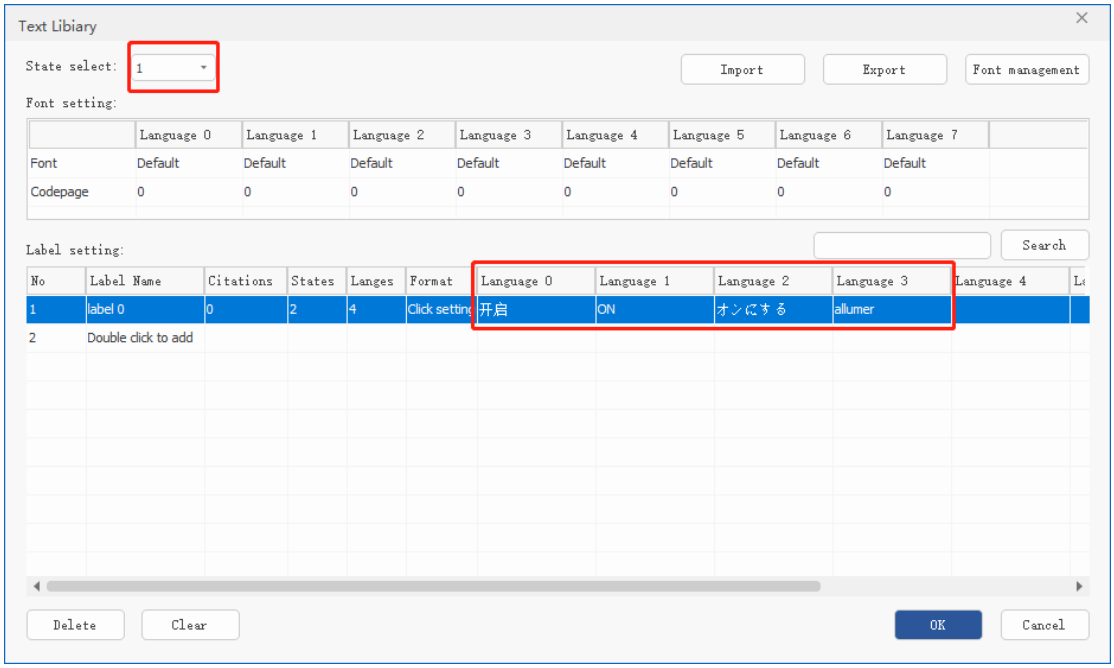
C. Select the state, and add the text content to be displayed to the corresponding number of languages in the selected state.

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



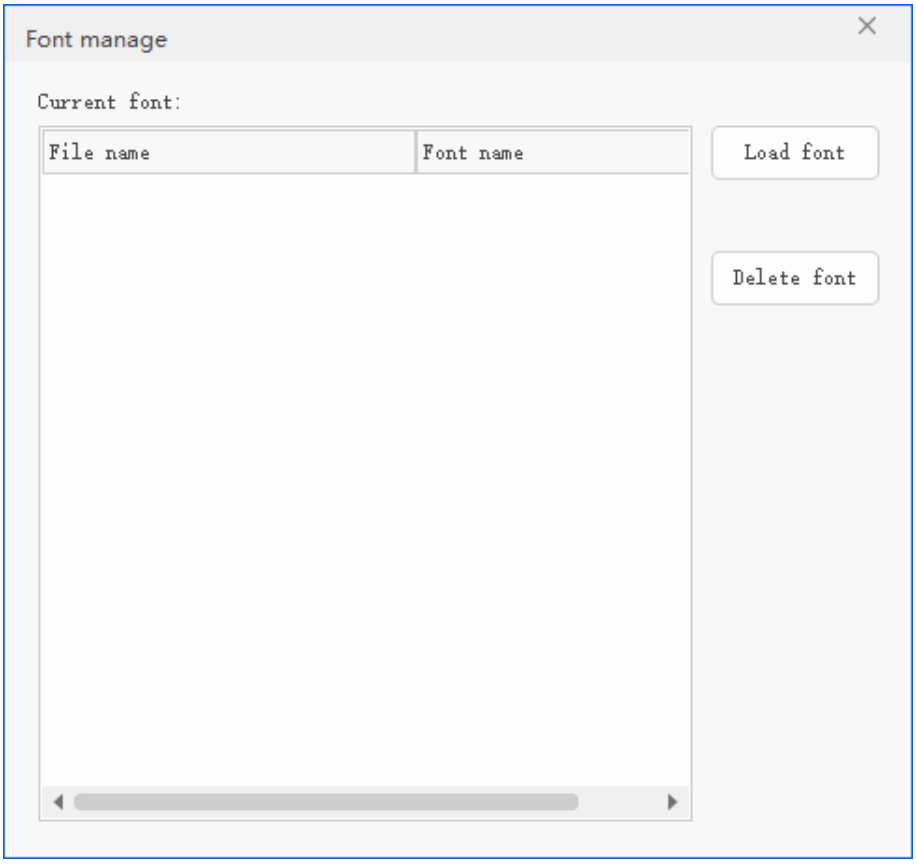
Note: under Chinese system, for the text showing in Chinese, no need to enter the codepage, but for others, please enter, for example, Japanese, it should be 932.

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).

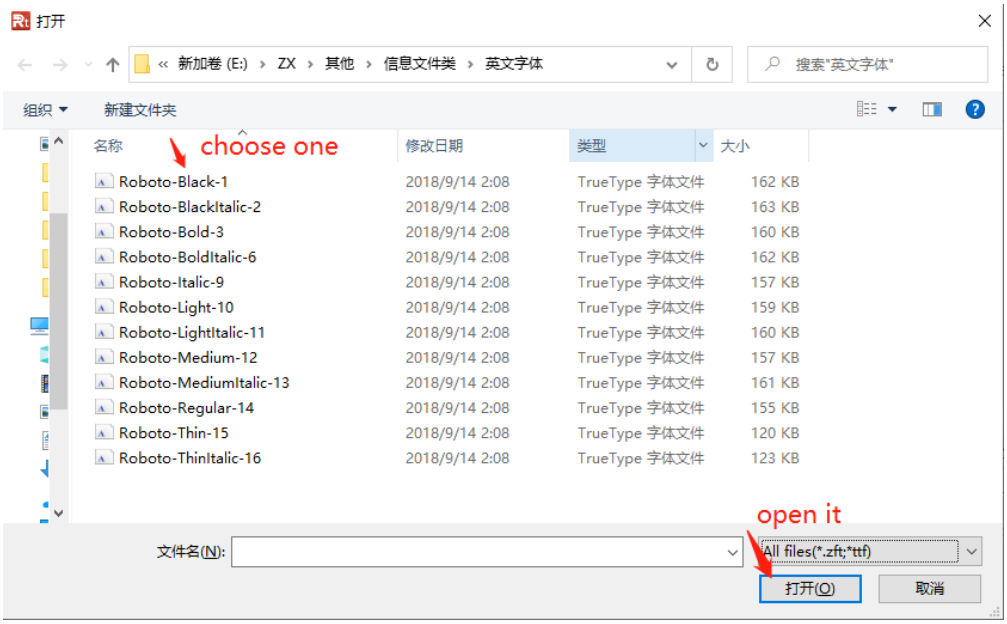


E. After setting the label and language text, if no need to set font, click [OK] to save.

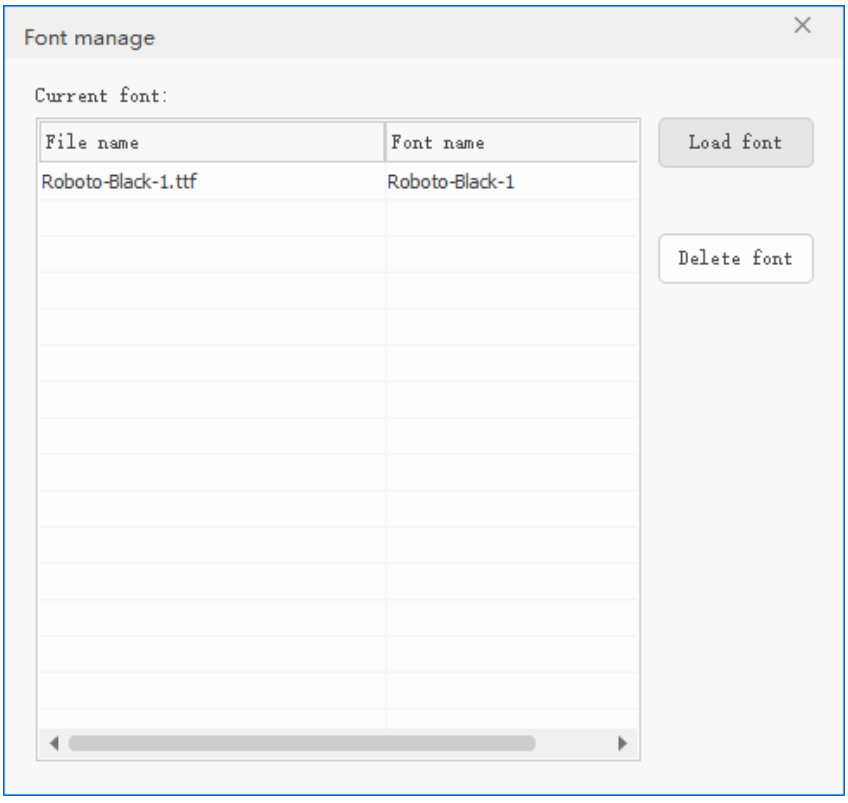
F. If you need to set the font, please add the font file to the project firstly. After importing the font, you can set the font for each language in the [Font Settings], then call the corresponding text library and run it to display.



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



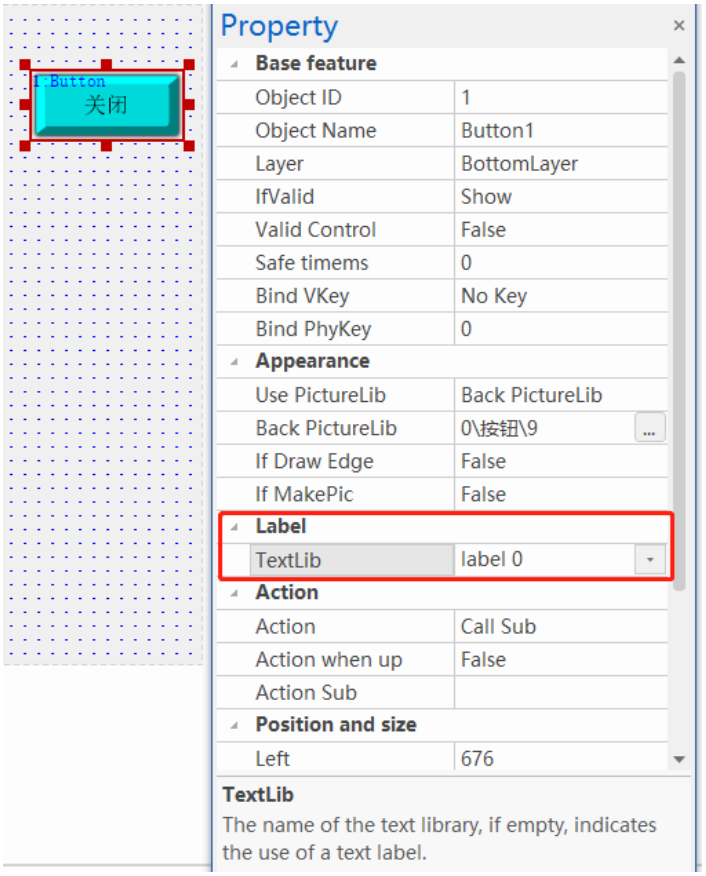
H. After loading, corresponding file name and font name will be shown in the list (only for .ttf format, font name can be shown).



I. After that, you can set each language’s font in “font setting”, when you call it, showing in set font.

2) Text Lib Calling

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



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.

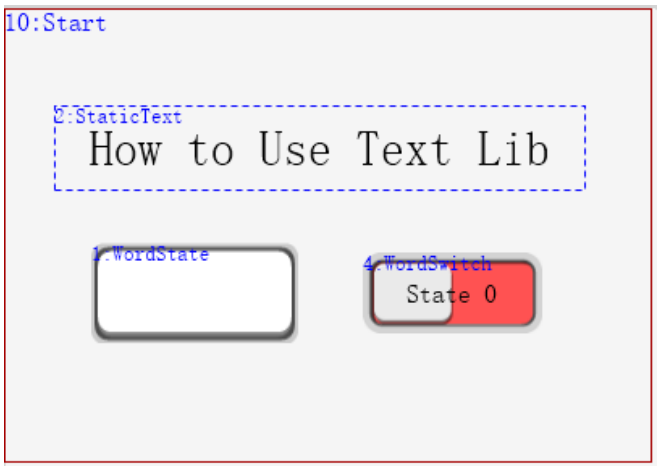
Next, let’s see one routine:

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:

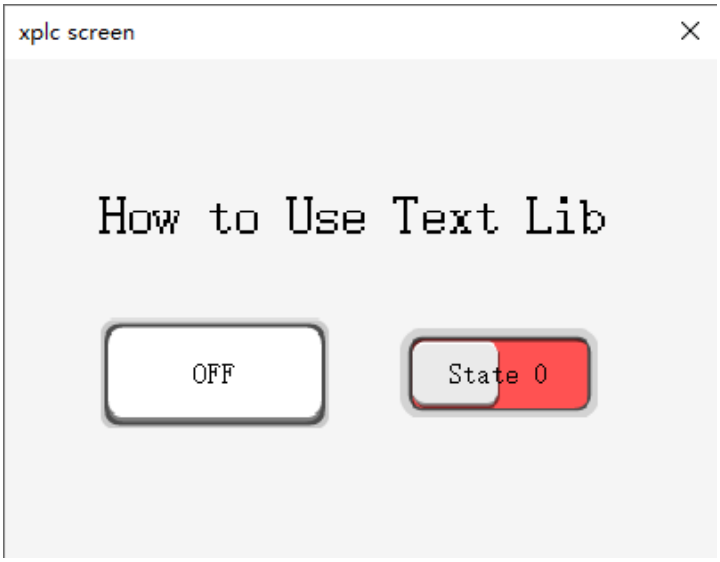
  --in “property” window, set state num as 2, and add corresponding texts--

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

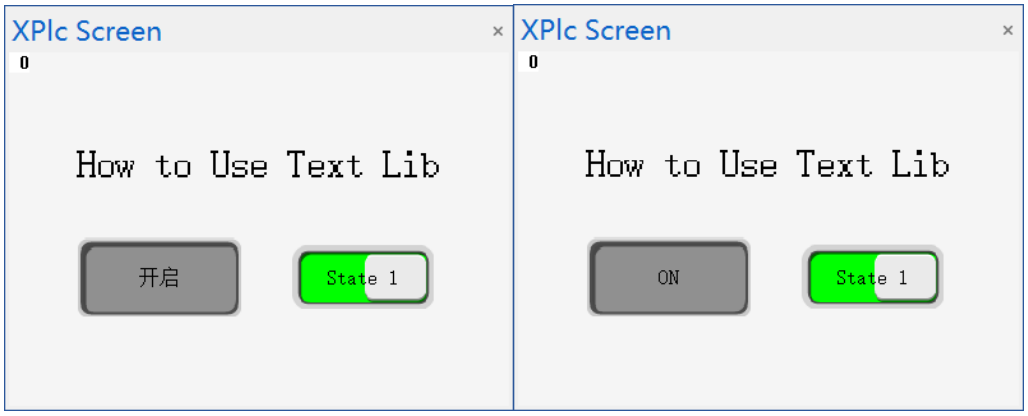
- **How the Effect is:**

  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 the language, let’s switch to language 1, it will be “OFF”.



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



2.2.3. Picture Lib

(1) What is “Picture Lib”

A library that integrates and stores Hmi component style images or custom 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.

Note: The maximum number of image libraries is 512!

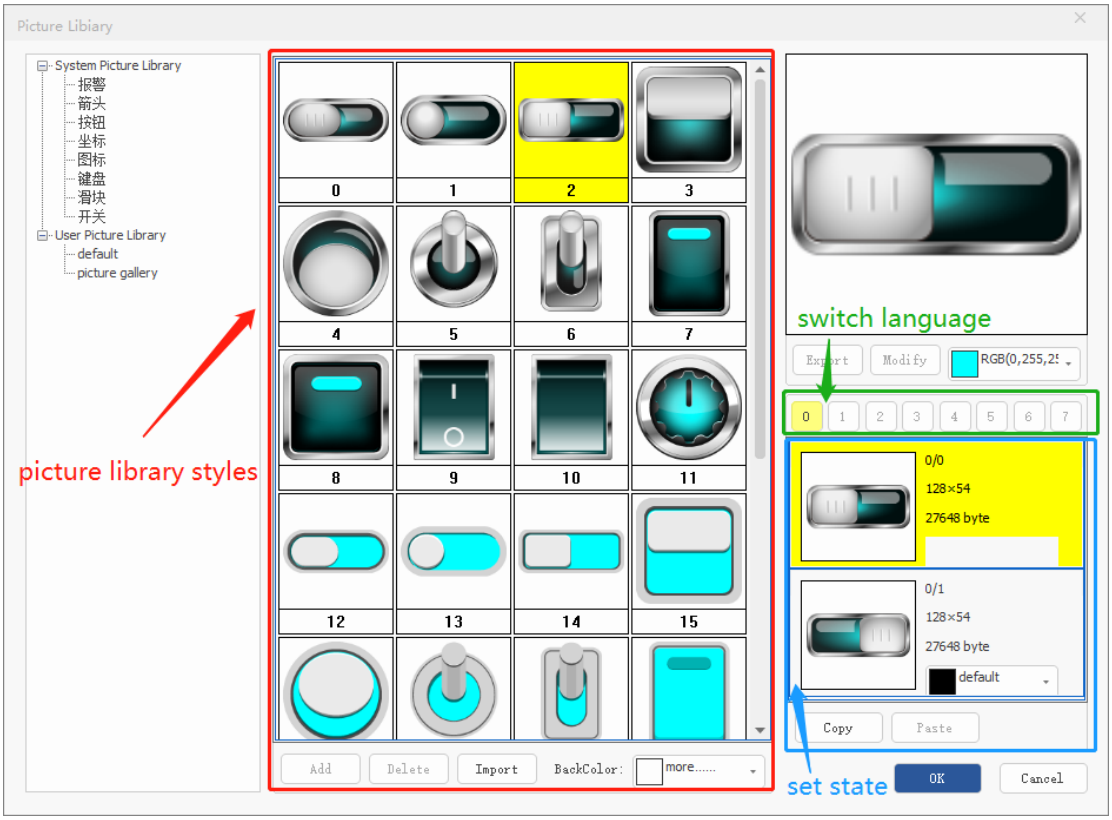
- System Picture Library

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.

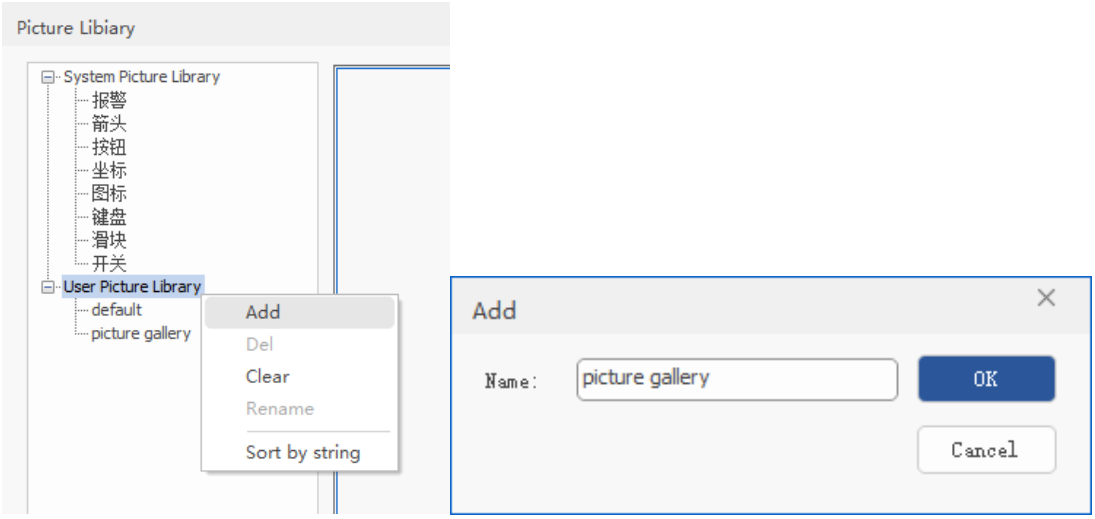


(2) How to Use (take the example of "switch on and off the led")

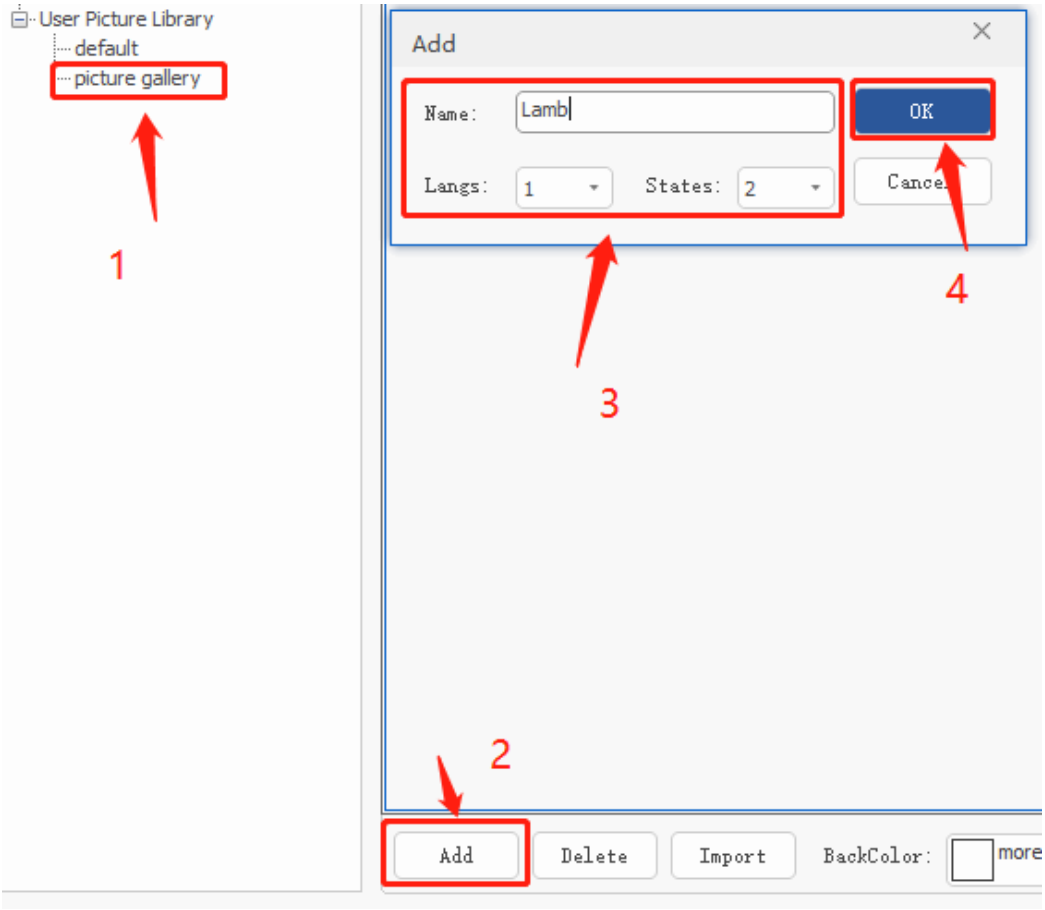
Step 1: [How to Build New Picture Lib]

A. Import "Lamb" Picture

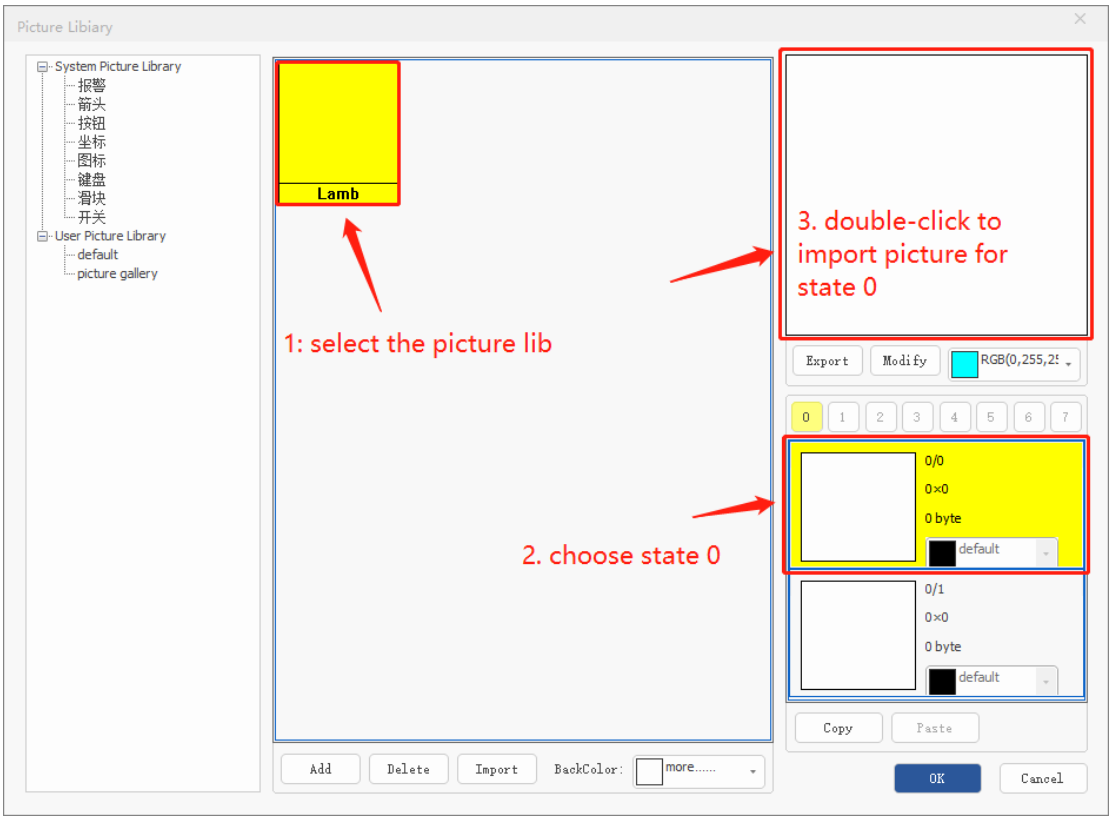
a. Right-click "user picture lib", and select "add", then enter new picture lib name in the popped window, then click OK.



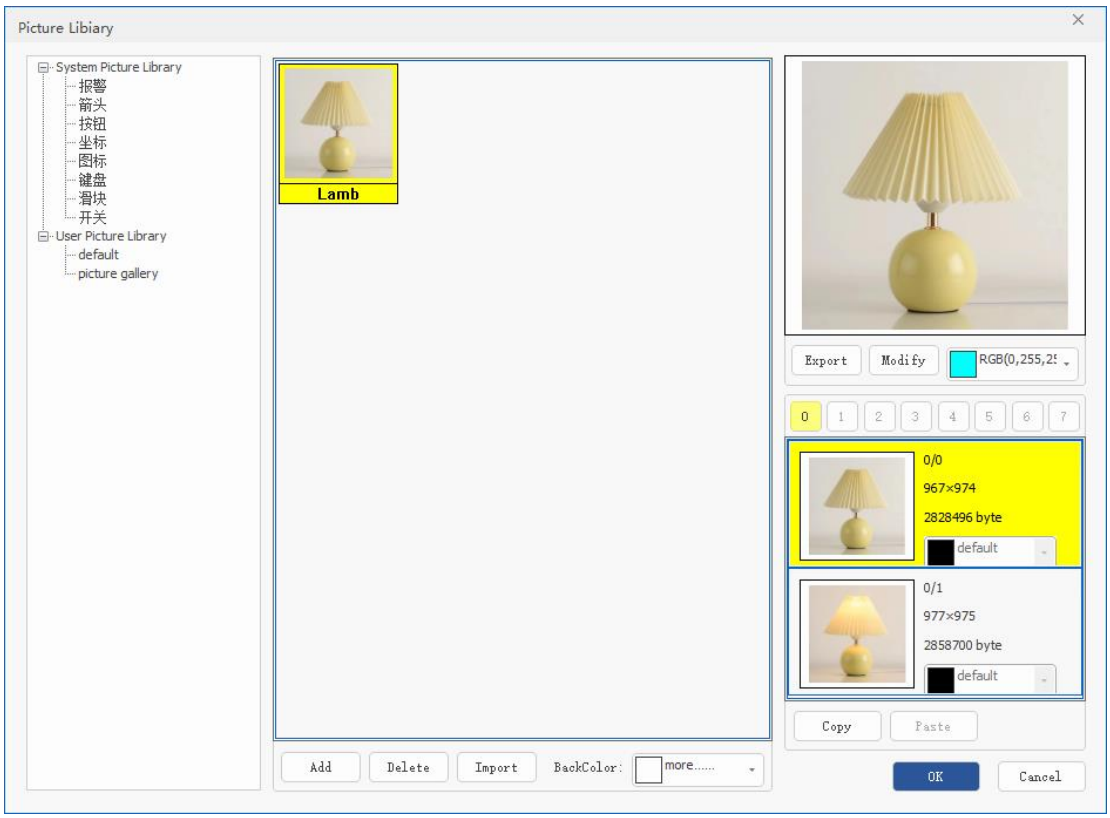
b. Double-click new created picture lib “picture gallery”, then click “add”, in new appeared window, enter the name, set well corresponding language numbers and state numbers.



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.

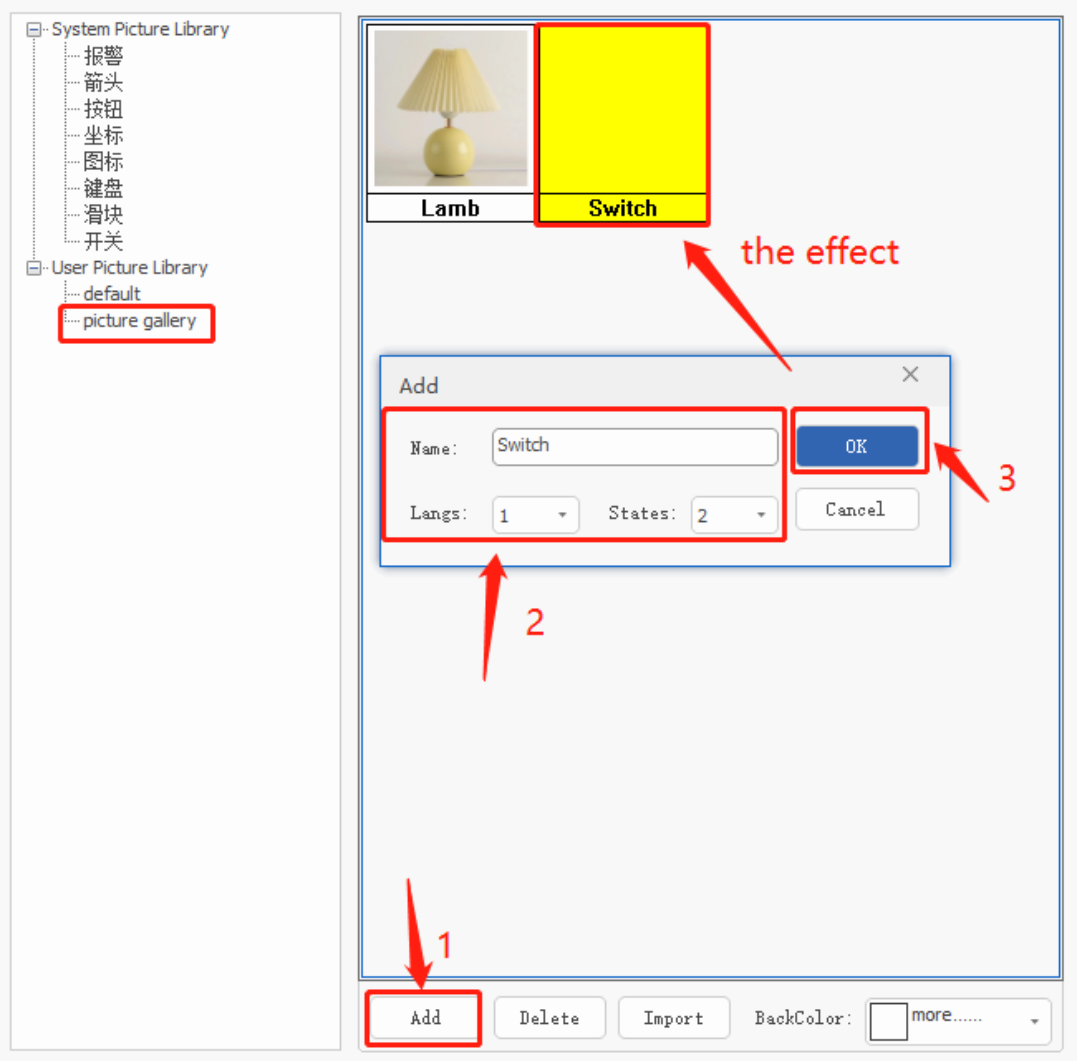


d. According to “c” to import the picture for state 0, and keep adding the picture of state 1. When both added, please click OK.

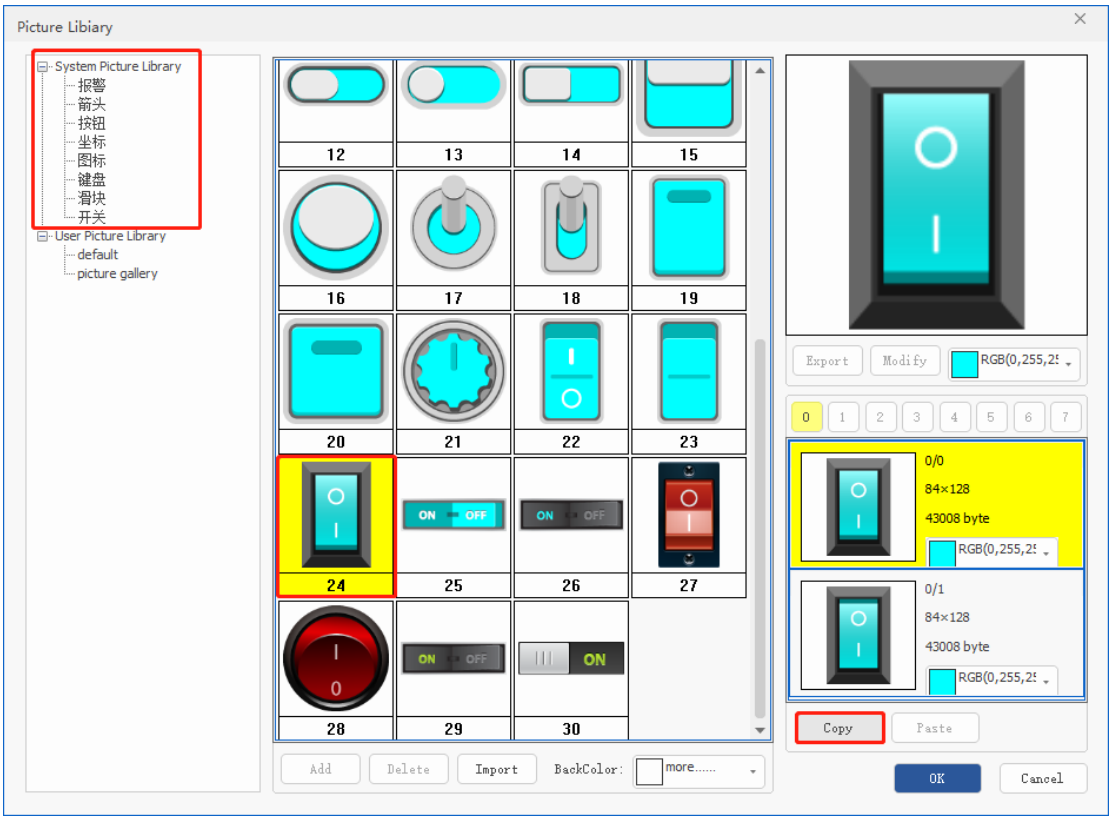


B. Copy Switch Button Styles

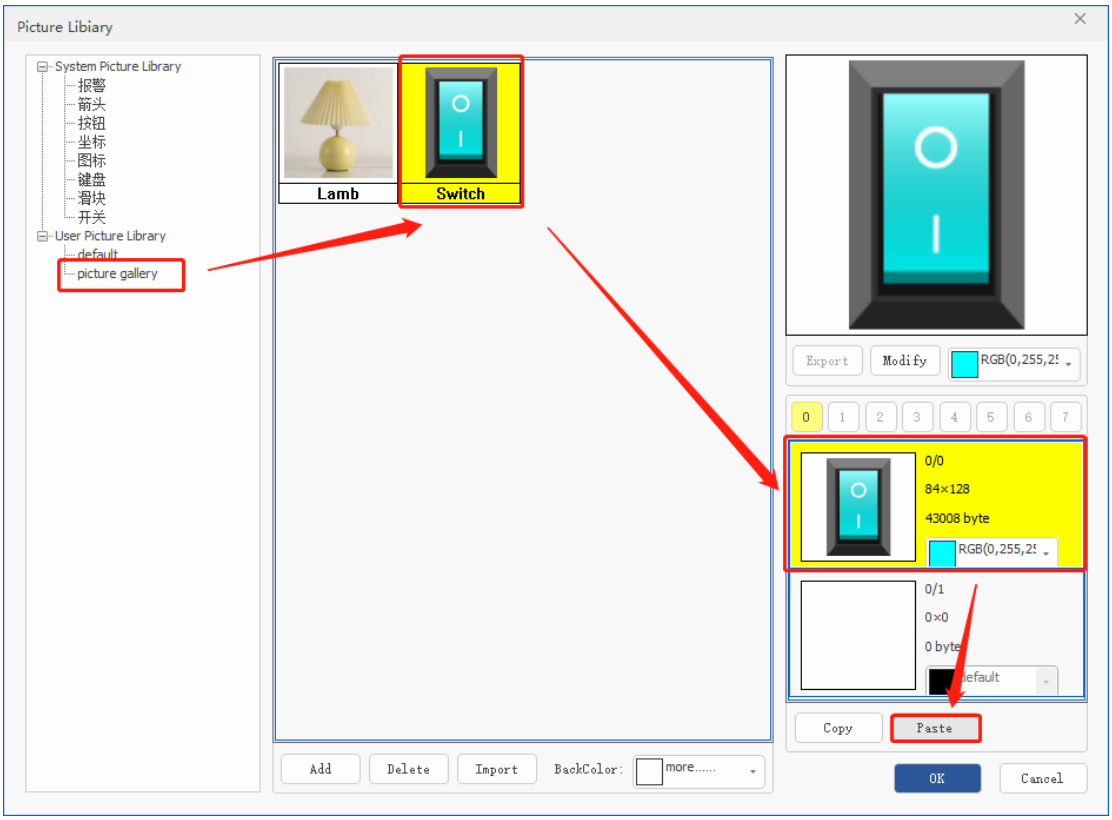
a. In new created “picture gallery”, then add one picture lib whose name is “switch”, also, set language numbers and state numbers.



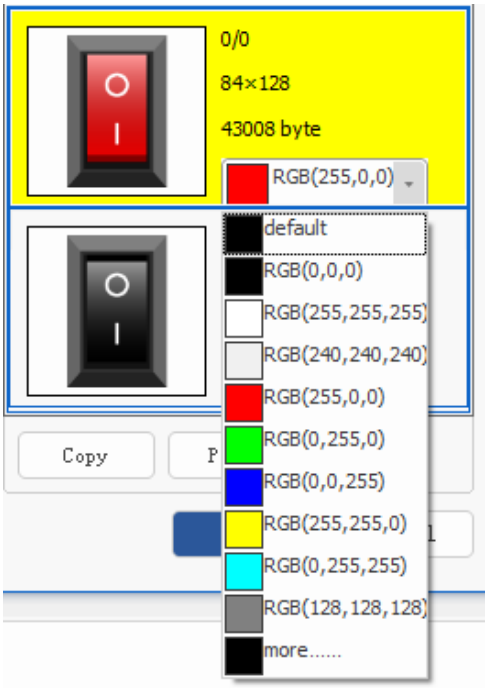
b. In “system picture library”, you can select needed format, and click “copy”. If there is no needed in system picture library, please add your own picture according to above “import”.



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.

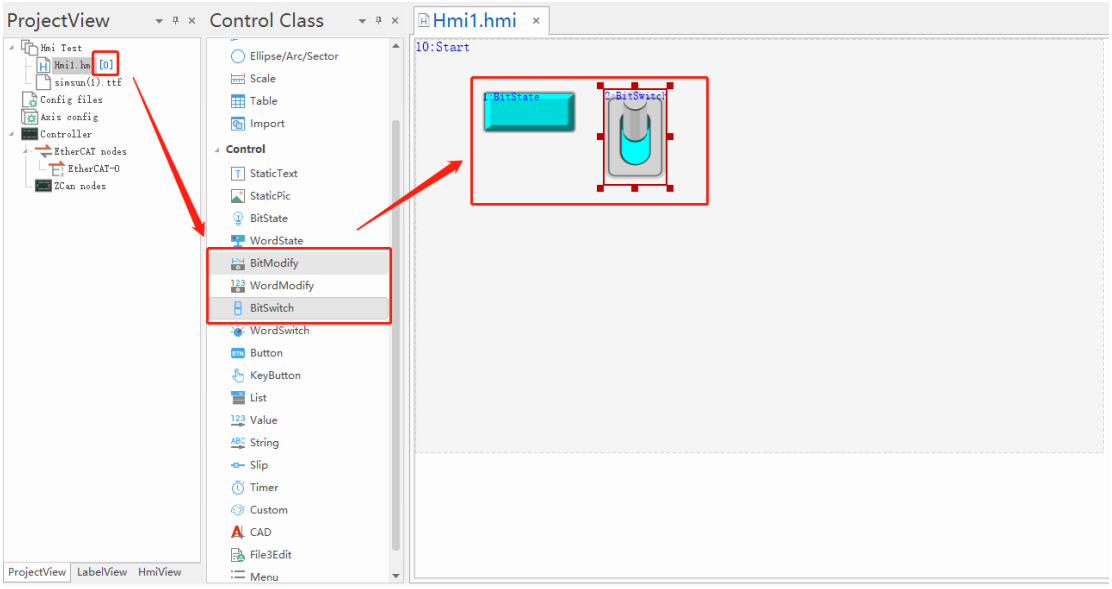


Step 2: [How to Apply Picture Lib]

A. Apply Directly

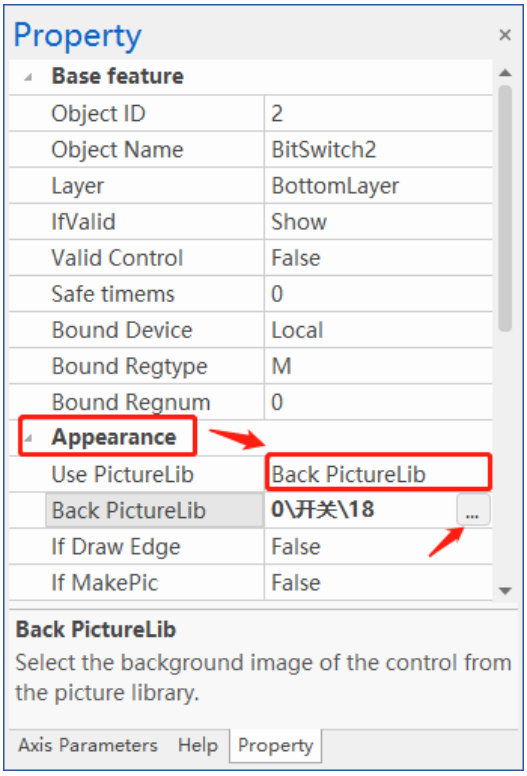
a. After building HMI file, set auto run task No. for HMI file, then open the HMI file. Next, set HMI system property, add HMI components in HMI window.

b. In control class, select and add “BitState” & “BitSwitch” into window 10.

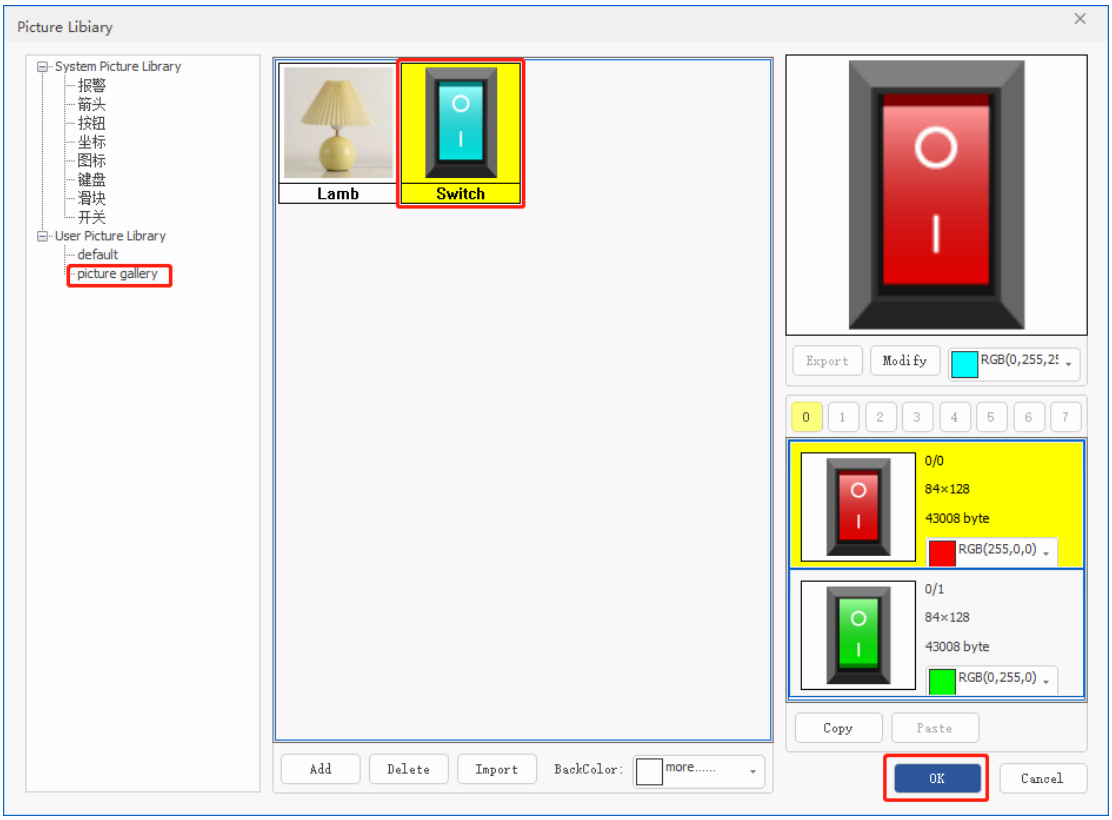


c. Click component 2 “BitSwitch”, then in the right side, its’ property window will appear. Then

in “Appearance” – “Use Picture Lib”, select “Back Picture Lib”, then in next item, click “…”, corresponding picture library will open.



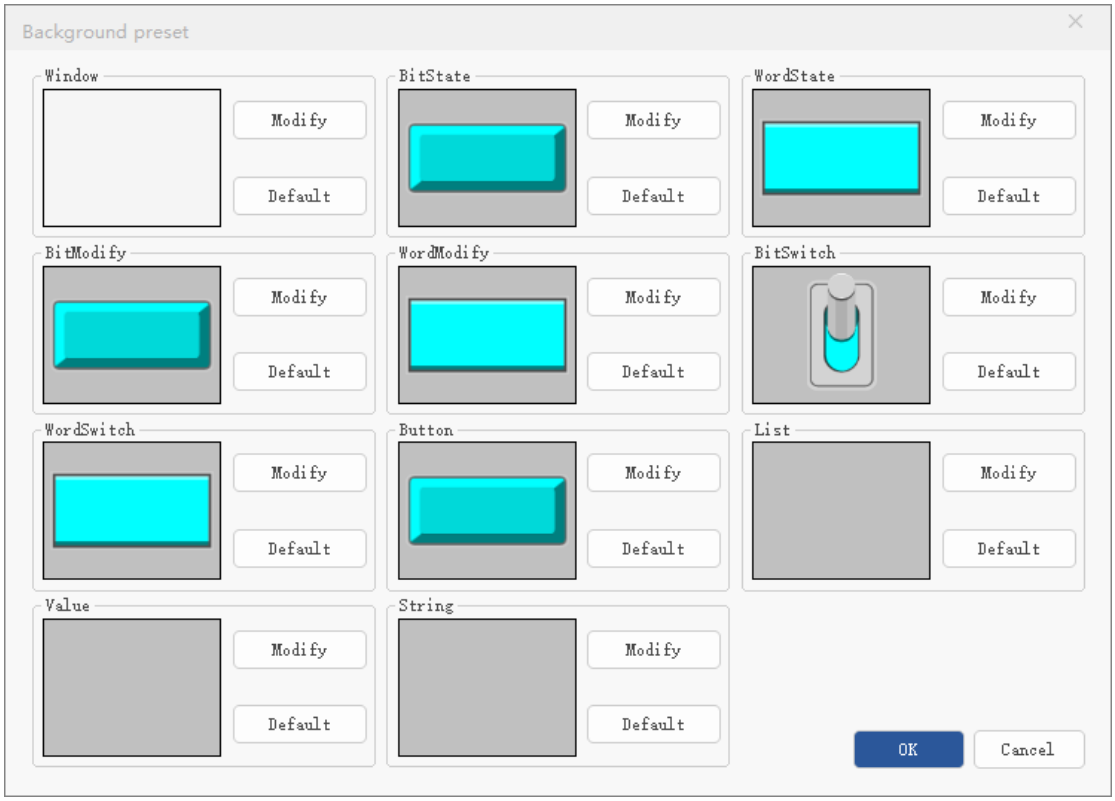
d. In appeared picture library, you can find needed one, and click OK, then the picture is applied directly. And you can adjust the size of component as needed (component 1 is same as this, in “Bound RegType”, please set as M2)



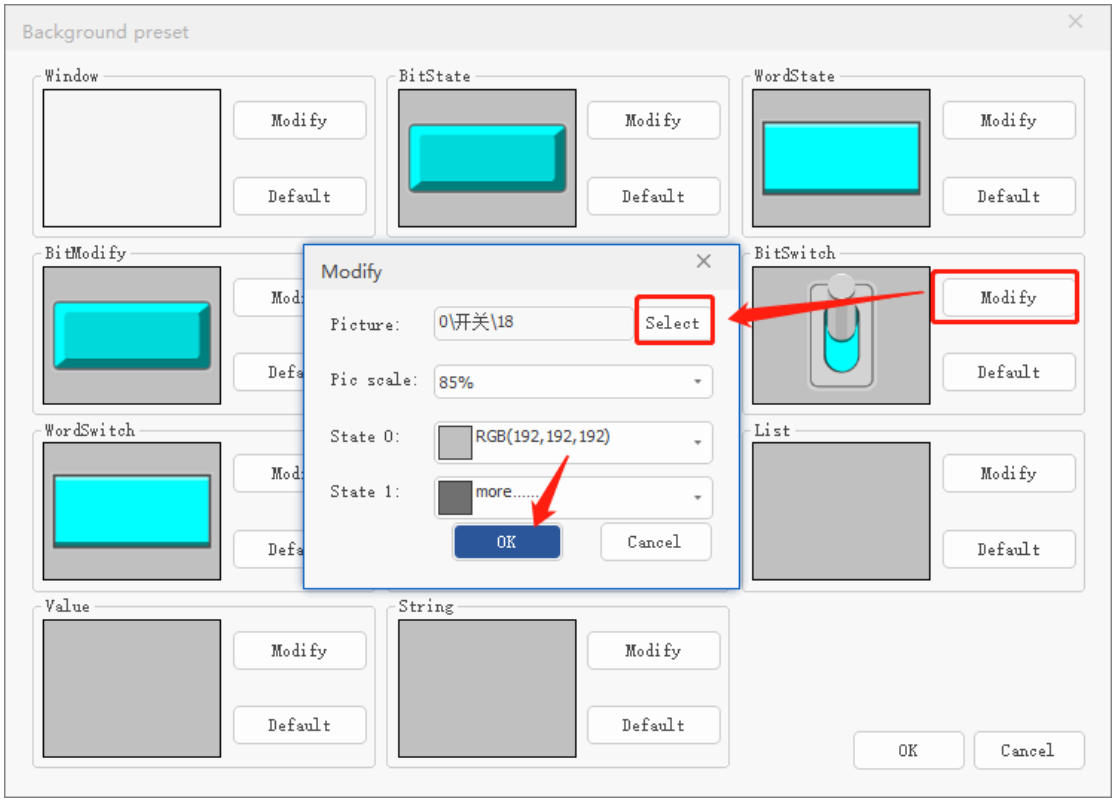
After that, apply successfully:

B. Apply by Background Preset

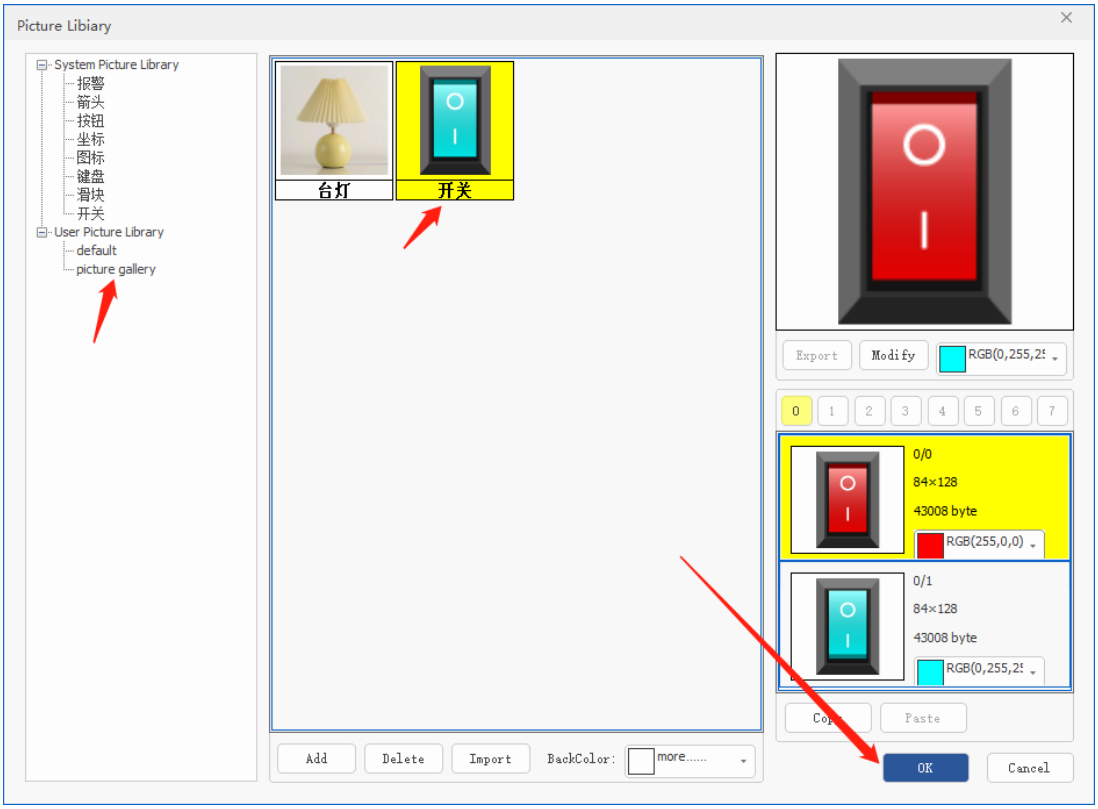
a. Open HMI file, then open below window through “HMI” – “Backgroud Preset”, you can directly modify the component formats here.



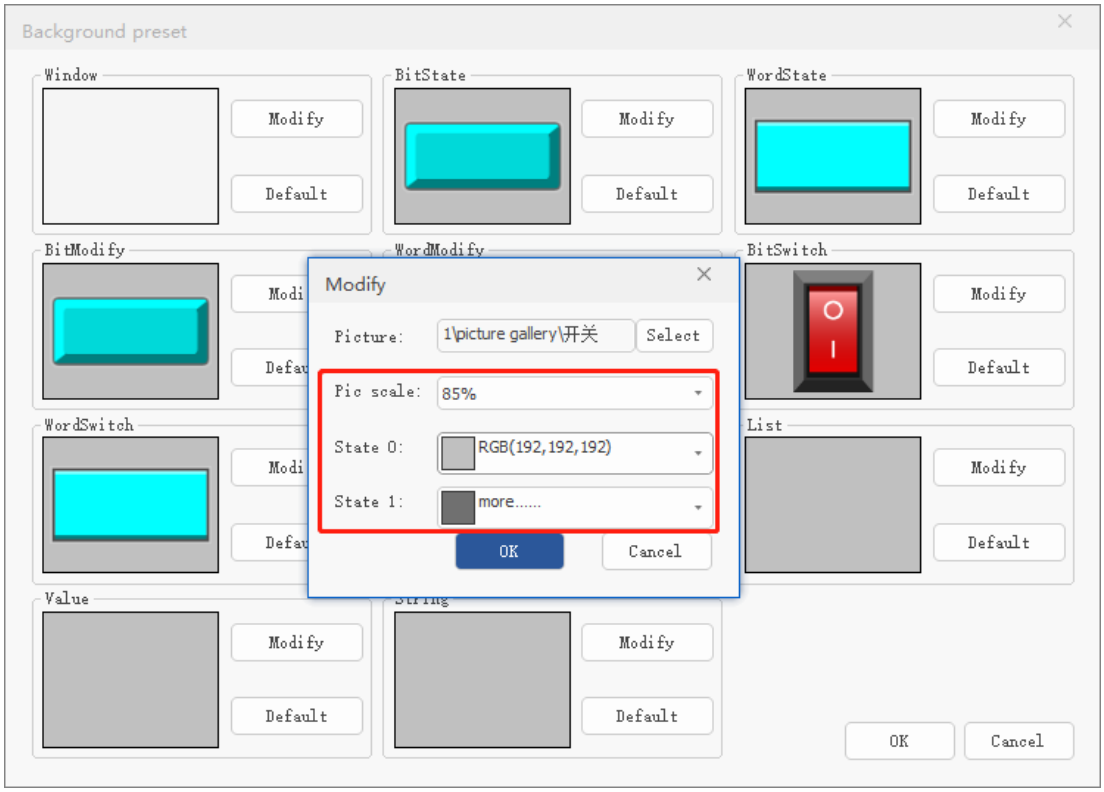
b. Select needed format (here takes the example of “BitSwitch”), click “Modify”, then select in below window.



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



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).



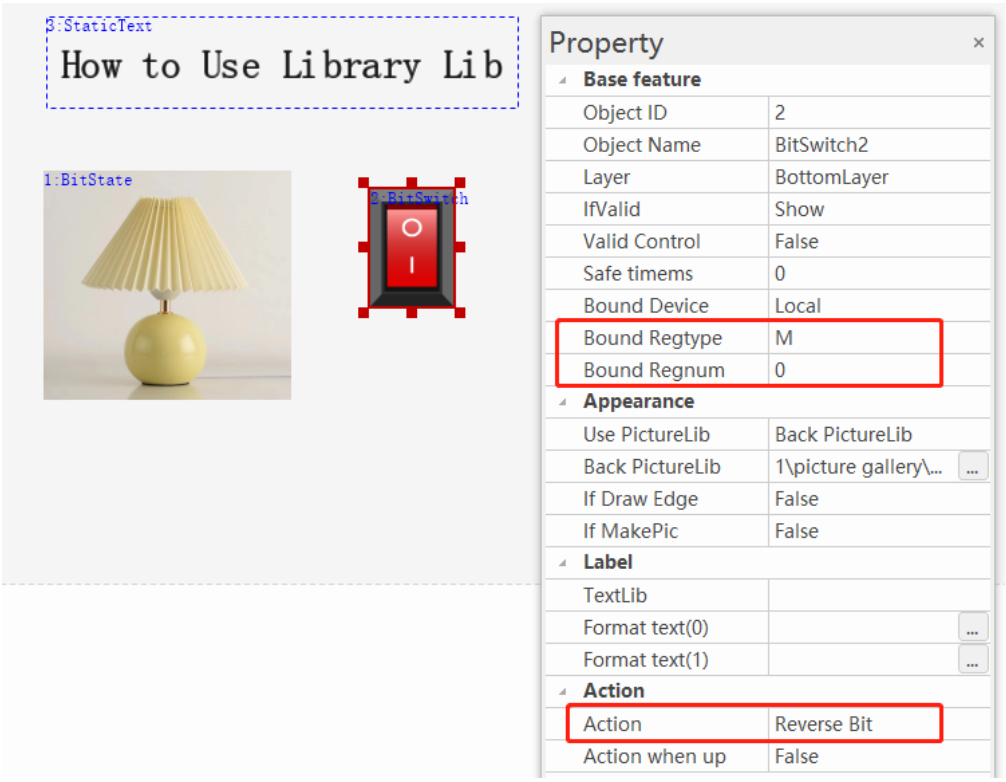
Step 3: [How to Call Sub-function]

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".



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.

2.2.4. 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.

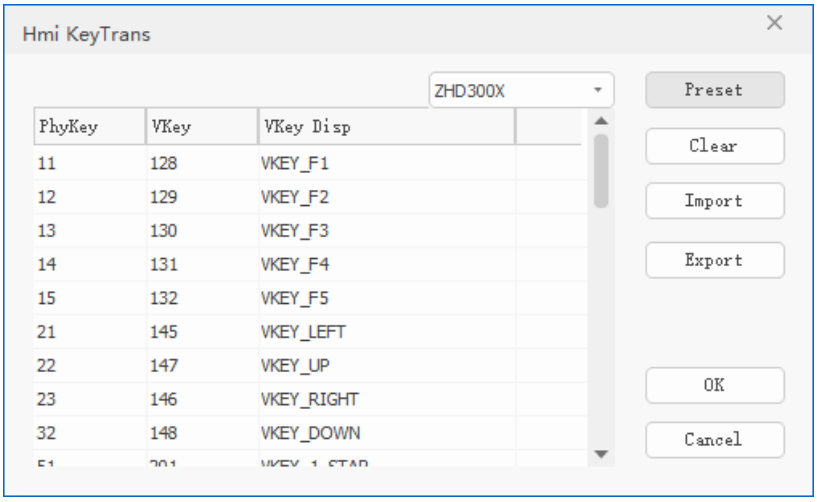
Step 2: in right upper, there is one menu you can select ZHD300X or ZHD400X.

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.



2.2.4.1. Physical Keys

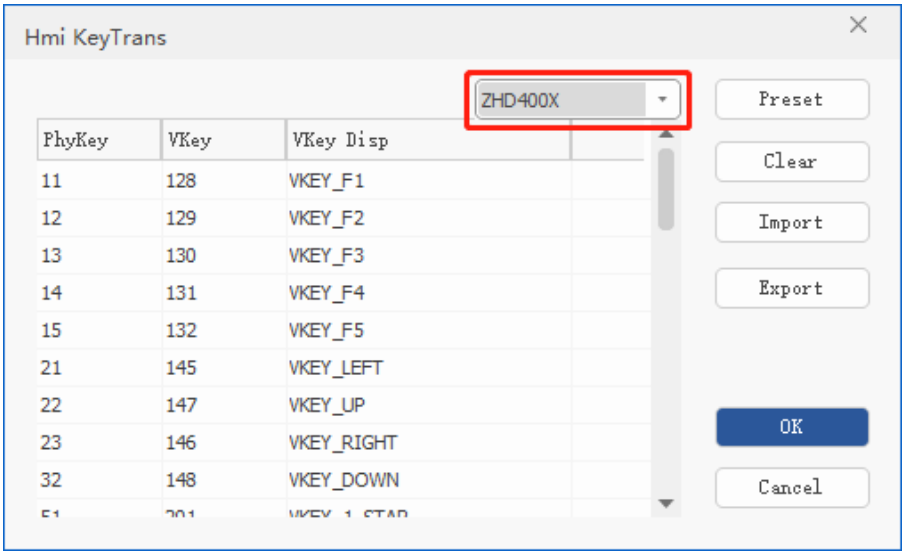
Physical keys refer to actual keys on external devices. Each key has a unique code value. When pressed, a message is sent. This message is the code value of the key.

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”



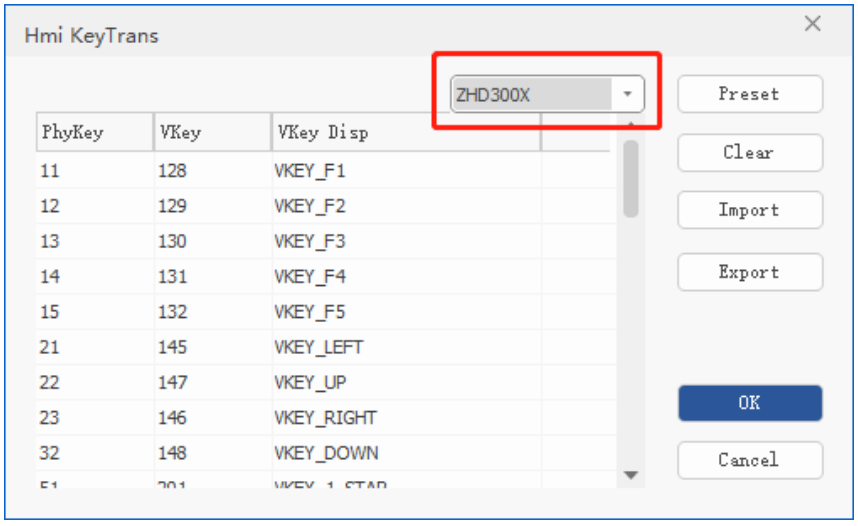
The encoding of the ZHD300X physical keys is composed of rows and columns, and the key value = row No. (1-10) × 10 + column No. (1-5).

- **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”



2.2.4.2. 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.**

2.2.4.3.Edit “HMI KeyTrans”

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

- **1. Menu Selection**

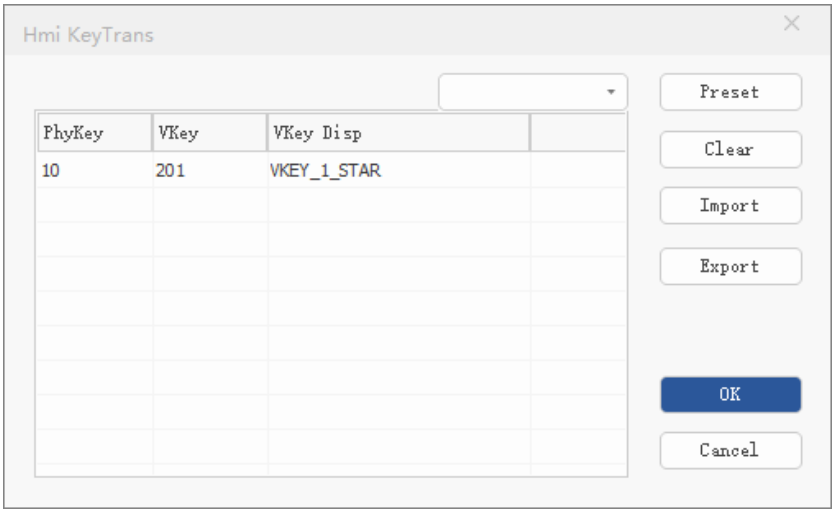
This tool has already set ZHD300X and ZHD400X conversion form, you can pull down the list to select.

- **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 axis 201:



➢ **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.

2.2.4.4.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".

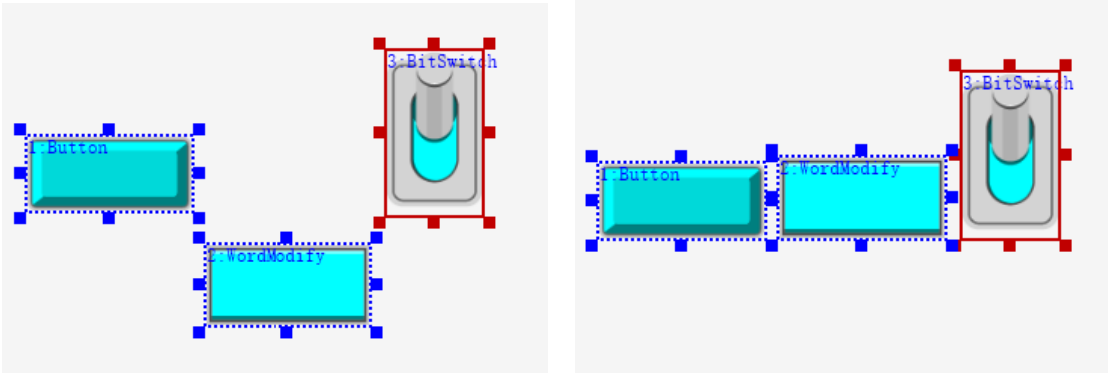
2.3. Arrangement

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

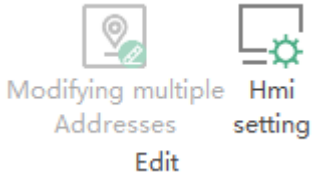
For example, make 3 components align in bottom:



--Notes--

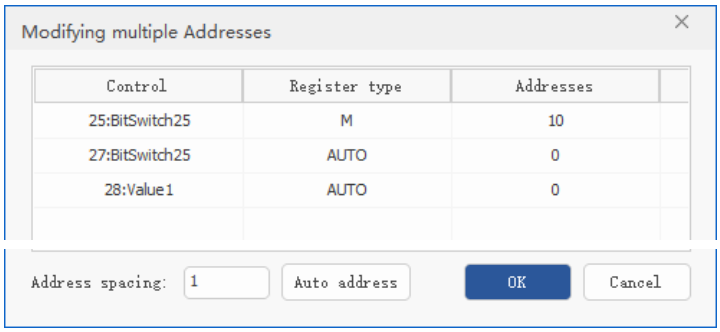
- It is used when you need to operate ≥ 2 components.
- The object that is selected will appear one red frame, and align it according to the object created sequence, not related to object No. sequence.
- If you need to custom the target object, please select the object at first, then press “Ctrl”, single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



- 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

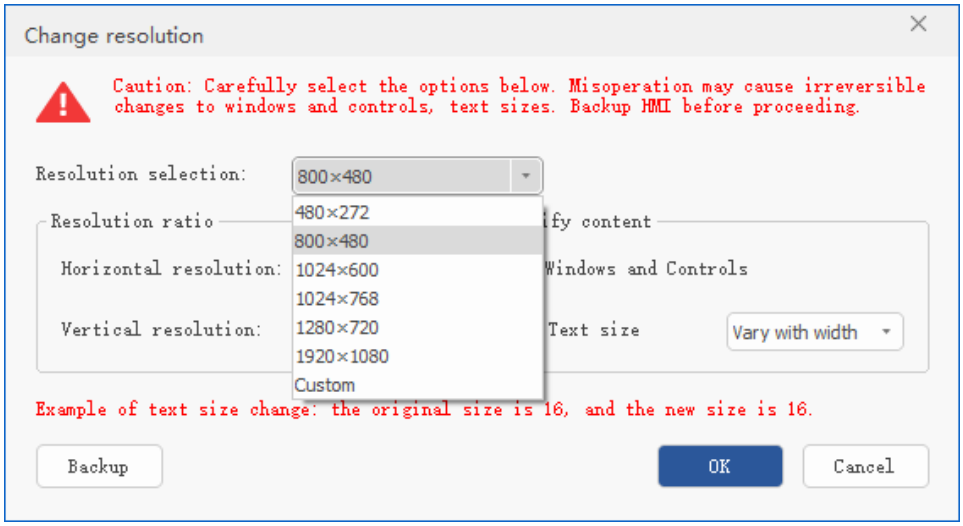
Perform initial property settings for the HMI system, and modify properties such as the HMI window resolution and the initial basic window. For details, see the table below. You can open the property window of the HMI system settings through the menu bar "HMI" → "Hmi Setting", or you also can click blank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

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

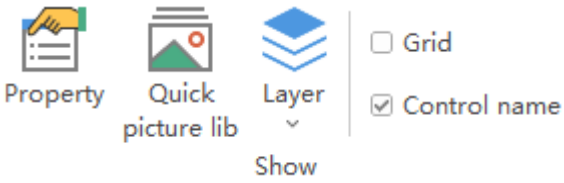
Property	Function	Description
Lcd number	Set the LCD screen No.	While connecting to HMI, select which HMI file content is shown by this No.
Backlight time	The actual backlight time of the teaching box	/
Screen time	Set screen saver time	/
Startup base window	Set the HMI initial base type window	Windwo 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.

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 control format text False: use text library format text	/
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.

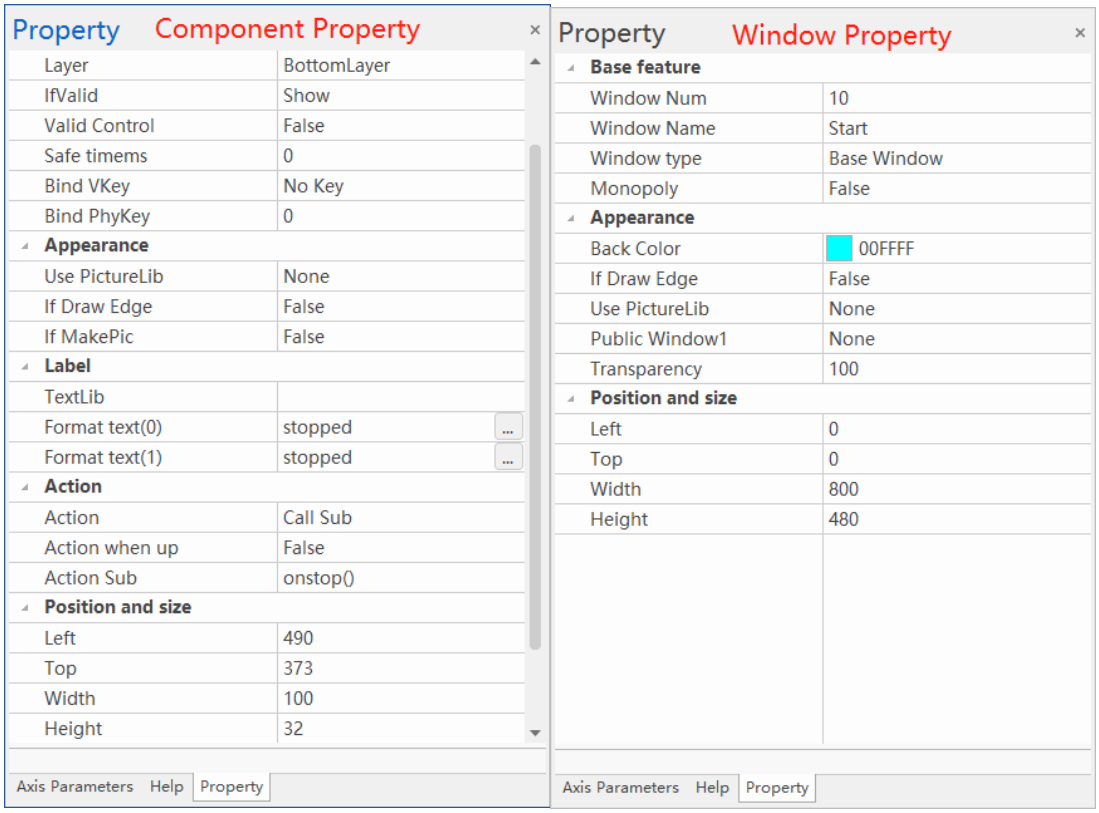


2.5. Showing Settings



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”.



- **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
Appearance		
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 controls of the public window. Up to 3 public windows can be set.
Transparency	Background transparency	Reserved.

Position and size		
Top	The X coordinate of the upper left corner of the window display	Can’t exceed vertical resolution
Left	The Y coordinate of the upper left corner of the window display	Can’t exceed Horizonical resolution
Width	Current window’s showing width	/
Height	Current window’s showing height	/

2.5.2. Quick Picture Lib

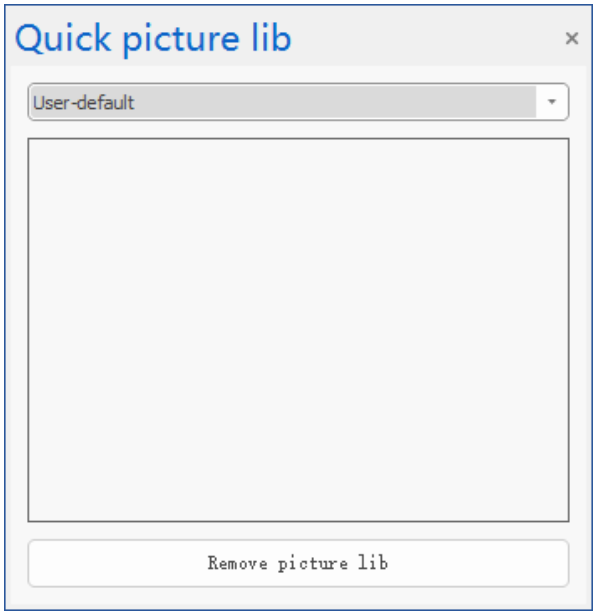
Used to quickly apply or remove styles from the picture library to HMI components. You can open this window through the menu bar "HMI" → "Quick Picture Library".

- **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**

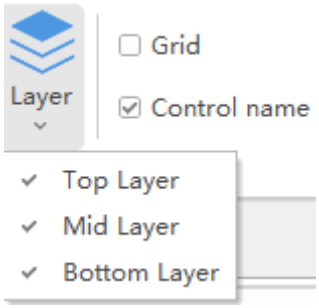
Choose the component, then remove it.



2.5.3. Show / Hide Layer

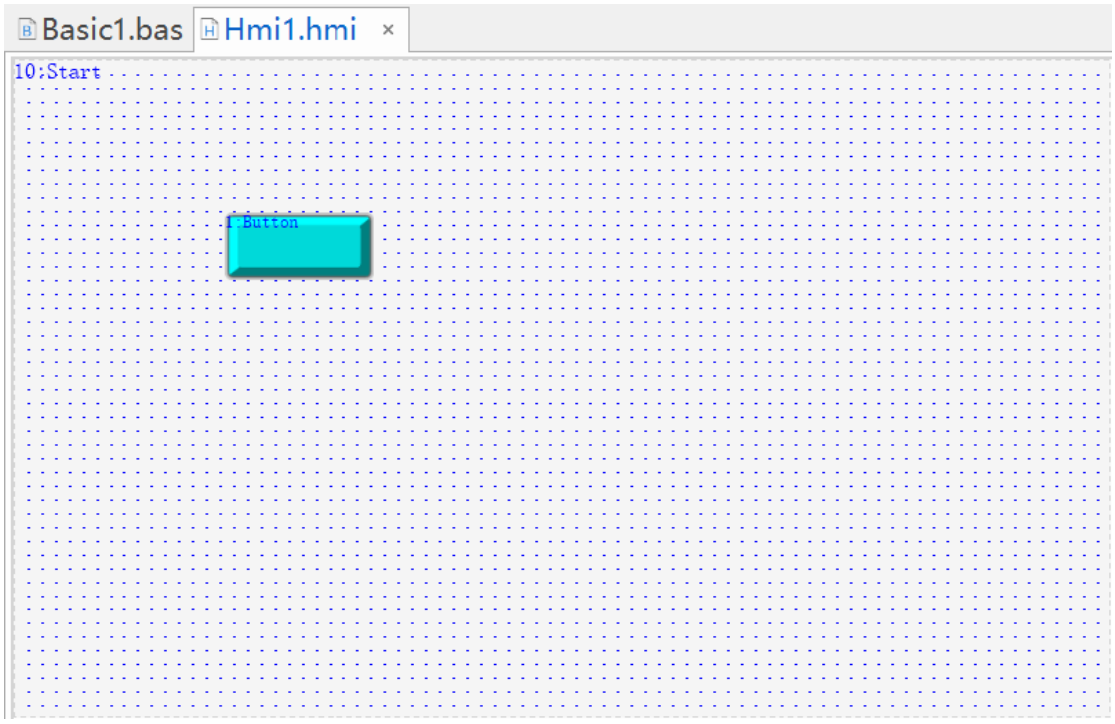
Show or hide components of different layers (HMI – Layer)

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

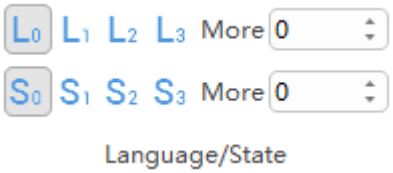


2.5.4. Grid & Control Name

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}.

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.)

When the configuration component exceeds the window size, the component can also be triggered normally.

3.1.2. Window Property

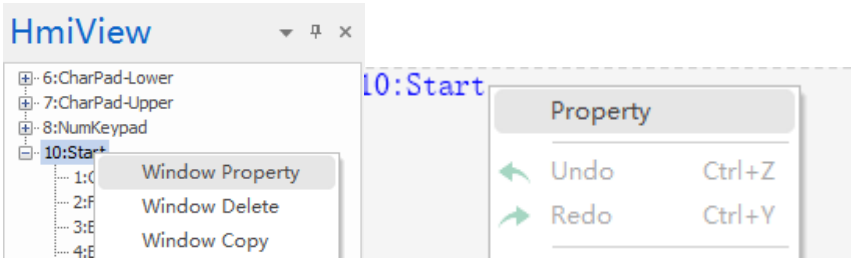
Window properties include the type, appearance, size, etc. of the window. You can set and modify the selected window through the "Properties" window;

- **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. Window Operations

3.2.1. Create the Window

HMI showing must use a basic window as the bottom window, which is used as the background screen of other windows, because components need to be attached to the window display, and multiple windows of different types can be created under one configuration 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.

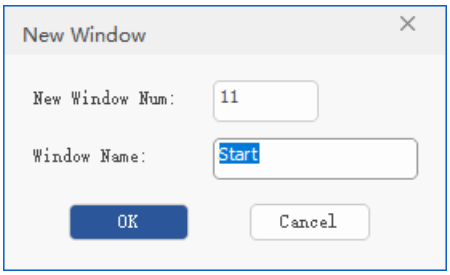
- **Create New Window**

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.

- **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.



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.

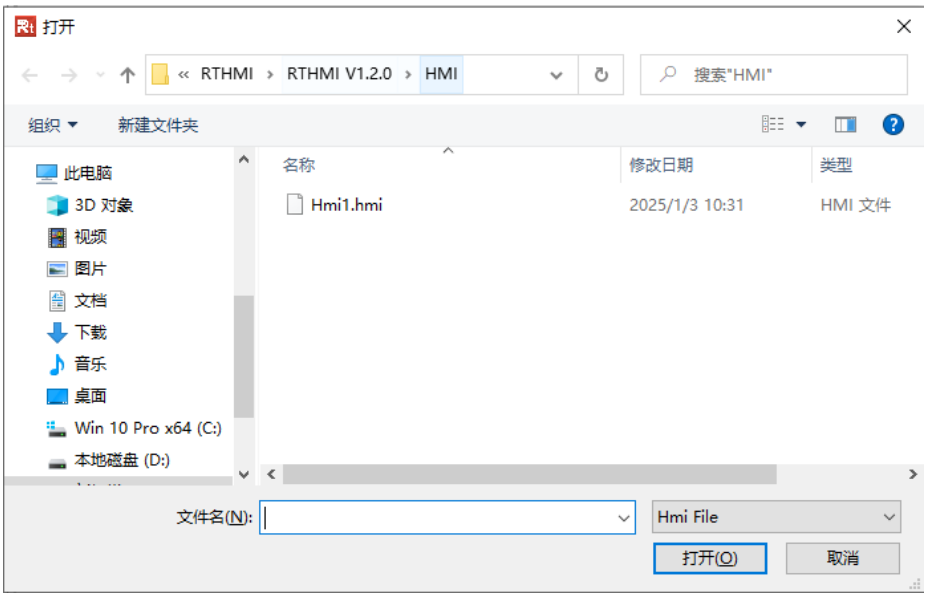
3.2.2. Import Window

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”.

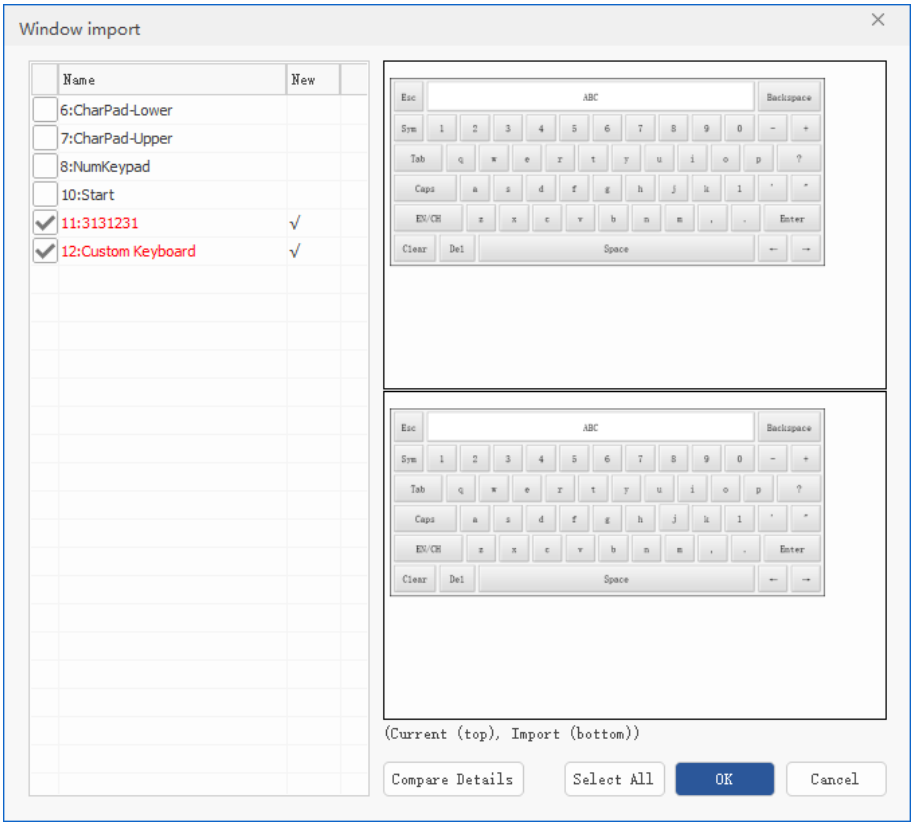
(2) In new popped window, select which one window (.Hmi file).



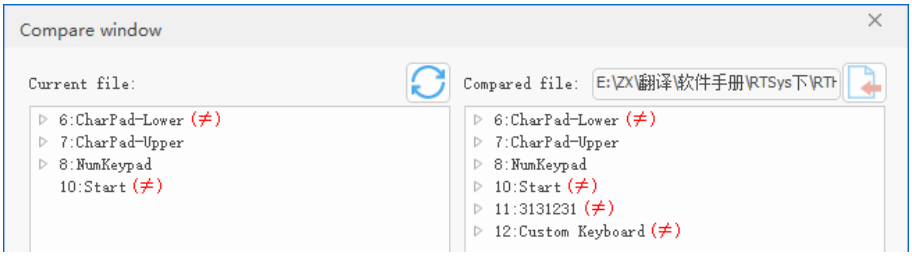
(3) Open needed HMI file, then “window import” window appears. In left side, it shows window list. In right side, it shows window comparison graphics.

Note: if the window ID and window name in the imported Hmi project are inconsistent, the window

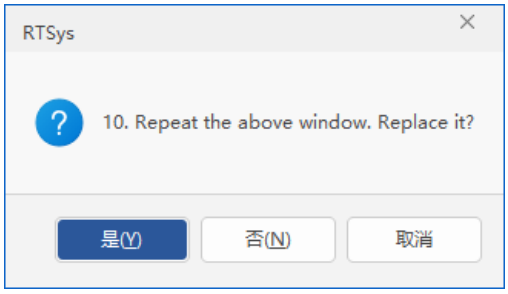
name will be displayed in red, and the window without the same window ID in the current project will be checked by default. You can choose whether to import / replace the original window. (Select "Yes" to replace, select "No" to import only the non-conflicting window.



After clicked "compare details", new window of "compare window" will appear.



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.

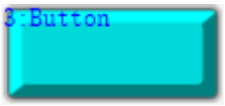


3.2.3. Call Window

There are two methods to call the window.

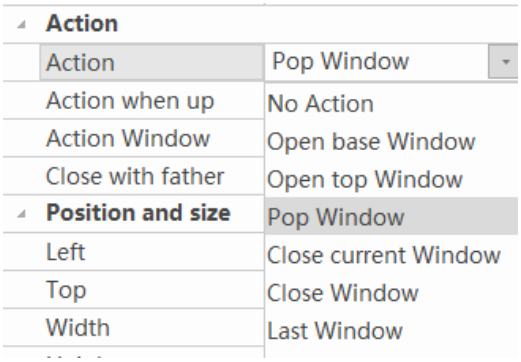
■ **Method 1: Call Functional Key**

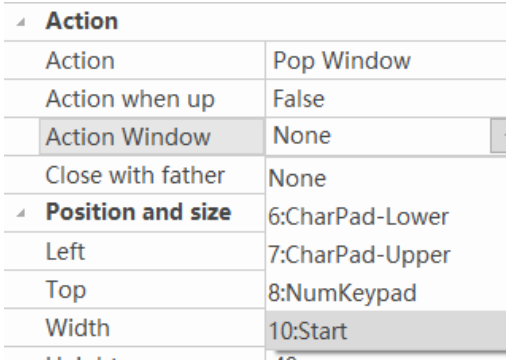
(1) Select “Control Class” – “Button”.



(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**

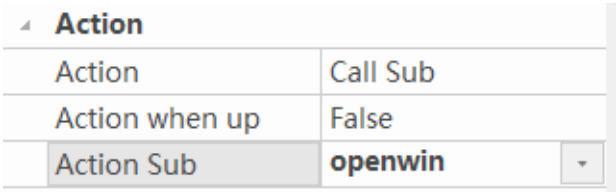




Note: if the window type to be opened is inconsistent with the selected window type, the window type will be forcibly converted.

■ **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.



--Corresponding Basic Codes--

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

3.2.4. Close Window

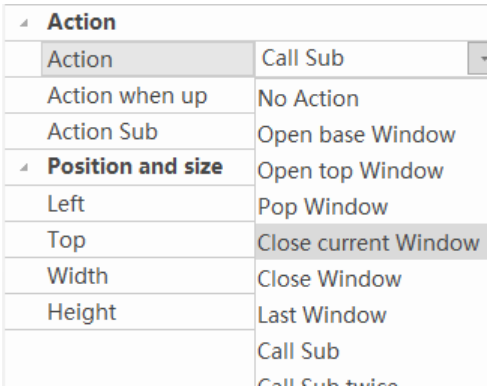
There are two methods to close the window, you can refer to below information.

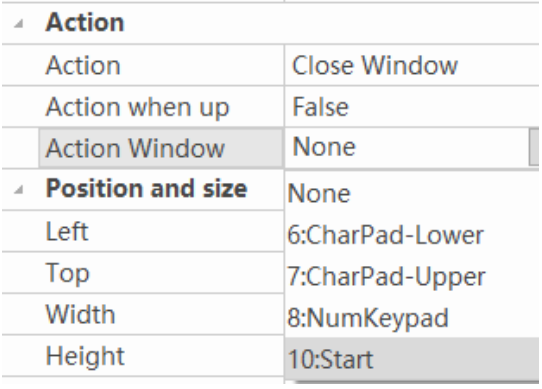
Note: base window can't be closed!

■ **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).





■ **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
```

3.2.5. Public Window

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

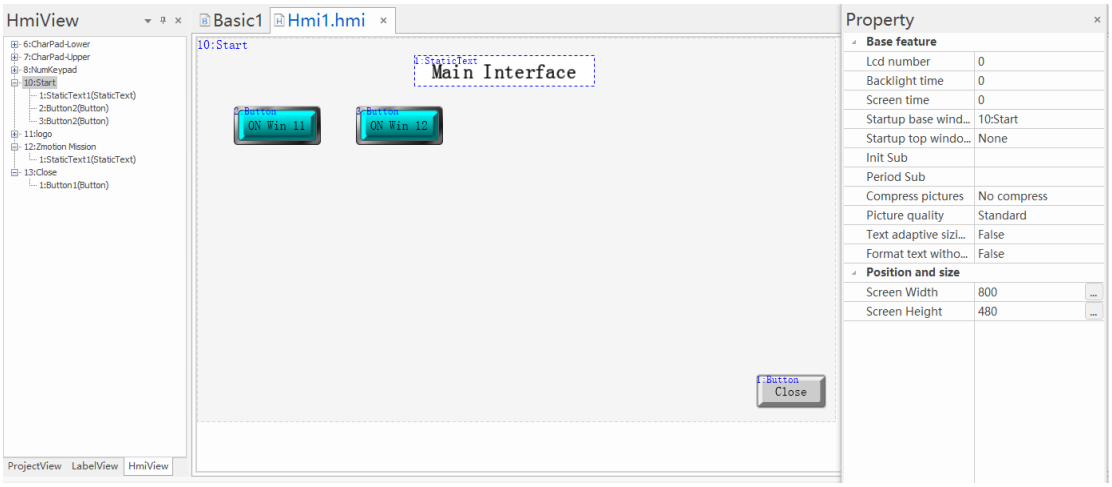
You can select current as public window in "Property". One window can set 3 public windows at most.

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.

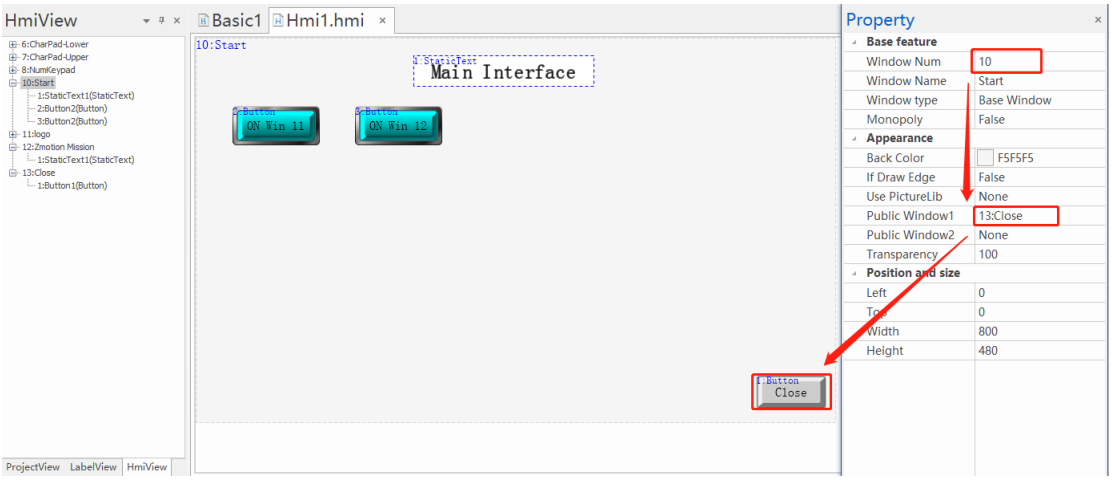
Appearance	
Back Color	F5F5F5
If Draw Edge	False
Use PictureLib	None
Public Window1	11:logo
Public Window2	12:Zmotion Mission
Public Window3	**13:Close**
Transparency	100

For Example:

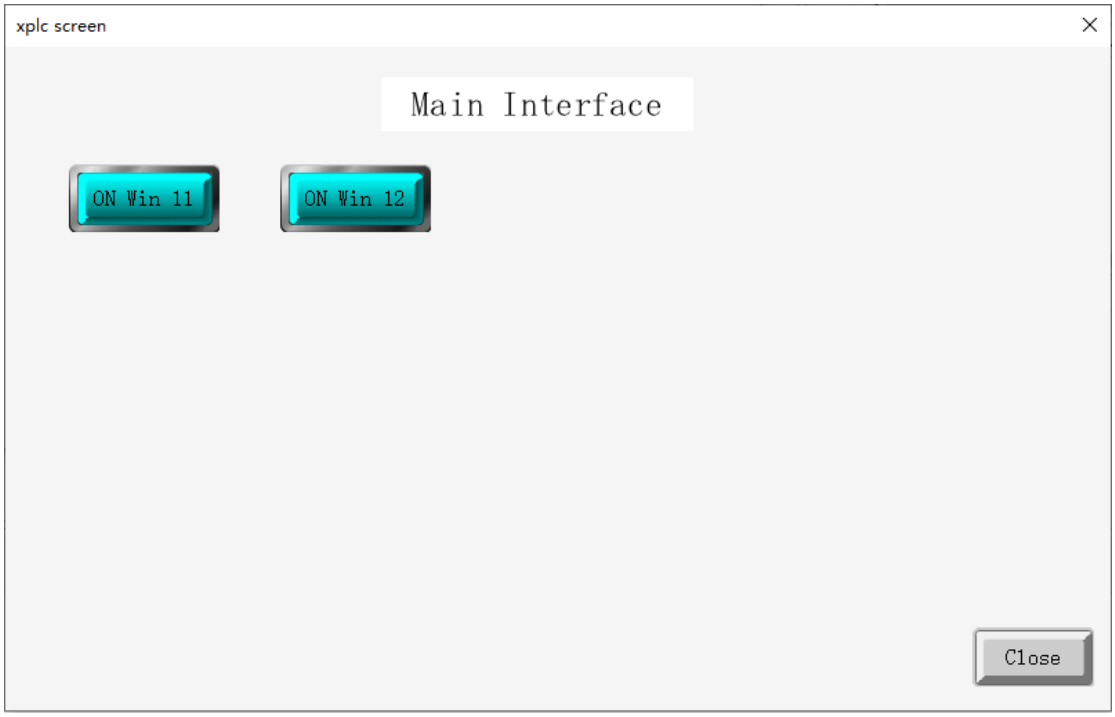
- See below, if no public window is added, each window only shows current window's component.



- 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.



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



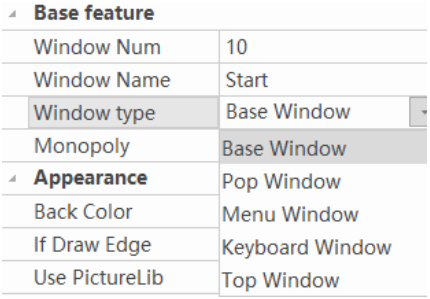
You can open corresponding window through operating main window (window 10), and the public window objects actions in window 10 can take effect.

When you open window 11 and window 12, main window shows window 100 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.



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.

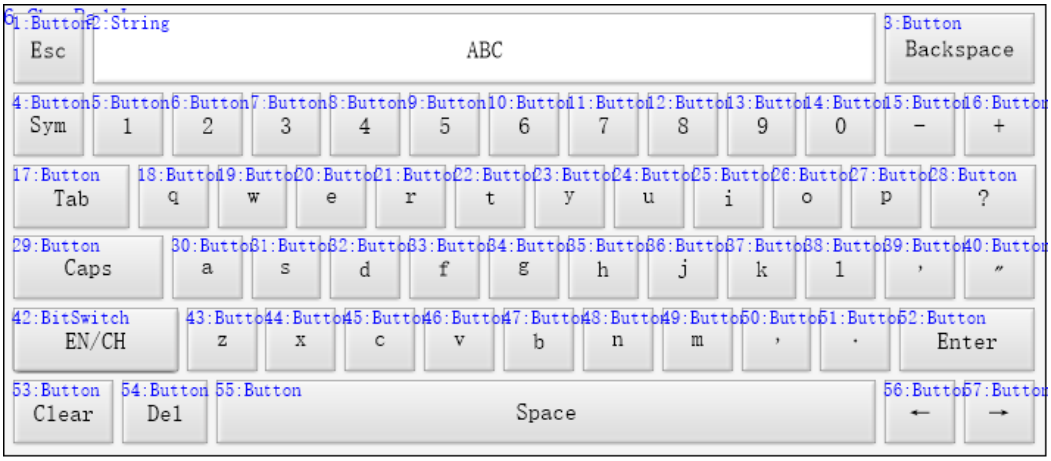
3.3.2. Keyboard Window

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

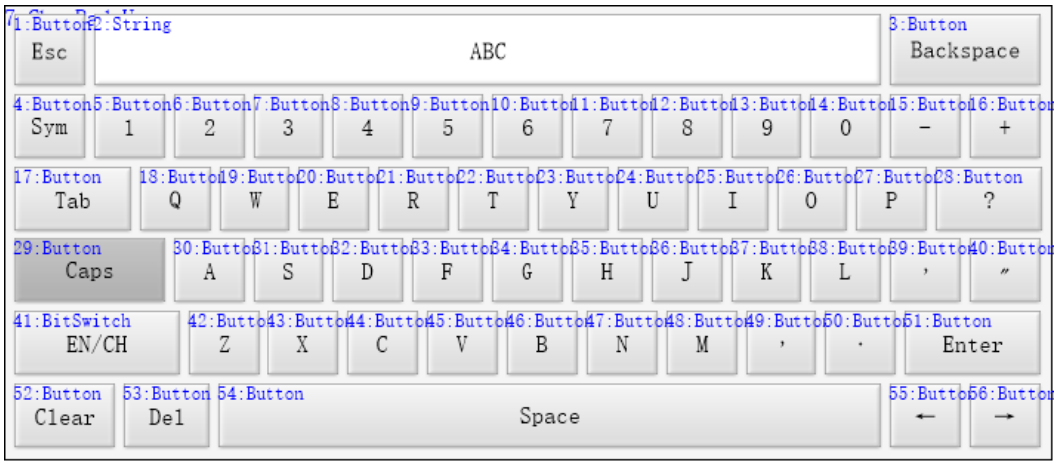
And there are 3 soft keyboard windows already inside the HMI file, you don't need to new build the HMI file. They are window 6, window 7, window 8. When you open one "value" property window, you can select "True" for "Allow Edit", then select one soft keyboard window No. in "Key Window".

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.

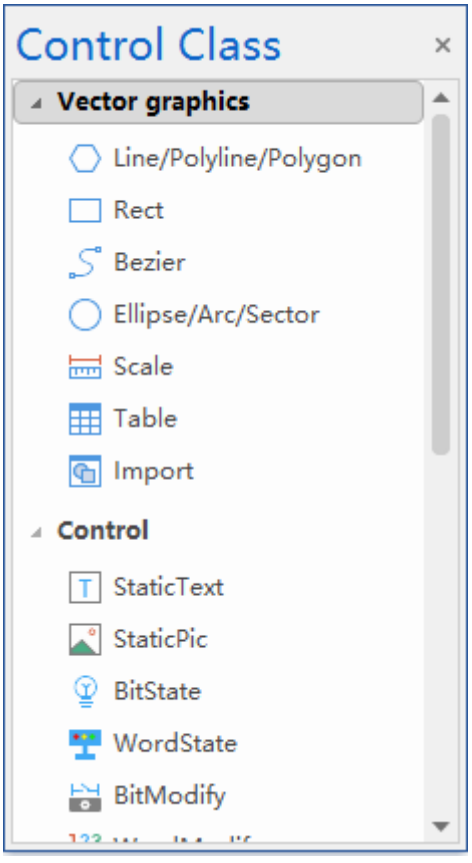
Chapter IV HMI Component

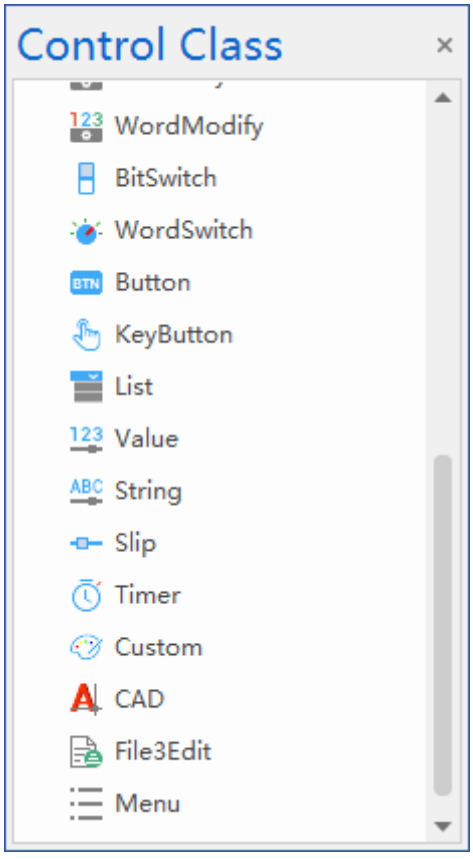
4.1. HMI Quick Tools

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 Images		
Line / Polyline /		Draw corresponding lines / polylines according to

Polygon		drawn points.
Rect		Draw one rectangle
Bezier		Draw a third-order Bezier curve with four points
Ellipse / Arc / Sector		Drag one to draw the full circle/ellipse/arc/sector.
Scale		Draw scale for equal space
Table		Draw 3*3 form, you can customize table format
Import		Import vector graphics
Control		
Static Text		Add static text in the window, you can customize corresponding properties.
StaticPic		Insert one image from system / background library
BitState		Show the state of bound bit register address value
WordState		Show the state of bound word register address value
BitModify		Set bit register address value according to component action state
WordModify		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		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 triggered by real key
List		Show multiple lists, switch to corresponding item by bound register value

Value		Edit and show value, and change corresponding bound register value
String		Edit and show character string, and change word register value
Slip		Change word register value by dragging the slide
Timer		Timed refresh for repeated actions
Custom		Call basic function in the area of component.to achieve dynamic drawing
CAD		Show vector graphics
File3Edit		Develop the edit component of file 3 program in HMI
Menu		Reserved

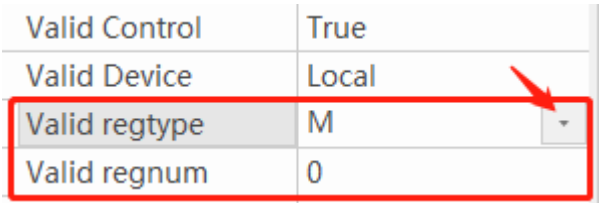
4.2. HMI Component General Property

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.



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.

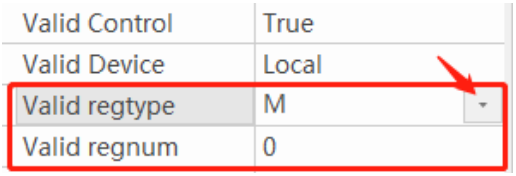
4.2.2. Action

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

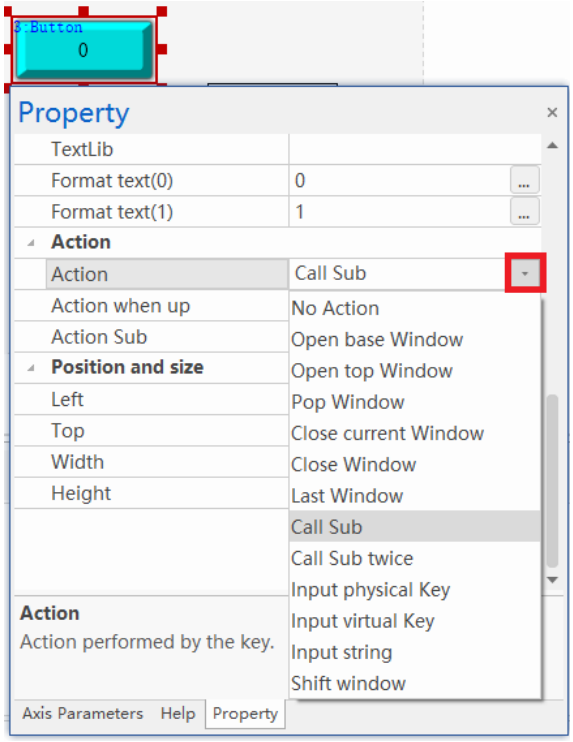
Action Name	Function	Description
Component General Actions		
No Action	/	/
Call Sub	Call SUB function defined by Basic	Function must be GLOBAL.
Functional Key “Button”		
Open Base Window	Open window with “Base Window”	In “Action Window”, select

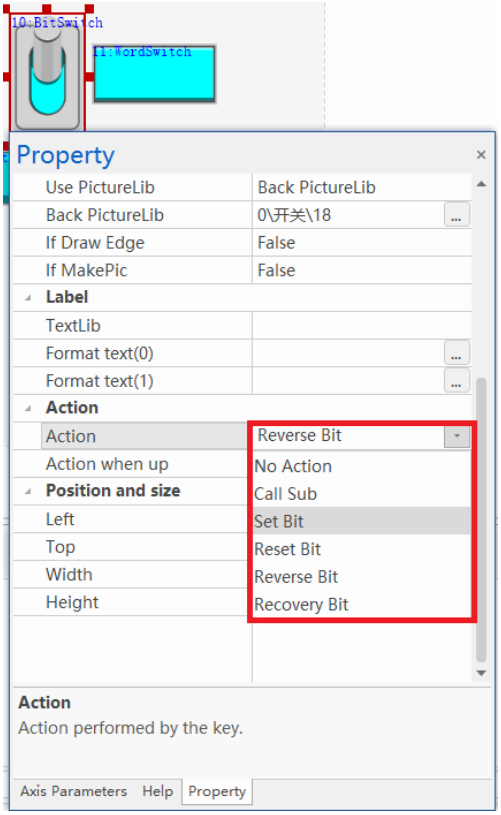
Open Top Window	Open window with “Top Window”	the No., for more window type info, please refer to “3.3”.
Pop Window	Open window with “Pop Window”	
Close Current Window	/	Close all windows of the current key button.
Close Window	Close selected / assigned window	In “Action Window”, please select the No.
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		
Reset Bit		
Reverse Bit		
Recovery Bit		
WordSwitch (Multi-State Switch) / WordModify (Multi-State Configuration)		
Data Write		
Data Plus		
Data Loop		

➢ Component General Actions:

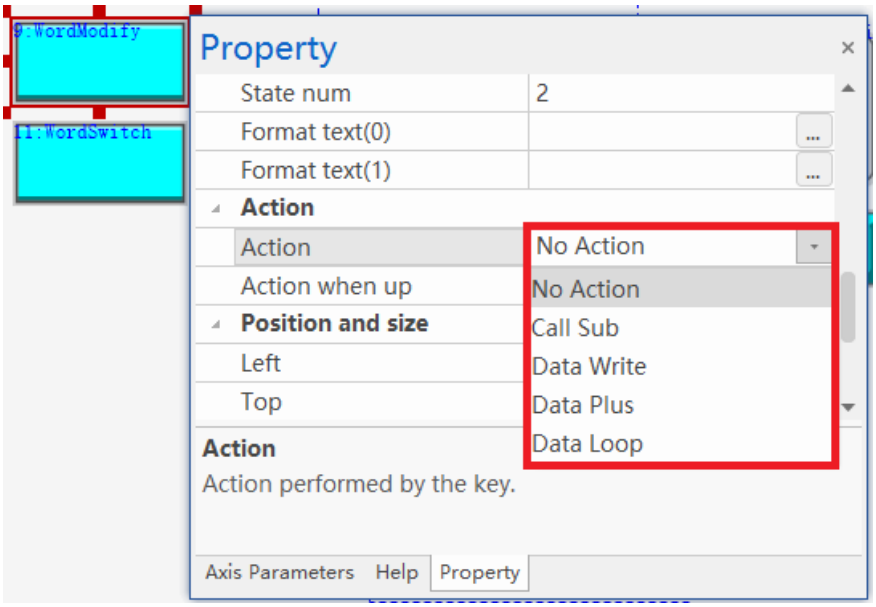


➢ Functional Key “Button” + BitSwitch & BitModify:

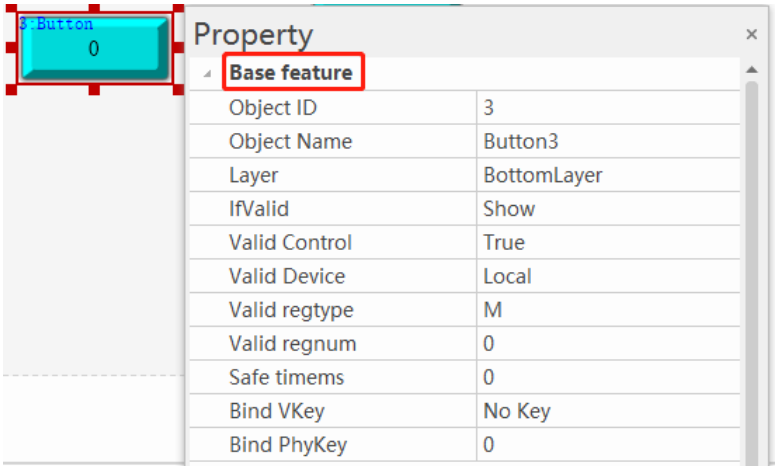




➢ WordSwitch & WordModify:

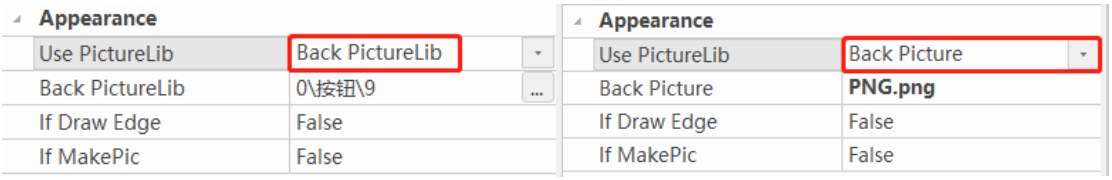


4.2.3. “Base Feature” of Object Property



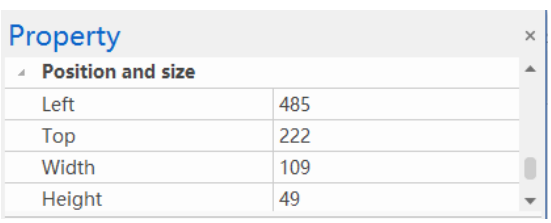
Base Feature	Description
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object’s display layer ● TopLayer: the surface, it shows the most external layer, and covers below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

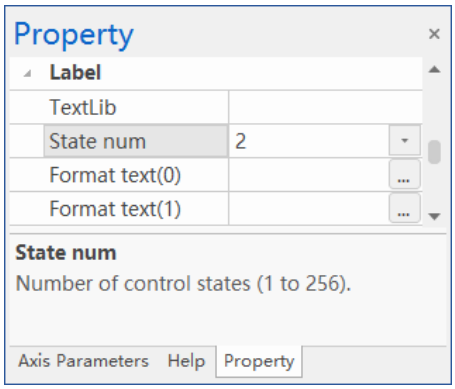


Position and Size	Description
Top	The horizontal distance of the component in the Hmi window. The upper left corner of the Hmi window is (0,0).
Left	The vertical distance of the component in the Hmi window. The upper left corner of the Hmi window is (0,0).
Width	Set component's displaying width
Height	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.



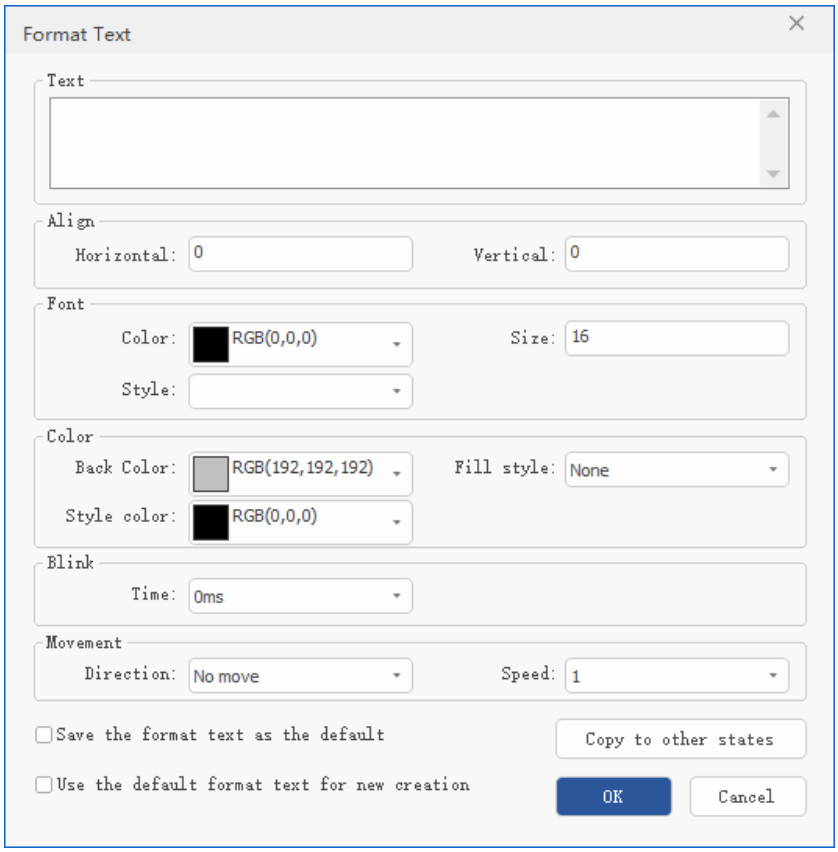
When you click the "…", one window will pop up, then you can enter needed text and set related formats.

- **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!

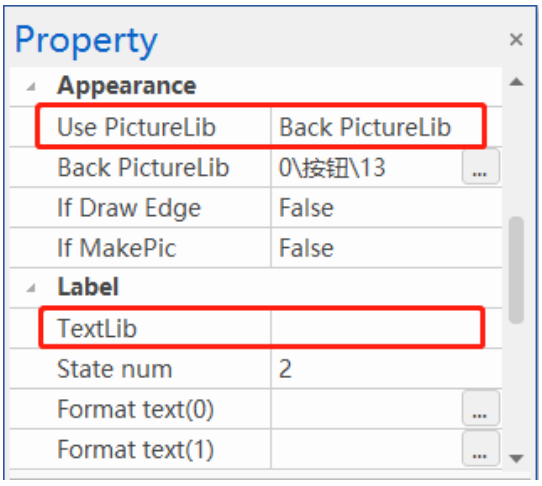
- **Use the default format text for new creation:** The newly created component uses the default format text that has been set. Check it to take effect.



Operation	Function	Description
Text	Enter needed text info	/
Align	Align-Horizontal	0: center alignment (default) >0: left / upper alignment, the value indicates the distance from the left / up <0: right / bottom alignment, the absolute value indicates the distance from the right / bottom
	Align-Vertical	
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”



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].

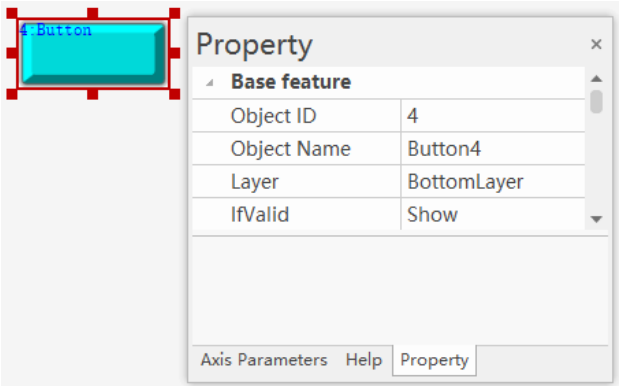
4.3. Components in Control Class Introduction & Usage

➢ **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.



➢ **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. <u>What is It?</u>

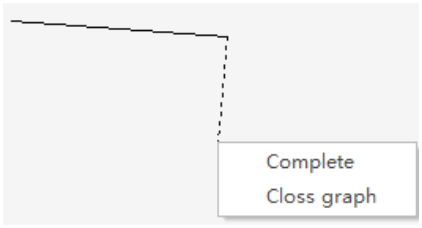
"Line" is the line segment, whose length and slope by yourself. Because width and height can be

set by dragging directly or in property window.

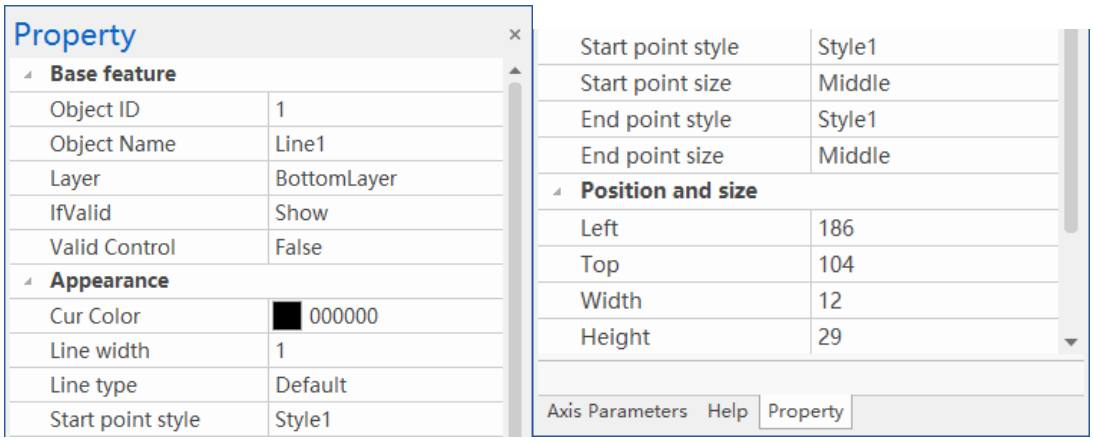
B. How to Use?

Click “Control Class”, 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



D. “Property” Description

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.
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.
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 "Control Class", 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".

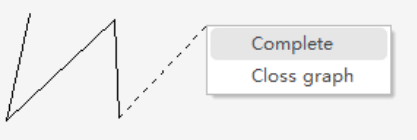
Notes:

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

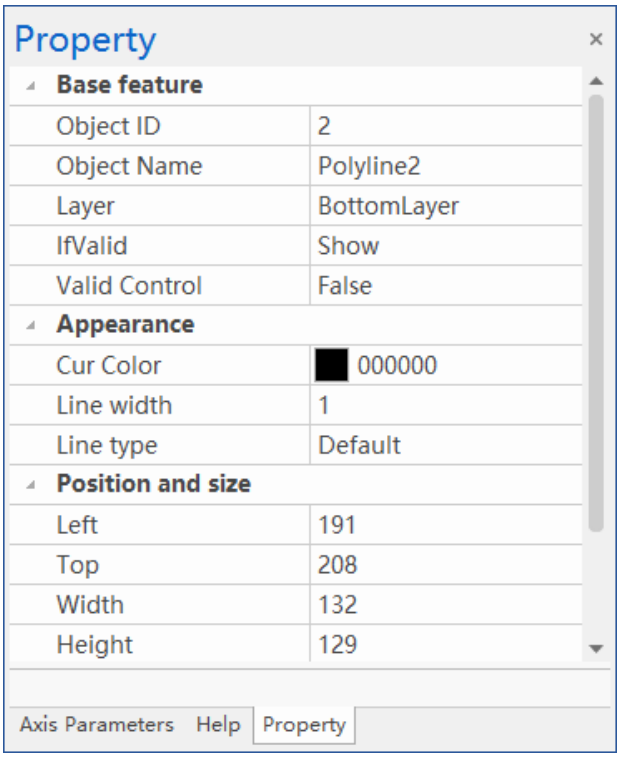
to determine the direction and length.

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



D. "Property" Description

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.
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.
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.

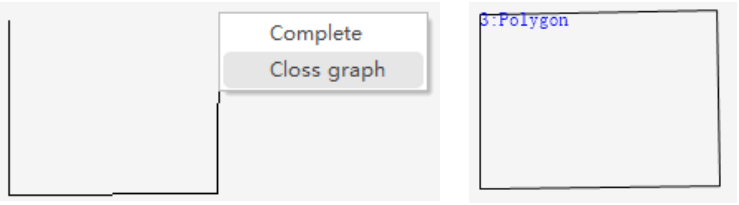
B. How to Use?

Click "Control Class", 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".

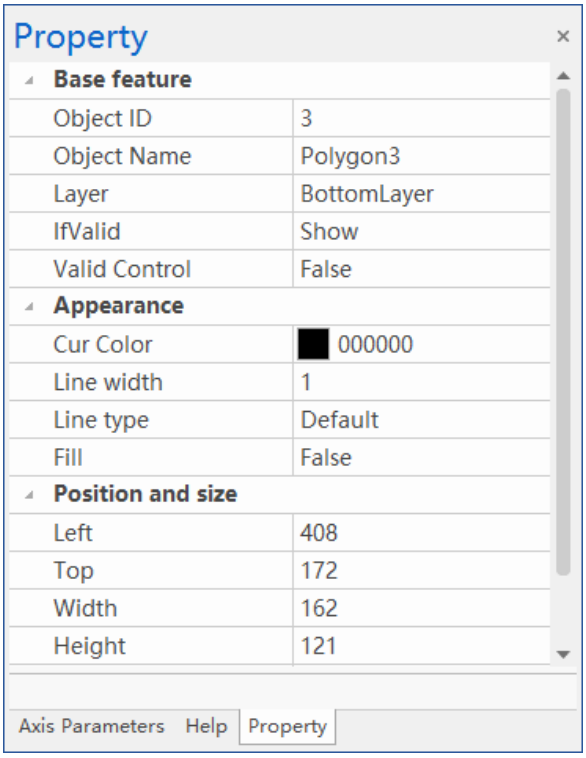
Notes:

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



D. "Property" Description

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.
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
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.

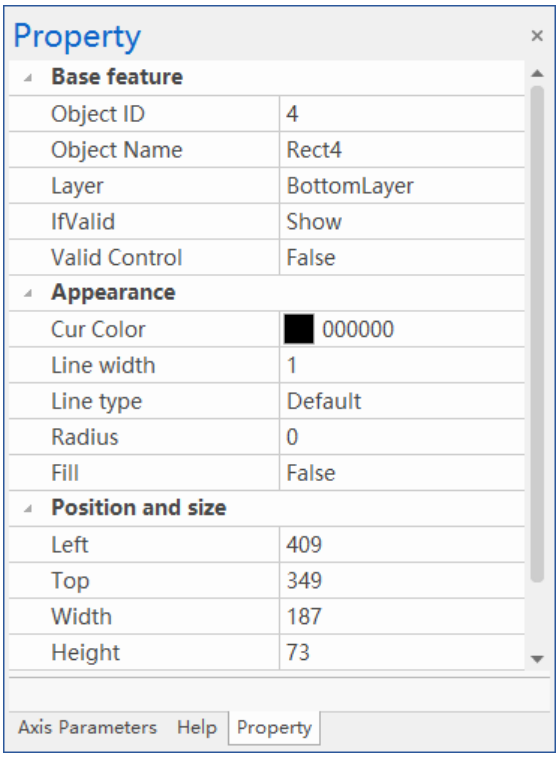
B. How to Use?

Click "Control Class", 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



D. "Property" Description

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.
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
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?

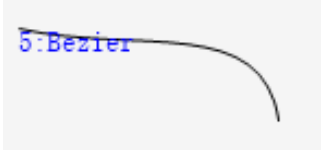
The vector mathematical curve based on a two-dimensional plane.

B. How to Use?

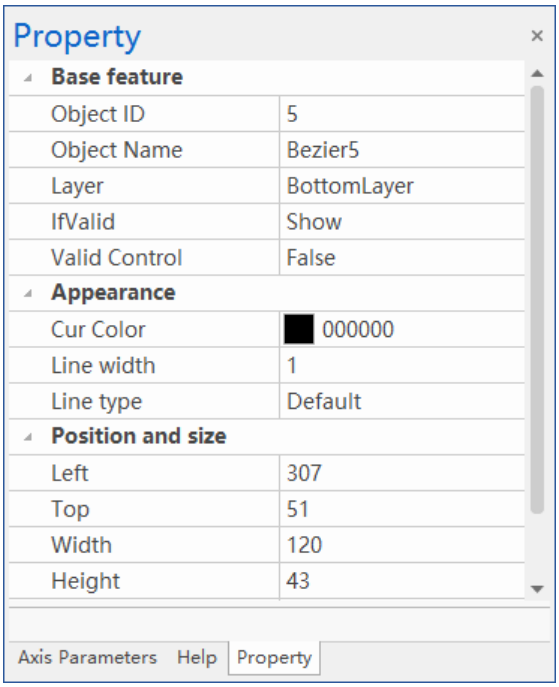
Click "Control Class", select "Bezier" from "Vector graphics". When drawing, you only need to determine 4 points, then you can automatically create the Bezier.

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



D. <u>"Property" Description</u>

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.
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.
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 "Control Class", 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:

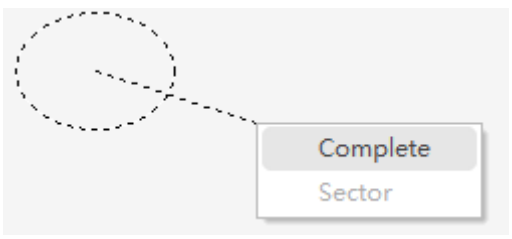
*Draw Ellipse: you only need to drag the mouse, confirm size, then right click "complete" (no need to confirm position).

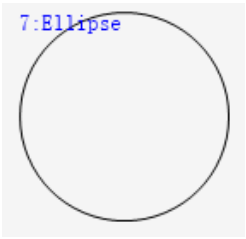
*Draw Circle: when you drag the mouse to determine the size, press Ctrl button at the same time, when confirmed, release mouse and button, then right click "complete" (no need to confirm position).

Notes:

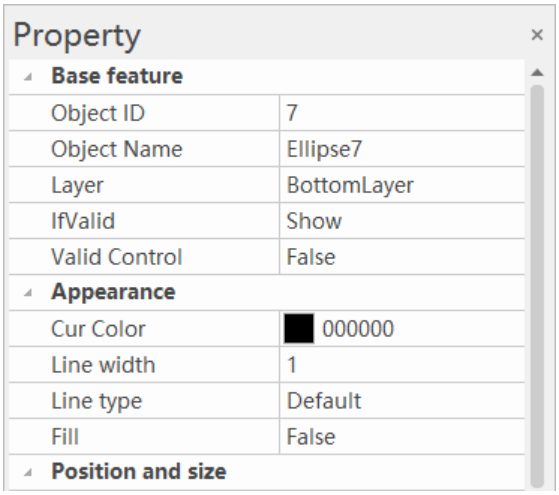
a. when drawing a horizontal or vertical line, just hold down the "Crtl" key and move the mouse to determine the direction and length.

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



D. "Property" Description

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.
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
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?

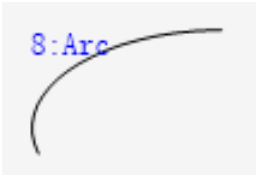
The curve between any two points on a circle is an arc.

B. How to Use?

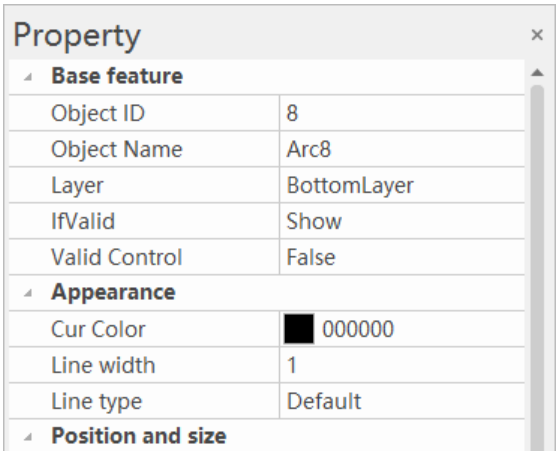
Click "Control Class", select "Ellipse / Arc / Sector" from "Vector graphics". At first, draw one rectangle to confirm the size, then determine the arc starting point and end point, next, draw the arc from the starting point to end point clockwise, at last, click "complete".

Note:

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



D. "Property" Description

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.
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.
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?

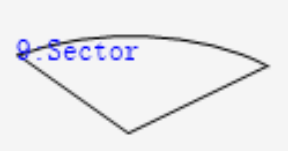
A closed object consisting of an arc and two radiuses.

B. How to Use?

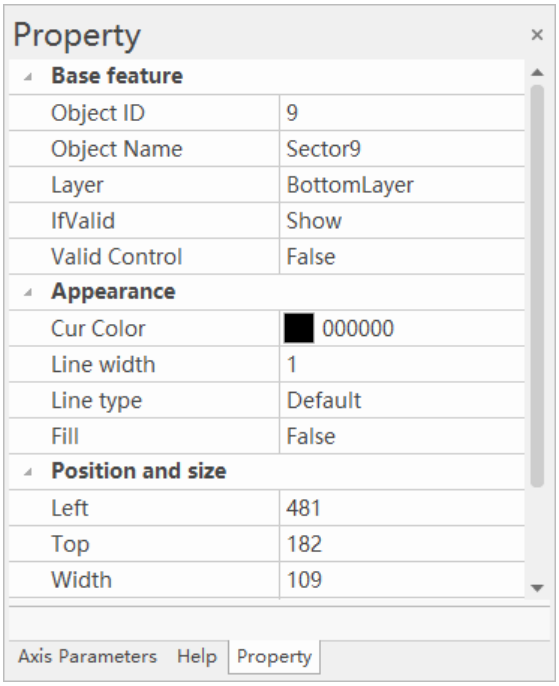
Click “Control Class”, select “Ellipse / Arc / Sector” from “Vector graphics”. At first, draw one rectangle to confirm the size, then determine the arc starting point and end point, next, after that, right click “sector”, it will generate one sector at the clockwise direction.

Note:

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



D. "Property" Description

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.
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
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?

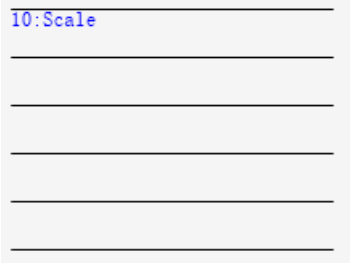
Draw multiple rows of contour lines to serve as auxiliary marks for component alignment, etc.

B. How to Use?

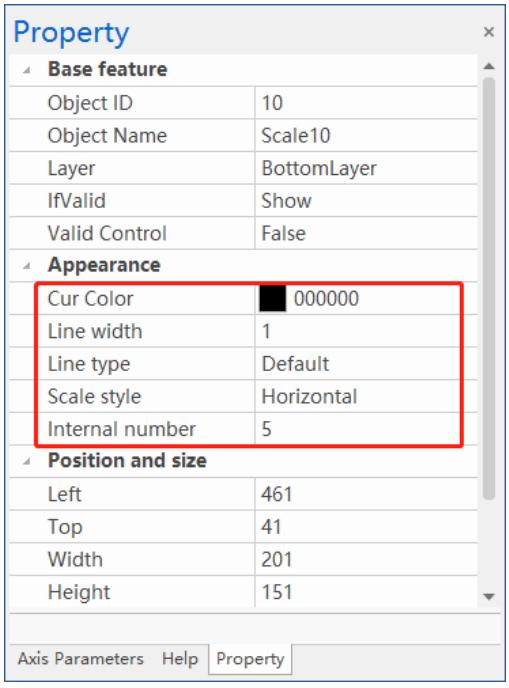
Click “Control Class”, select “Scale” from “Vector graphics”. You only need to move the mouse to any position in the window, then click. You can adjust the size, paragraph, line direction in the property window.

Note:

a. 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



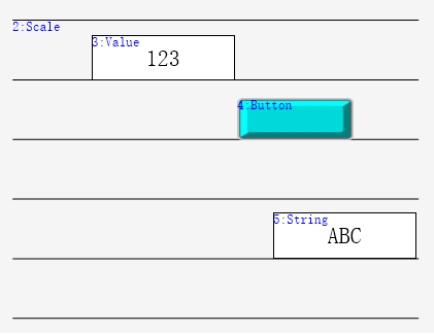
D. “Property” Description

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.
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.
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. <u>For Example</u>

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?

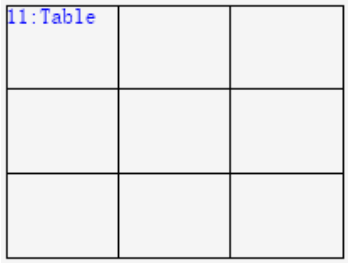
Click “Control Class”, select “Table” from “Vector graphics”. You only need to move the mouse to any position in the window. When you move the object to table position, it will automatically align, and there are many kinds of alignment methods.

Note:

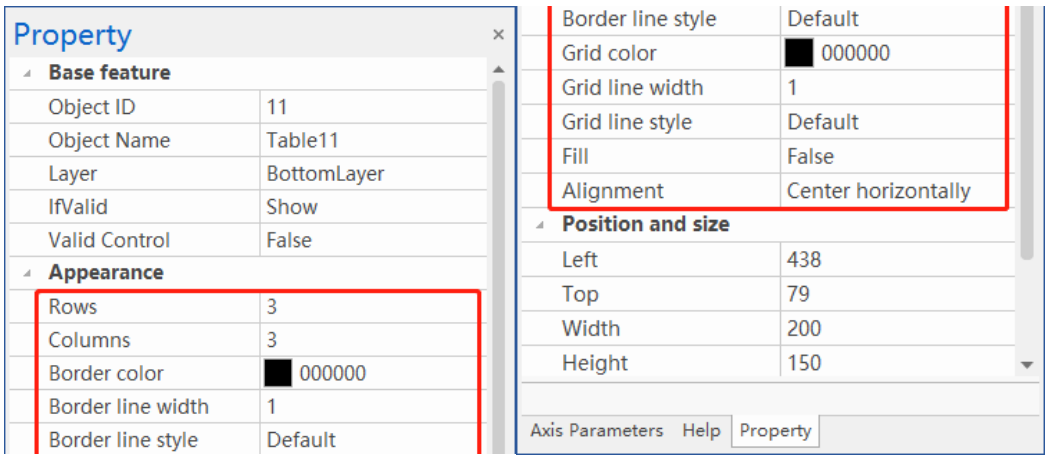
a. One table only can set one alignment method.

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



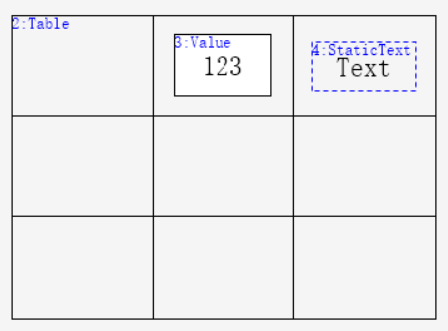
D. “Property” Description

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.
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
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. <u>For Example</u>

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.



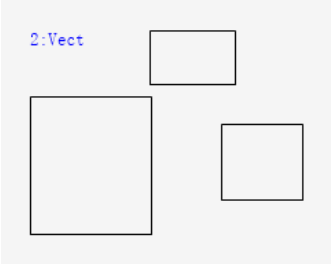
4.3.7. Import Vector Graphics

A. What is It?

Import the vector graphics.

B. How to Use?

Click “Control Class”, 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.



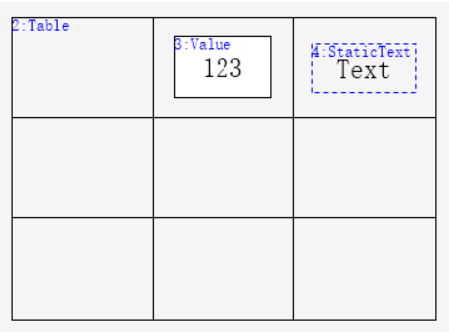
C. “Property” Description

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.
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.
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. <u>For Example</u>

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.



4.3.8. Static Text

A. What is It?

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

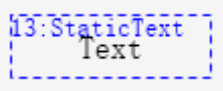
B. How to Use?

Click “Control Class”, select “Static Text” from “Common Controls”. When you selected, put the rectangle frame at any position, then you can enter the text directly.

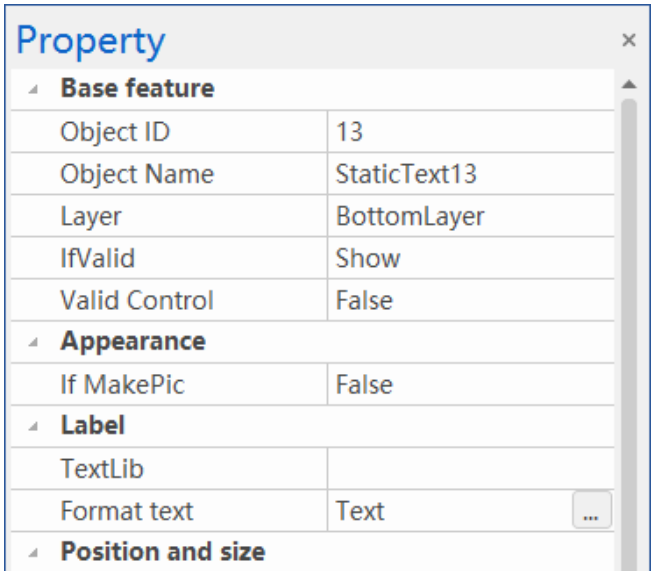
Notes:

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 4.2.6.

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



D. “Property” Description

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.
If MakePic	Set whether the component changes as graphic, the default is False.
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.
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 and show static picture.

B. How to Use?

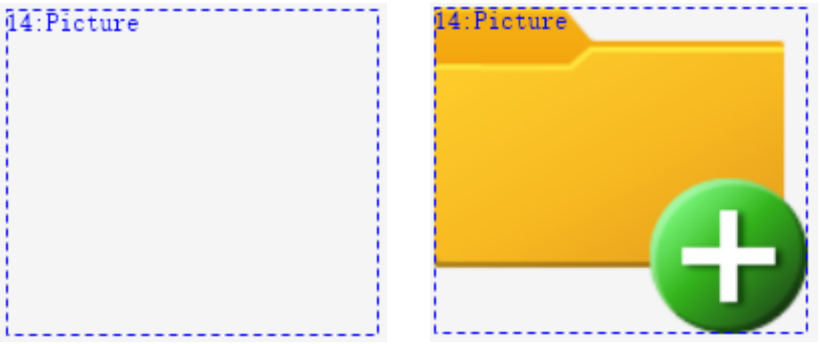
Click "Control Class", select "Picture" from "Common Controls". Add one "picture" component in the window, then in corresponding property, you can add the picture from Back picture lib or Back picture.

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

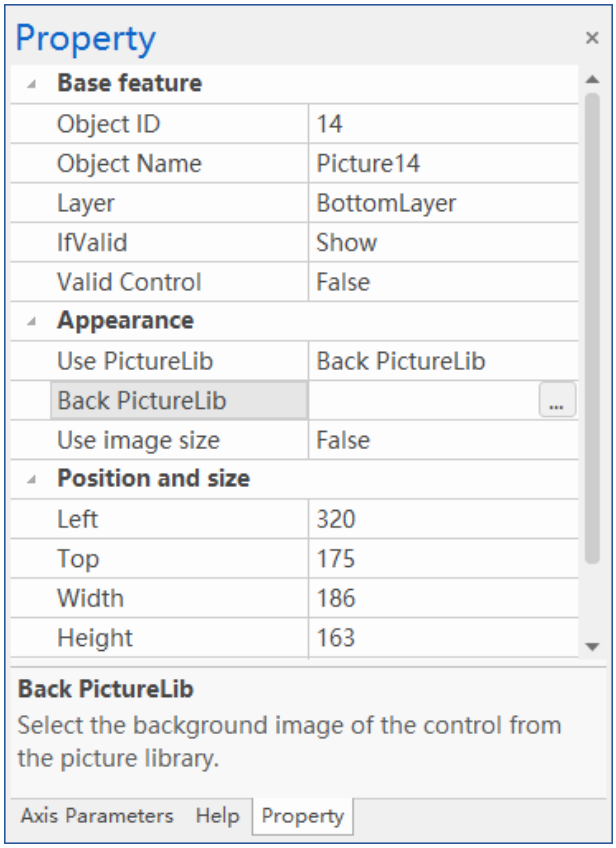
**from Back Picture:* add the picture into "Project View" at first, then select "Back picture", and find needed one.

Notes:

a. 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. "Property" Window



D. "Property" Description

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.
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.
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

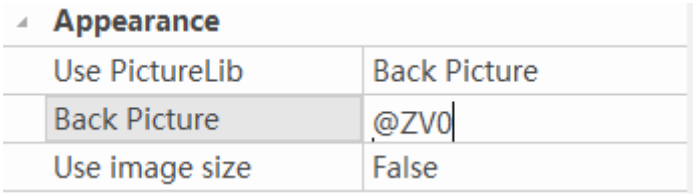
Step 1: put the picture into controller / simulator flash folder.

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.



Step 4: 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 “Control Class”, select “Bit State” from “Common Controls”. 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.

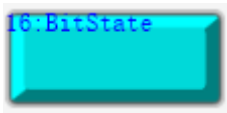
Method 2: in “output” window, enter register address and state value, and click “send”, for example, enter “MODBUS_BIT(0)=1”.

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).

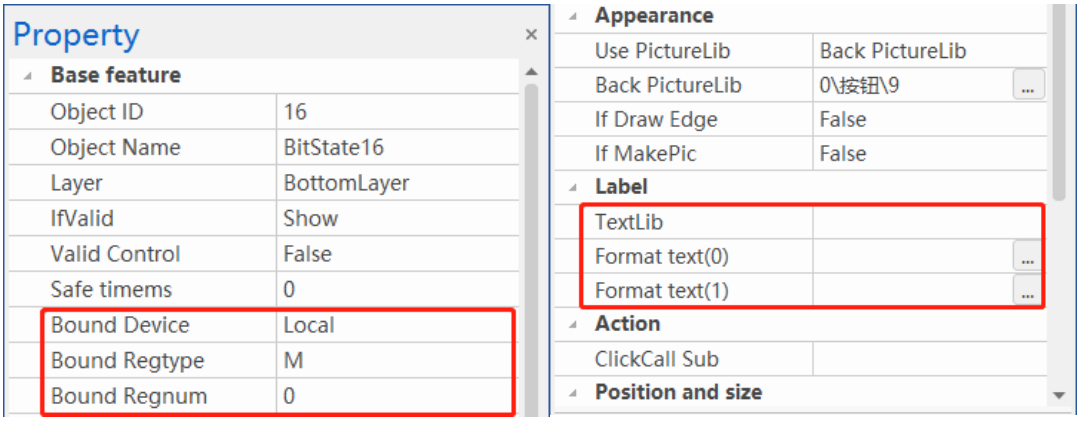
Notes:

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



D. "Property" Description

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

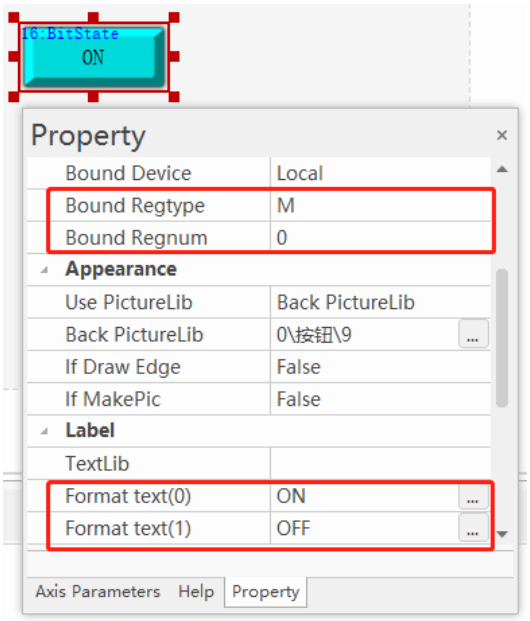
	is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
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
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)	
ClickCall Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Use Image Size	Whether to use picture original size, False: use component size, True: use picture original 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., M0 corresponds to MODBUS_BIT(0).

  Step 2: enter corresponding texts under different states.

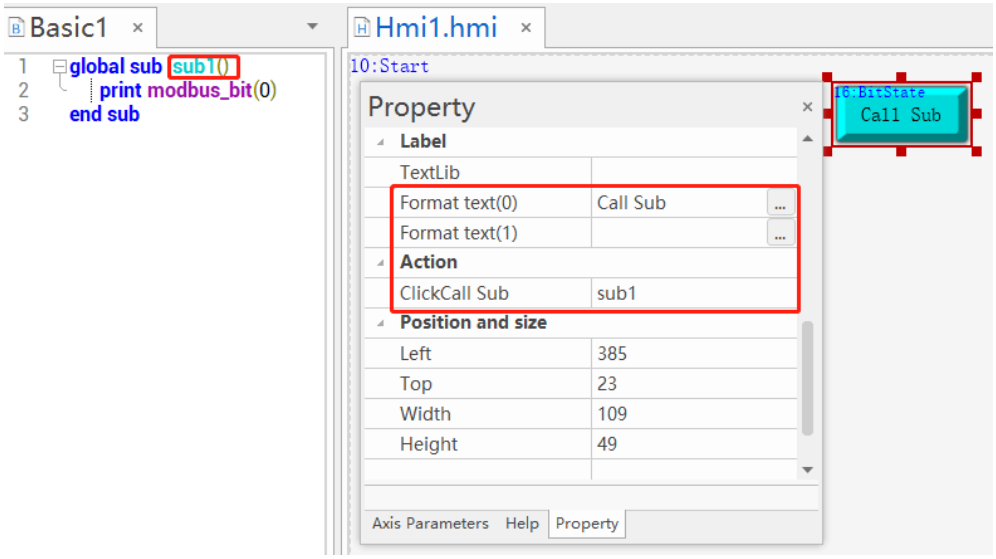


Step 3: check the effect: when MODBUS_BIT(0) = 0, it will show ON. When MODBUS_BIT(0) = 1, it will show OFF.

- **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.



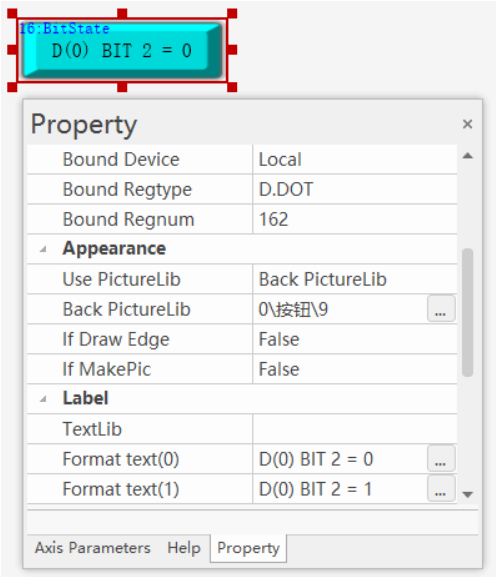
Step 3: check the effect: when the Call Sub function is pressed, it will call BASIC SUB function, each press, call function once.

- **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.



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 “Control Class”, select “Word State” from “Common Controls”. 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”.

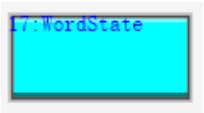
Method 3: in “action”, set call sub, and write related register state switching program in Basic sub function, after that, press the component to call the sub to achieve switching (if no action is set,

no way to achieve that directly).

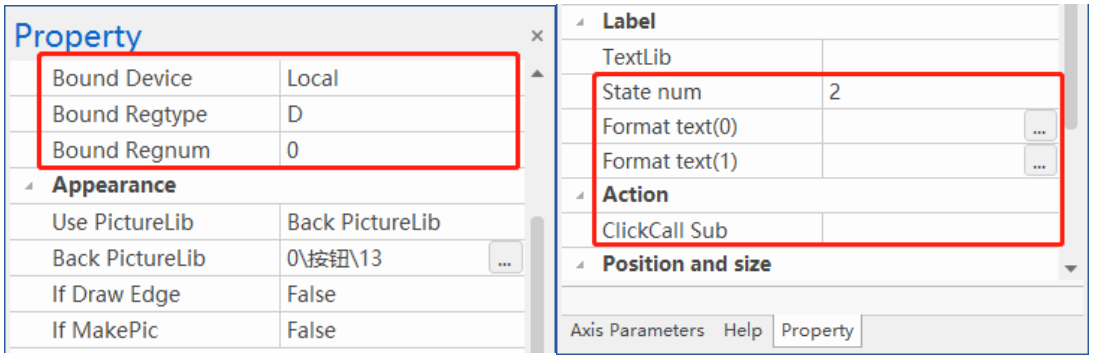
Notes:

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

b. 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



D. "Property" Description

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.

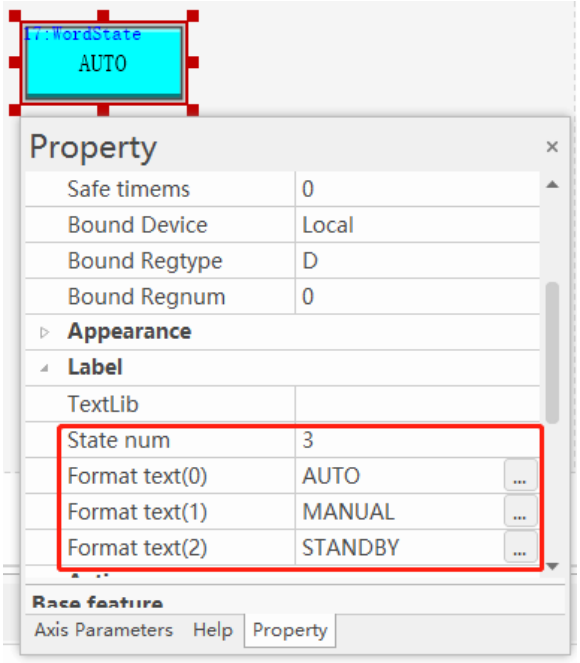
	If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned 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.
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
TextLib	Text library name, if no set, it shows “Text”.
State num	How many states you need, can be 1-256.
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)	
ClickCall Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Use Image Size	Whether to use picture original size, False: use component size, True: use picture original 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.

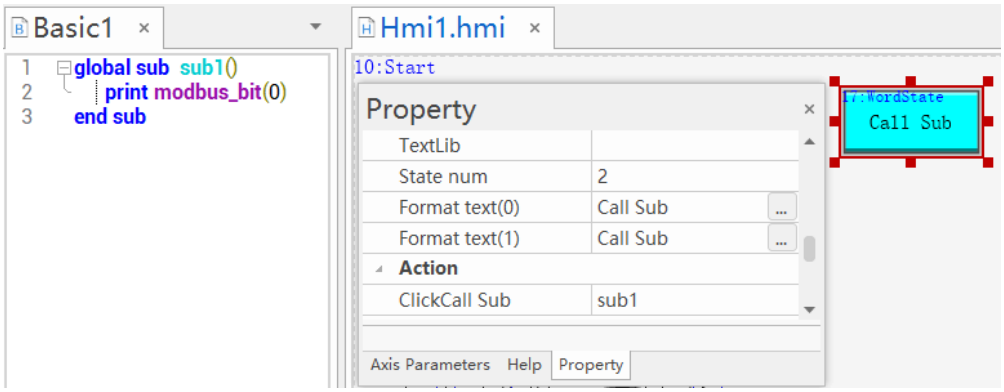


Step 3: check the effect: when MODBUS_REG(0) = 0, it will show AUTO. When MODBUS_REG(0) = 1, it will show MANUAL, when MODBUS_REG(0) = 2, it will show STANDBY.

- **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.



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. 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, when pressed, it will show text 1.

B. How to Use?

Click “Control Class”, select “Bit Modify” from “Common Controls”. 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--

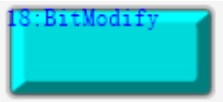
Method 1: click menu Tool – Register, select bound register type and address, then click read.

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)”.

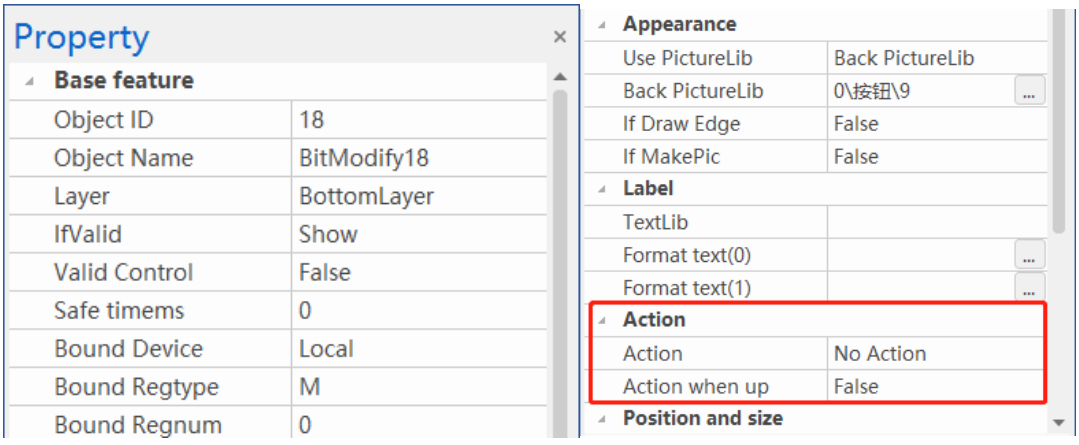
Notes:

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” Window



D. “Property” Description

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	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.
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
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)	
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.

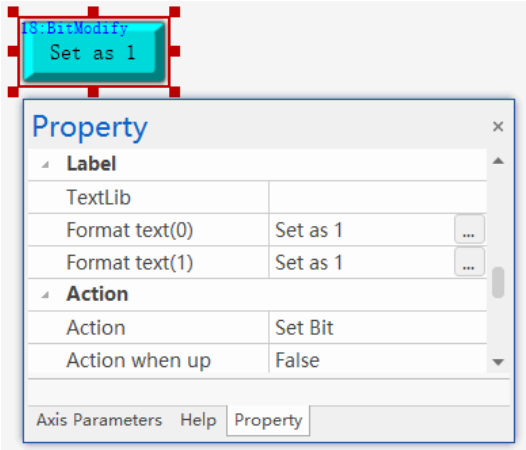
ClickCall Sub	When the button is pressed, the function is called. You can select function name from the drop-dowm list.
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.
Width	Object width
Height	Object height

E. Examples

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

Step 1: select register type and No.

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



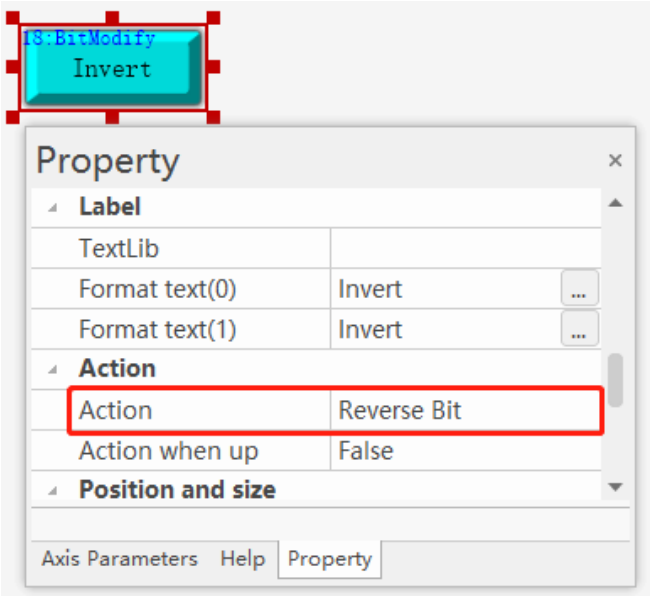
Step 3: check the effect: when set register value as 1 (that is, MODBUS_BIT(0) = 1), at this time, the object shows state 1, when released, object shows state 0, but the register value is still 1. If you selected "True" for "Action when up", that is, after the object pressed and released, MODBUS_BIT(0)=1,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

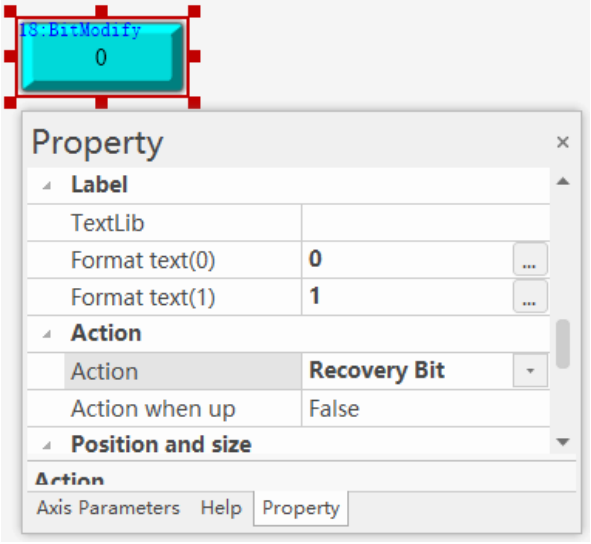


Step 3: check the effect: if initial value of MODBUS_BIT(0) is 0, when the component is pressed, the value will be inversed, that is, MODBUS_BIT(0)=1, when pressed again, MODBUS_BIT(0)=0.

- **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”.



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.

4.3.13. 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 "Control Class", select "Word Modify" from "Common Controls". 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--

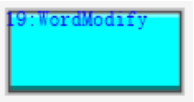
Method 1: click menu Tool – Register, select bound register type and address, then click read.

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)".

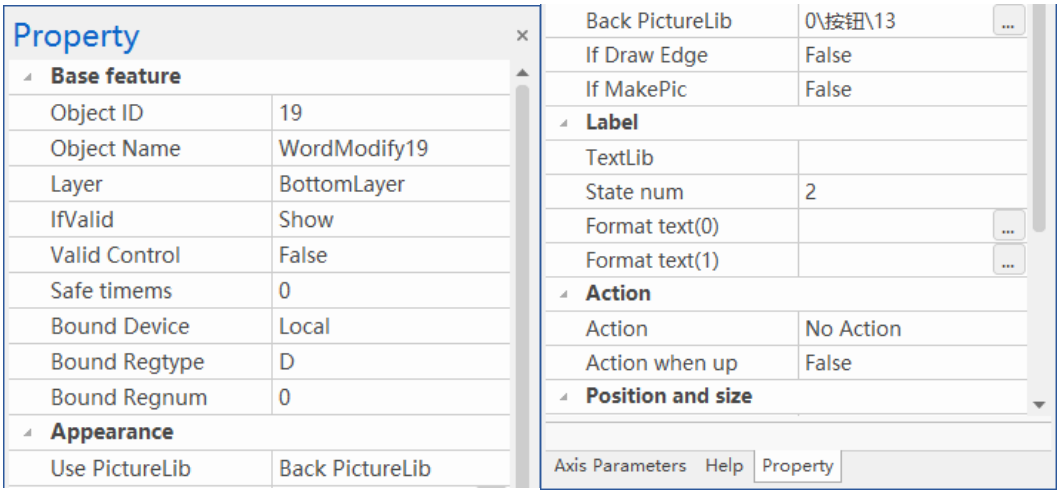
Notes:

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" Window



D. "Property" Description

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	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.

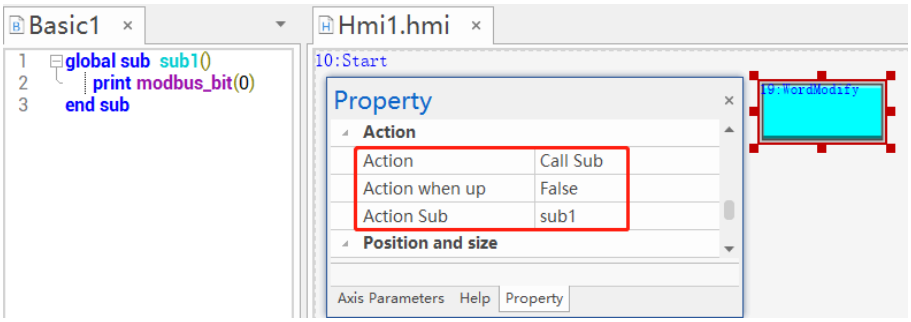
Use Picture Lib	None / use picture library / use back picture
Back Picture Lib	Select one picture from background picture library
Back Picture	Select one picture from background picture
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
TextLib	Text library name, if no set, it shows “Text”.
State Num	How many states for one component (1-256).
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)	
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	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.
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: 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.



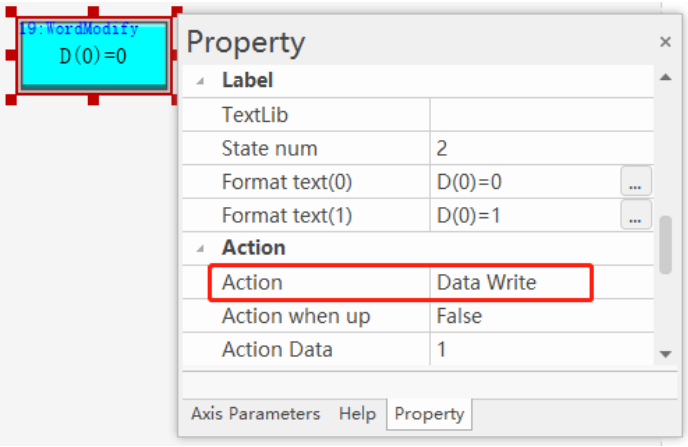
Step 3: check the effect: when the component is pressed, execute SUB function of Basic.

➢ 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.

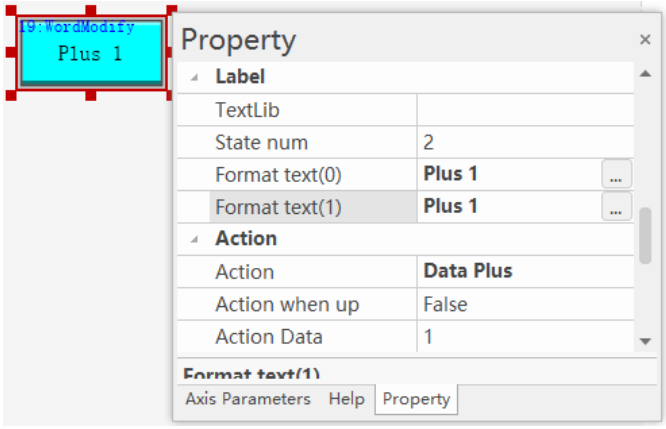


Step 3: check the effect: when pressed, write data 1 to register MODBUS_REG(0). If set “action data” as other values, then write corresponding values into register.

- **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.

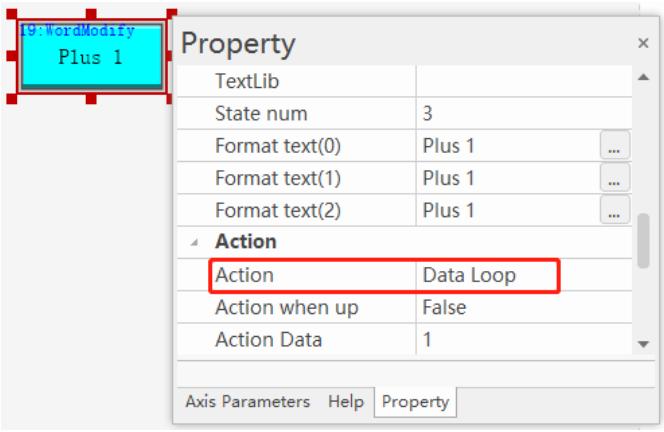


Step 3: check the effect: when the component is pressed once, MODBUS_REG(0)=original value + 1.

- **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.



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.

4.3.14. 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 “Control Class”, select “Bit State” from “Common Controls”. Put the component at suitable position, then in its property window, you need to select register type and address, and enter needed

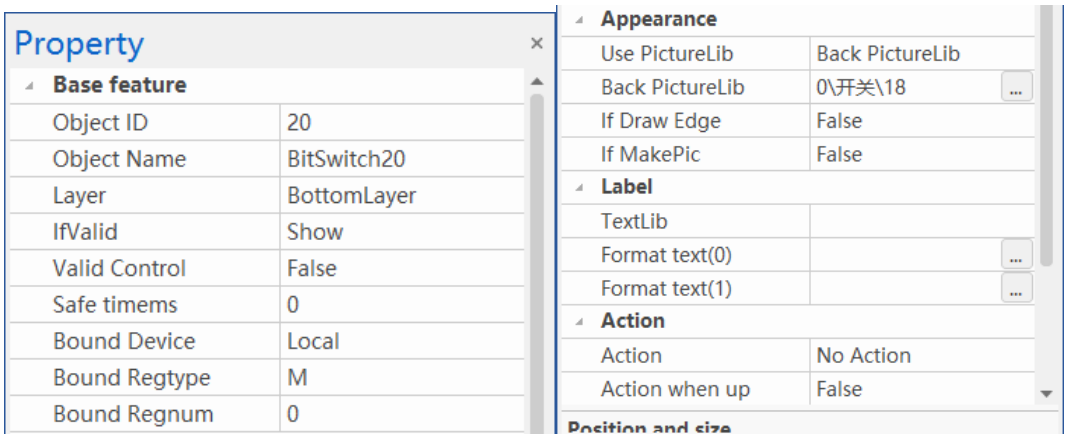
info of different states. And set “action” type. It will set bit register value according to action type.

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” Window



D. “Property” Description

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)

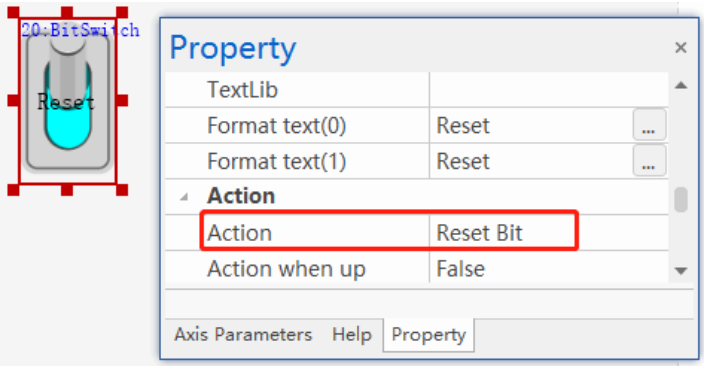
IfValid	Confirm whether this object shows in the interface. ● Show: Objects will be shown and can be called after downloading. ● Hide: not show after downloading ● Show & Disable: show but can't use after downloading.
Valid Control	Determine object is shown or not through register. Default is False. 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	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.
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
TextLib	Text library name, if no set, it shows "Text".
State Num	How many states for one component (1-256).
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)	
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.
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: assign value 1 for register**

Step 1: select register type and No.

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



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.

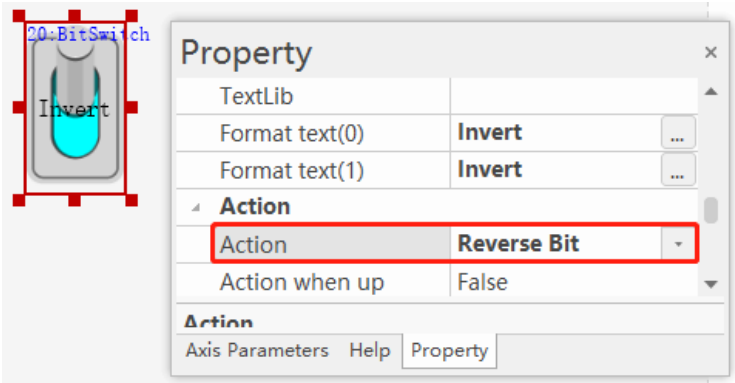
If you selected “True” for “Action when up”, that is, after the object pressed and released, MODBUS_BIT(0)=0, MODBUS_BIT(0) value keeps value 0.

(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”.

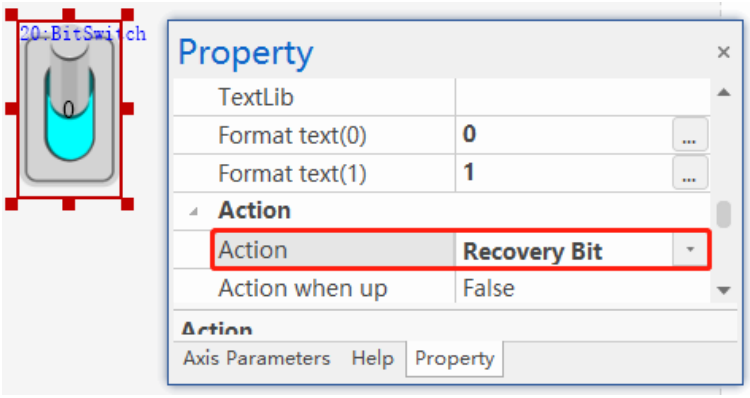


Step 3: check the effect: if initial value of MODBUS_BIT(0) is 0, when the component is pressed, the value will be inversed, that is, MODBUS_BIT(0)=1, when pressed again, MODBUS_BIT(0)=0.

➢ **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”.



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

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

4.3.15. 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 “Control Class”, select “Word Switch” from “Common Controls”. 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.

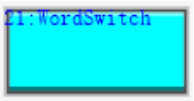
Notes:

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.

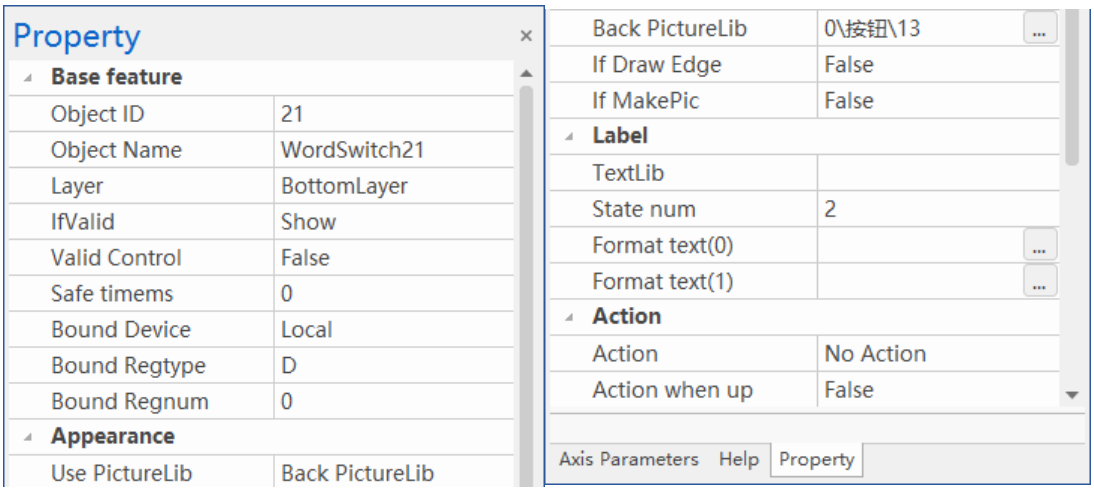
b. You can set how many states for texts in Property – state num, the range is 1-256.

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" Window



D. "Property" Description

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

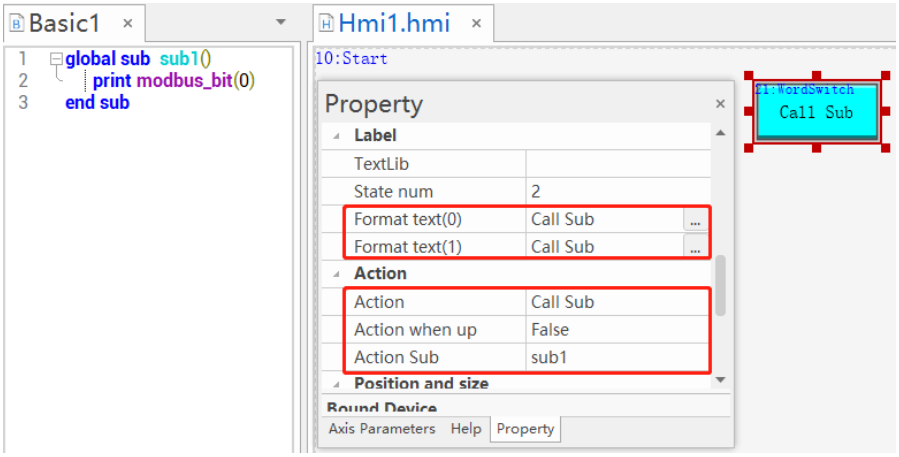
	● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Safe timems	The min button time, the unit is ms
Bound Device	Assigned device, default is local
Bound Regtype	Select register type, you can select from the list
Bound Regnum	Set register No., values obtained from register, then control different states of component.
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
TextLib	Text library name, if no set, it shows “Text”.
State Num	How many states for one component (1-256).
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)	
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	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.
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: 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.

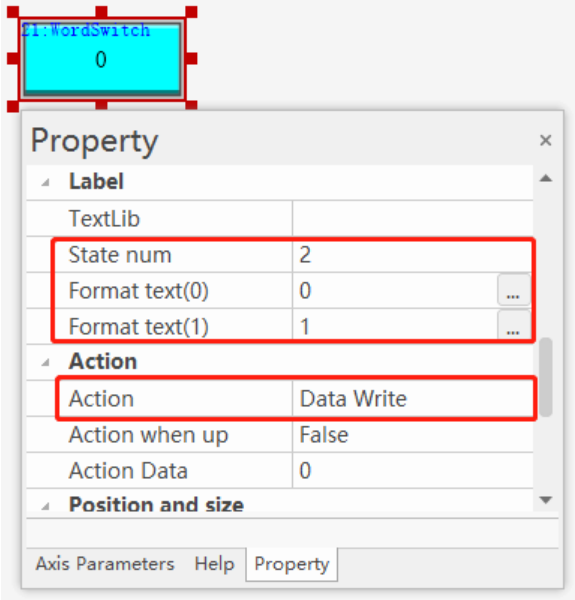


Step 3: check the effect: when the component is pressed, execute SUB function of Basic.

- **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.

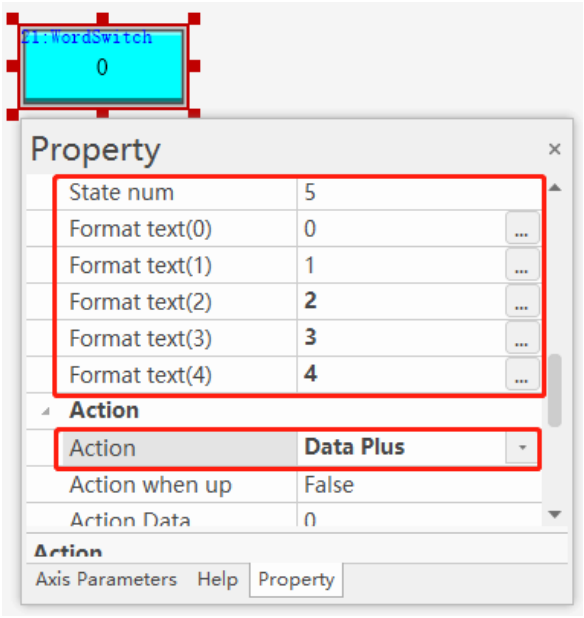


Step 3: check the effect: when pressed, write data 1 to register MODBUS_REG(0), that is, MODBUS_REG(0)=1, and it shows text 1 at the same time.

- **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”.



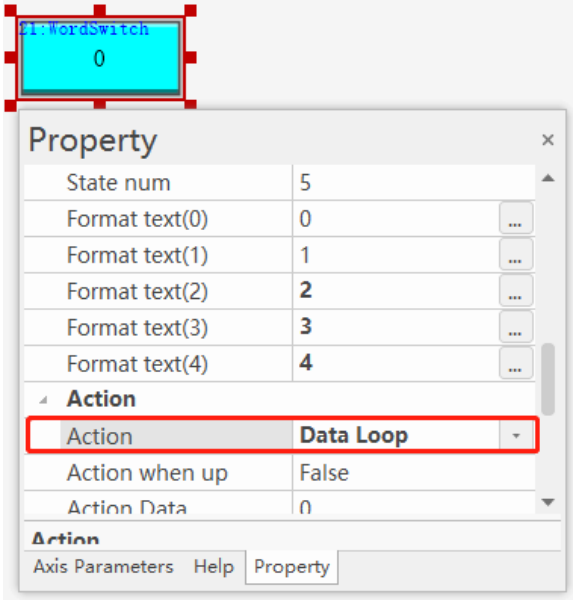
Step 3: check the effect: when the component is pressed once, MODBUS_REG(0)=original value + 1.

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.



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.

4.3.16. 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.

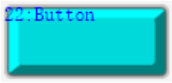
B. How to Use?

Click “Control Class”, select “Button” from “Common Controls”. 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.

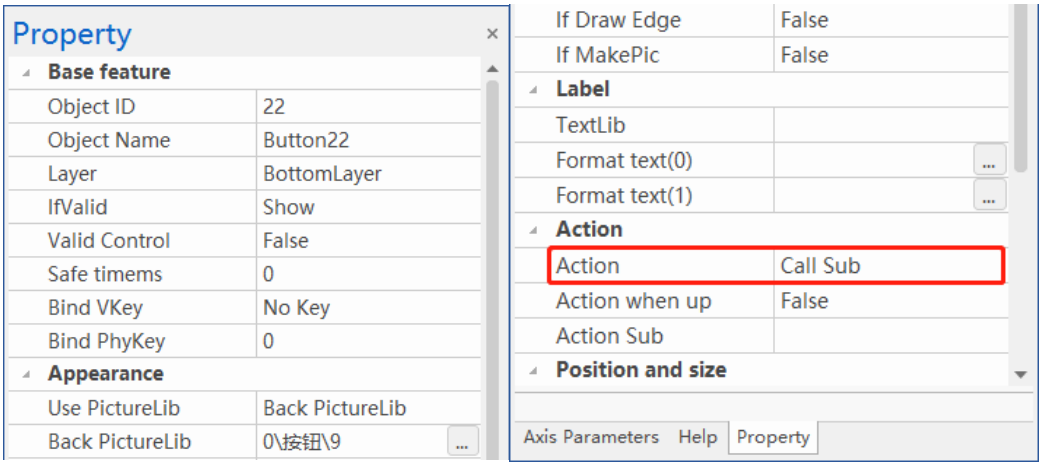
“Button” main functions: call basic function, open / close window, input button / character, switch keyboards. Select them through “property” – “action”.

Notes:

a. Button doesn’t support binding with register.



C. “Property” Window



D. "Property" Description

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	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.
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

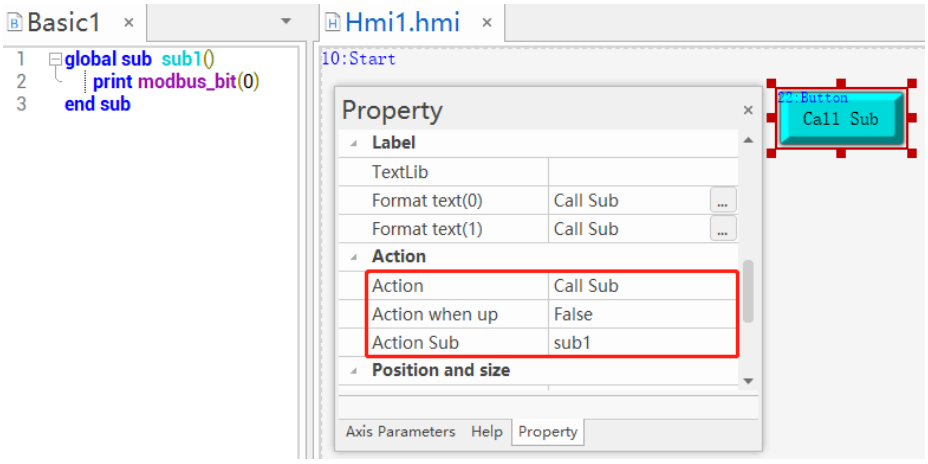
If Draw Edge	Whether to draw the draw
If MakePic	Whether to make the component as graphic, default is False
TextLib	Text library name, if no set, it shows “Text”.
Format Text (0)	Text will be shown when opened, register 0 shows text 0, register value is not 0, it will show text 1.
Format Text (1)	
Action	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.
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: Call Sub

Step 1: set button’s action as “Call Sub”.

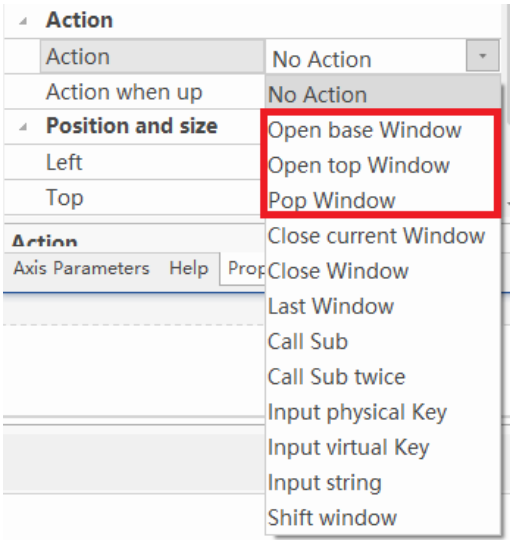
Step 2: then select which one sub function to be called, set it in “action sub”

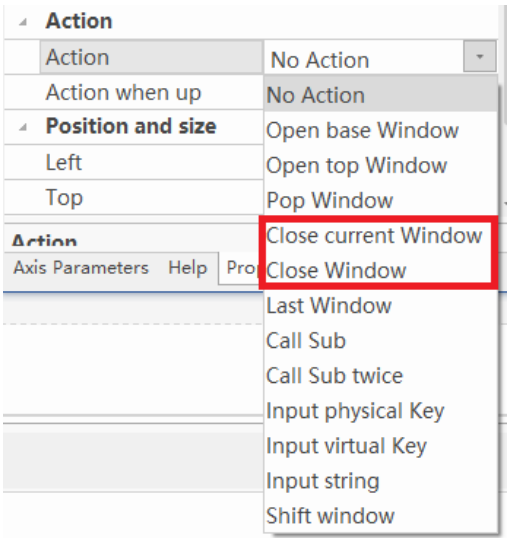


Step 3: check the effect: when the component is pressed, call Basic sub function “sub 1”.

➢ Example 2: Open / Close Window

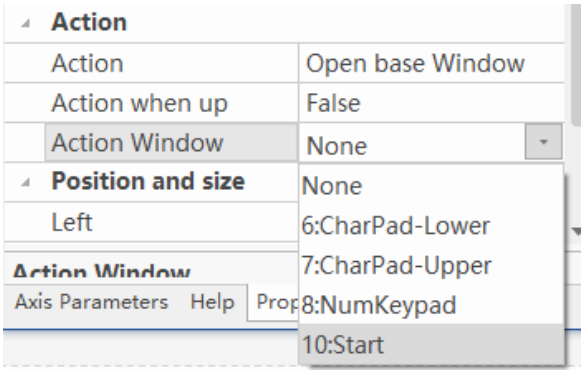
Step 1: in “action” selection of button, you can select below these:





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.



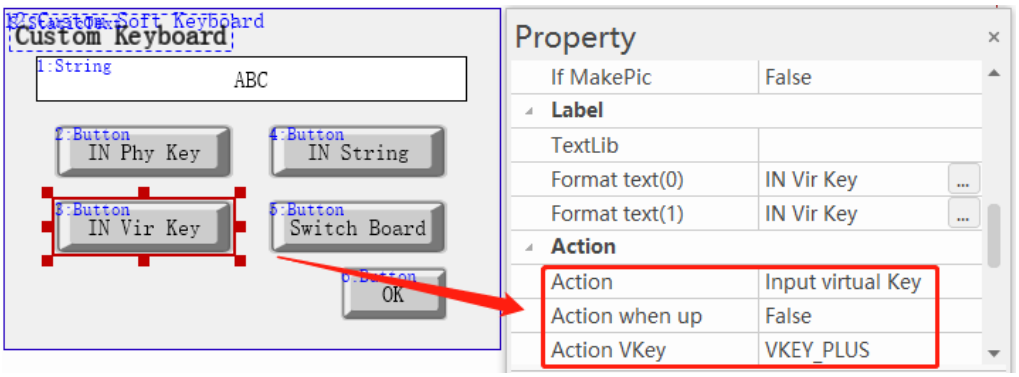
Step 3: check the effect: when pressed, window 10 will be opened.

➢ **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.

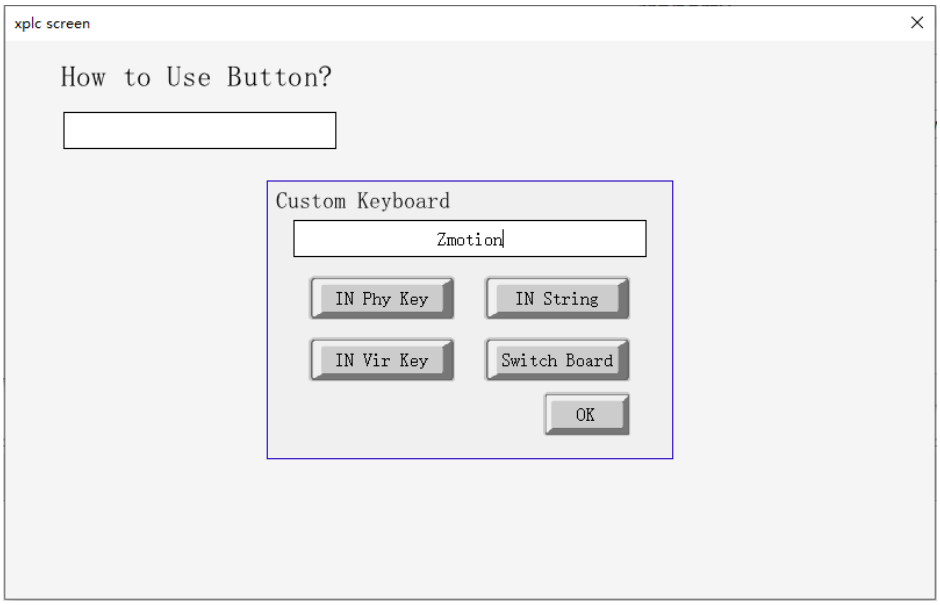
Step 2: in base window, add one “string” component, and in string property, select “true” for “Allow Edit”, then select the new created soft keyboard window No.

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.



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.



4.3.17. 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 “Control Class”, select “Key Button” from “Common Controls”. 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

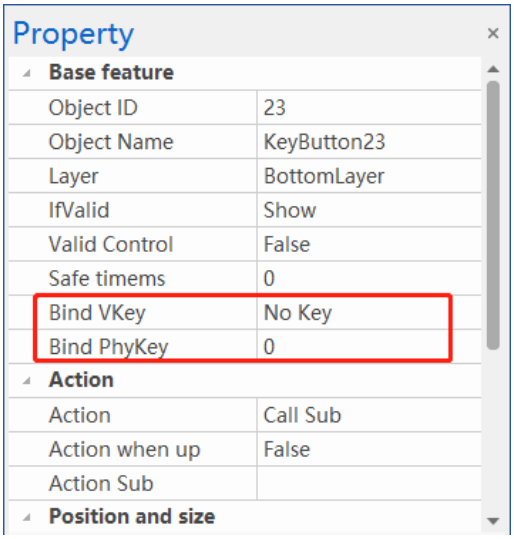
through virtual keys / real hardware buttons (for example, if it binds with one certain button of teach pendant key, select “Call Sub” for action, then when you press the teach pendant button, you can call corresponding sub function).

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” Window



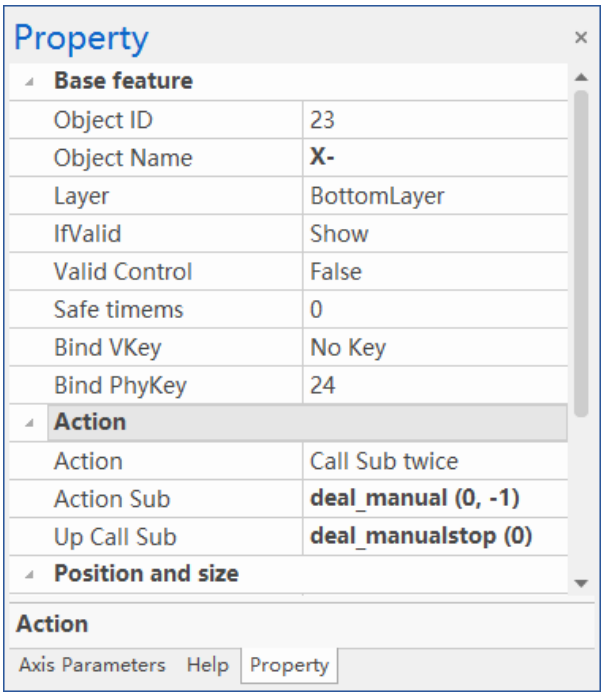
D. “Property” Description

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	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.
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 virtual key code, default: no key.
Action String	Input character string, it only can be used in soft keyboard window.
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. <u>For Example</u>

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.



4.3.18. List

A. What is It?

Show multiple states through list. The number of list rows can be set by state num (1-256).

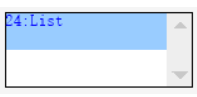
B. How to Use?

Click “Control Class”, select “List” from “Common Controls”. 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.

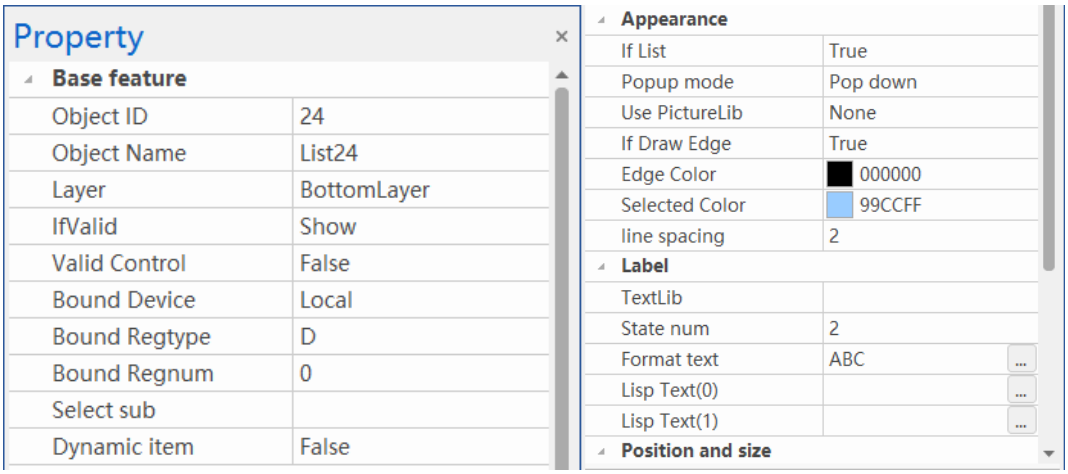
Notes:

a. When you adjusted the object height, all list can’t be shown, when running, it will automatically show the scroll bar.

b. When you select one row text, bound register value corresponds to text state value.



C. “Property” Window



D. "Property" Description

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.
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.
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
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
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

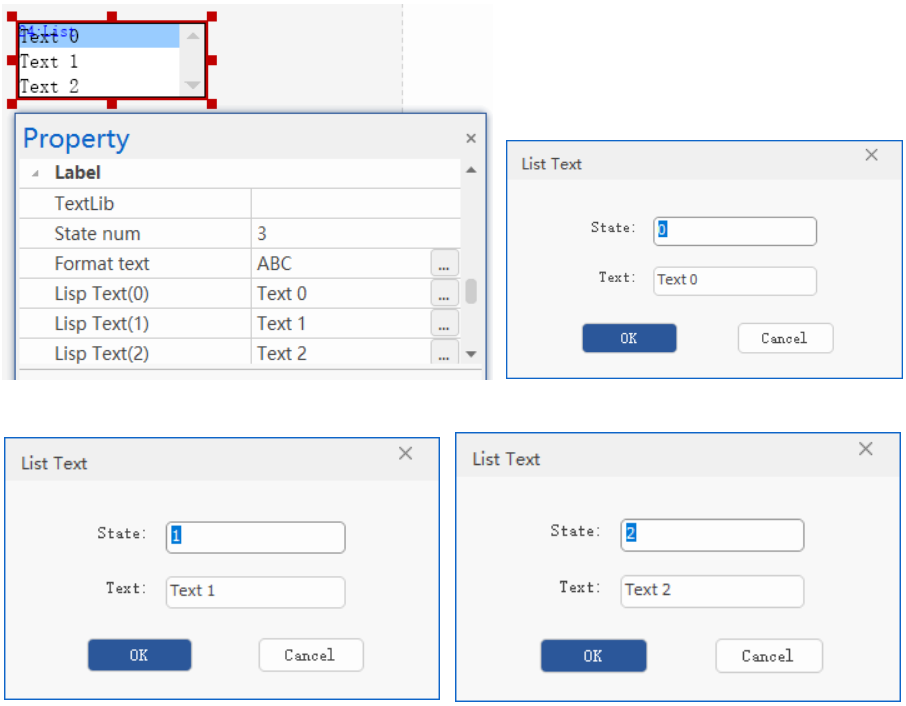
Step 1: select register type and No.

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.



Effect:

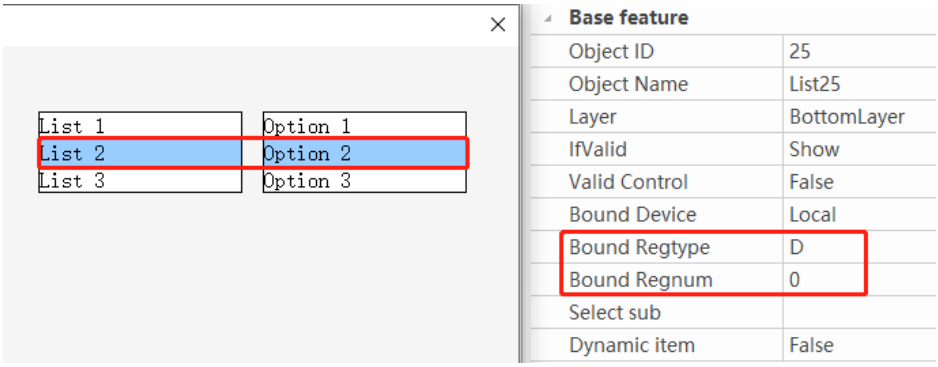
When text 0 line is pressed, MODBUS_REG(0)=0 (value of the first state 0). When text 1 line is pressed, MODBUS_REG(0)=3 (value of the second state 1). When text 2 line is pressed, MODBUS_REG(0)=1 (value of the second state 2), and selected line showing color is set color. Inversely, if you want make MODBUS_REG(0)=0, text 0 line will be selected.

If state values are the same, no way to switch between same states.

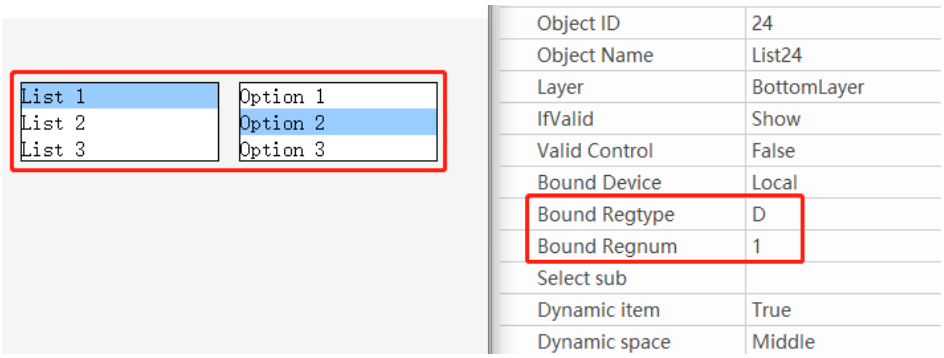
Special Note:

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.



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.



4.3.19. Value

A. What is It?

It can edit and show value, also put the value into bound register.

B. How to Use?

Click “Control Class”, select “Value” from “Common Controls”. 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 Value Component:

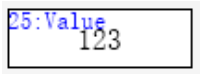
*show register content, only can show non-character type data.

*call keyboard window to set bound register value directly.

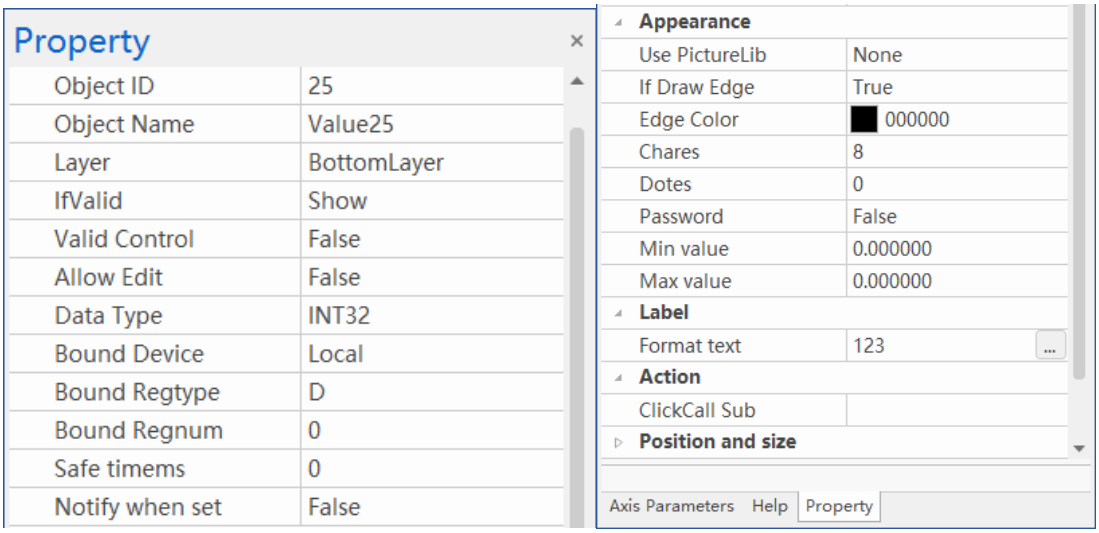
*call SUB function.

Notes:

a. When you set “True” for “allow edit”, and after selecting call keyboard window, call sub can’t be used.



C. “Property” Window



D. “Property” Description

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

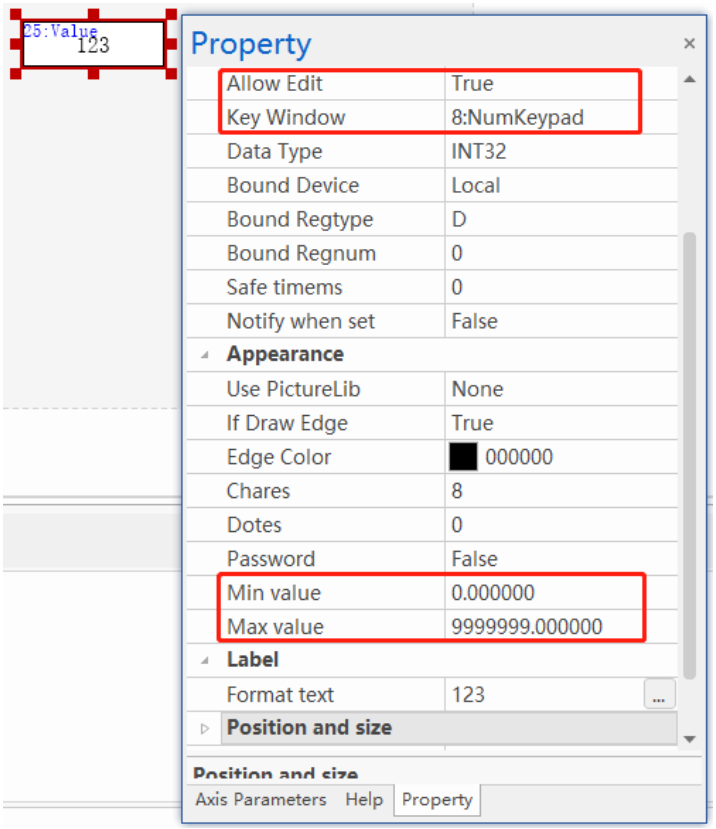
	● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Allow Edit	Whether ON input the data component. Default is False, if you select True, it will keyboard input.
Data Type	Set “value” component data type, default is INT32.
Bound Device	Assigned device, default is local.
Bound Regtype	Set 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 times	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.
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
Format text	Open the setting window, you can set the content of component, and the number of texts is determined by state num.
ClickCall Sub	Call Basic function.
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. <u>For Example</u>

Step 1: select the keyboard window to be called for “allow edit”.

Step 2: set register type and No.

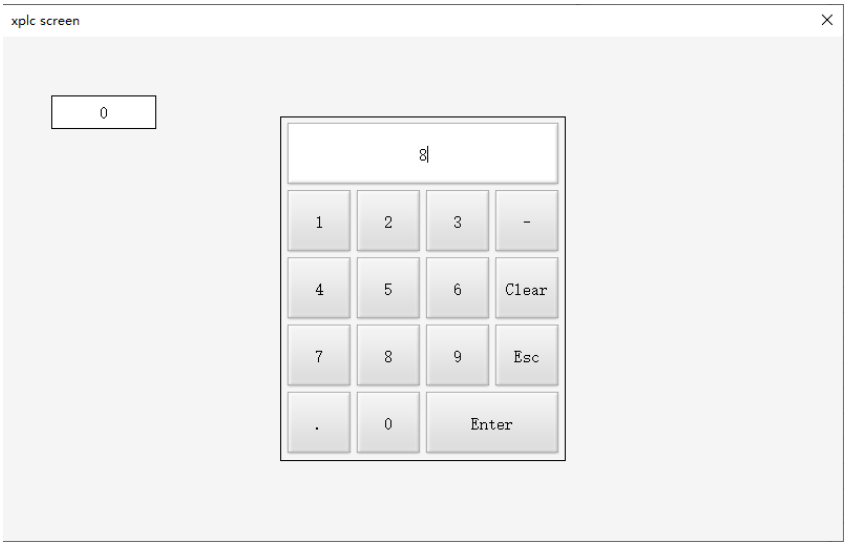
Step 3: set data type, max value, min value, etc. as needed.



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.



4.3.20. String

A. What is It?

It can edit and show all data that belong to character sting type.

B. How to Use?

Click “Control Class”, select “String” from “Common Controls”. 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.

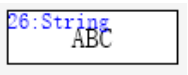
Notes:

a. The data of register type must be character string type.

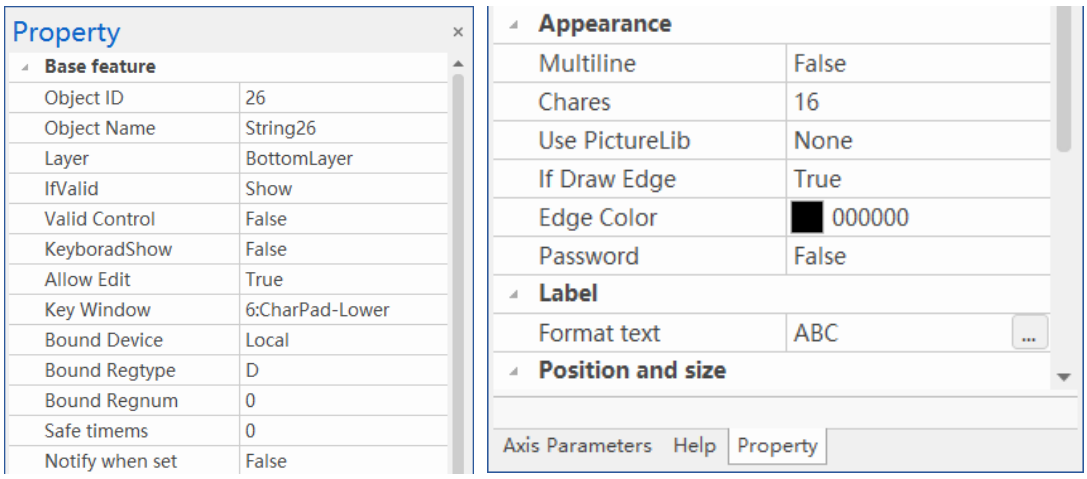
b. For “call keyboard window”, you can custom input data and show them. When the register type is custom @ and set variables are not arrays, keyboard window can’t be called and edited.

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” Window



D. “Property” Description

Base Feature	Description

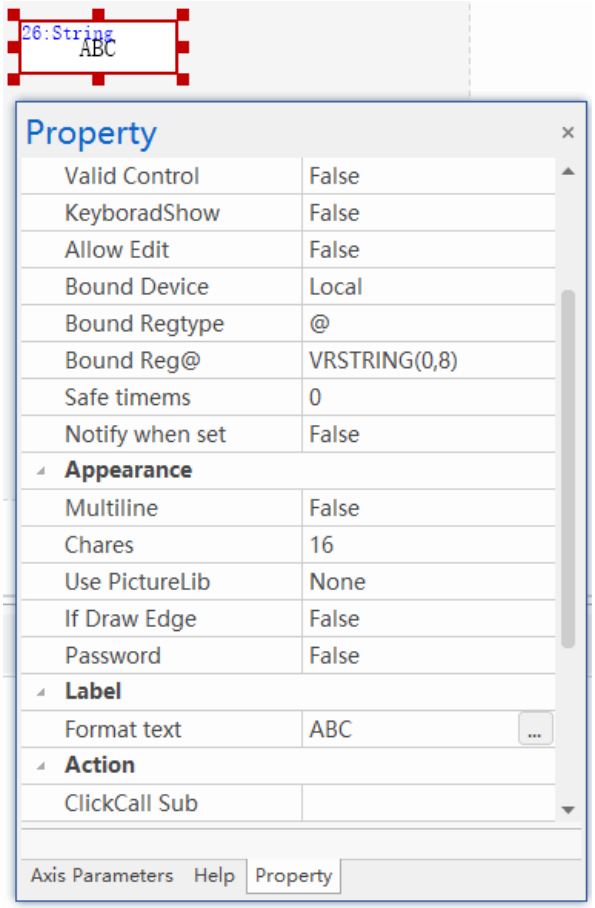
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below components. ● MidLayer: 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.
Allow Edit	Whether ON input the data component. Default is False, if you select True, it will keyboard 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 times	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.
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
If Draw Edge	Whether to draw the edge
Edge Color	Set the edge color, True – show the set color
Password	Default is False, when True, the component will show "*".

Format text	Open the setting window, you can set the content of component, and the number of texts is determined by state num.

E. For Example

➢ **Example 1: show character string saved in register**

Step 1: set register type and No., show character string type data.



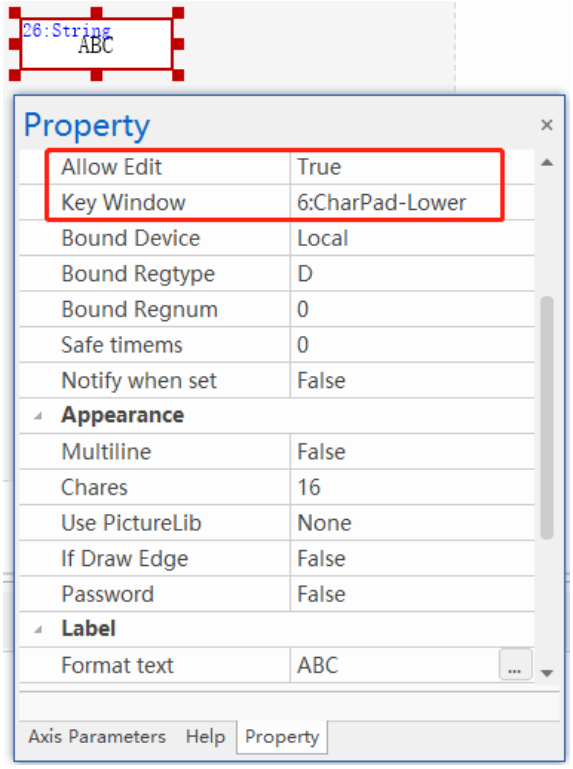
Effect:

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.



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”.

4.3.21. 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.

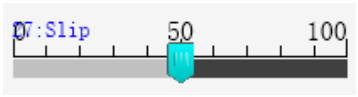
B. How to Use?

Click “Control Class”, select “Slip” from “Common Controls”. Put the component at suitable position, and in its property window, bind register and No., then it can control data by register variables.

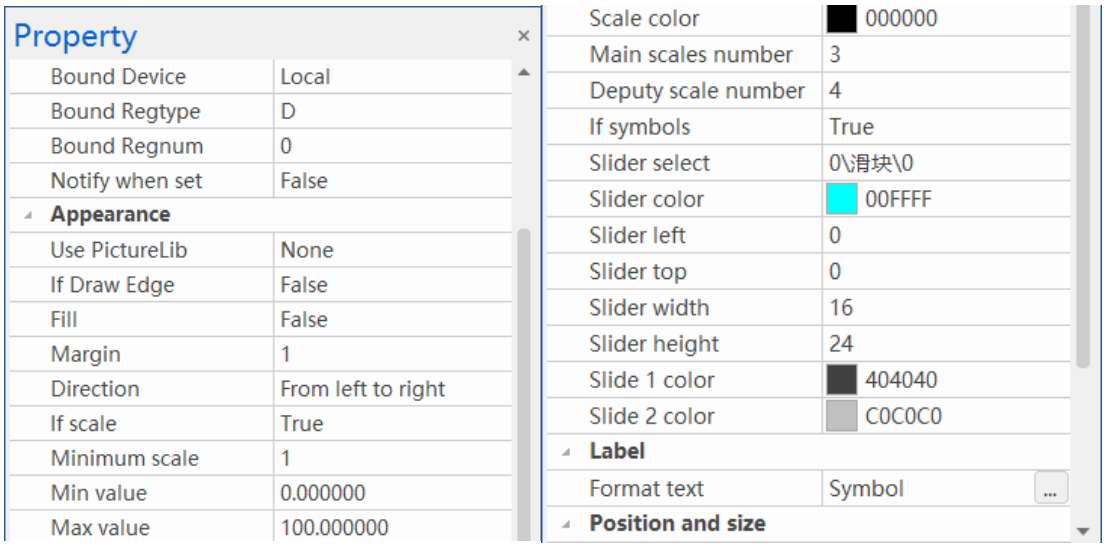
Notes:

a. Set “True” for “allow edit”, when selected the called keyboard window, no way to call sub.

b. 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” Window



D. “Property” Description

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.
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.
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

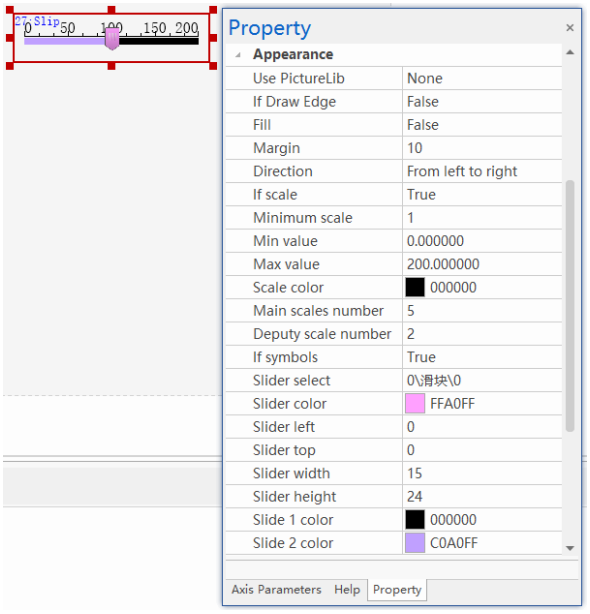
Main scales number	Set the number of long scale segments of the slider
Deputy scales number	Set the number of short scale segments of the slider
If symbols	Whether to show the scale
Slider select	Select slider's format
Slider color	Select slider's color
Slider left	Set silder's horizontal position, when 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

E. 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.



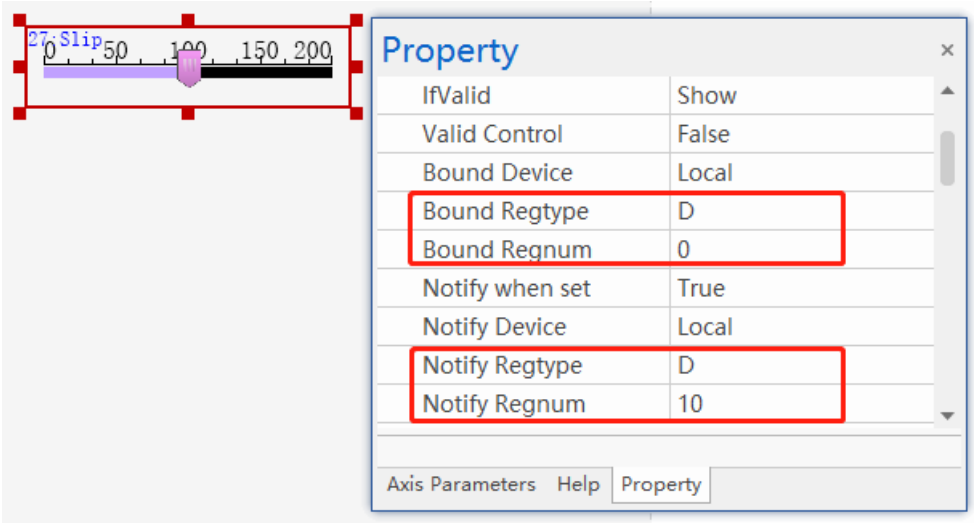
➢ **Example 2: bind register with slip**

Step 1: add “slip” component

Step 2: in its property, please select suitable register type and enter register No., here, select DT0.

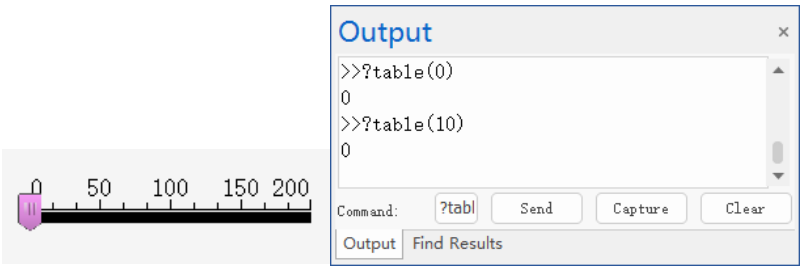
(Note: when you select TRUE for notify when set, notified device No. can’t be same as device No. register)

Step 3: in Basic function, set register initial value, if not set, default value is 0.

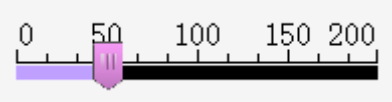


Effect:

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



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.



4.3.22. Timer

A. What is It?

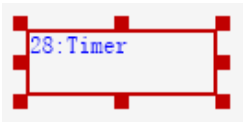
It can refresh regularly and repeat actions.

B. How to Use?

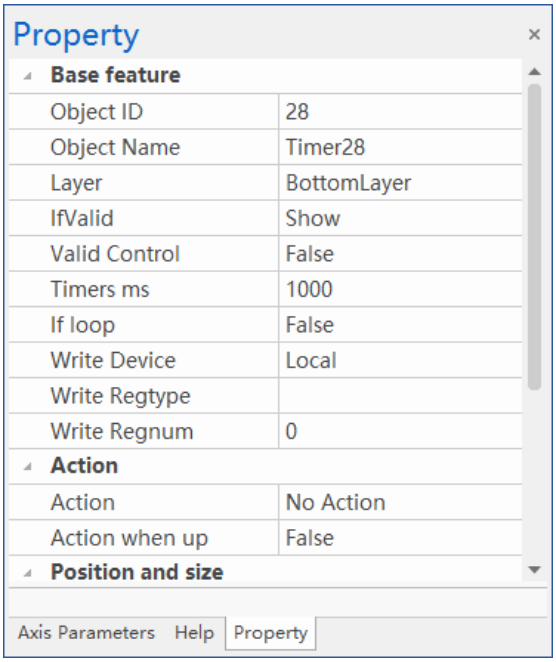
Click “Control Class”, select “Timer” from “Common Controls”. Put the component at suitable position, and in its property window, set timer on space time and action content. Also it can call sub function to assign register.

Notes:

a. HMI interface is shown while editing, not show when running.



C. “Property” Window



D. “Property” Description

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.
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.
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.

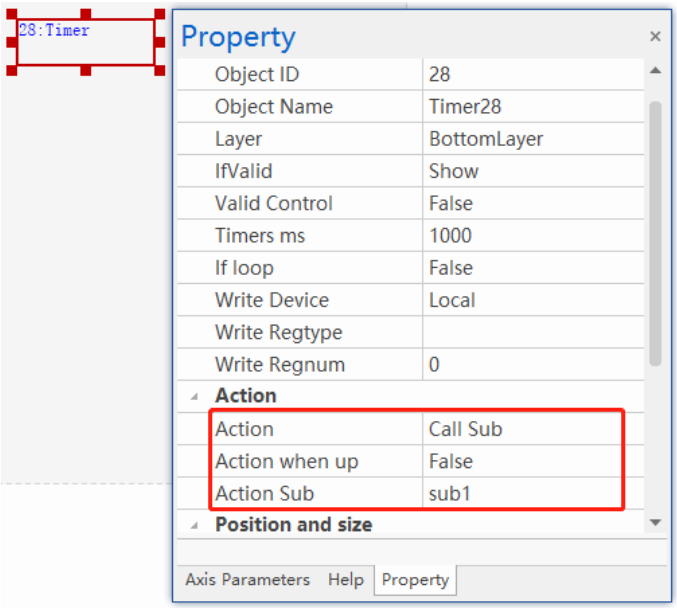
E. For Example

➢ **Example 1: Call SUB Function**

Step 1: in Basic file, edit one global sub function.

Step 2: in component property, select Call Sub for Action, and select corresponding sub name for Action Sub.

Step 3: fill how long the time space for calling sub in “timer ms”.



Effect:

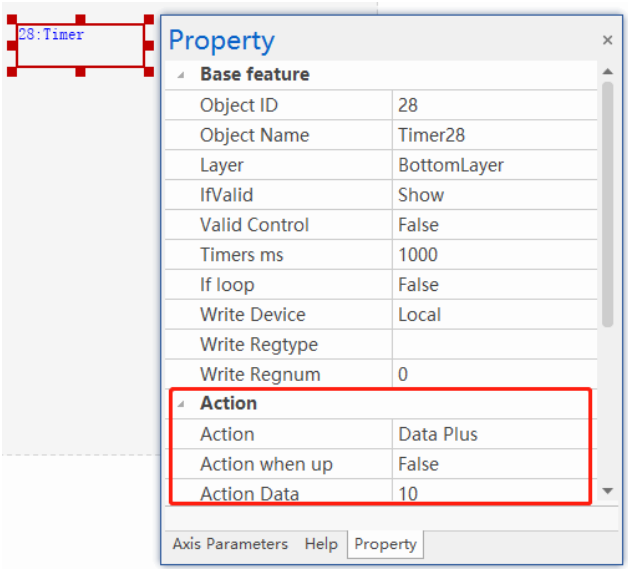
After HMI running, the object will not show, it will call one sub function “sub1” once each span 1000ms.

- **Example 2: register original value + action data value**

Step 1: select register type and No.

Step 2: select “data plus” for action, and 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.



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)".

4.3.23. Custom

A. What is It?

A "custom" component is a component whose behavior is defined 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 "Control Class", select "Custom" from "Common Controls". Put the component at suitable position, adjust the size to determine the drawing area. Then, in BASIC file, please define global defined refresh sub and draw sub at first, next in it property window, find refresh sub and draw sub, and select corresponding sub name.

Notes:

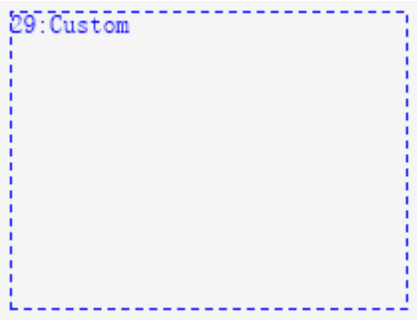
a. The coordinates of the drawing sub are relative to the upper left corner of the component as the zero point.

3 special properties of "custom" component:

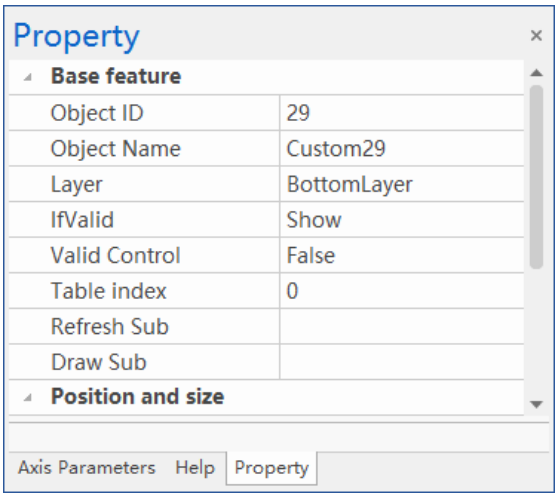
(1) Table Index: usually used for TABLE No., which can indicate where is now component data in TABLE.

(2) Refresh Sub & Draw Sub: it indicates component's SUB functions.

(3) For the usage routine, please refer to website – download – Development Example – Quick Start – HMI.



C. "Property" Window



D. "Property" Description

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.
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.

4.3.24. CAD

A. What is It?

It can show vector graphics.

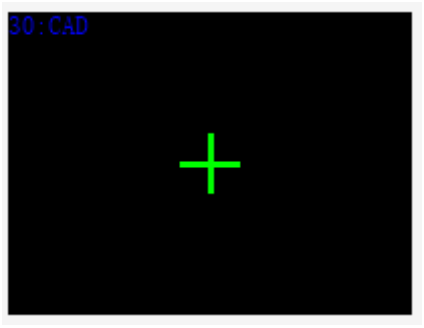
B. How to Use?

Click “Control Class”, select “CAD” from “Enhanced Controls”. Put the component at suitable position, and select the channel No. for importing file 3 (there are only channel 0 and 1), then set layer, path, empty motion color, etc. For import / close graphic, please use together with CAD commands.

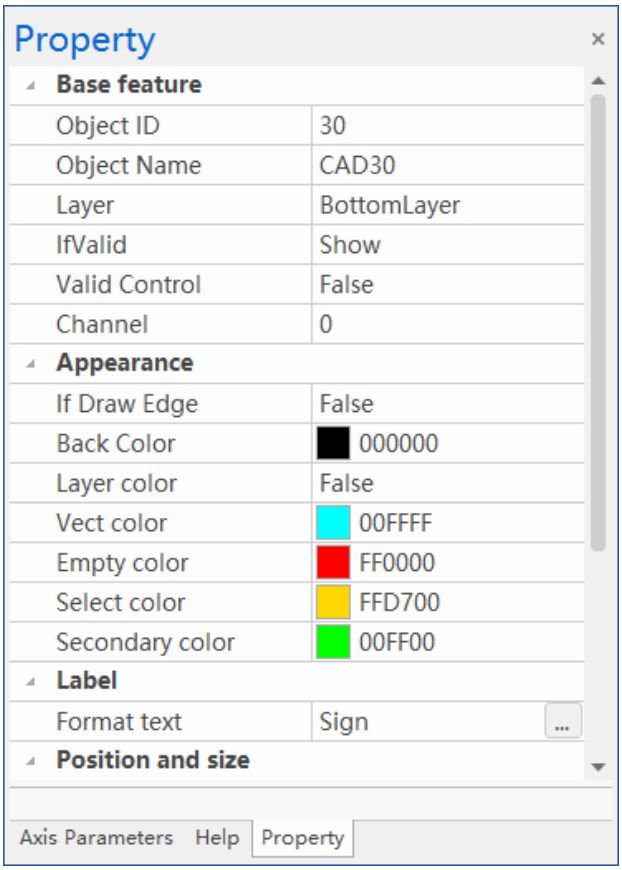
Notes:

a. The vector graphic must be saved into flash folder.

b. Now supported graphic format: .dxf/.ai/.dst/.nc/.cnc.



C. “Property” Window



D. <u>"Property" Description</u>

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.
Channel	Set channel No., only 2 chanels 0 / 1.
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.

For Quick Start CAD, please refer to Chapter VI How to Import Vector Graphic by CAD.

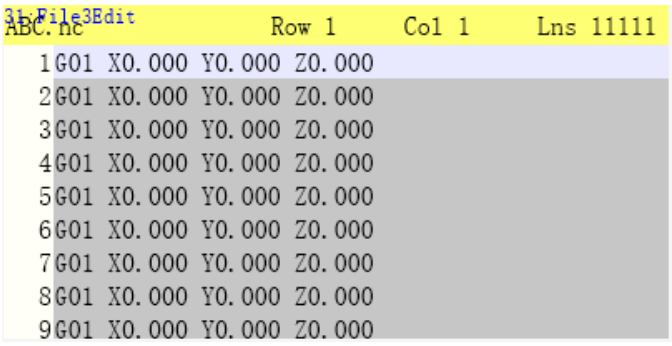
4.3.25. File 3 Editor

A. What is It?

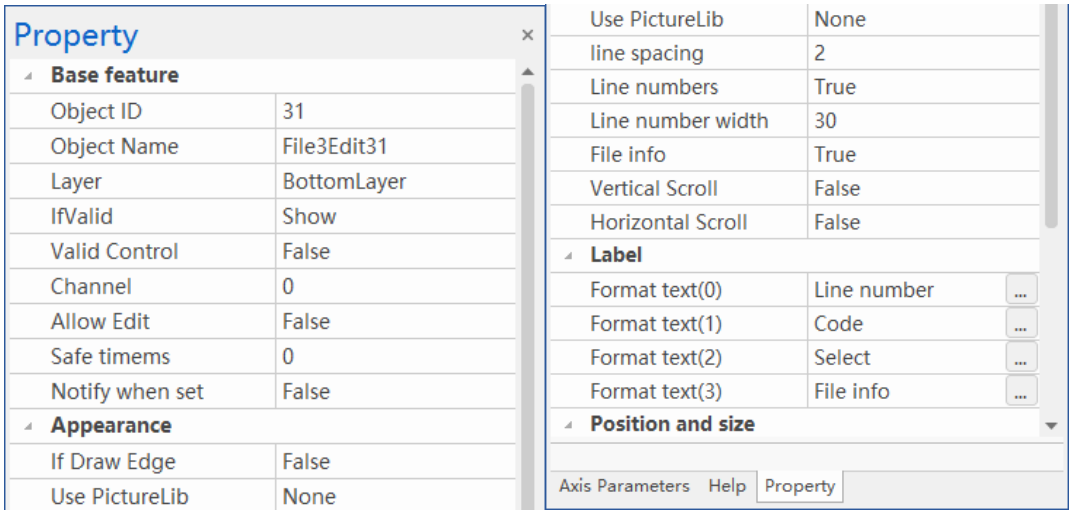
It can develop file 3 program in HMI interface, specifically, edit and show the program. It needs using together with FILE3 commands.

B. How to Use?

Click “Control Class”, select “File3 Editor” from “Enhanced Controls”. 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).



C. "Property" Window



D. "Property" Description

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.
Channel	Set channel No., only 2 chanels 0 / 1.
Allow Edit	Whether ON input the data component. Default is False, if you select True, it will keyboard input.

Safe times	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.
If Draw Edge	Whetehr to draw the edge, default is False.
Use Picture Lib	None / use picture library / use back picture
Line Spacing	The distance between two lines, default is 2.
Line numbers	Whether to show line No.
Line number width	Set line No. width, when TRUE, please set.
File info	Whether to show title line
Vertical Scroll	Set whether to use the vertical scroll bar
Horizontal Scroll	Set whether to use the horizontal scroll bar
Format text 0	Set fromat of line number (text name can not be changed)
Format text 1	Set fromat of code (text name can not be changed)
Format text 2	Set fromat of selected code area (text name can not be changed)
Format text 3	Set fromat of file info (text name can not be changed)

For Quick Start CAD, please refer to Chapter VI Use CAD & File 3.

4.3.26. Report View

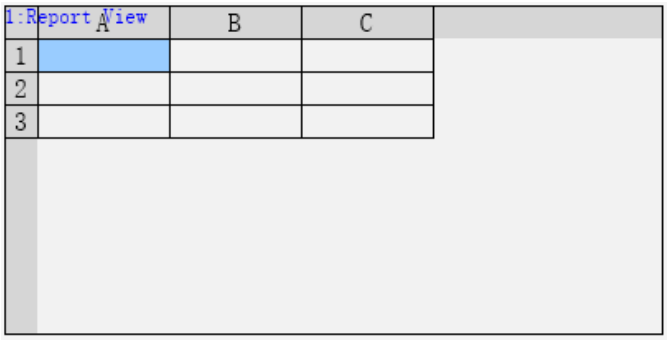
A. What is It?

[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.

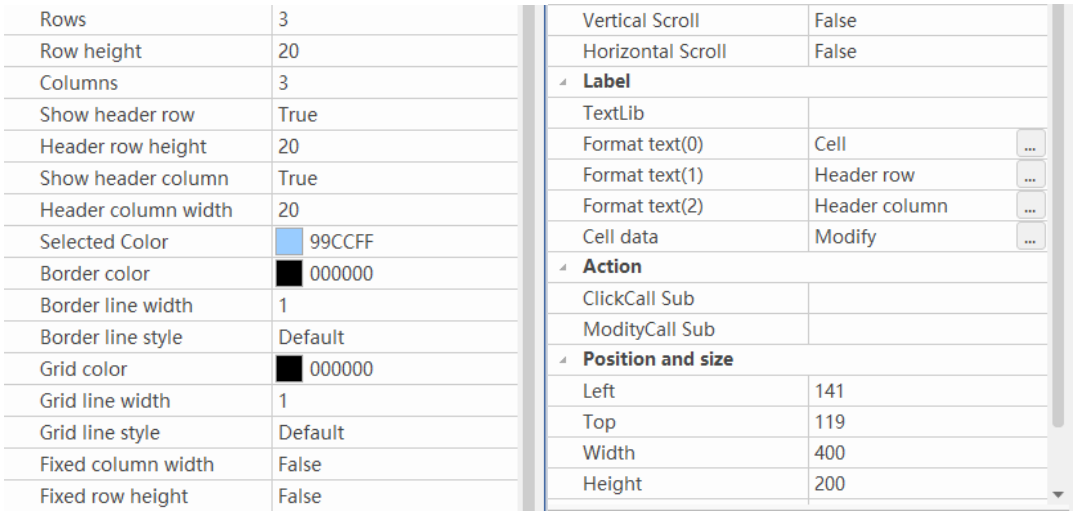
Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

B. How to Use?

Click “Control Class”, select “Report View” from “Enhanced Controls”. 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" Window



D. "Property" Description

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.

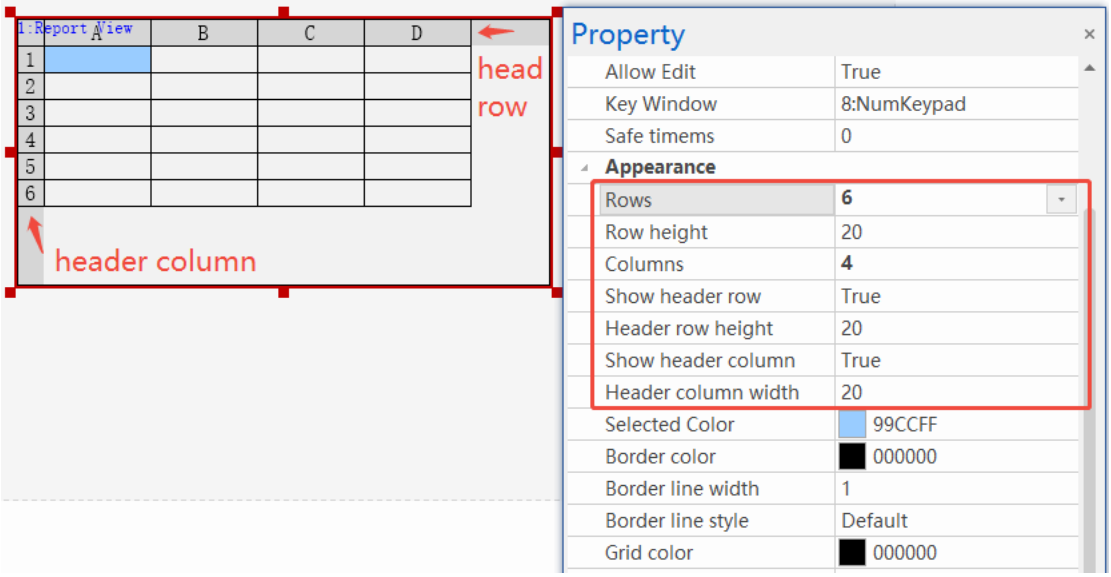
	If TURE, register type and No. must be set (below 3). When register is set as 0, this object will be hiden, if non-0, will be shown. ● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Allow Edit	Whether ON input the data component. Default is False, if you select True, it will keyboard input.
Safe times	The min key press time, the unit is ms.
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
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.
ClickCall Sub	SUB function that was defined by BASIC will be called when pressed. The SUB must be GLOBAL function.
ModifyCall Sub	SUB function that was defined by BASIC will be called when modified. The SUB must be GLOBAL function.
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

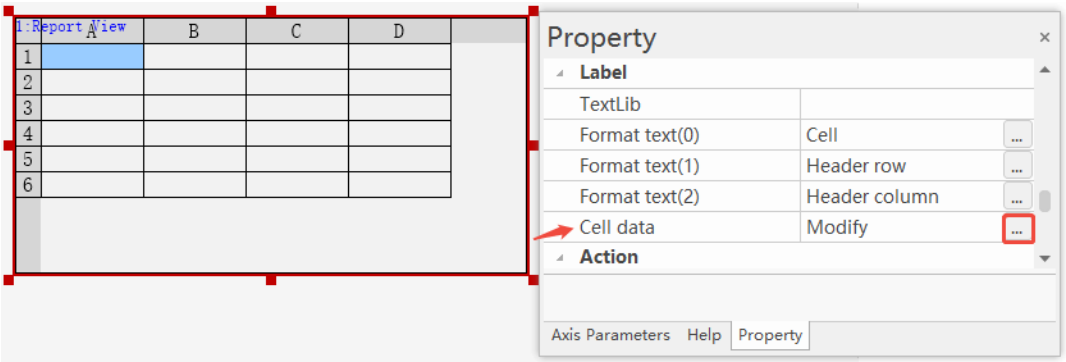
E. For Example

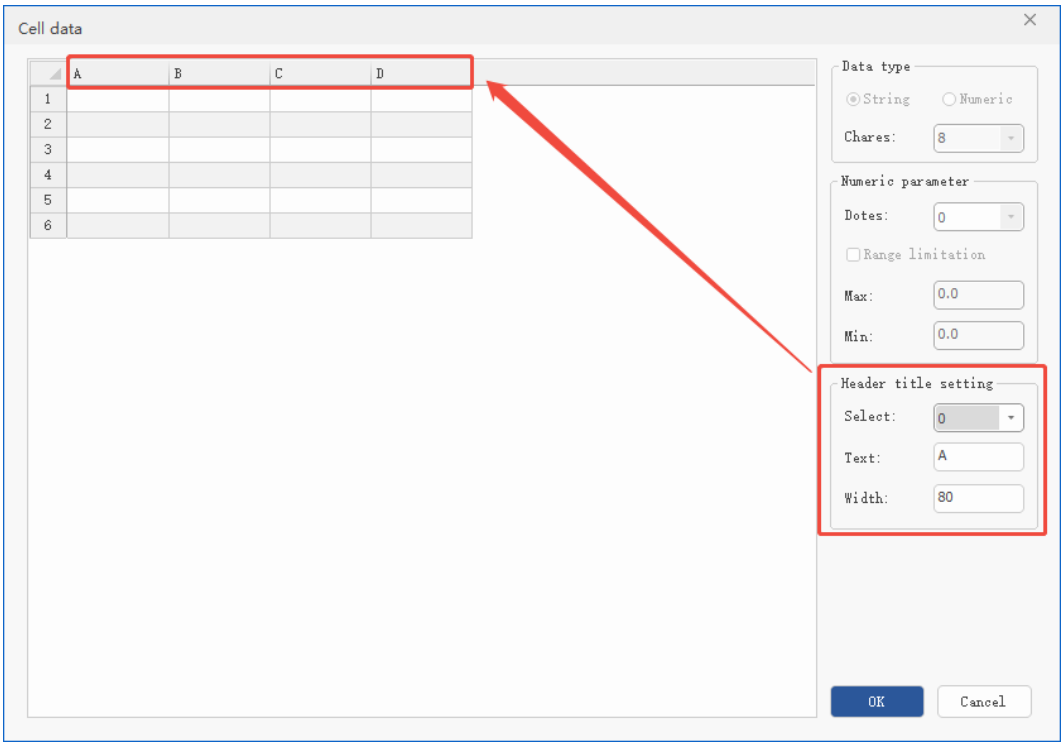
a. In report view property – appearance, set rows, columns, row height and column width (not include head row and head column).

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.

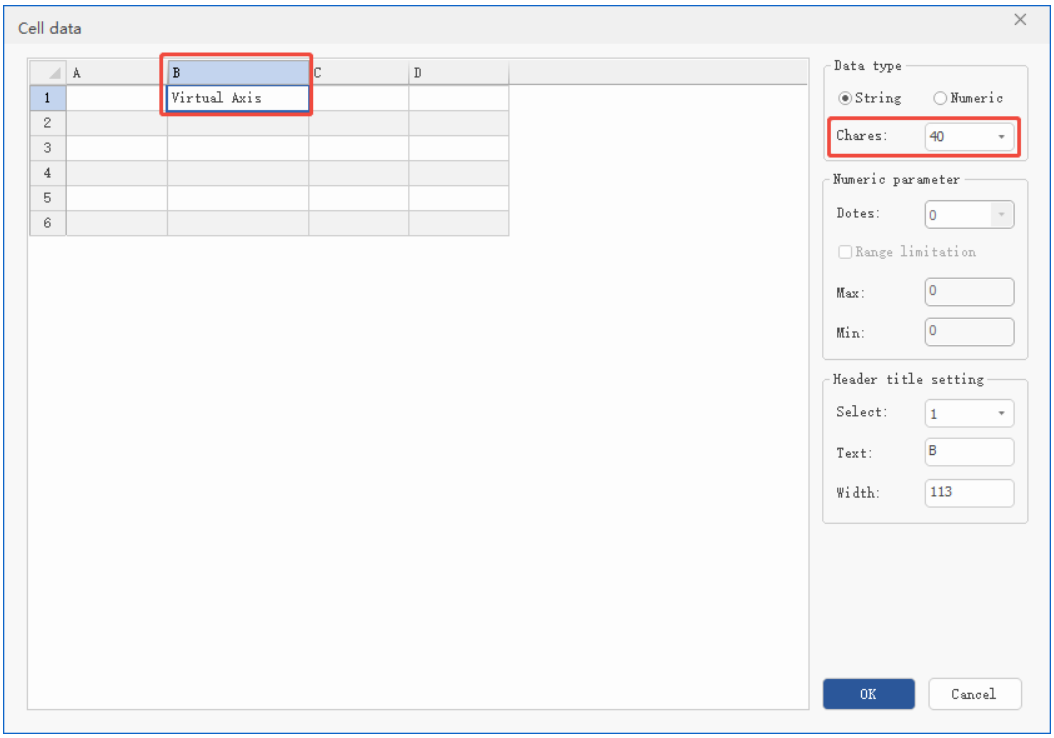


c. Click "cell data", set the header (header row) text.

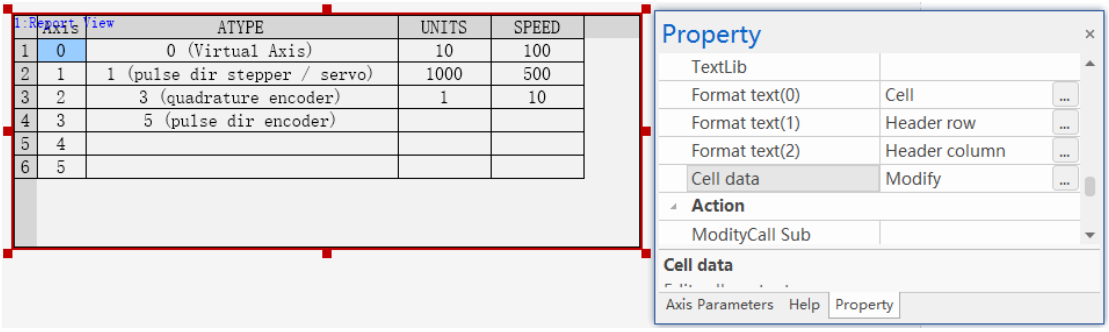




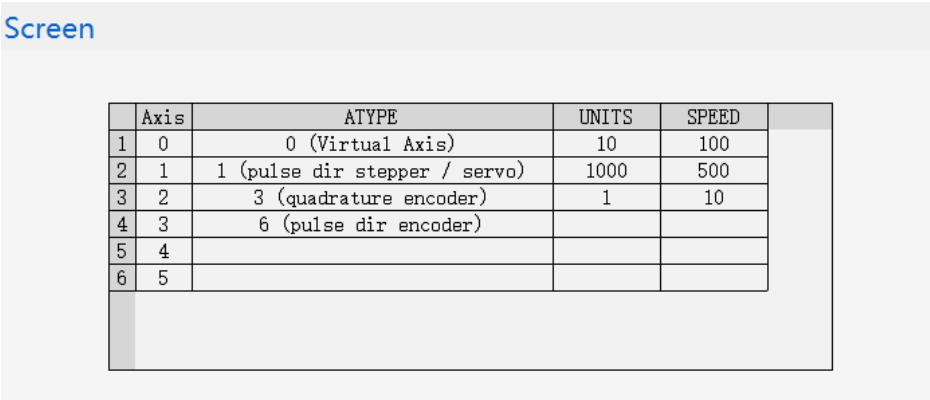
d. Edit the content. In data type, when you select “string”, you only need to set characters. When you select “numeric”, you can set the dotes, minimal value and maximum value. If dotes is set, but no dotes is edited or 0 is edited, the parameter’s dotes is only shown in downloaded simulation interface.



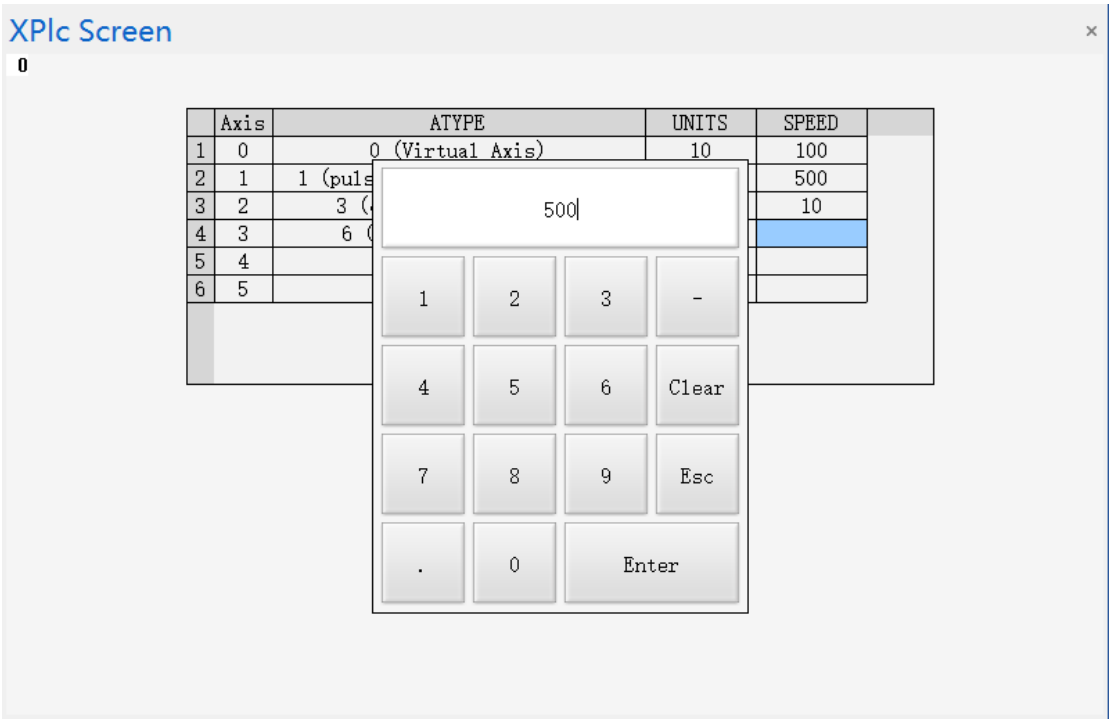
e. In HMI window, report view is shown as below.

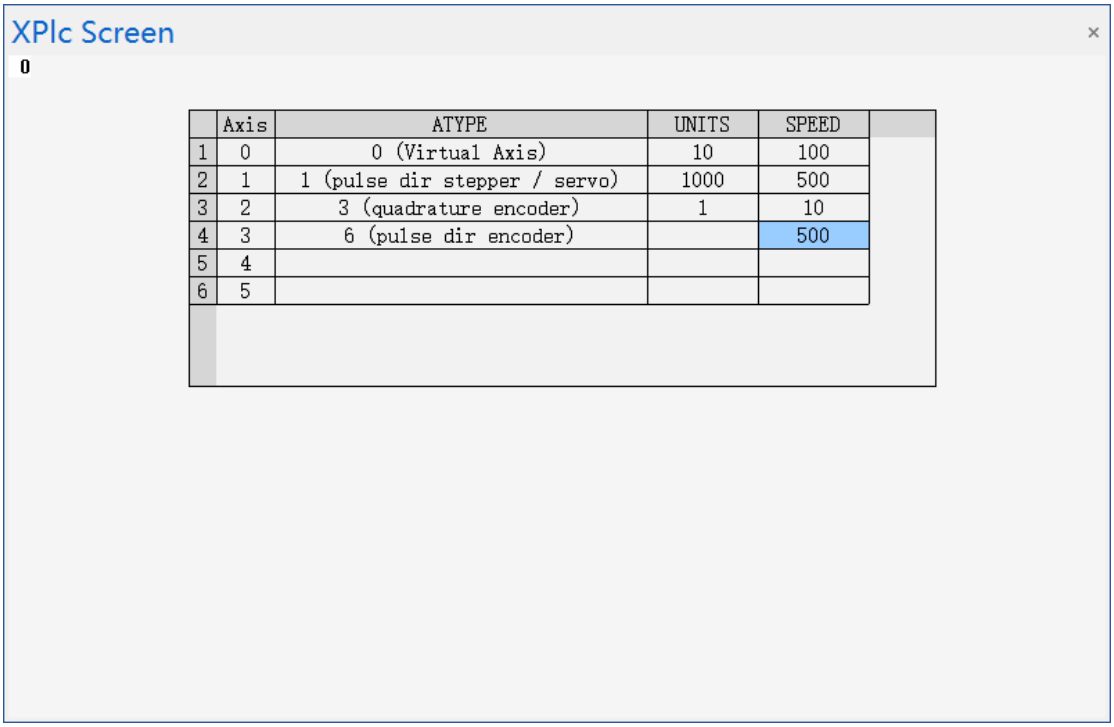


f. Connect to controller, download HMI file into controller, then, you can see the showing effect in xplc screen window. When the “fixed column width (row heigh)” is false, in downloaded simulation interface, you can put the mouse at header row to adjust the each column’s width, or put the mouse at the header column to adjust the each row’s height.



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!





4.3.27. File Browser

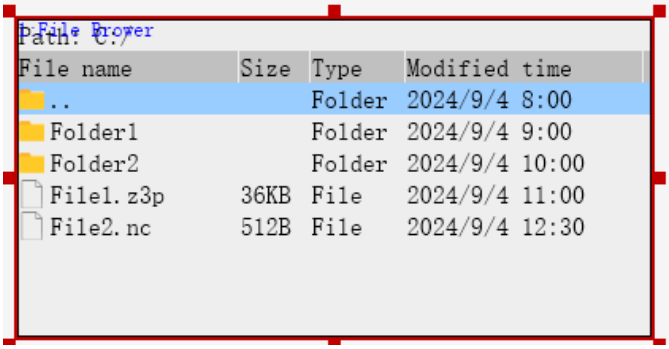
A. What is It?

[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 open it (call assigned SUB).

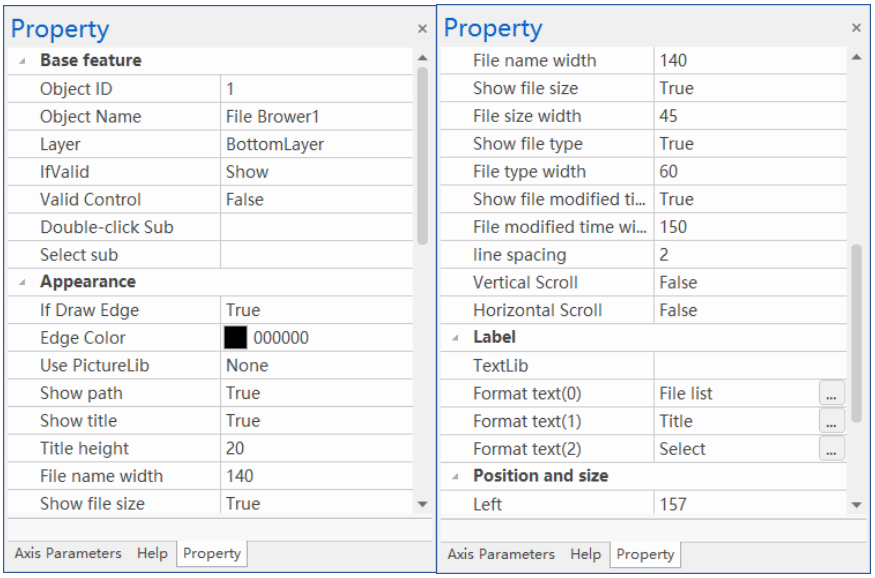
Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

B. How to Use?

Click “Control Class”, select “File Browser” from “Enhanced Controls”. Put the component at suitable position, call assigned global SUB program.



C. “Property” Window



D. "Property" Description

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.
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.
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.
Vertical scroll	Set whether to use the vertical scroll bar
Horizontal scroll	Set whether to use the horizontal scroll bar
Format text 0	Set the text format. Text name can't be modifyed.
Format text 1 (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 2 (Title)	Set the font format of the file list title row, file name can't be modified.
Format text 3 (Select)	Set the font format of the file list selected row, file name can't be modified.
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].

4.3.28. Menu

A. What is It?

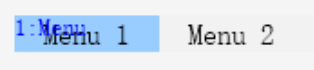
[Menu] component is similar to windows menu bar, it only can embed 5 layers of menu. Each menu item can set enable (not enabled, gray showing). When click one menu item, it can trigger corresponding SUB action, also can set whether to select the state. You can add the delimiter in any menu item.

Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

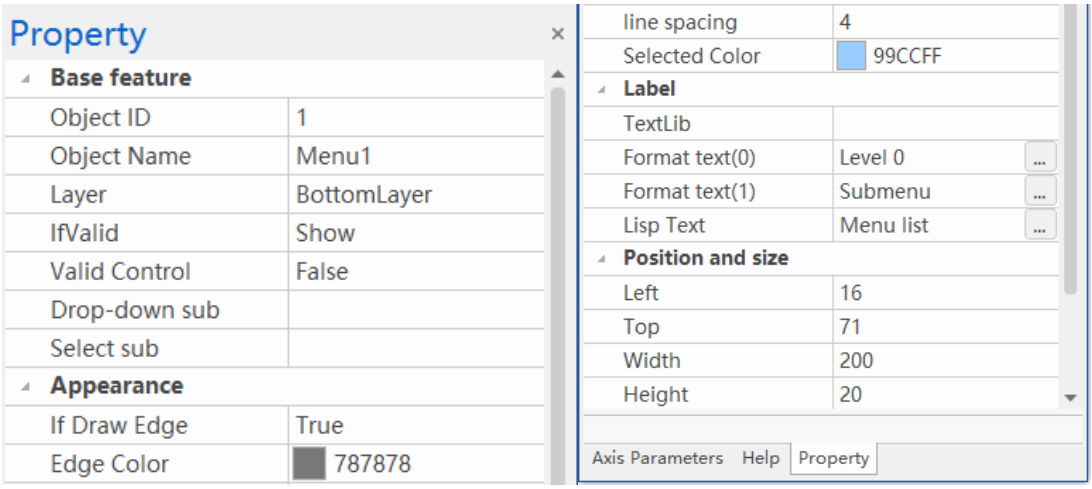
B. How to Use?

Click "Control Class", select "Menu" from "Enhanced Controls". 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., each item's operations can be achieved. At last, in menu property, call edited SUB function to complete menu function.



C. "Property" Window



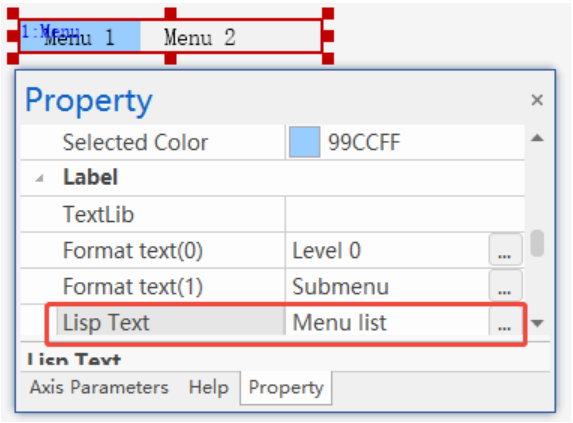
D. "Property" Description

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.

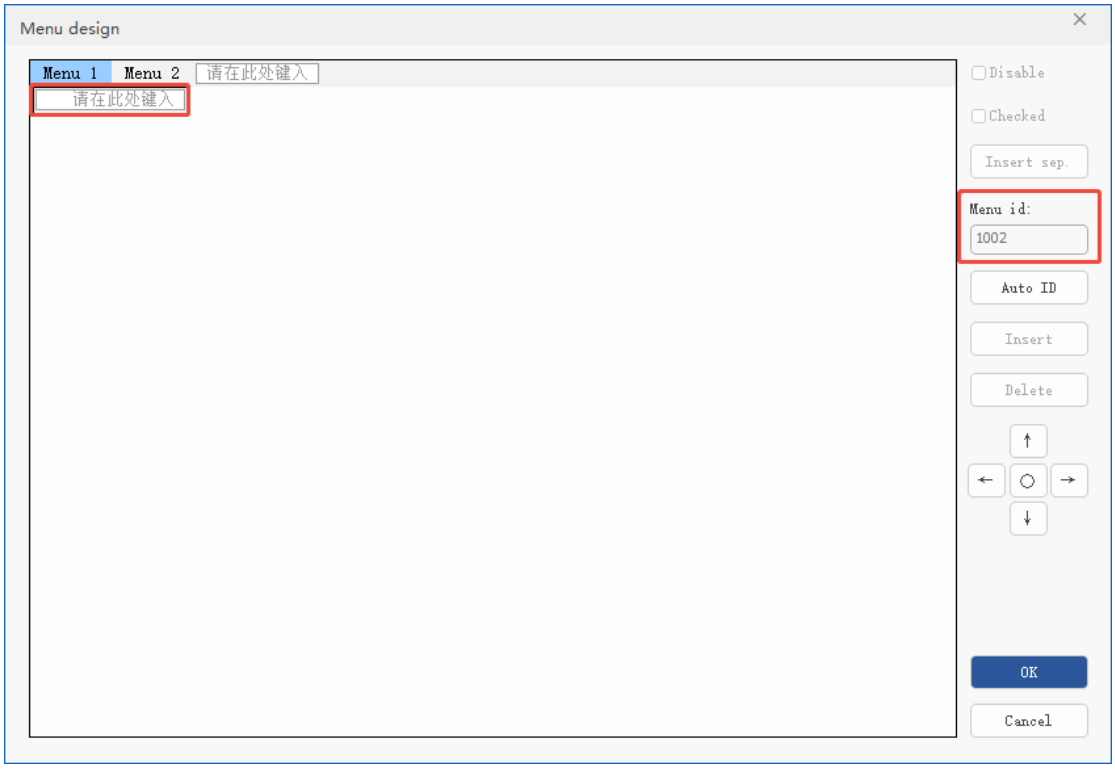
	● Valid Device: Default is local ● Valid regtype: Select from the list ● Valid regnum: Unit is ms.
Drop-down sub	Trigger and call SUB function that was defined by Basic when doing pull down operation, the SUB function must be GLOBAL type. Note: it can manage sub menu gray state, selected state in this action sub.
Select sub	When the menu item area pops up, click and select any one valid menu item, trigger corresponding SUB, the SUB function must be GLOBAL type. Note:
If Draw Edge	Whether to draw the edge.
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.
Selected Color	The main menu button and all sub menu items can be set background color, that is, when the mouse puts at one menu button or one menu item, the background color will be highlightened, the default is blue.
TextLib	The text library name, if it is empty, the text label is 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.
Left	Horizontal and vertical starting position of the component (<horizontal & vertical resolution)
Top	
Width	Width of the component
Height	Height of the component

E. For Example

a. In menu property, open “lisp text”.



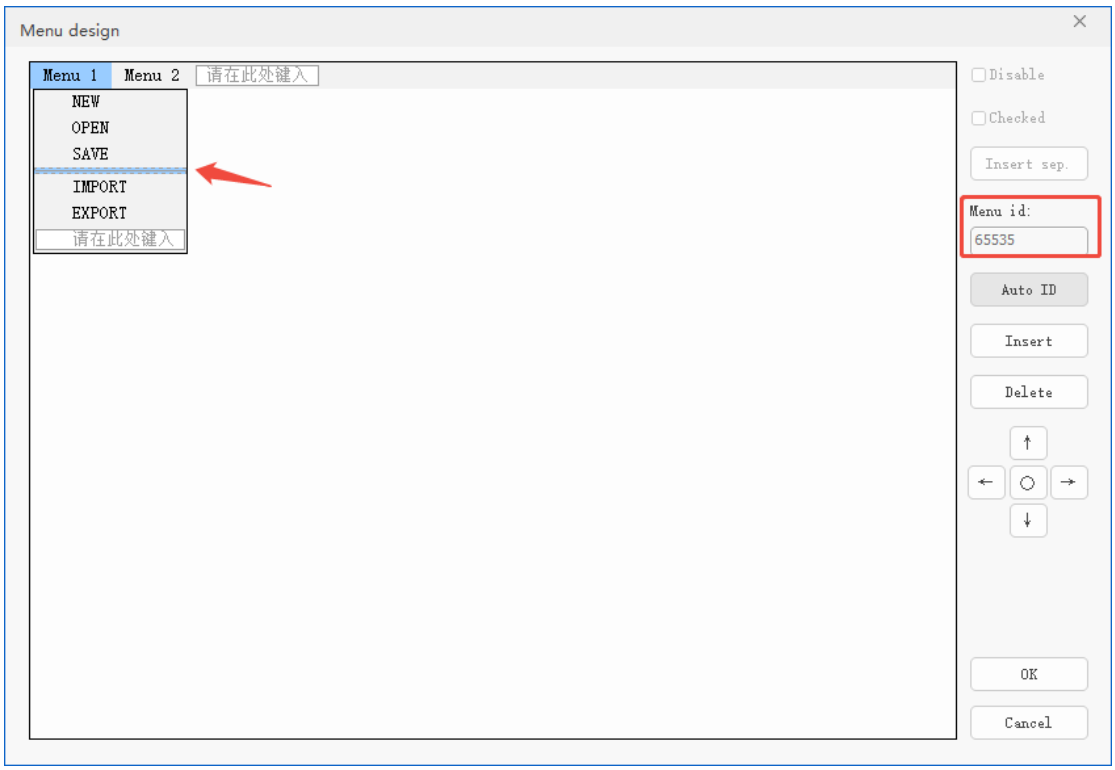
b. Edit menu item content, each item has one unique menu No., and they are numbered as order.



Note: menu No. can be set manually or automatically, but already 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.

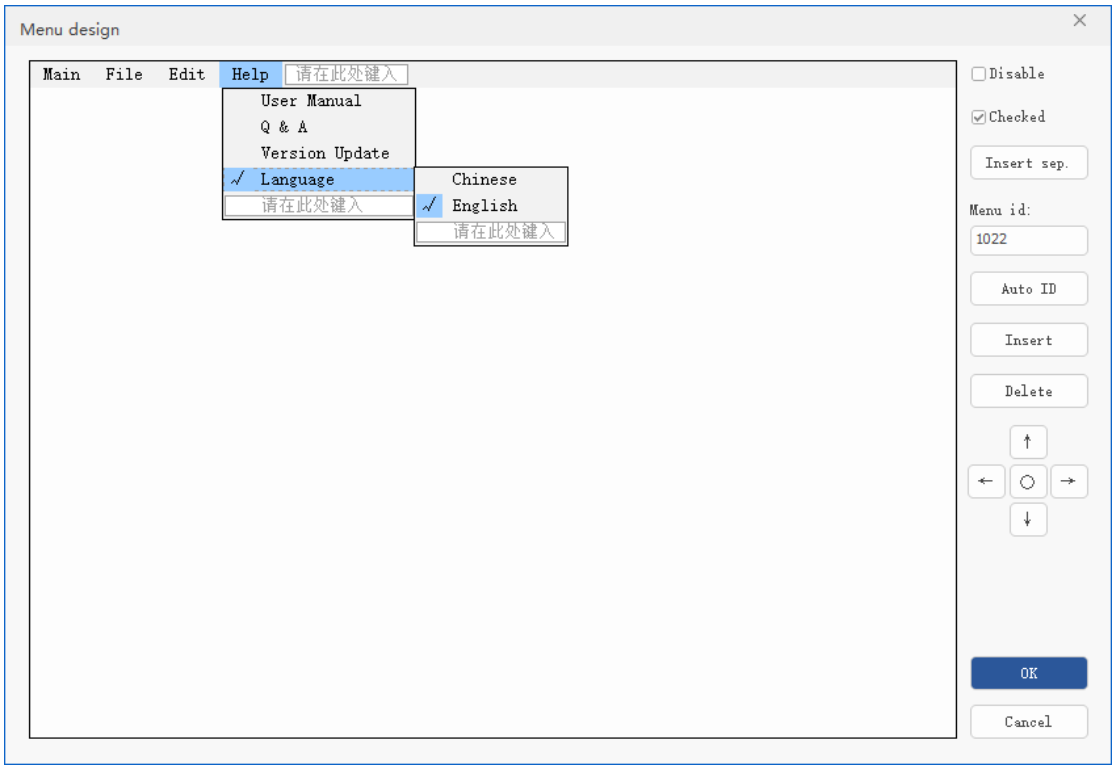
--Insert one Delimiter--

Select one item, click “insert sep”, then the delimiter will be inserted into above the selected item, the menu No. is fixed value 65535.



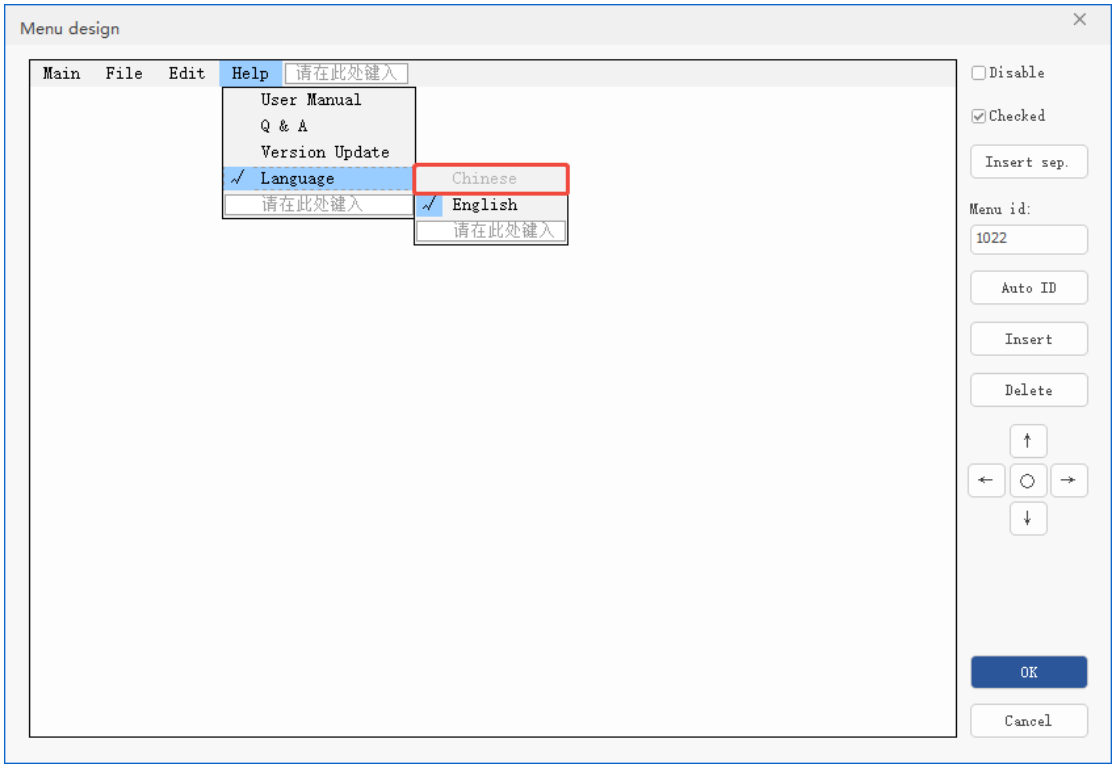
--Set Checked State--

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.



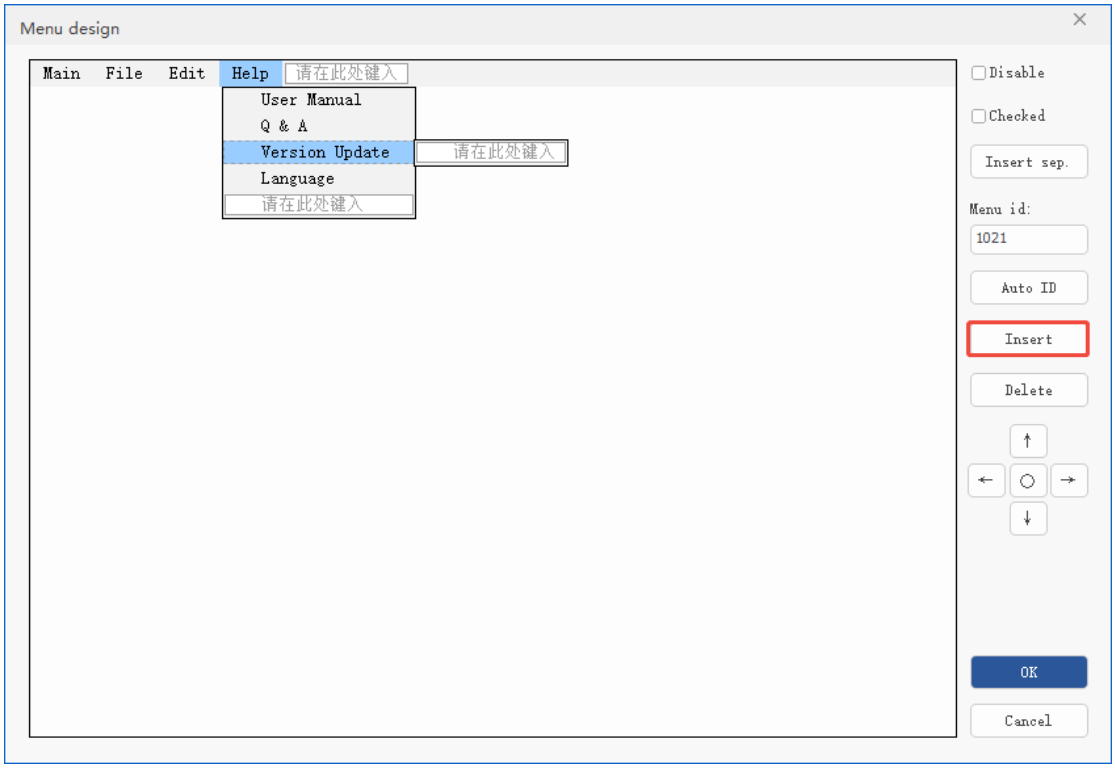
--Disable one Item--

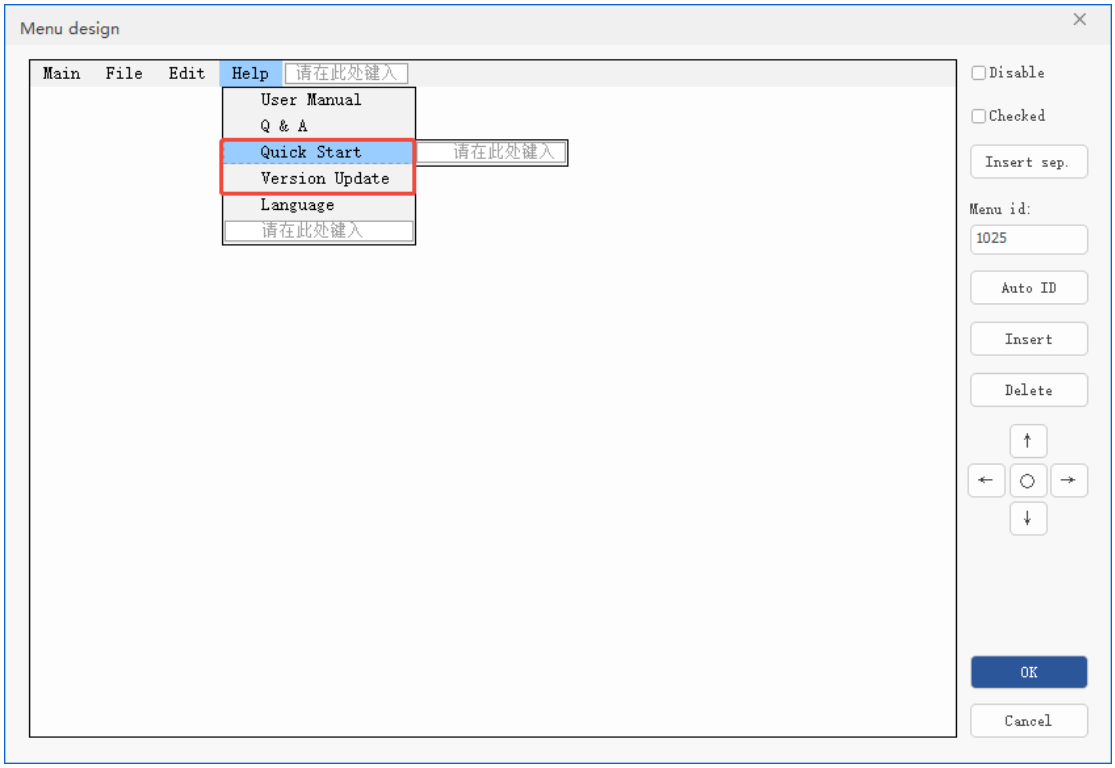
Select one item, then check “disable”, then the selected one will become gray. In this way, in real operation, no way to select it. You also can set through HMI_MENUITEM command.



--Insert one Item--

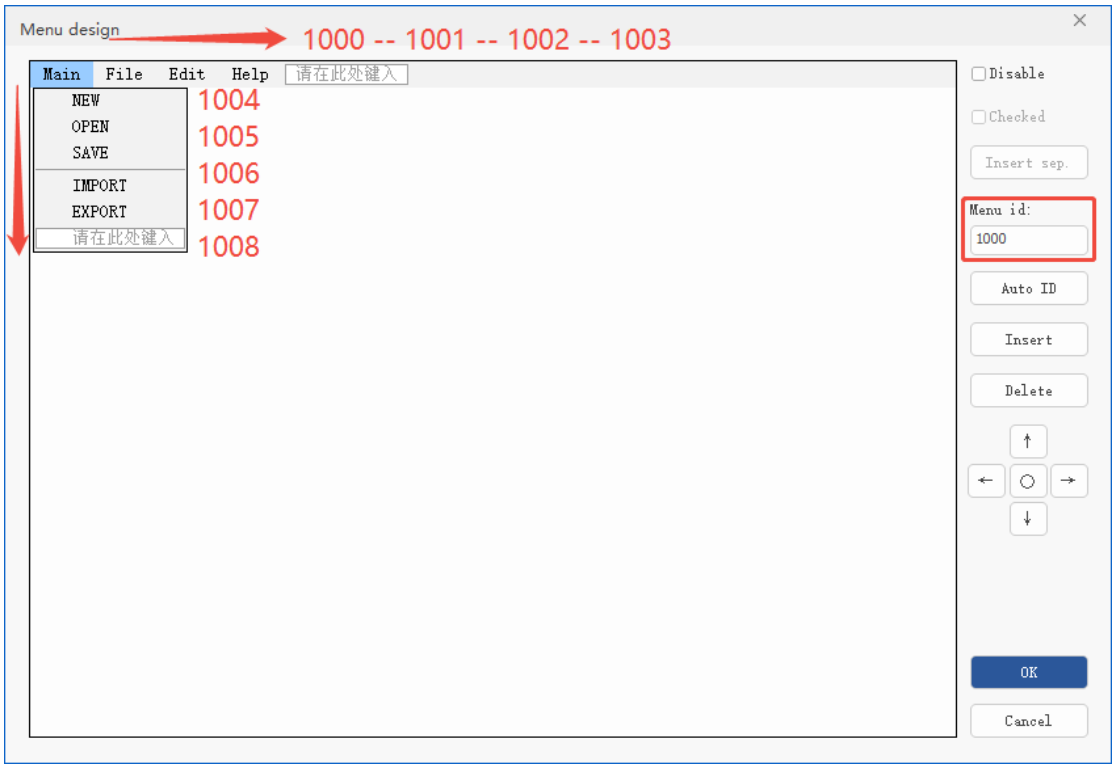
Select one item, then check “insert”, new one item will be inserted above it.



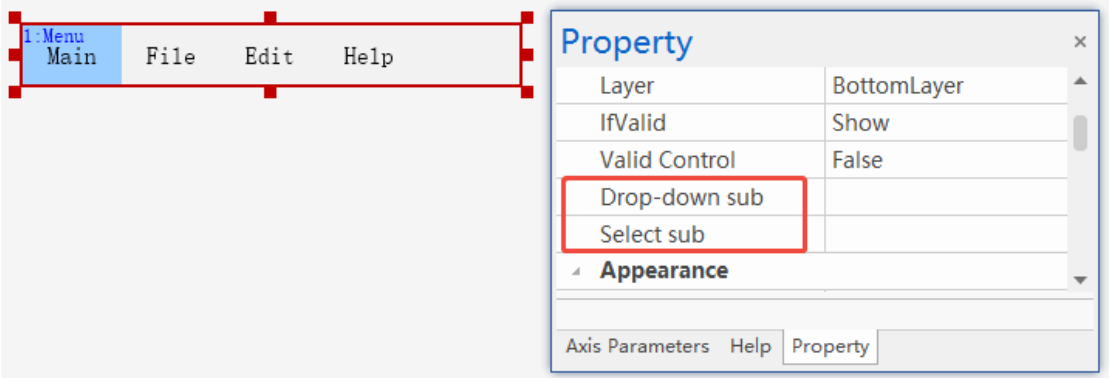


--Auto ID--

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.



c. 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.



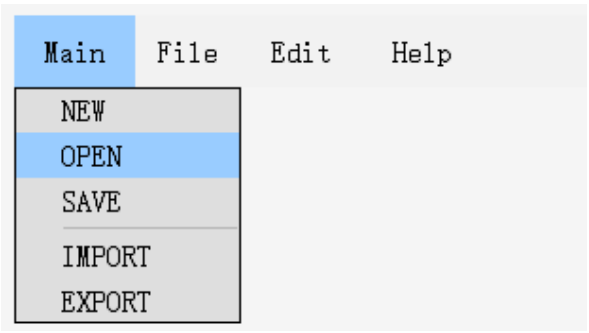
Example of Select Sub:

```
end        "main program ends


'selected sub for the menu item (some functions)
'transfer one parameter (menu No.)
```

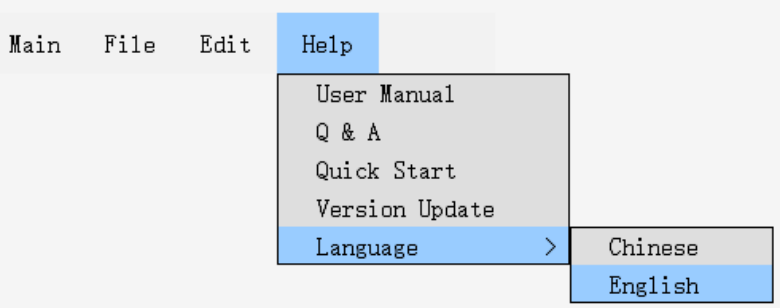
```
global sub Sub_MenuSelect(menuid)
    ?"Menu Item Clicked",menuid
    'process menu item clicking event
    if menuid = 1005 then
        HMI_SHOWWINDOW(11) "open the "main" menu
    elseif menuid = 1027 then
        HMI_SHOWWINDOW(12) "open file browser
    elseif menuid >= 1031 AND menuid <= 1034 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
```

d. Connect to controller, and download HMI file into the controller. It will be shown in xplc screen.

--open Main--

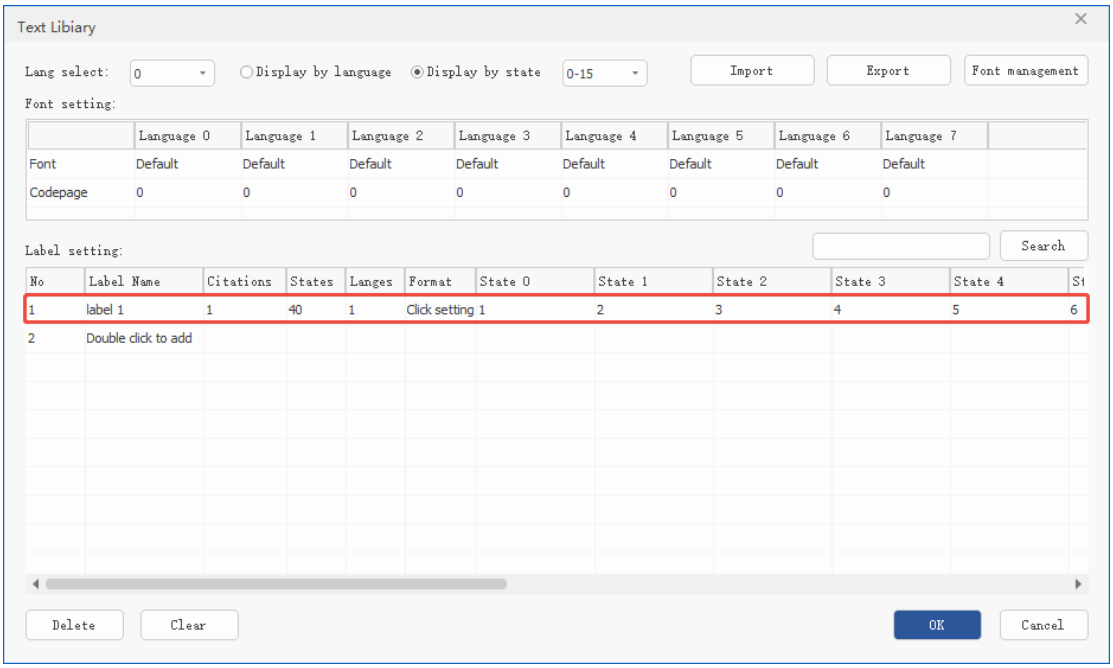


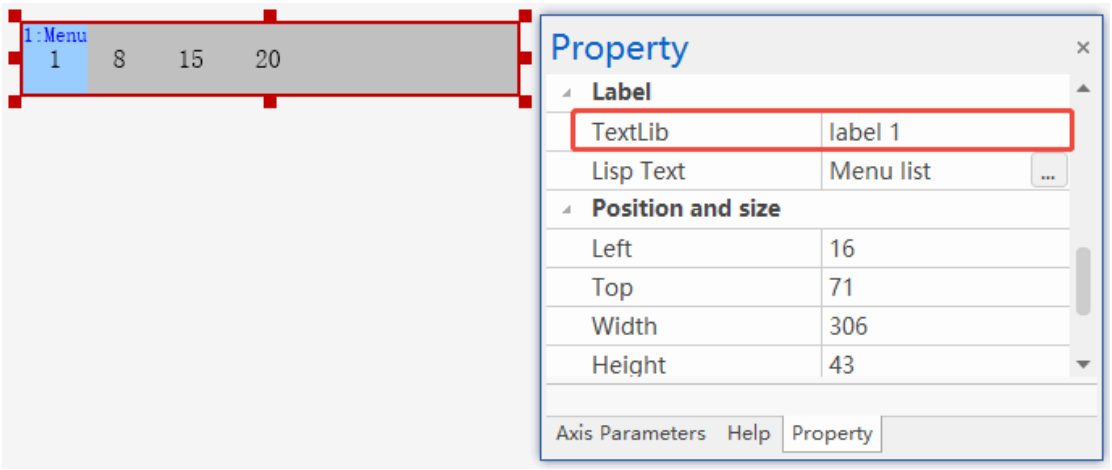
--Help – Language--

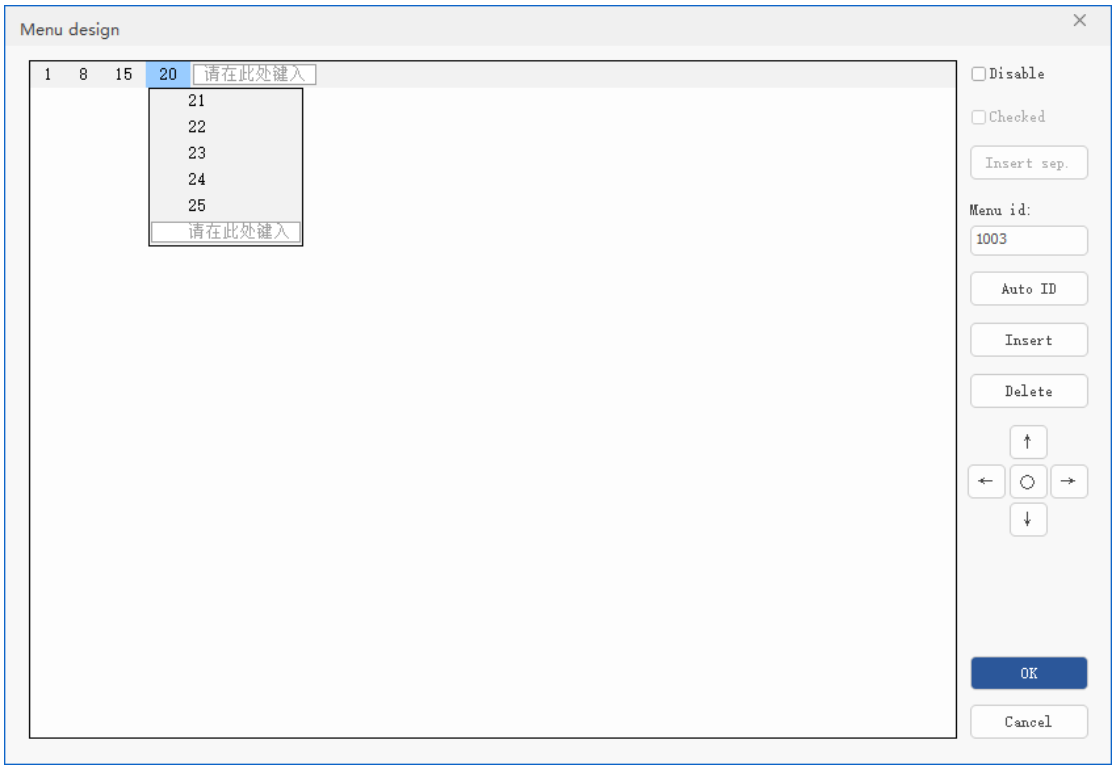


It only can select one language, disabled menu item can't be selected.

e. If it used 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).







4.3.29. Tree

A. What is It?

[Tree] component is the control that 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.

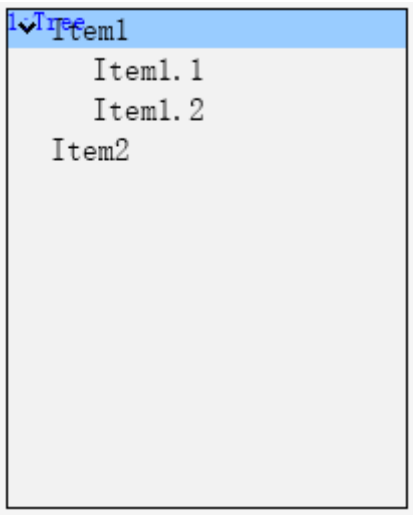
Note: when using this control, please make sure RTHMI version be V1.3.0 or above & RTSys version be V1.2.02 or above.

The current version of the tree diagram control does not support 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.

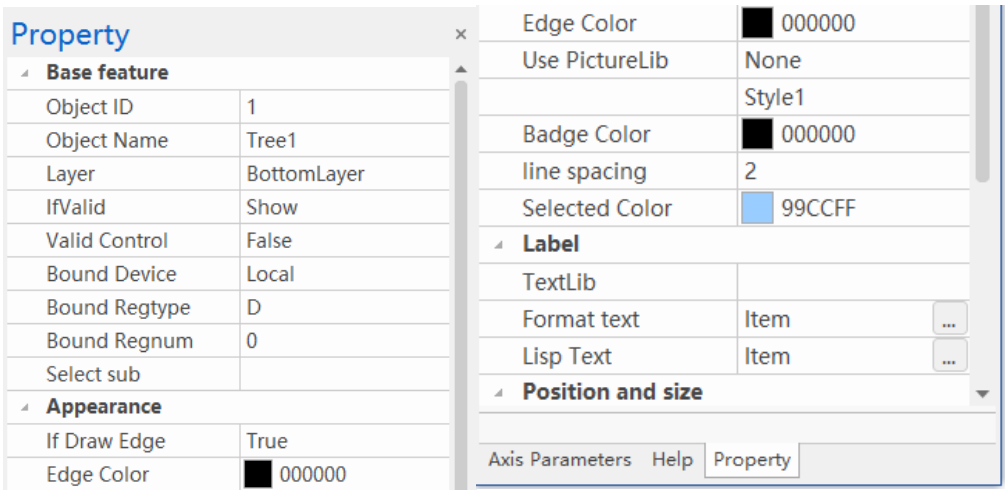
B. How to Use?

Click “Control Class”, select “Tree” from “Enhanced Controls”. 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" Window



D. "Property" Description

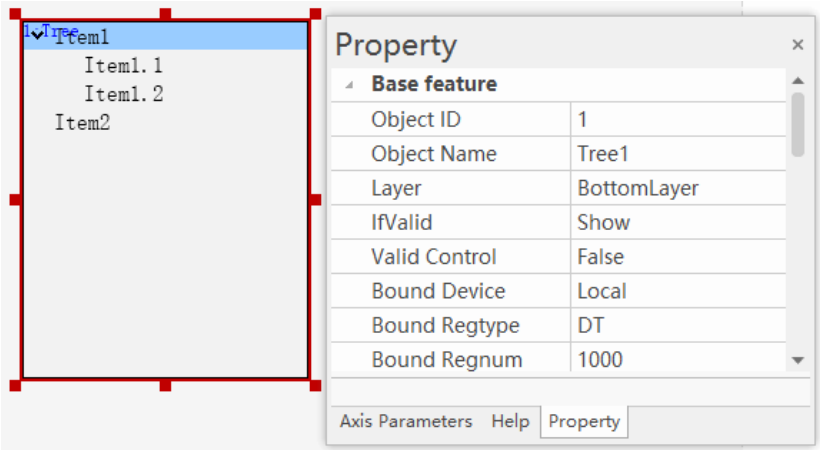
Base Feature	Description
Object ID	Start to make number for this window according to adding order.
Object Name	Name + No., you can modify by yourself.
Layer	When there are several objects, you can set the object's display layer ● TopLayer: the surface, it shows the most external layer, and covers below 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.
Bind Device	Assign the device, default is local.
Bind Regtype	Select the register type, there is one list, you can select direcly.
Bind 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.
If Draw Edge	Whether to draw the edge. If TRUE, draw it.
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
Level Mark Style	The display type of the parent tree corner mark, supporting styles 1 to 3.
Badge Color	The color of the corner mark.
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 is 2, 0-100 can be set.
Selected color	The main menu button and all sub menu items can be set background color, that is, when the mouse puts at one menu button or one menu item, the background color will be highlightened, the default is blue.
TextLib	Text library name, if no set, it shows "Text".
Lisp text (Item)	Display text and ID of all tree (parent and child) nodes. The currently selected item will be associated with the ID number of the tree node.
Left	Object horizontal starting position, don't exceed X resolution.
Top	Object vertical starting position, don't exceed Y resolution.

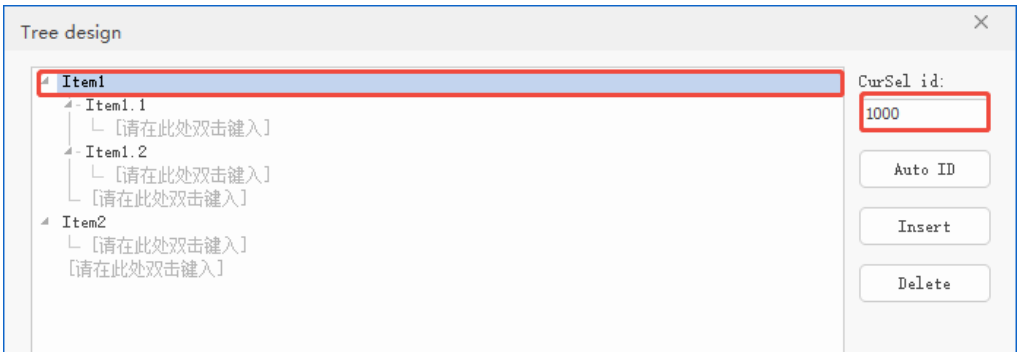
Width	Object width
Height	Object height

E. For Example

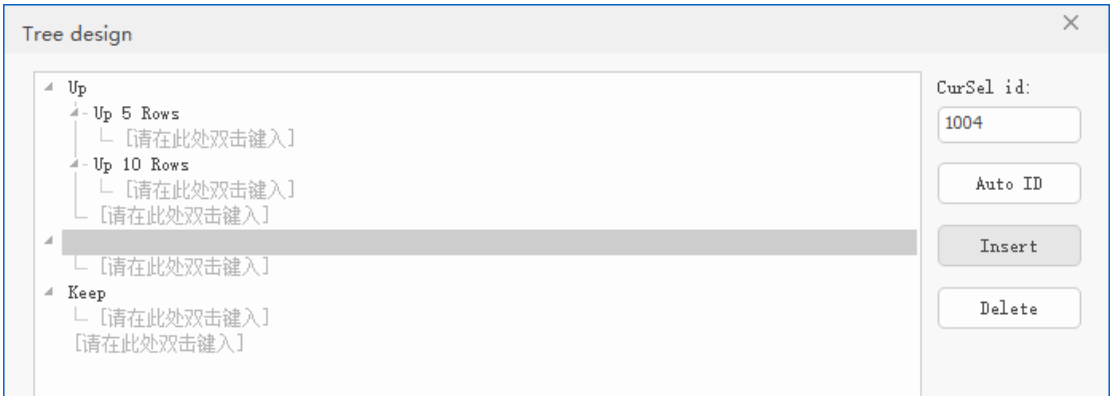
a. In Tree basic property, select register type and No. (here select DT register and set the No. as 1000, that is, bound register address is **TABLE(1000)**).



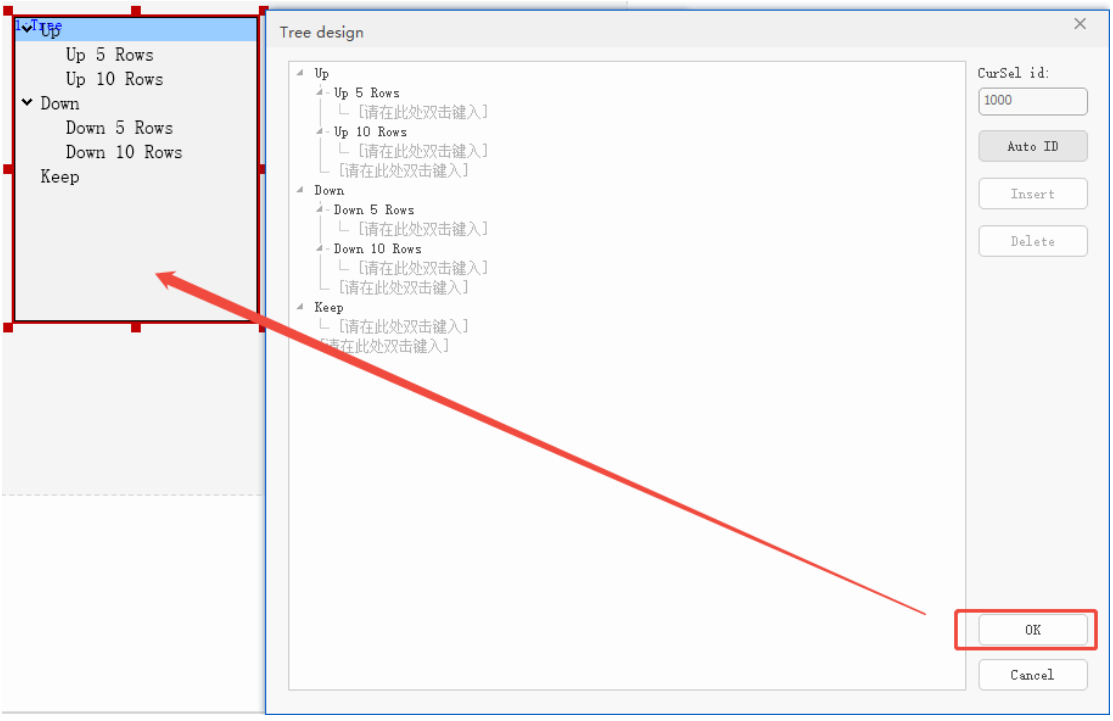
b. In tree property – lisp text, set the tree branch structure and edit the content. Each tree node binds with one unique ID, ID of tree nodes are numbered according to editing sequence. After edited, if you click “Auto ID”, then they will be numbered from the up to bottom.



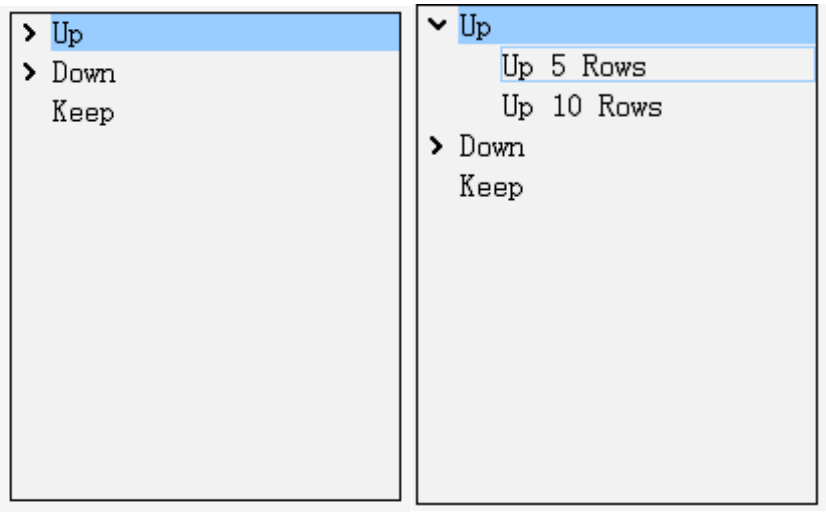
--insert: add one same level item above selected item--



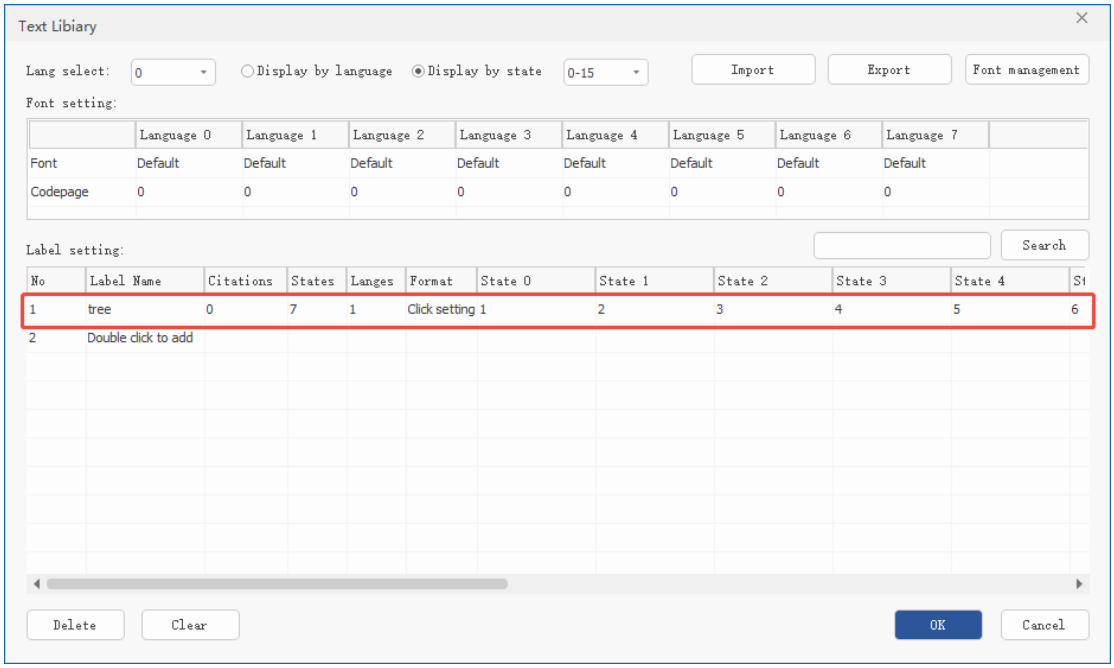
c. After edited, click OK.

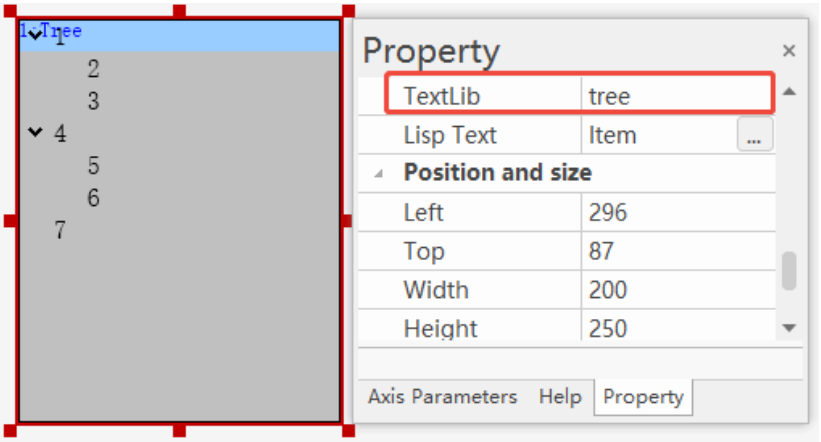


d. Connect to controller, download HMI into controller, open xplc screen. It only shows the first layer information, you can click the arrow > to show sub items.



e. If it used text library for the menu control, then contents will be filled into each menu item according to the layer structure.





f. For state showing color, line spacing, pop up method, please set in property.

g. For specific usage of SUB calling by tree, please refer to Chapter VIII [File Browser Usage].

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.

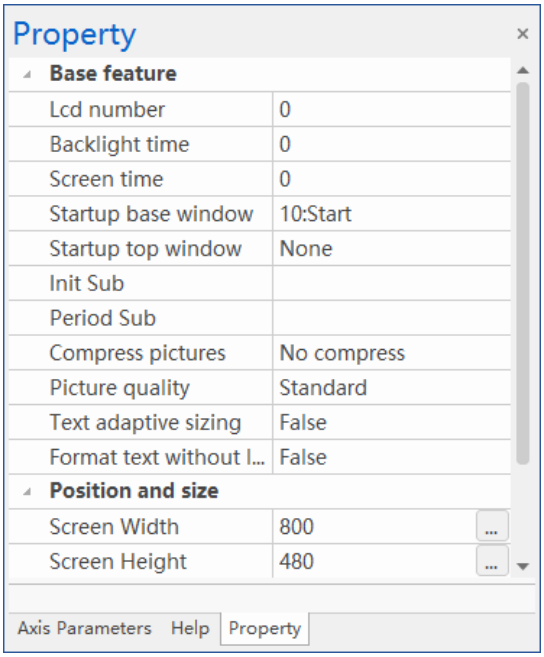
(2) In defined SUB function, edit this function’s program.

(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.

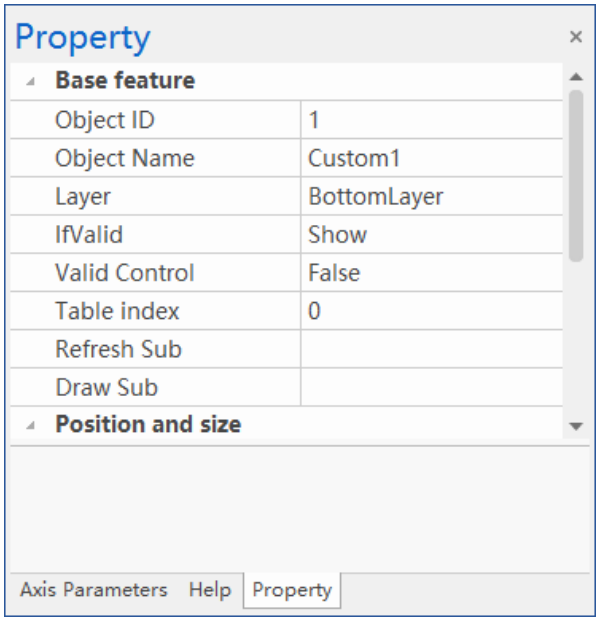


- **Init Sub:** when powered on, it is the function that is only called once. Generally write initial related parameter definitions into defined initialization global SUB function, and it will be called in HMI setting.
- **Period Sub:** when powered on, it is the function that is called cyclically.
- Note: init sub and period sub both select SUB function defined by GLOBAL in Basic.

5.2. Call Basic Function SUB in Custom Component

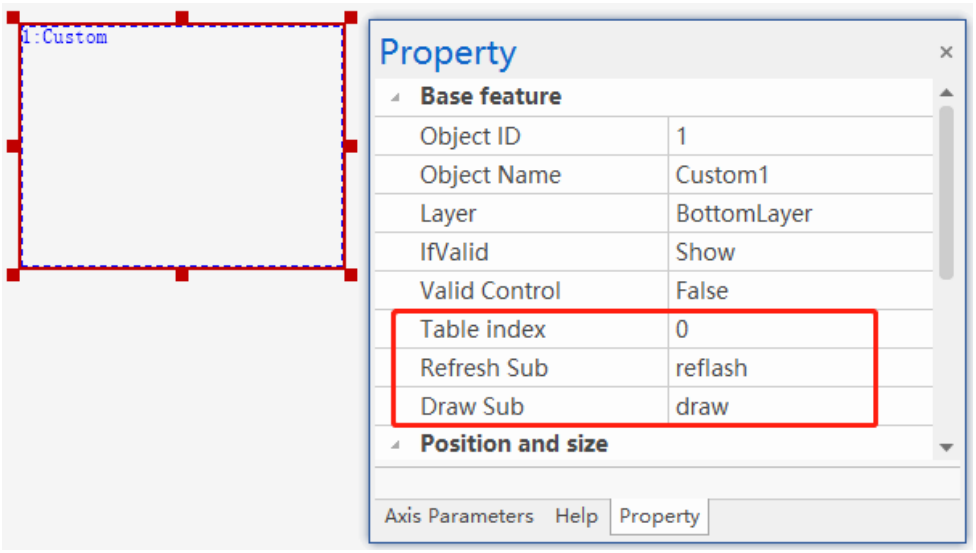
For "Custom" component, it can add draw sub and refresh sub, and these two sub functions are global functions defined in Basic.

- **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.



- **Examples:**

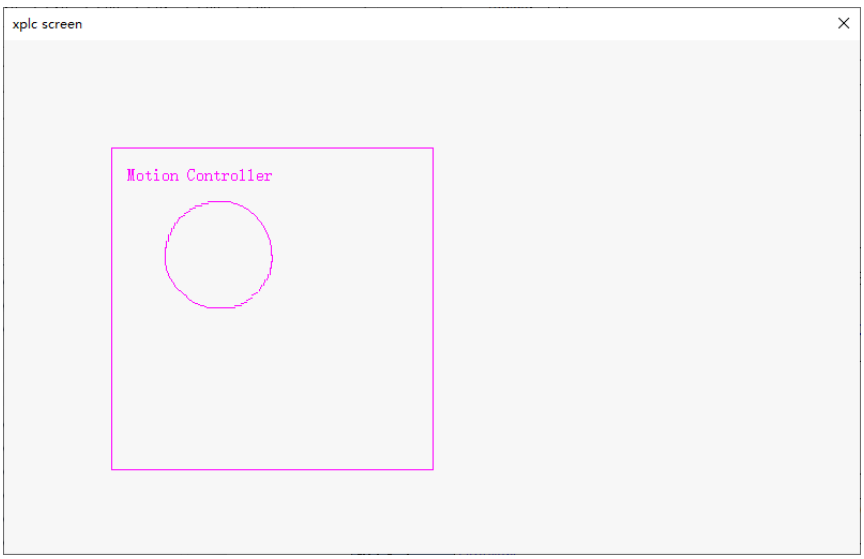
Example 1: in HMI window, create one Custom component, open the property window, set Table index, draw sub, refresh sub.



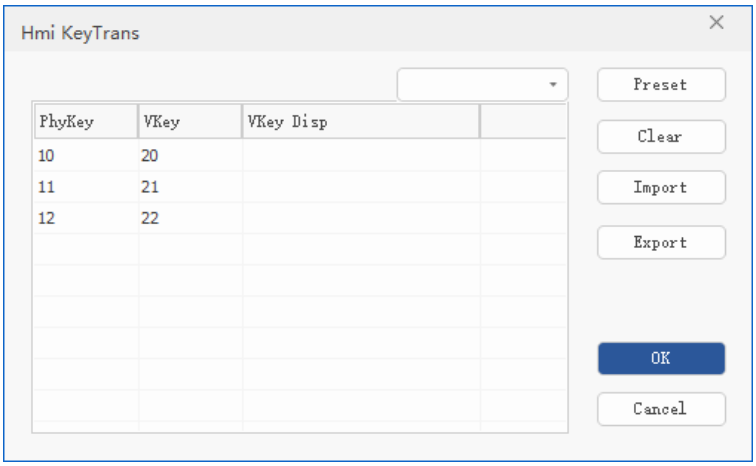
--Basic Program--

```
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.



--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.

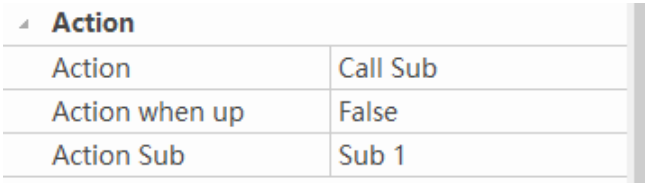
When physical key 11 pressed, assign TABLE(10) as 100.

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.



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

Type	Input and output functions
Description	**Scan data changes.** Return Value: 1 — Data from 0 to non-0 Return Value: -1 — Data from non-0 to0 Return Value: 0 — BOOL no changes of data

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_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 ZHMI.
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

6.3.2. 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.

6.3.3. LCD_WDOGTIME – Displayer Offline Process Time

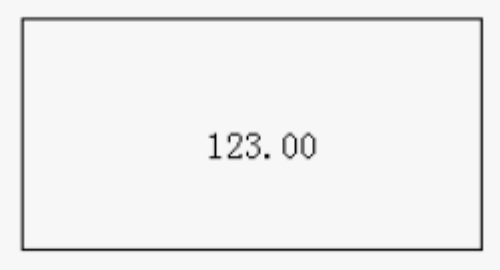
Type	System Screen Parameters
Description	Set the screen offline processing time, the unit is ms. When the screen is operated without communication during this time, the emergency stop switch (physical key number = 5) is automatically pressed, and when it is equal to 0, this function is not enabled. 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 ZHMI.
Example	LCD_WDOGTIME(0) = 100

6.3.4. 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 ZHMI.
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 
Instructions	DRAWNUM2

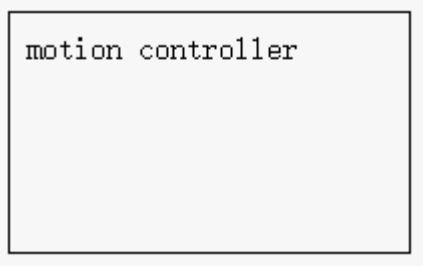
6.3.5. DRAWNUM2 – Show Value in Specified Position

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 
Instructions	DRAWNUM

6.3.6. 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.

	
Instructions	DRAWTEXT2

6.3.7. 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 
Instructions	DRAWTEXT

6.3.8. 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

6.3.9. 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

6.3.10. DRAWREVERSE – Draw Square

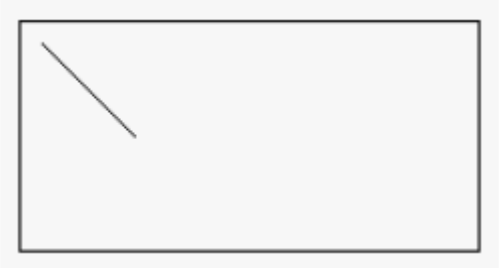
Type	Display instructions.

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 

6.3.11. 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

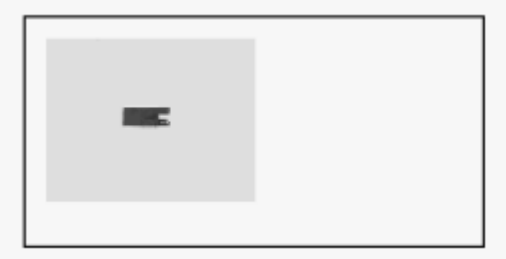
6.3.12. DRAWLINE – Draw Segment

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

6.3.13. DRAWCLEAR – Clear Content in Specified Area

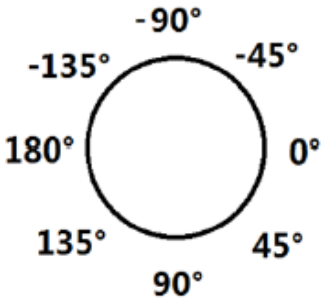
Type	Display instructions.
Description	Clear assigned reading area. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “Custom”.
Grammar	DRAWCLEAR ([x1,y1,x2,y2]) x1, y1: clear the coordinate position of upper left corner x2, y2: clear the coordinate position of bottom right corner all is cleared when there is no parameter.
Controller	Controllers that support ZHMI.
Example	DRAWCLEAR (10,10,50,50)

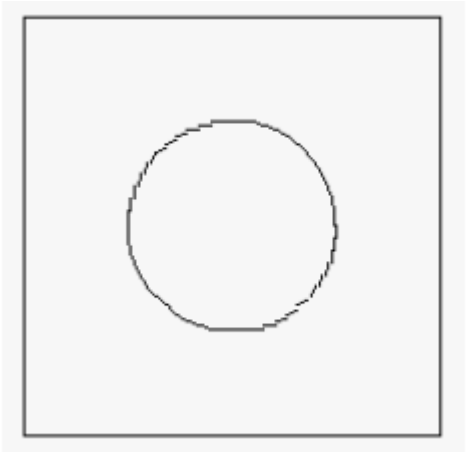
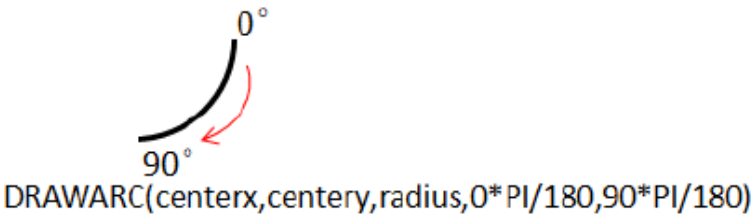
6.3.14. DRAWPIC – Insert PNG File

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

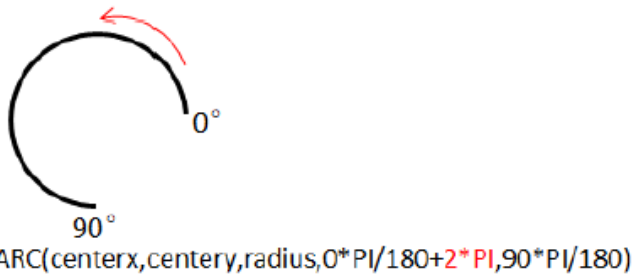
6.3.15. DRAWARC – Draw Arc

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

<table>
<tr><td></td><td>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</td></tr>
<tr><td>Controller</td><td>Controllers that support ZHMI.</td></tr>
<tr><td>Example</td><td>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
 

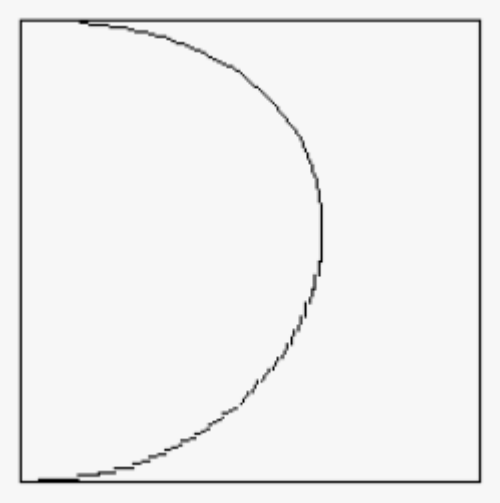
Example 3: When it is necessary to draw a superior arc segment, add 2*pi to the smaller angle between the starting angle and the ending angle.</td></tr>
</table>

	
Instruction	DRAWLINE, DRAWEX_ARC

6.3.16. DRAWLIBPIC – Insert PNG Picture

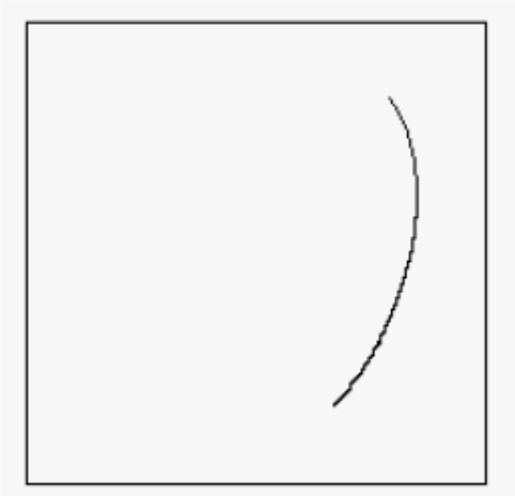
Type	Display instructions.
Description	To draw a picture, the picture file must be added to the project first, and the picture should be added in the file view. Note that the picture takes up more space, and do not add unnecessary pictures to the project. This command only can be used in “Draw Sub” of “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

6.3.17. 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

6.3.18. 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 

6.3.19. 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

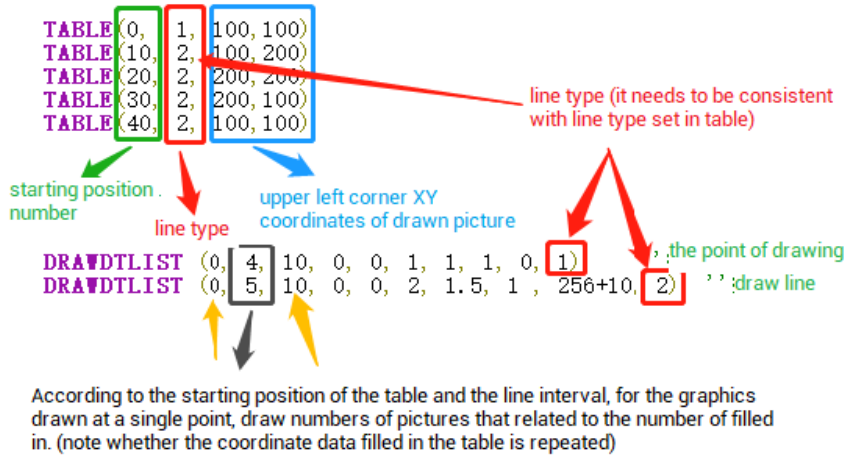
imode: the storage format

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

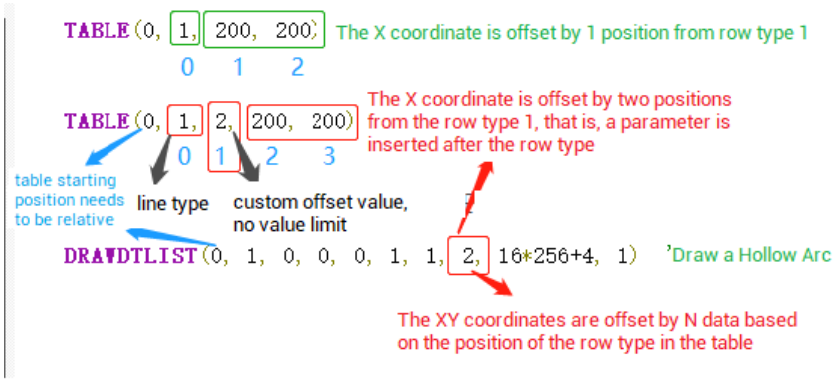
drawtype: display type

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>Controller</td><td>Controllers that support ZHMI (It is recommended to use the latest firmware).</td></tr>
<tr><td>Example</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>



How to use imode (1-9) storage format for parameters:

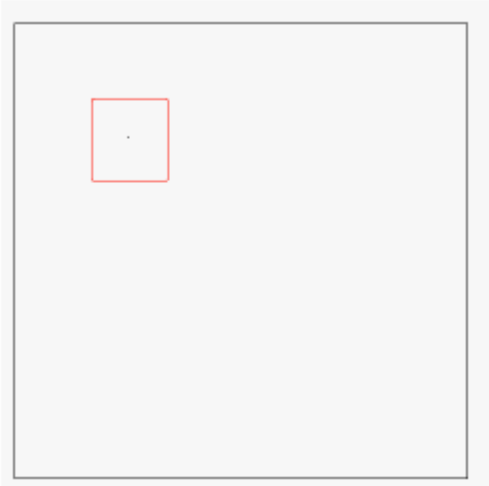


Example 1: draw one point

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)



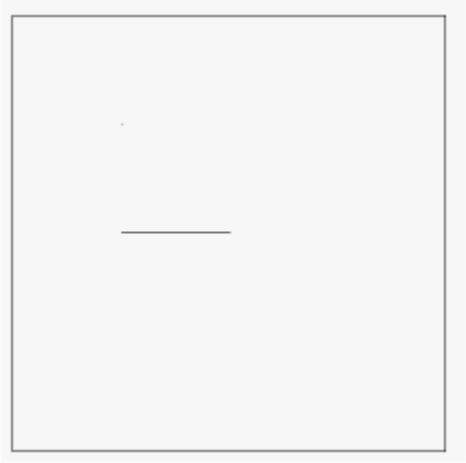
Example 2: draw a segment of straight line

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)

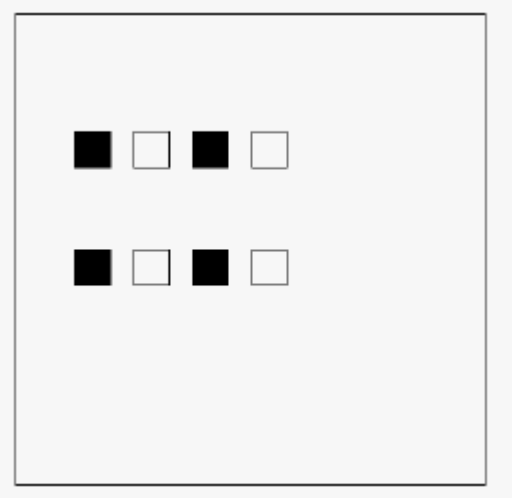
TABLE(10,2,100,200)

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)

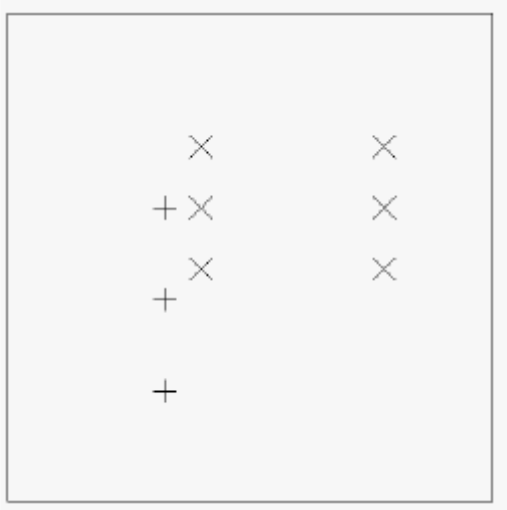
TABLE(15,1,200,200)

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:

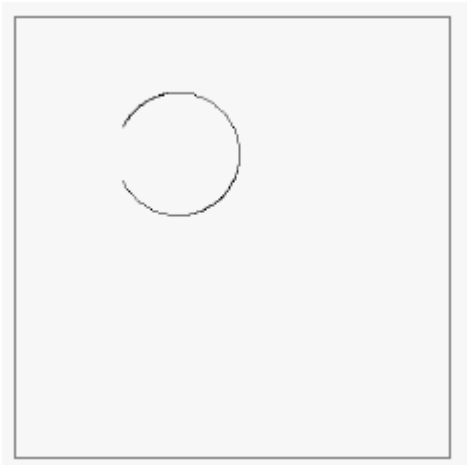
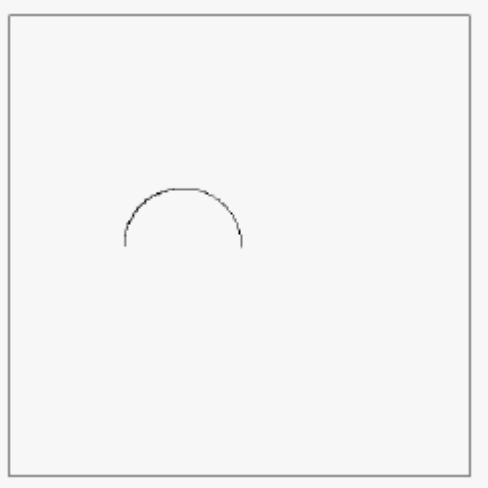
TABLE(0,1,100,100) 'the starting point of the arc

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

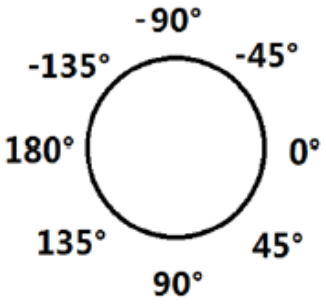
6.3.20. DRAWEX_LINE – Draw Segment (with format)

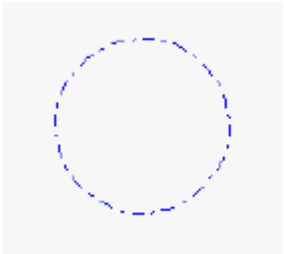
Type	Drawing instructions.
Description	Draw a straight line. The difference with DRAWLINE is that DRAWEX_LINE can set the line width and line type, but DRAWLINE does not support it. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom 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

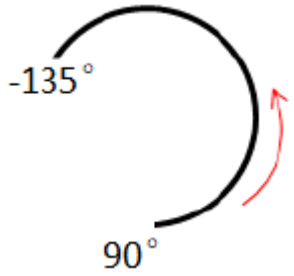
6.3.21. DRAWEX_ARC – Draw Arc (with format)

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:

<table>
<tr><td></td><td> 

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</td></tr>
<tr><td>Controller</td><td>Controllers that support ZHMI. (4XX series controllers and above)</td></tr>
<tr><td>Example</td><td>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</td></tr>
</table>

	
Instruction	DRAWARC, DRAWEX_LINE, SETEX_LINE, SET_COLOR

6.3.22. 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

6.3.23. DRAWEX_BSPLINE – Draw B Type Spline Curve (with format)

Type	Drawing instructions
Description	Draw a B-spline curve. The difference with DRAWBSPLINE is that

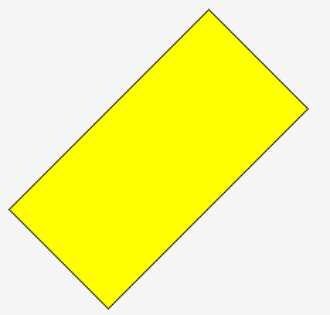
	DRAWEX_BSPLINE can set the line width and line type, but DRAWBSPLINE does not support it. The DRAWEX drawing command will also limit the drawing area. In principle, it is not allowed to draw outside the custom control, but the DRAW related commands are unlimited. This command only can be used in "Draw 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

6.3.24. DRAWEX_RECT – Draw Rounded Corners Rectangle (with format)

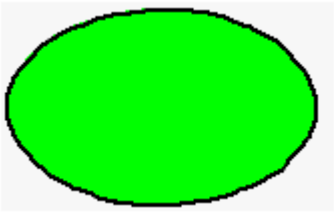
Type	Drawing instructions
Description	Draw a rounded rectangle in the specified area. The difference with DRAWRECT is that DRAWEX_RECT supports setting line width, line type and filling drawing, while DRAWRECT does not. The DRAWEX drawing command will also limit the drawing area. In principle, it is 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

6.3.25. DRAWEX_ROTRECT – Draw Rotary Rectangle (with format)

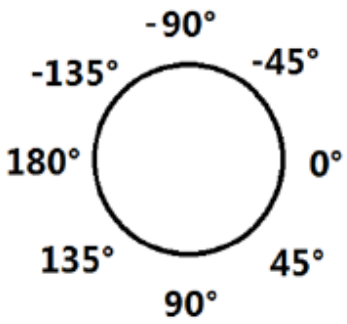
Type	Drawing instructions
Description	Draw a rotary rectangle in the specified area. This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in “Draw Sub” of “Custom” object, please refer to “Draw Sub” of “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

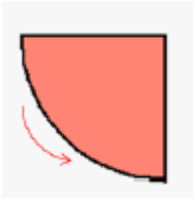
6.3.26. DRAWEX_ELLIPSE – Draw Ellipse Rectangle (with format)

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

6.3.27. DRAWEX_SECTOR – Draw Sector (with format)

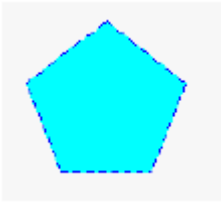
Type	Drawing instructions

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

6.3.28. DRAWEX_POLYGON – Draw Polygon (with format)

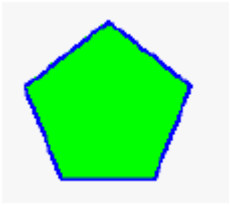
Type	Drawing instructions
Description	Draw a polygon in the specified area. This command supports setting line width, line type and fill drawing, but does not allow drawing outside custom controls. This command only can be used in "Draw Sub" of "Custom" object, please refer to "Draw Sub" of "Custom".
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

6.3.29. DRAWEX_POLYGON2 – Draw Polygon (save in TABLE)

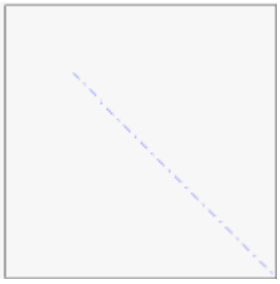
Type	Drawing instructions
Description	Draw a polygon in the specified area.

	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

6.3.30. 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)

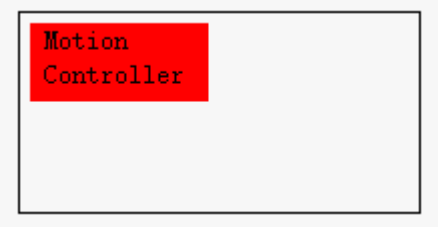
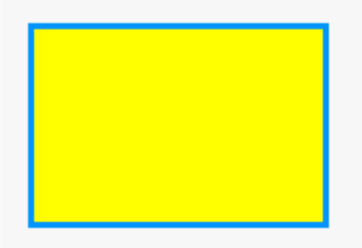
	
Instructions	SET_COLOR

6.3.31. 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

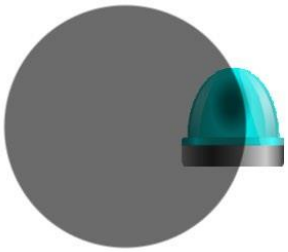
6.3.32. SET_COLOR – Set Color

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

6.3.33. 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) 

6.3.34. 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

6.3.35. 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

6.3.36. 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	/

6.3.37. 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	/

6.3.38. 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

6.3.39. 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

6.3.40. 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

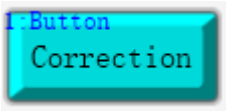
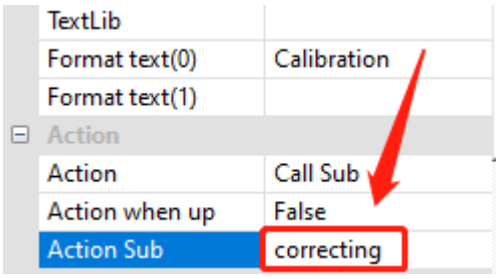
6.3.41. SCROLLBAR_DRAW – Draw Scroll Bar

Type	Refresh instructions
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

6.4. 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.  (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 ZHMI
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

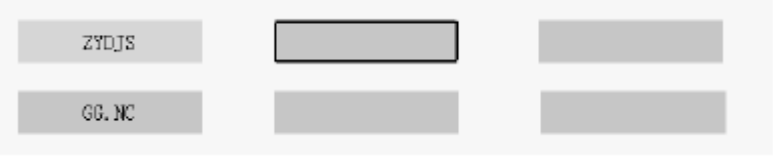
Type	Virtual key button instructions
Description	Read the state change of the physical button, 1-pressed, -1-released, 0-unchanged. This command can only be used in the refresh sub of the "custom" object.
Grammar	KEY_EVENT (keynum) keynum: key number
Controller	Controllers that support ZHMI
Example	num =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

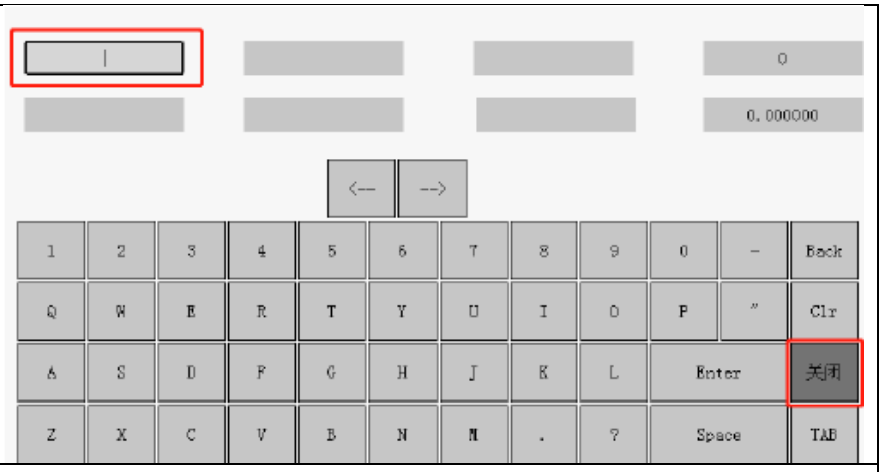
6.5.4. KEY_SCAN – Get Physical Key Encodes

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 ZHMI
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 ZHMI
Example	VKEY_MODE(1) ‘open virtual key input mode VKEY_MODE(1,10) ‘specify window 10 to open virtual key input mode

	
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 ZHMI
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 ZHMI

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 ZHMI
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 ZHMI
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 RHHMI
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 ZHMI
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 ZHMI
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 ZHMI
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	HMI_CLOSEWINDOW

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 ZHMI
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 ZHMI
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) 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 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) 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 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.

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 ZHMI
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 ZHMI
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 ZHMI
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 ZHMI
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

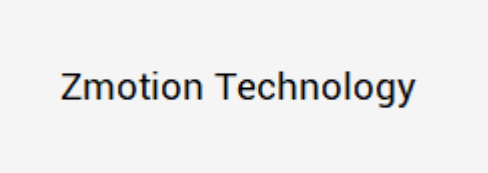
Type	Show operations.
Description	Get or set the enable of the object, which can be specified in HMI.
Grammar	value= HMI_CONTROLVALID ([winid, controlid]) HMI_CONTROLVALID (winid, controlid) = value winid: the window number in the HMI file controlid: object number, the default is the current "custom" object value: when it is 1, object touching is valid, when it is 0, no touching effect.
Controller	Controllers that support ZHMI
Example	HMI_CONTROLVALID(10,5)=0 'the fifth object of window 10 is invalid
Instructions	/

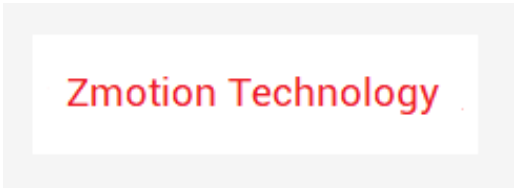
6.6.13. HMI_CONTROLSTRING – Get String Info

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 ZHMI
Example	RSTRING(0,10)= HMI_CONTROLSTRING (10, 2) 'data of the second object in window 10 is read into VR
Instructions	/

6.6.14. HMI_CONTROLATTR – Set / Get Component Property

Type	Show operations.
Description	Get/set the specified property value of the control. 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 "POSX": control X position "POSY": control Y position "SIZEX": width of the control "SIZEY": height of the control "MIN": the minimum value (only the single state control containing this property is valid, the same below) "MAX": the maximum value "FORE": foreground color "BACK": background color "PWD": password display "DECS": decimal places "STATE": control state "POPUP": pop up state (pull down list) "CURSORX": control's input cursor column "CURSORY": control's input cursor row "FOCUS": control focus state "MSELROWS": multi-selection row, positive = multiple select from the up to bottom, negative = multiple select from the bottom to up, 0 means no multi-selection. When it matched with "CURSORY" parameter, current selected row area can be known. 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 ZHMI
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:  ➢ The style after setting:



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:

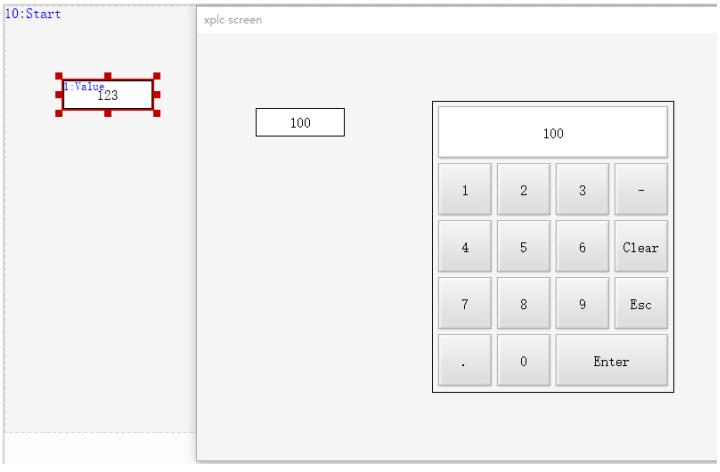
HMI_CONTROLATTR("MIN",10,10,6)

'set the minimum value of the value display control

HMI_CONTROLATTR("MAX",100,10,6)

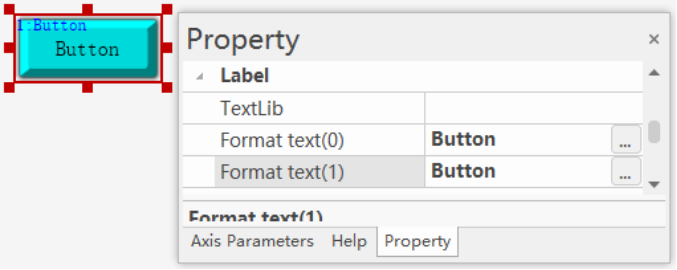
'set the maximum value of the display control

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”.



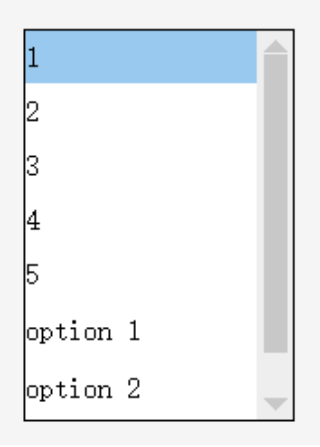
6.6.15. HMI_CONTROLTEXT – Modify Component Text

Type	Control operation command.
Description	Set the format text that will be shown in the component. It supports controls

	that are with format text. Note: The length of the modified text cannot exceed the current control text, otherwise it will be truncated.
Grammar	HMI_CONTROLTEXT (winid, controlid, state, strText) winid: the window No. in the HMI file controlid: object No. state: specifies which state the control text should be modified in strText: modify the format text
Controller	Controllers that support RTHMI
Example	HMI_CONTROLTEXT (10, 1, 0, “released”) ‘set text when button released HMI_CONTROLTEXT (10, 1, 1, “pressed”) ‘set text when button pressed --Before--  --after-- Left: state 0, showing released. Right: state 1, showing pressed. 

6.6.16. HMI_LISTTEXTS – Modify Text of List Object

Type	Control 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: 

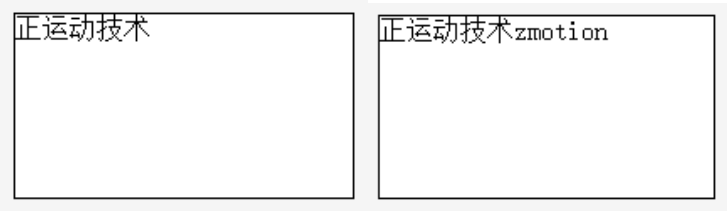
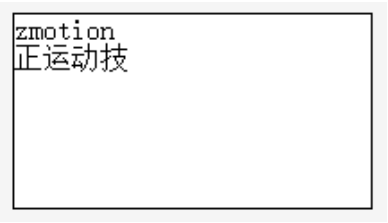
6.6.17. HMI_LISTITEM – Modify Assigned List Item Text

Type	Control 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

6.6.18. HMI_STRAPPEND – Object Text Append

Type	Control 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

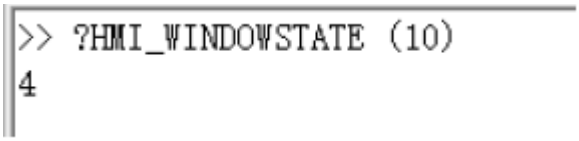
	ifNewLine: whether to append a new line, the default is 0 (no) Note: appending a new line occupies one character space by default
Controller	Controllers that support RTHMI
Example	**Example 1: HMI_STRAPPEND ("zmotion",10, 2, 0, 0)** 'append a line of string to the end of the component text  **Example 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 

6.6.19. 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

6.6.20. 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	 Read result: 4, which means window 10 is the base window.

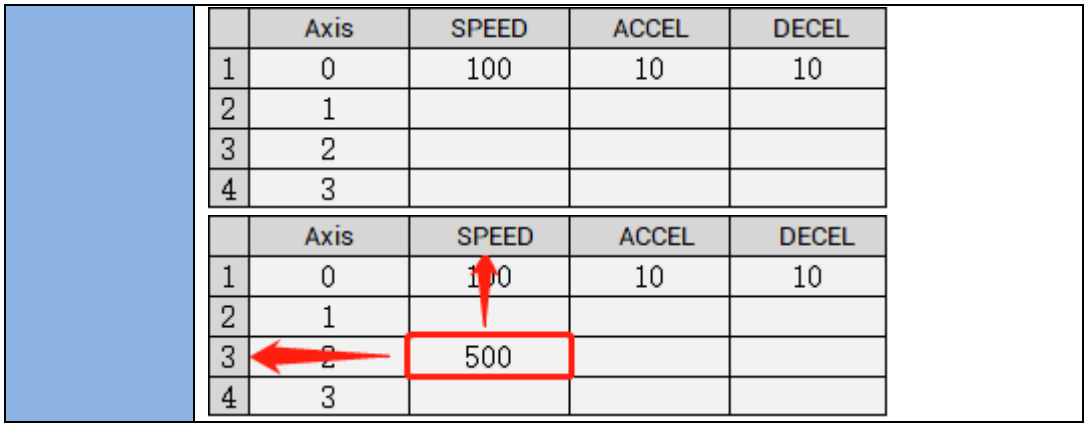
6.6.21. HMI_MOVEWINDOW – Move Assigned Window

Type	Window operation command
Description	Move specified window. Firmware versions after 20161112 are supported.
Grammar	HMI_MOVEWINDOW (winid, posx, posy [, sizex, sizey]) winid: window number posx: horizontal coordinate posy: vertical coordinate

	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)

6.6.22. 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			

	Axis	SPEED	ACCEL	DECEL
1	0	100	10	10
2	1			
3	2	500		
4	3			

6.6.23. HMI_TABLETEXT – Set / Get Table Content

Type	"report view" control operation commands
Description	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.
Grammar	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. aaa bbb ccc ddd eee fff ggg hhh iii Note: This command only can operate selected part. Row No. and column No. start coding from 0 (head row / head column is not included)!
Controller	Controllers that support RTHMI
Example	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.

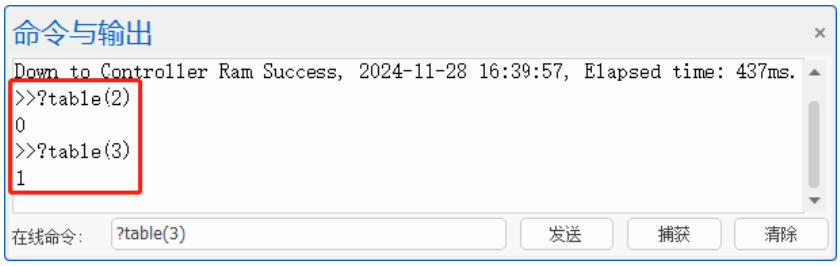
	Axis	SPEED	ACCEL	DECEL
1	0	100	10	10
2	1			
3	2			
4	3			

	Axis	SPEED	ACCEL	DECEL
1	0	100	10	10
2	1	x/t		
3	2			
4	3			

6.6.24. HMI_TABLECURSOR – Get Current Selected Row & Column

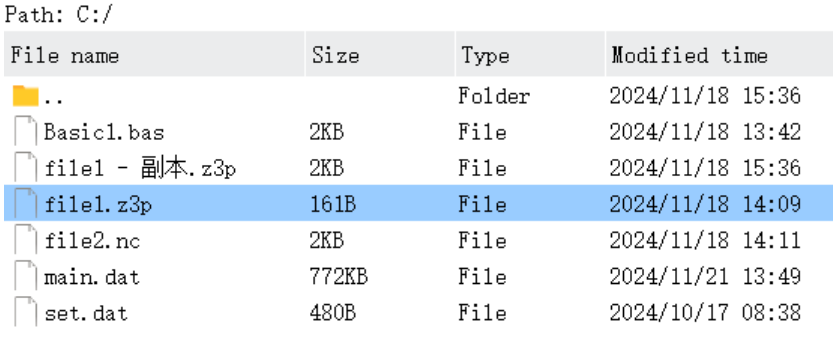
Type	"report view" control operation commands
Description	Get current row and column of assigned "report view" control, and save it into table.
Grammar	HMI_TABLECURSOR (winid, controlid, num) string = HMI_TABLETEXT (winid, controlid, row, col) winid: window No. in HMI file controlid: component No. num: 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 selecet. Note: Row No. and column No. start coding from 0 (head row / head column is not included)!
Controller	Controllers that support RTHMI
Example	HMI_TABLETEXT(16, 1, 2) 'get current selected row and column of window 10 first control, 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.

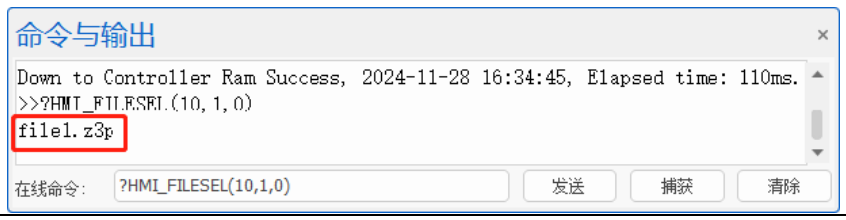
	Axis	SPEED	ACEEL	DECEL
1	0	100	10	10
2	1			
3	2			
4	3			



6.6.25. HMI_FILESEL – Get Selected File / Folder Name

Type	"file browser" control operation commands
Description	Get current selected file / folder name (without path) of "file browser" control.
Grammar	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.
Controller	Controllers that support RTHMI
Example	?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.





6.6.26. 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) 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.
Controller	Controllers that support RTHMI
Example	HMI_FILEPATH(10, 1, “A:/”) ‘set path of window 10 first control as A disk ➢ Current default path is C disk 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, the path is set as A disk. Path: A:/ File name | Size | Type | Modified time .. | | Folder | 2024/10/11 09:04

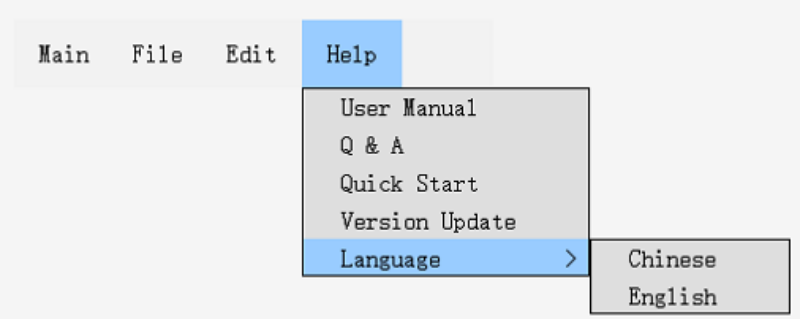
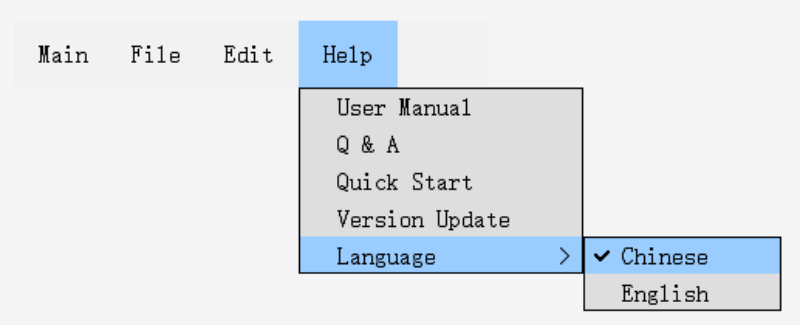
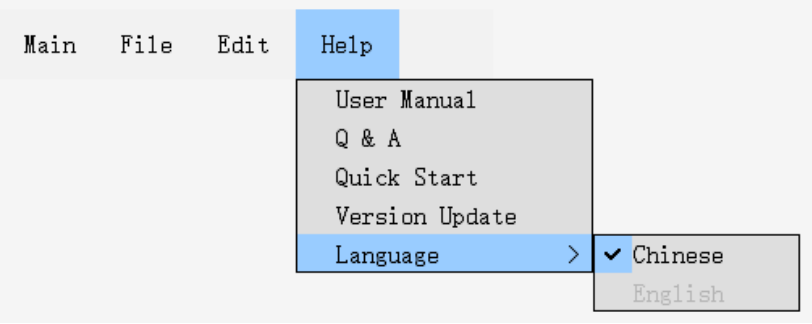
6.6.27. HMI_FILEFILTER – Set File Filter

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

<table>
<tr><td></td><td>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.</td></tr>
<tr><td>Controller</td><td>Controllers that support RTHMI</td></tr>
<tr><td>Example</td><td>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</td></tr>
</table>

6.6.28. 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. 

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

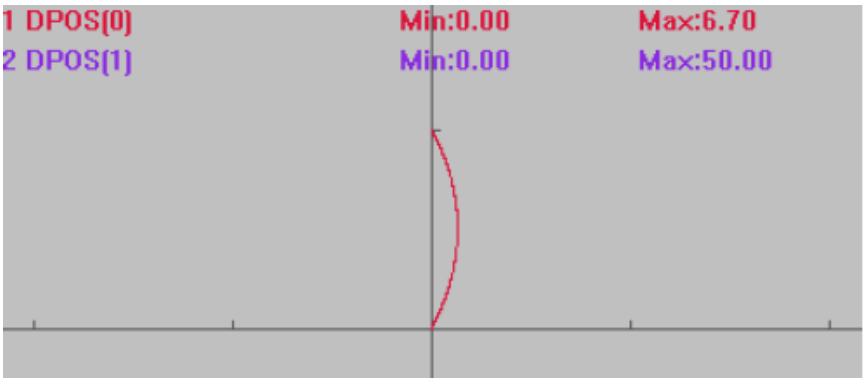
Type	DT Motion Functions
Description	Store the axis No. and movement distance in the TABLE table respectively, and perform linear movement through the TABLE list. Select the axis No. in the TABLE table using bit presence.
Grammar	MOVEDTSP(maximum number of axes, bit present, axis dt list starting number, 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

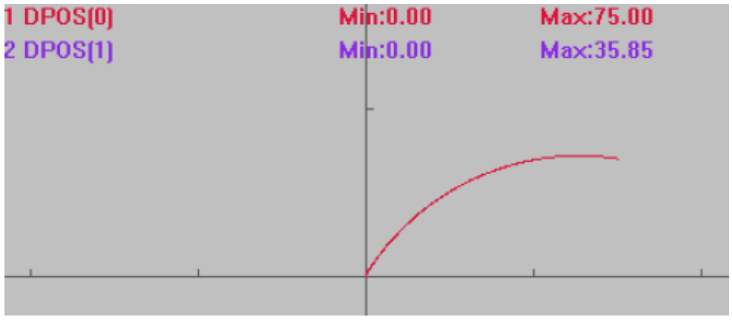
Type	DT Motion Functions
Description	Store the axis number and movement distance in the TABLE table

<table>
<tr><td></td><td>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.</td></tr>
<tr><td>Grammar</td><td>MOVECIRCDTSP(maximum number of axes, bit present, axis dt list, end point dt list, radius, cw 0/ccw 1)</td></tr>
<tr><td>Controller</td><td>General</td></tr>
<tr><td>Example</td><td>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
 
</td></tr>
</table>

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

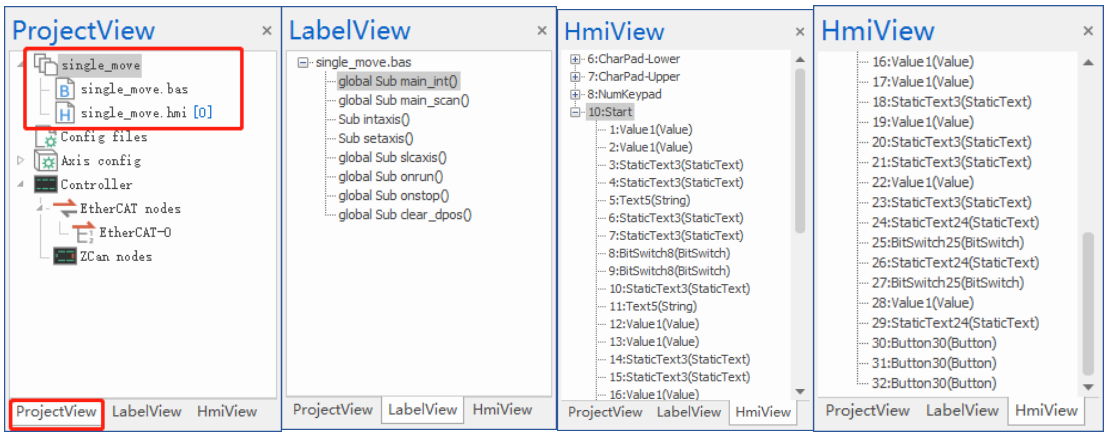
8.1. Single Axis Motion

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:**

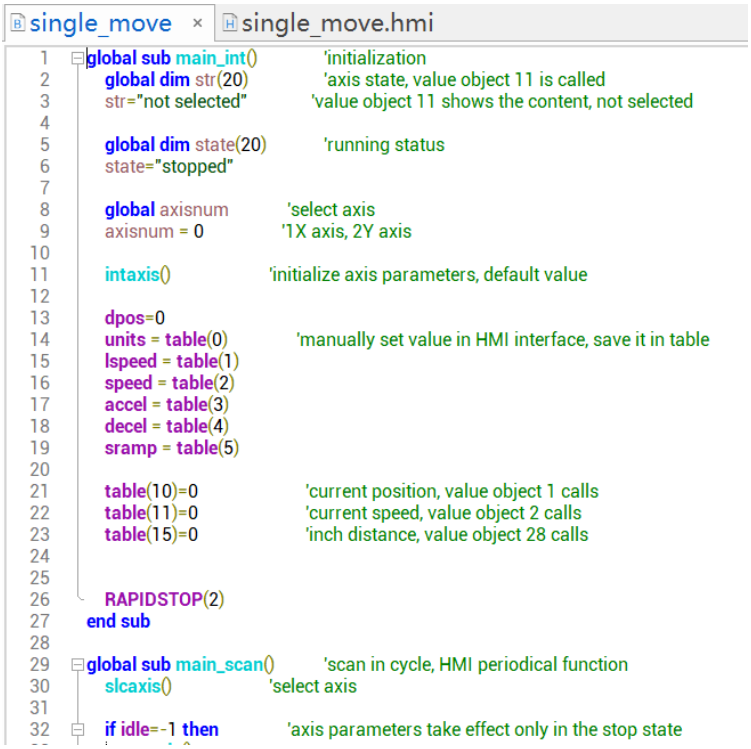
  Create Basic file and HMI file, then define needed GLOBAL SUB function in BASIC for HMI. According to below showing, add HMI component files.
  - FileView: the files contained in the project.
  - LabelView: SUB sub-process contained in each file.
  - HmiView: configuration windows and configuration object.

In HMI programming mode, it is often necessary to use the HmiView to switch configuration programming windows.



➢ **Routine Description:**

- **Basic Program Interface**



--Basic Program Codes--

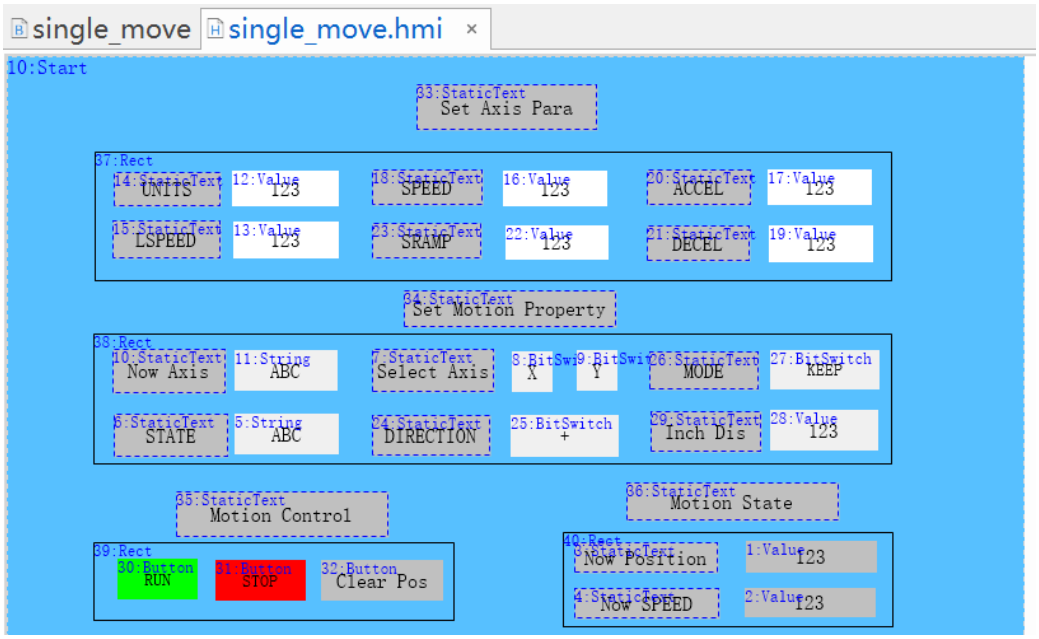
```
global sub main_int()                    'initialization
    global dim str(20)                   'axis state, value object 11 is called
    str="not selected"                   ‘value object 11 shows the content, not selected

    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)
```

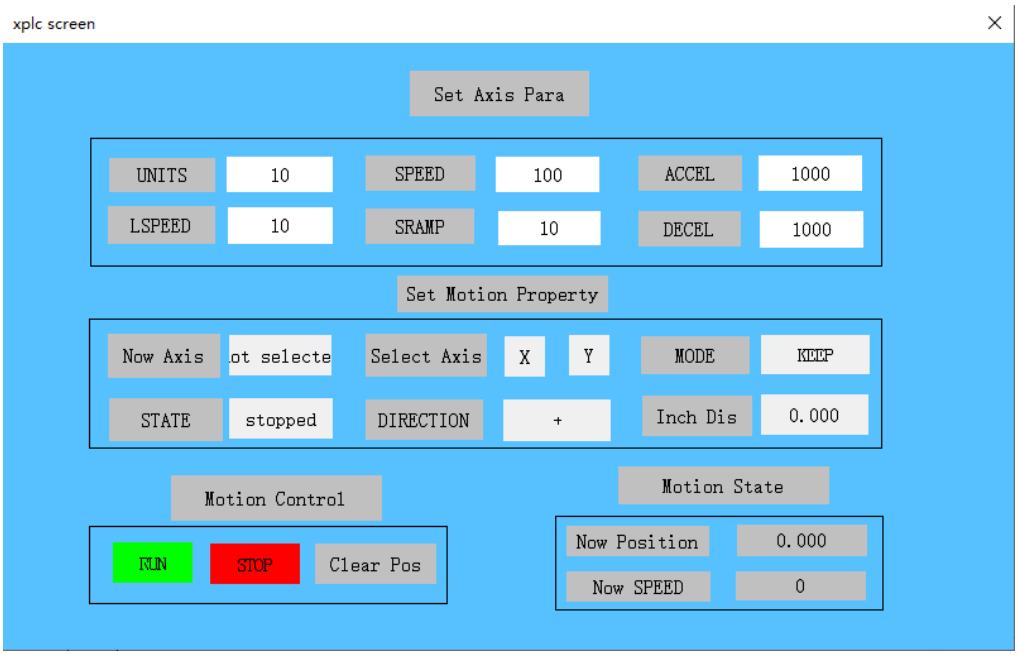
```
    accel = table(3)
    decel = table(4)
    sramp = table(5)

    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**





- 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.

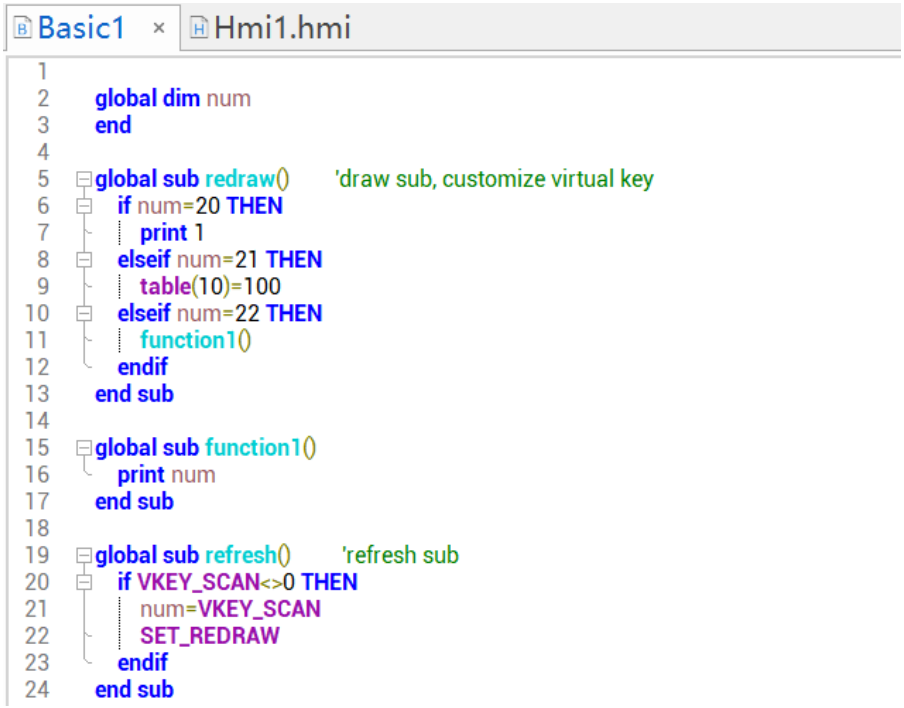
➢ **How to Do?**

- 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.



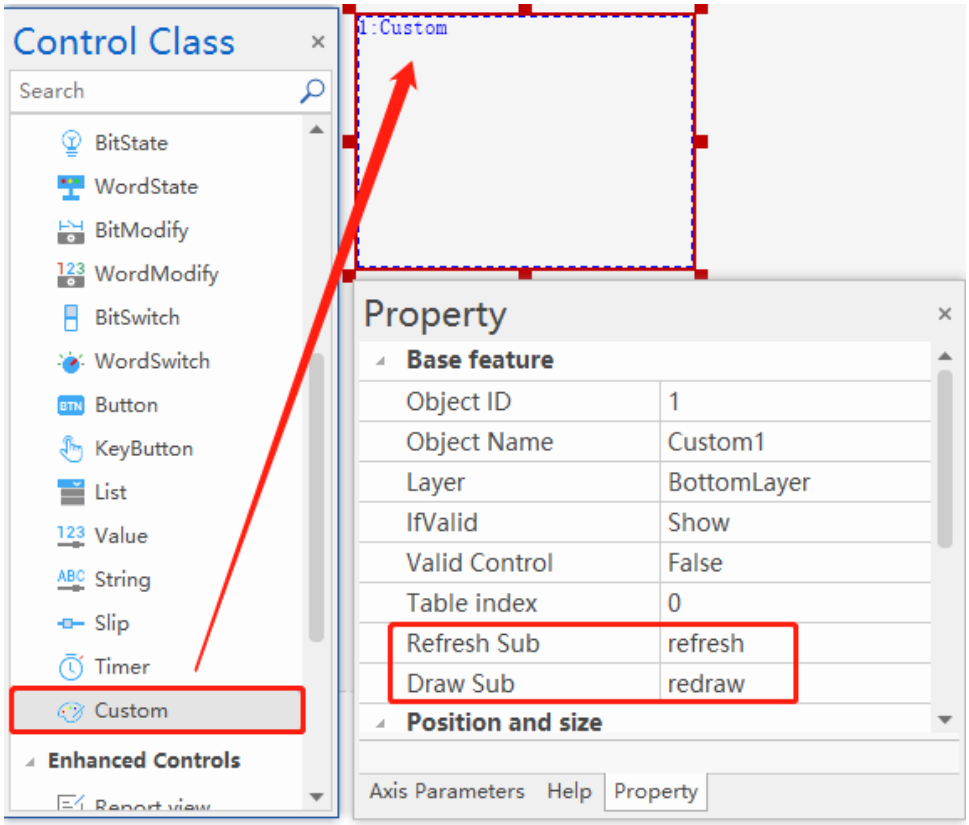
--BASIC Program Codes--

```
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
```

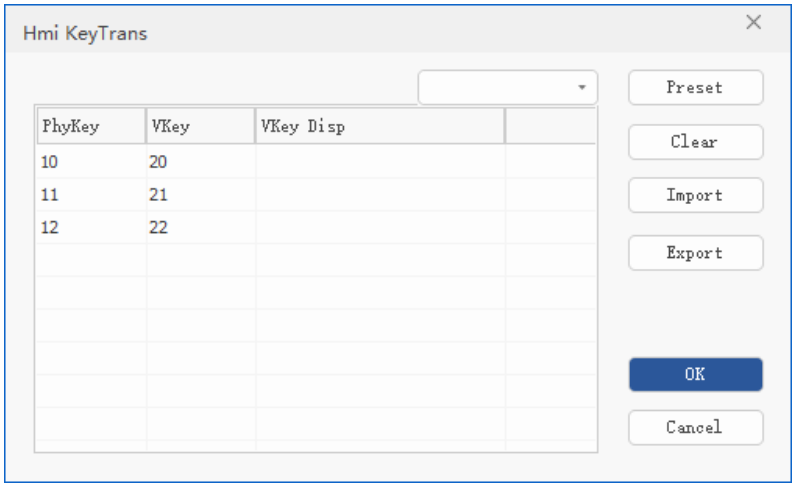
➢ **HMI Interface:**

- 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.



- 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.



➢ **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.

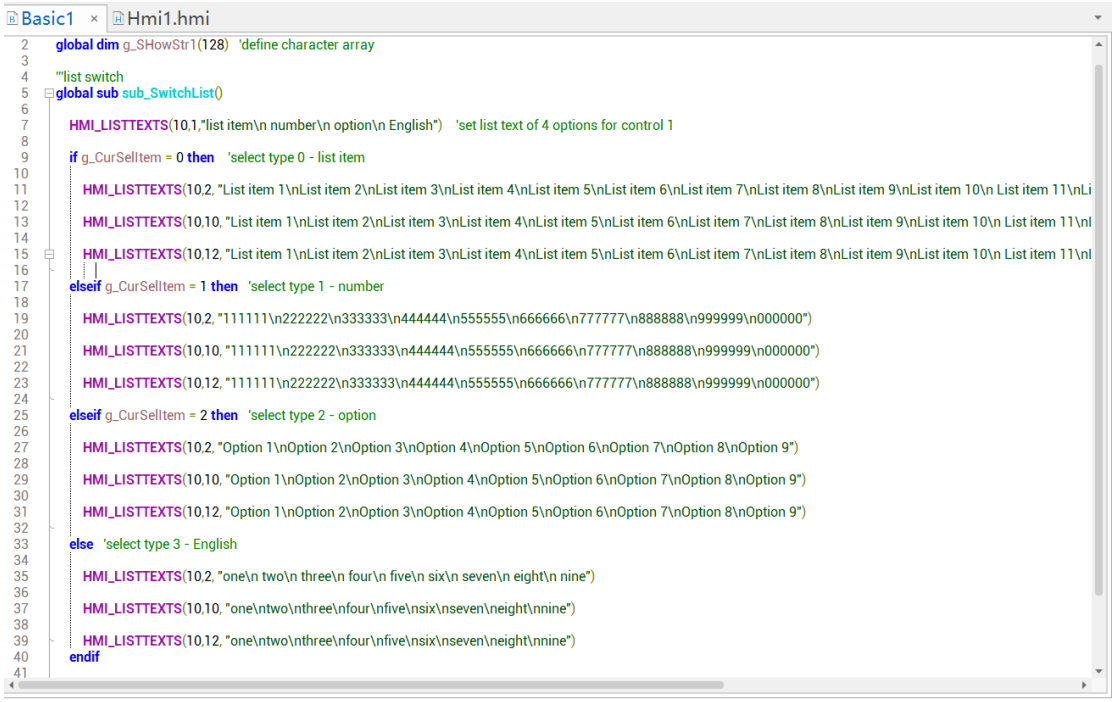
➢ **How to Do?**

- 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.



--BASIC Program Codes--

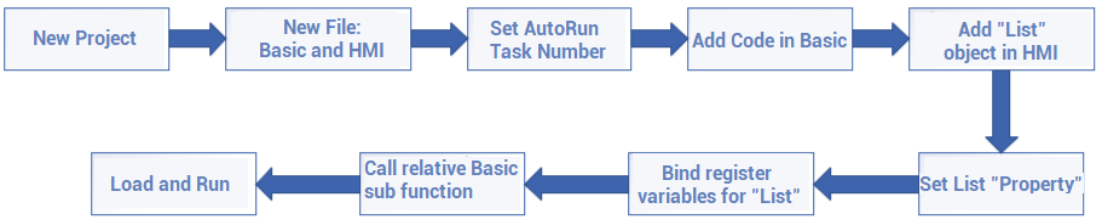
```
global dim g_CurSelItem          'define list item types, 0-list item, 1-number, 2-option, 3-English
global dim g_SHowStr1(128)       'define character array

'"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
```

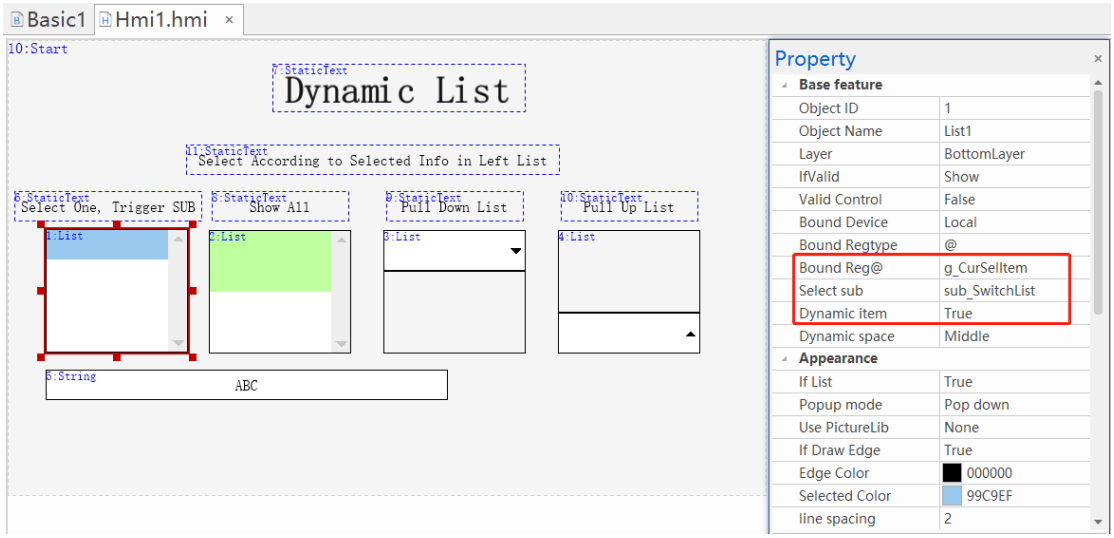
```
        HMI_LISTTEXTS(10,10, "List item 1\nList item 2\nList item 3\nList item 4\nList item 5\nList item 6\nList item 7\nList item 8\nList item 9\nList item 10\n List item 11\nList item 12")    'set list text for control 10

        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
```

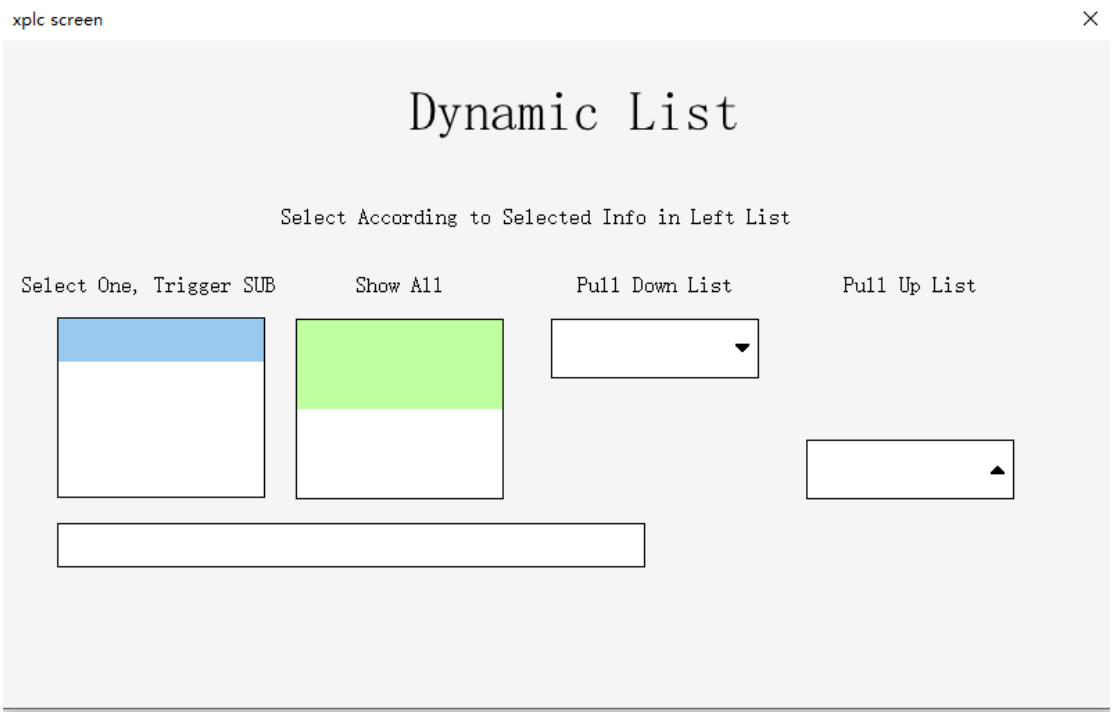
➢ **HMI Interface:**

- 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.





➢ **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:**

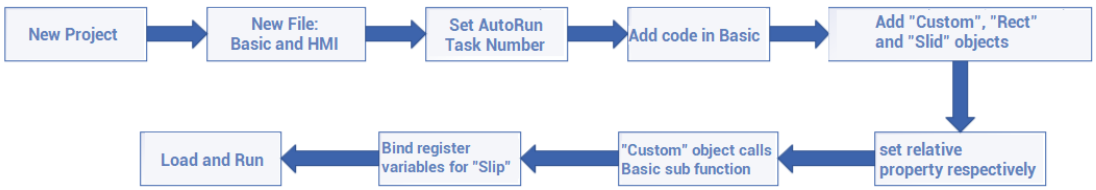
In “custom” object, use the command to draw one rectangle, then you can use the slider to zoom in and out rectangle.

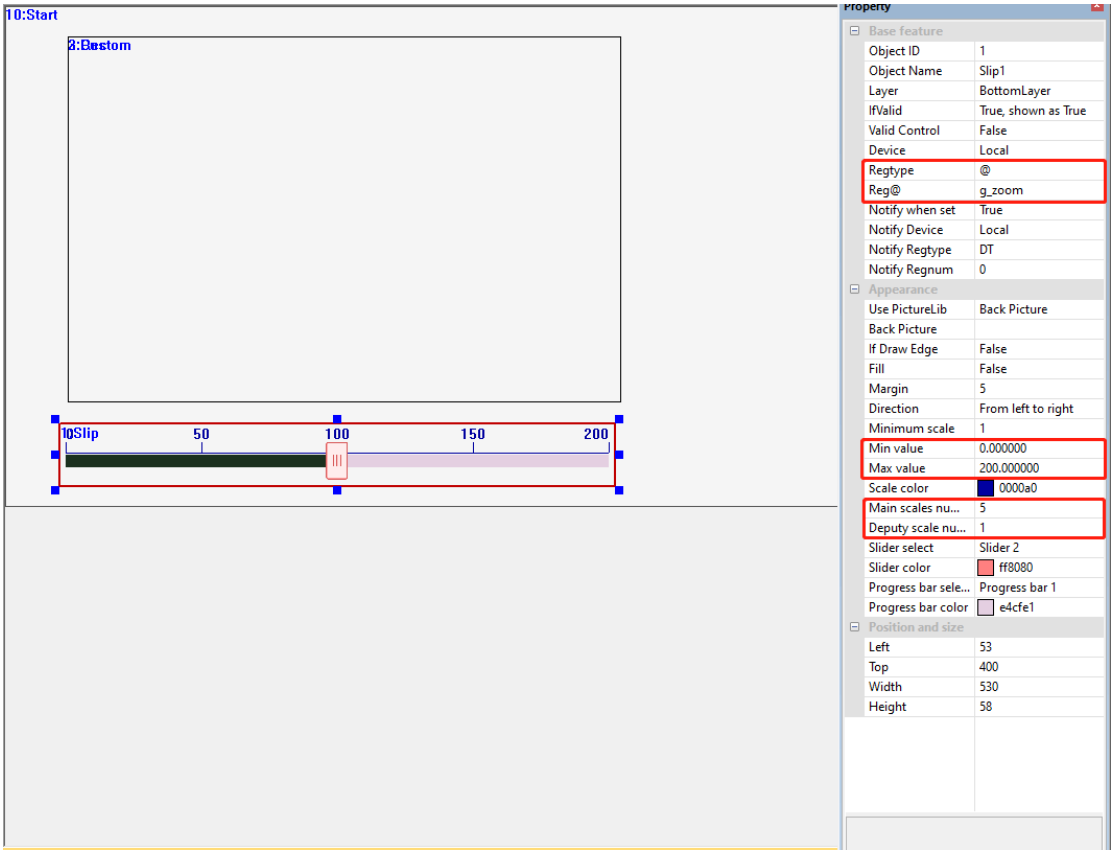
Step 1: create a new HMI file, set the auto-run task number, and open the HMI window.

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.

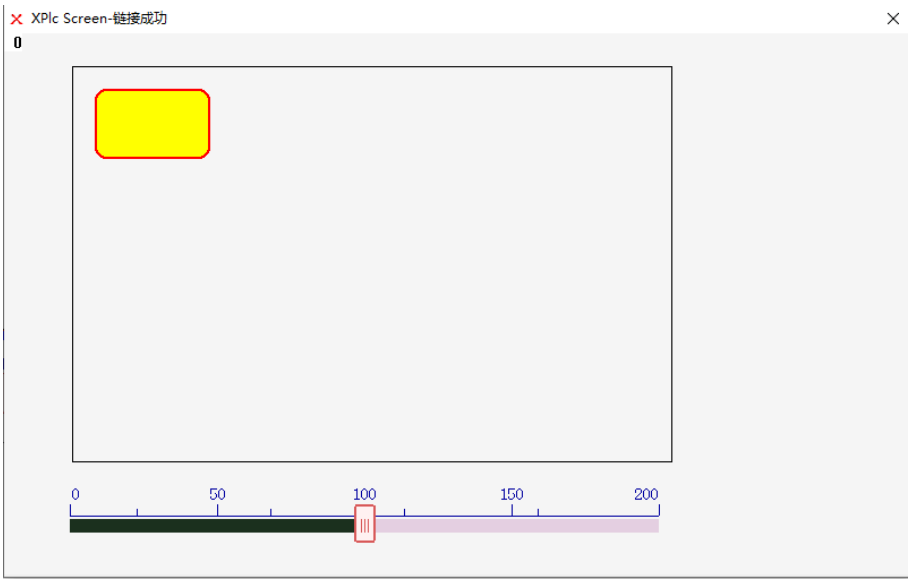
Step 4: draw the "slip" object, adjust the style, size, display value range, etc. Scales can be set through “Main scale number” and “Deputy scale number” in the "Property" (below: main scale is 5, and deputy scale is 1), and put in suitable position (it is recommended to put outside the “custom” object), at the same time, bind register variables.



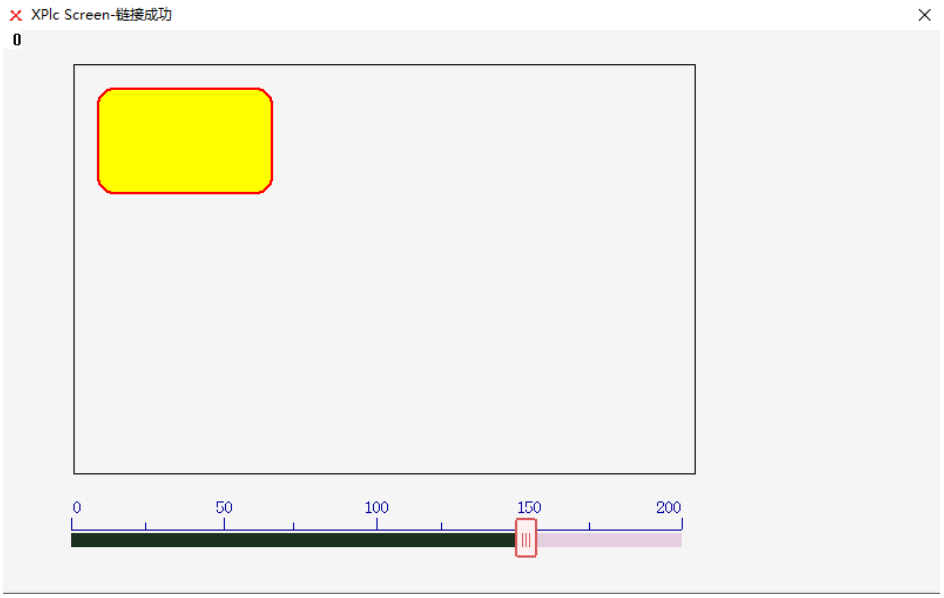


➢ **Effects:**

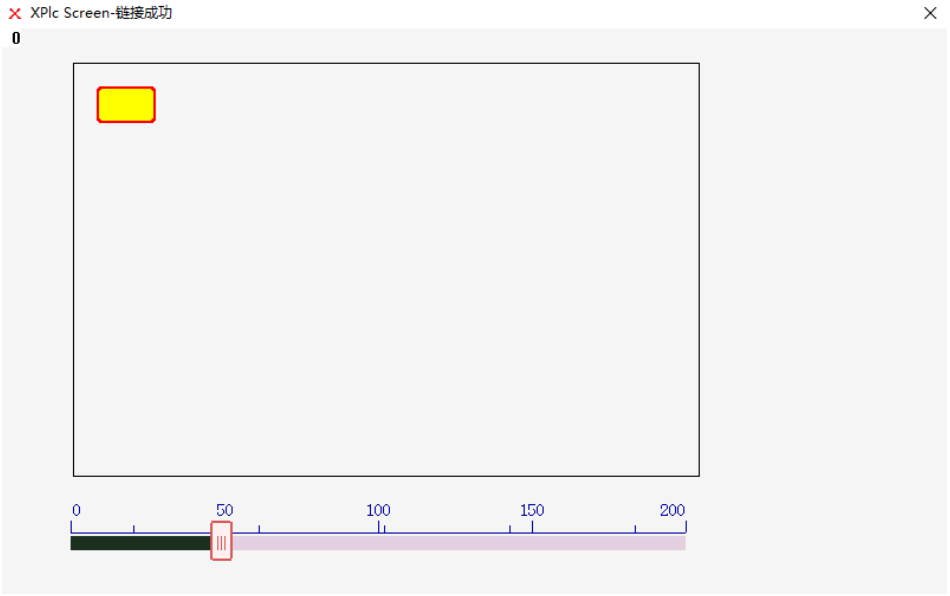
After running, since the initial value of the setting is 100, the slider is in the middle position by default, that is, at the scale of 100, and the initial size of the graph is as shown in the figure.



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.

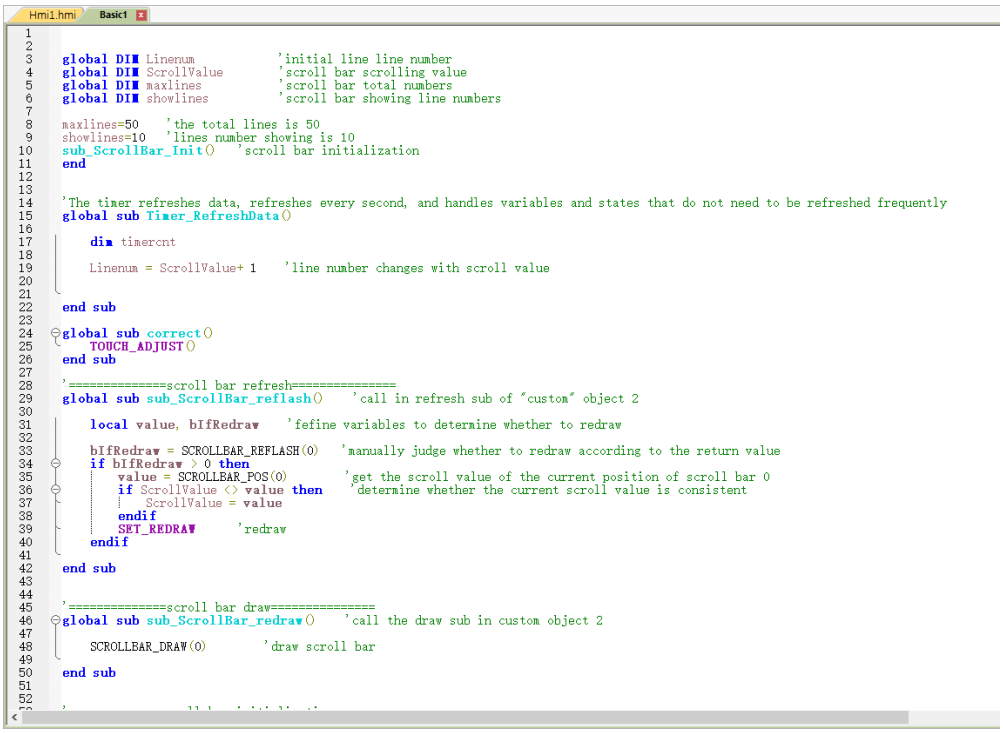
➢ **Applications**

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.



➢ **--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:**

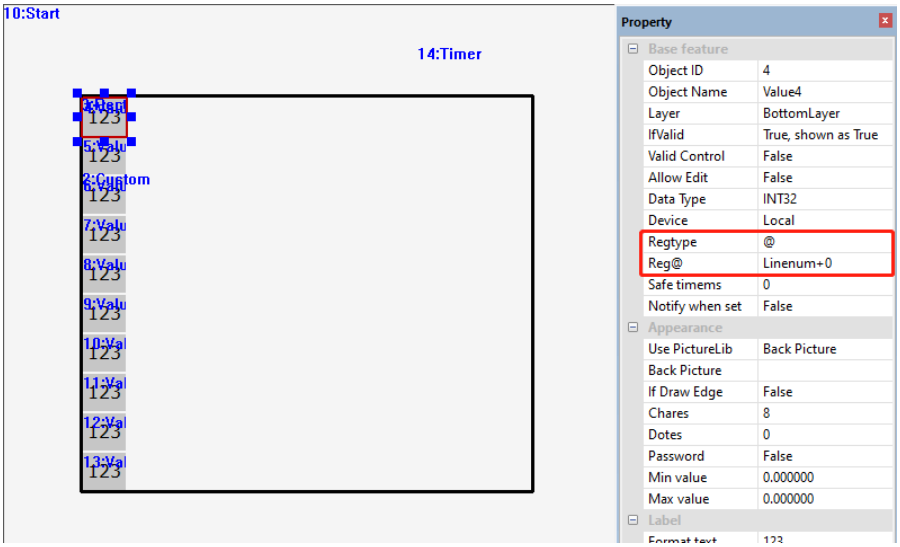
In “custom” component, draw the scroll bar, and use “value” component to show the line No., also use “timer” to refresh the line No.

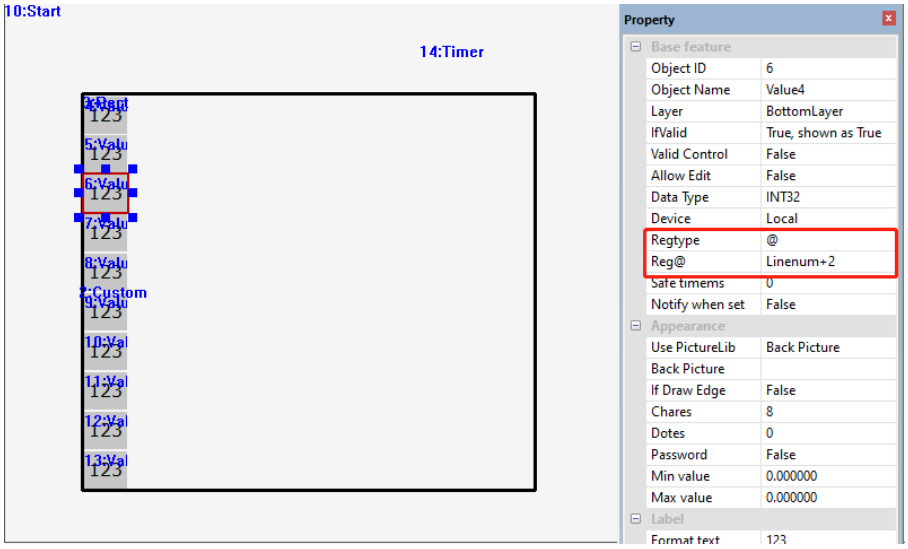
Step 1: create a new HMI file, set the auto run task No., and open the HMI window.

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.

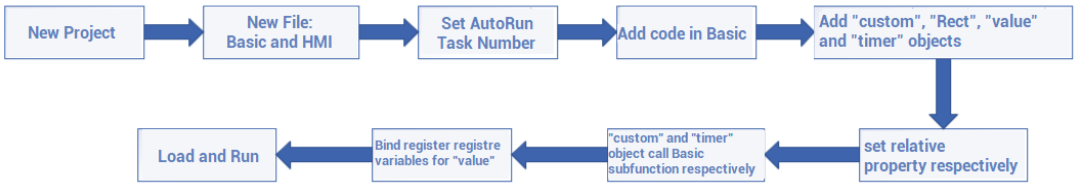
Step 3: add a "Rect" object as a frame, adjust its size to be the same as that of the "custom” object, and place it at the same position as the "custom” object. (to determine the position of the borderless custom object).

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.



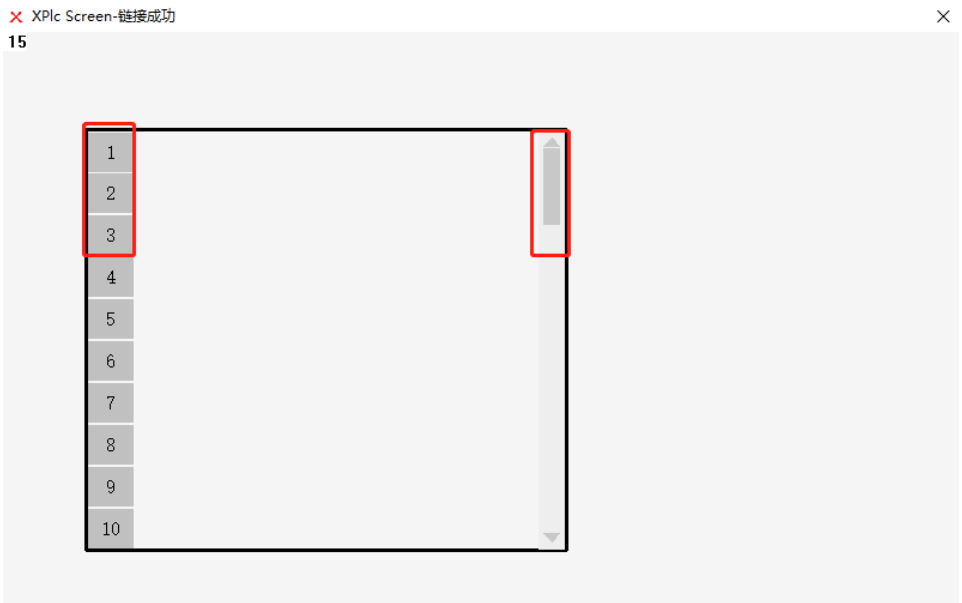


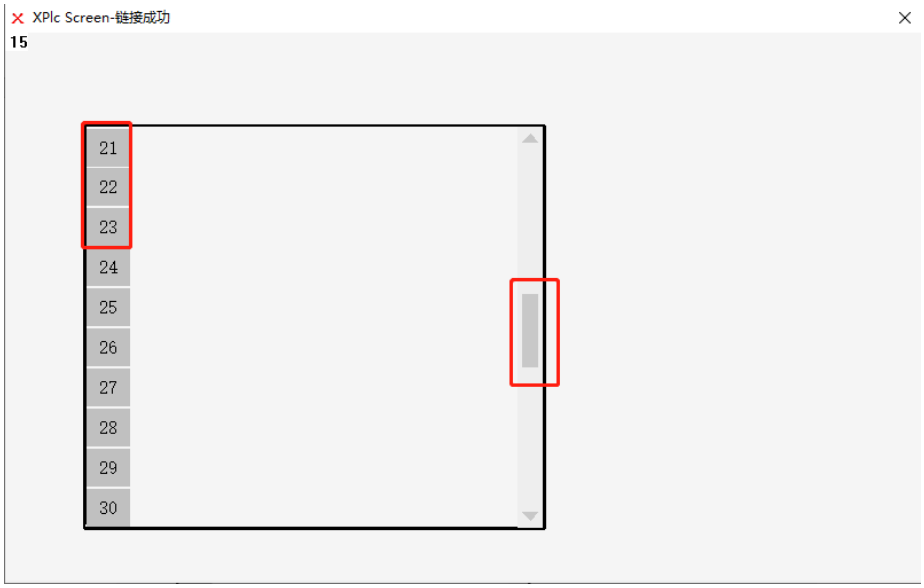
Step 5: add a "timer" object, place it in a suitable position, and adjust the "times ms" in the "property" appropriately. The lower the value, the shorter the trigger period of the timer. Select call sub in "Action" of "Property", and call the Basic subfunction at "Action Function".



➢ **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.

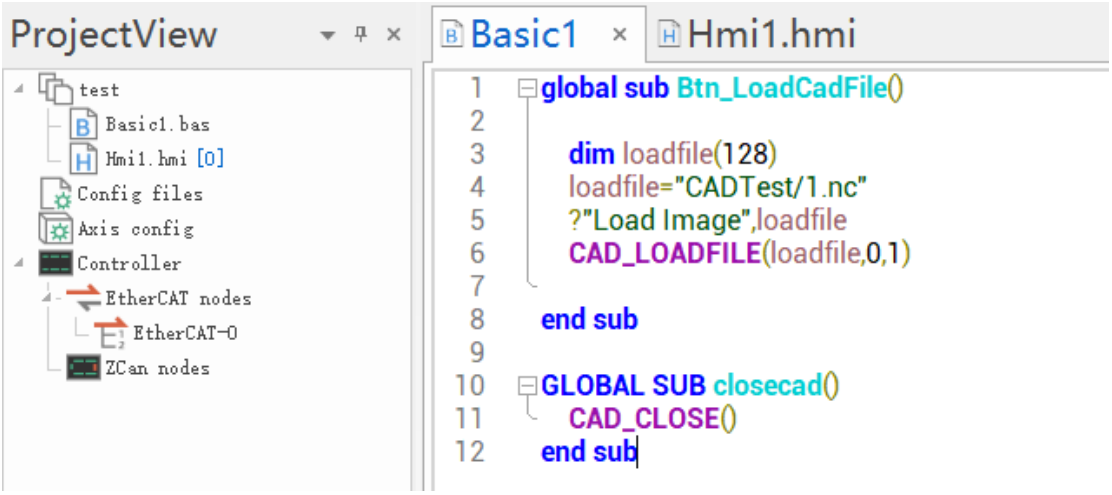
    define “import CAD graphic” function

    “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.



➢ **--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.

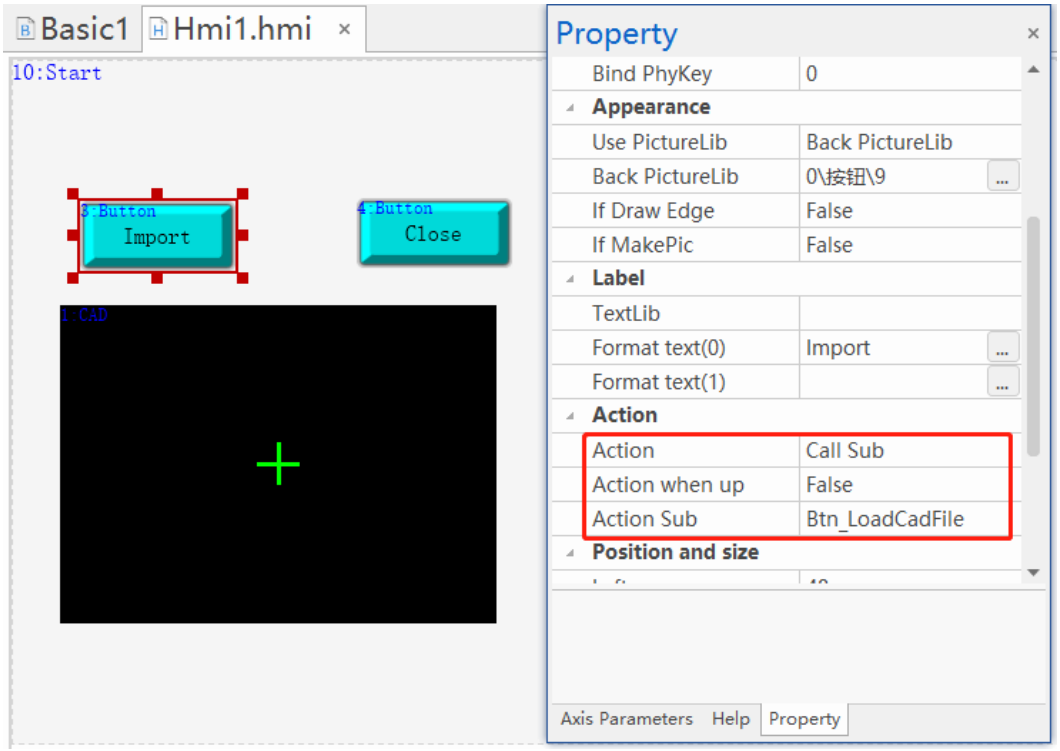
Step 3: add one CAD file, and adjust its size, put it at suitable position.

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.

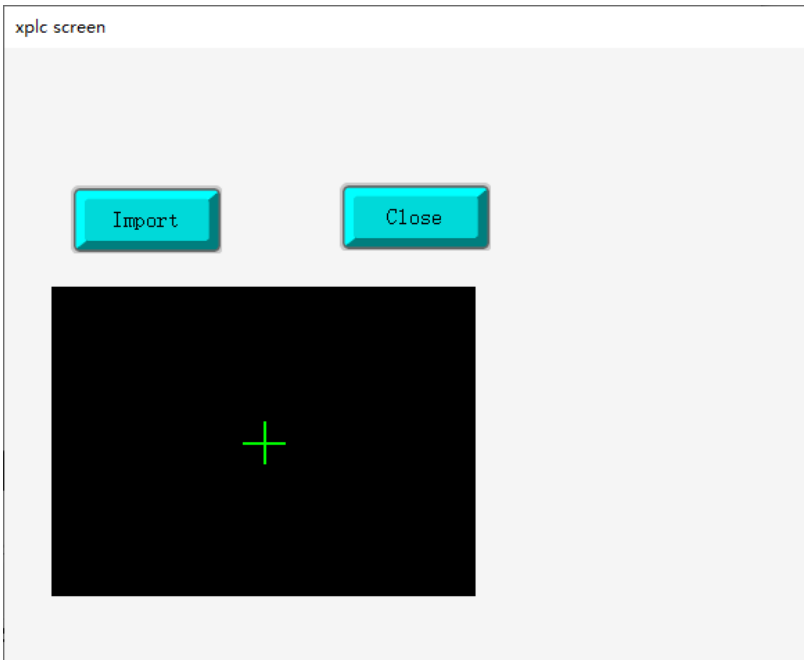
Step 5: in HMI file, add one “Button” file, adjust size, put suitable position, set the format text as “import”, and set CALL SUB for it, the action sub selects the SUB you edited in step 4.

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.



➢ **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.

➢ **--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.

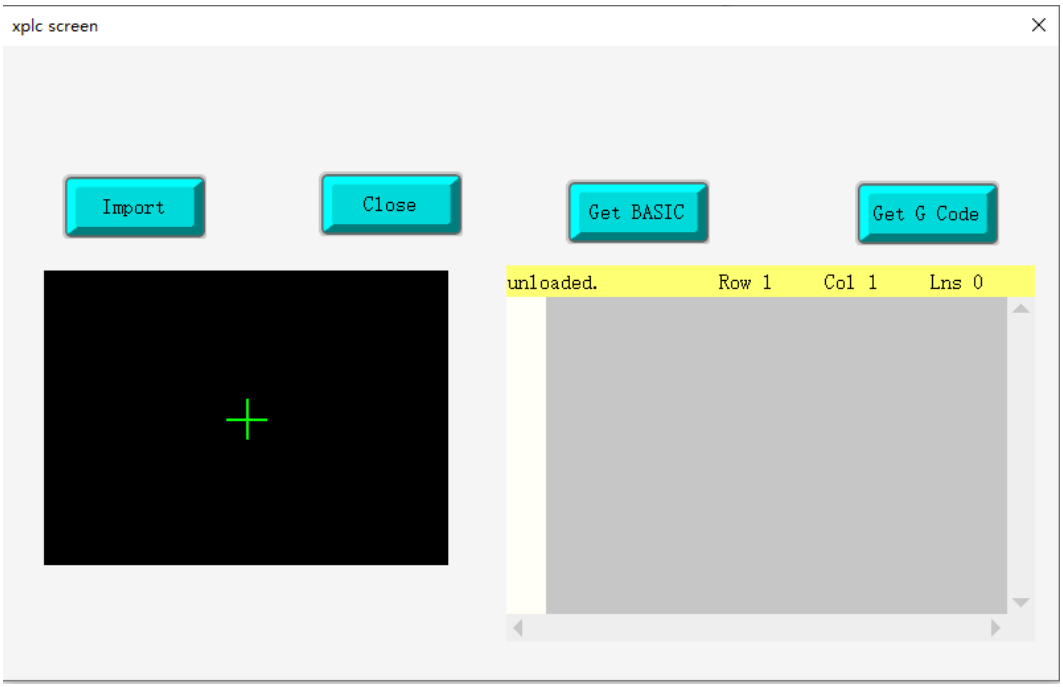
Step 2: In BASIC file, use CAD_TOCODE command to edit “export / get BASIC code” function, also set its code file name, type (type=0), and export layer.

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:**



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.

Basic1 × Hmi1.hmi

```
1
2   CAD_LOADFILE("CADTest/3D-玫瑰花.dxf",0)   'load graphic from the file
3   CAD_TOTABLE(10000, -1)                    'export TABLE data, when exported table(10000)
4   CAD_CLOSE()                               'close CAD loading
5   CAD_LOADTABLE(10000)                      'load graphic from TABLE, start to load the graphic from TABLE(10000)
6
7   end
8
9   "draw sub
10  GLOBAL SUB drawing1()
11  '
12      CAD_DRAW(0, 0, 1)             'draw CAD graphic
13  '
14  END SUB
15
16  "refresh sub
17  GLOBAL SUB refresh1()
18  '
19      CAD_MOUSE(2)                  'set mouse mode
20      CAD_SETCOLOR(0,RGB(255,0,255))   'set CAD drawing color
21      CAD_DISPLAY(3)                'set CAD showing method
22  '
23  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:**

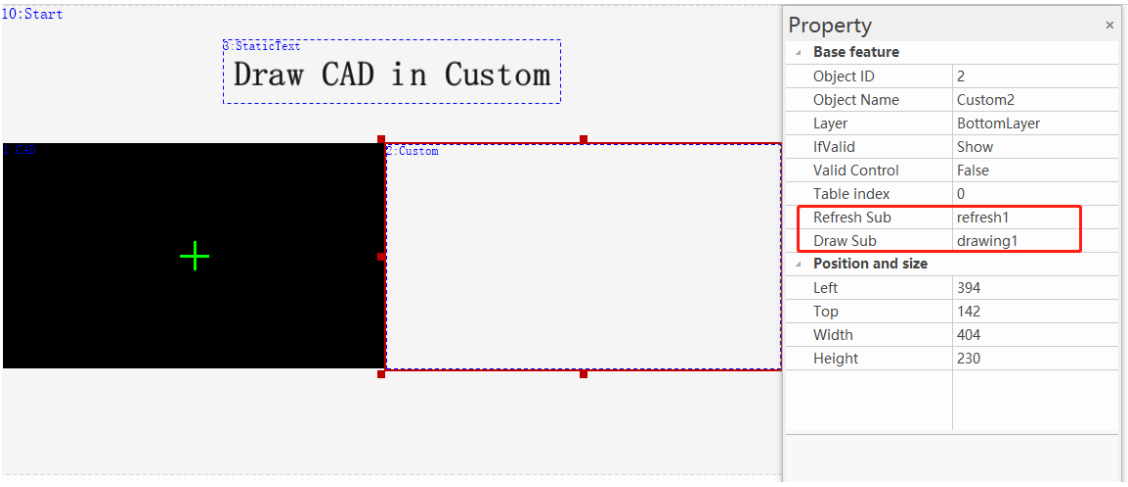
Step 1: new build one HMI file, set one auto run task No. for it, open HMI window.

Step 2: under controller / simulator flash content, insert one folder that is with vector graphic.

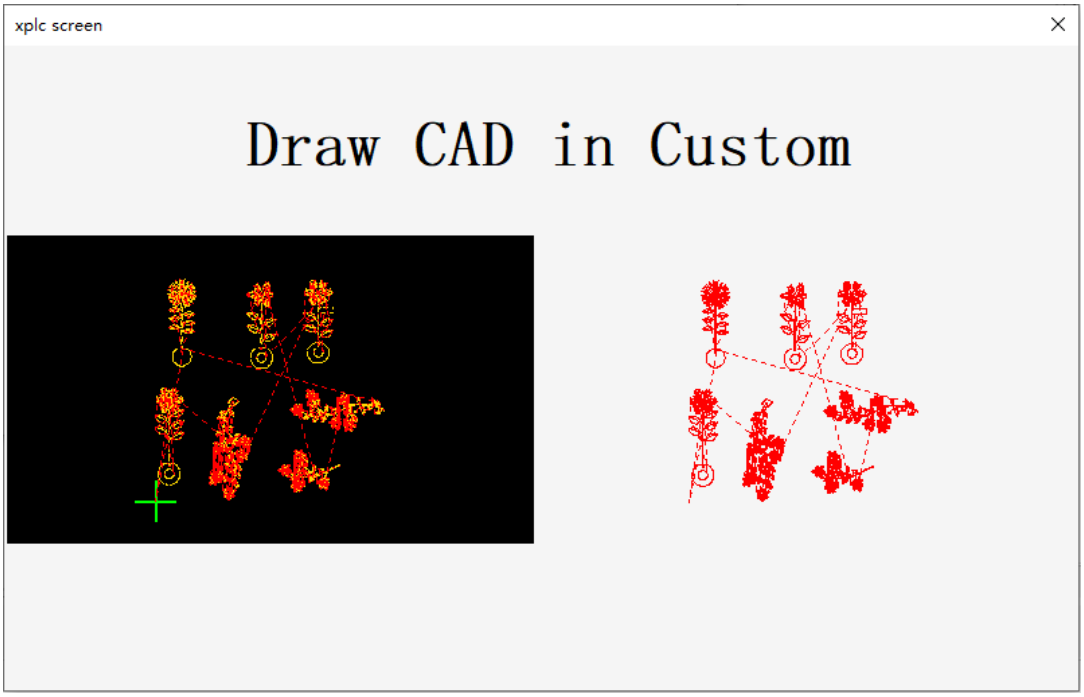
Step 3: add one CAD component and one custom component, and adjust size, put at suitable position.

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.



- 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?]

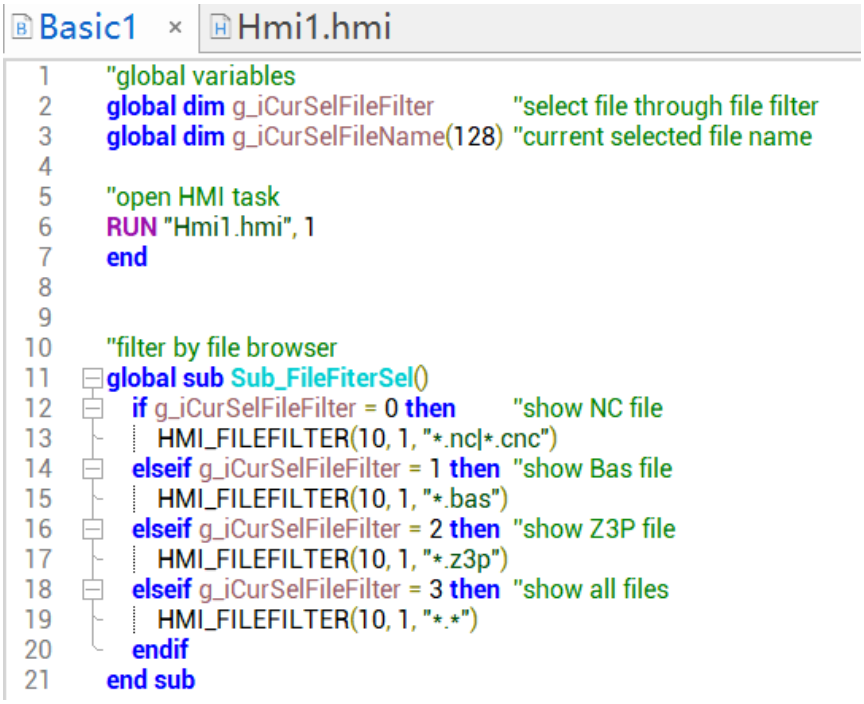
- In Basic file, use HMI_FILEFILTER, FILE_FILESEL, HMI_FILTERPATH commands to edit the program and define global SUB function.
- Use “file browser” component to call Basic SUB function to filter file type and select file.
- Use “list” component to bind with the register that corresponds to file type. 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.



--Basic Program--

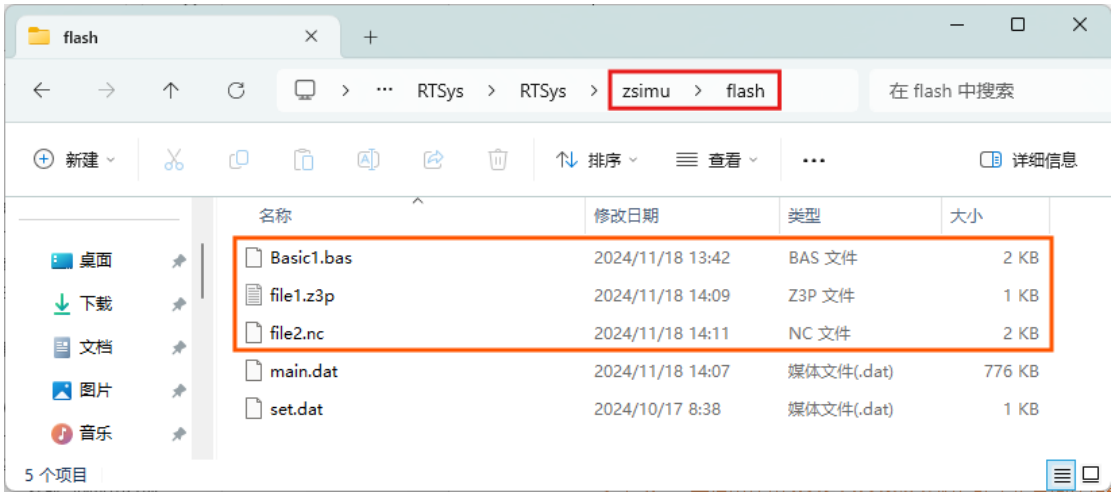
```
"global variables
global dim g_iCurSelFileFilter                "select file through file filter
global dim g_iCurSelFileName(128)     "current selected file name


"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
```

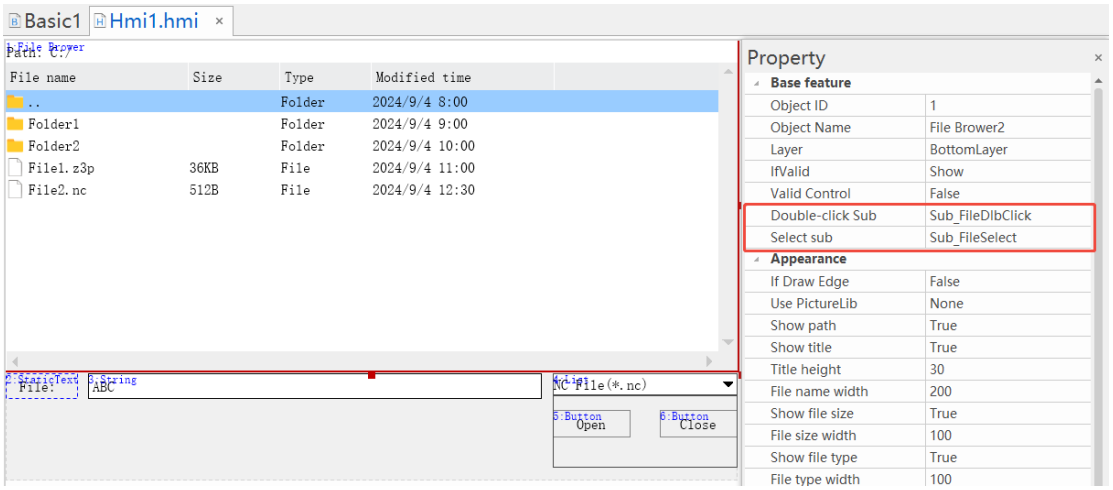
Step 2: HMI Interface

(1) Under controller / simulator flash, insert some files with types of basic, z3p, nc.



(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.

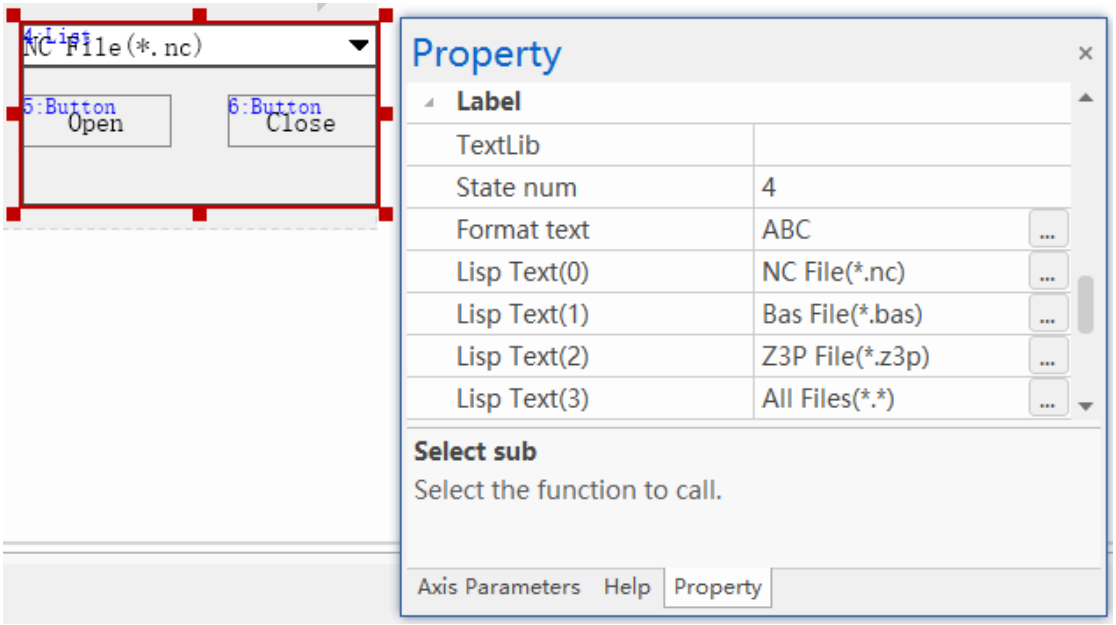


(4) In “file browser” component, add one “static text” component, one “string” component, one “list” component and two “button” components, then, adjust the size, put them at suitable position.

a. Set “static text” component format text as “file: ”.

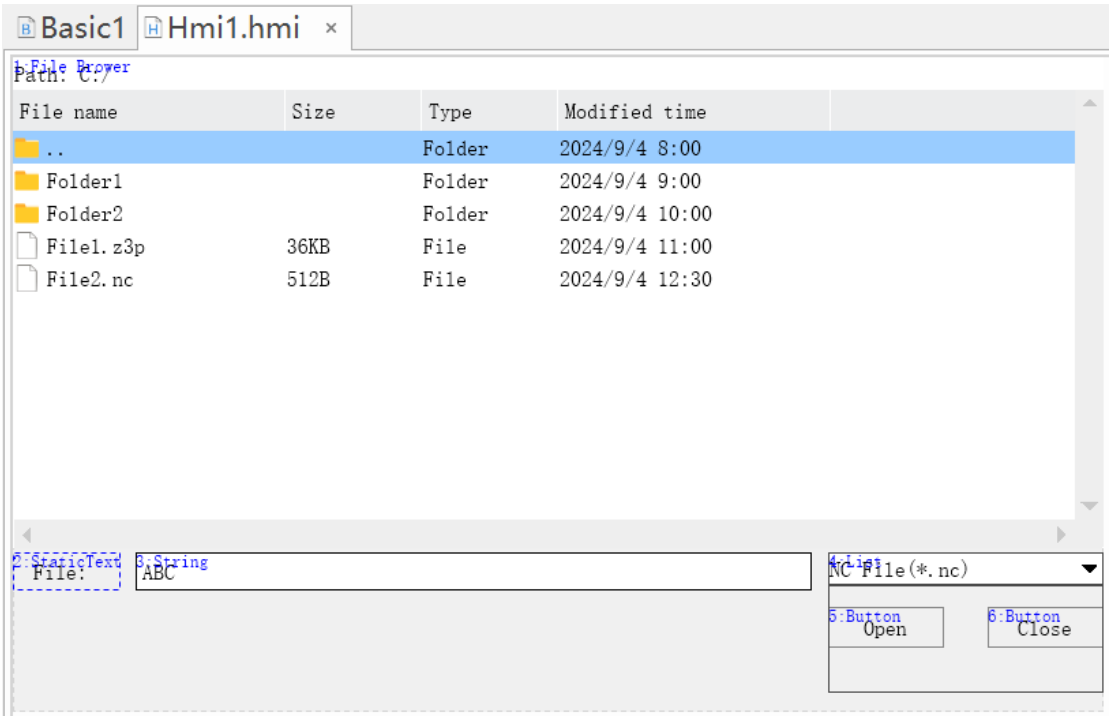
b. Bind “g_iCurSelFileName” register with “string” component to show selected file name.

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:

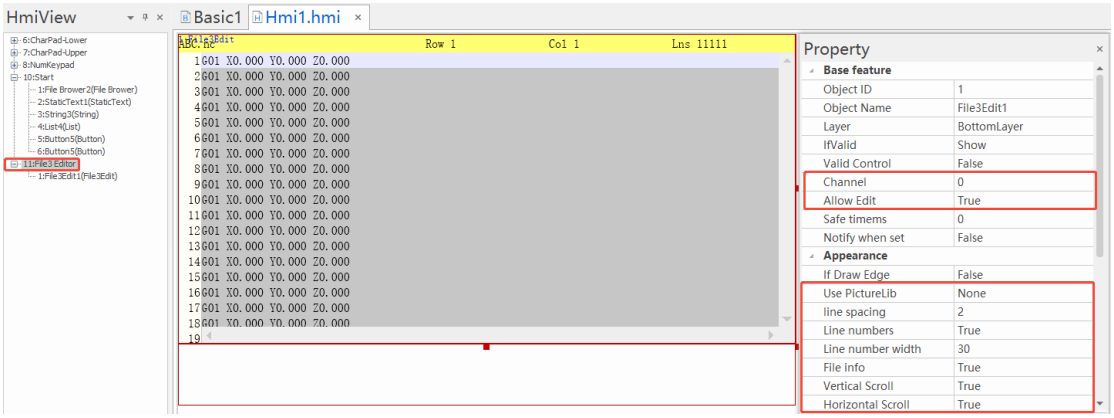


d. Call “Sub_FileOpen” function in “button” component to open file 3, select “close assigned window” action for another “button” to close file 3.

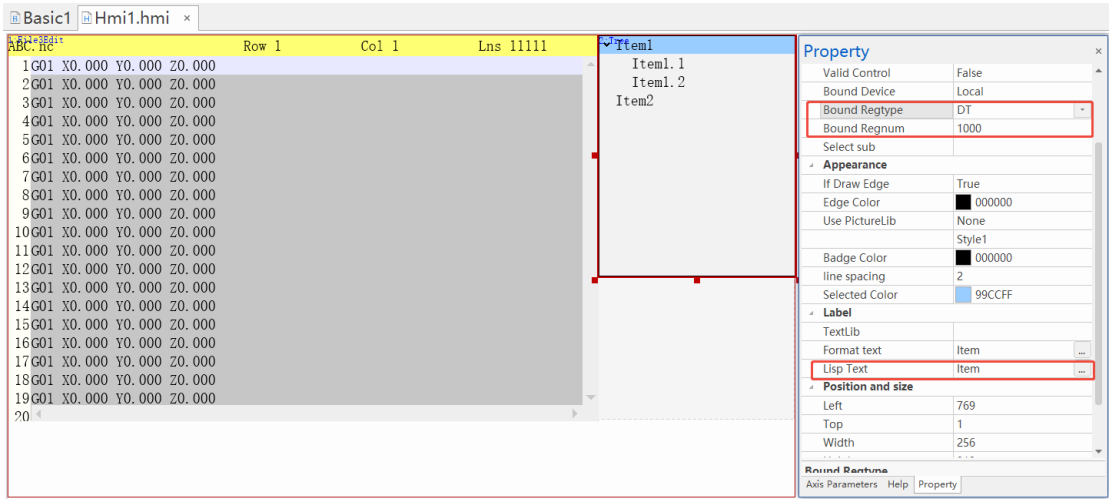
(5) The whole interface of base window 10 is shown as below:



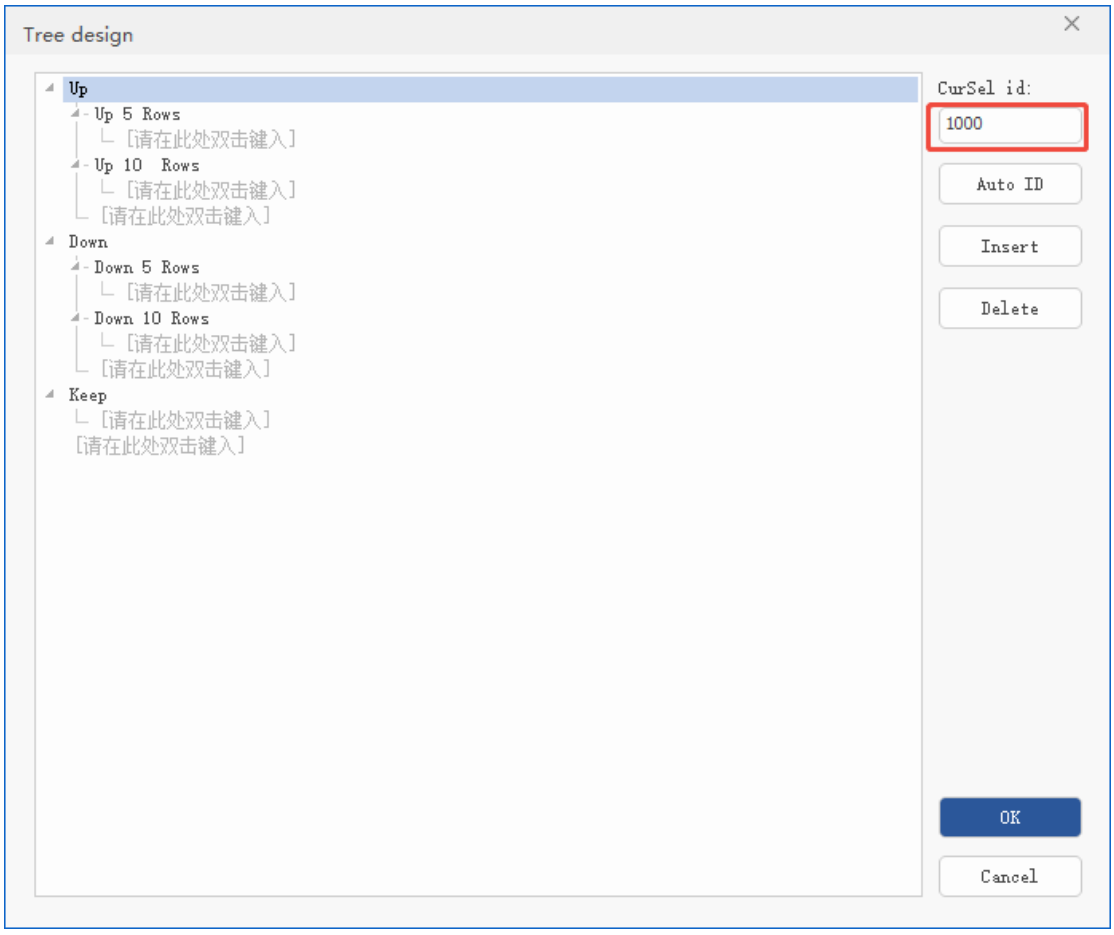
(6) Create one window 11 “popup window”, and add one “file3 editor” component, set the channel No., allow edit, and other properties. Use FILE3_BUFLOAD command to load assigned file3 into disk.



(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.



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.

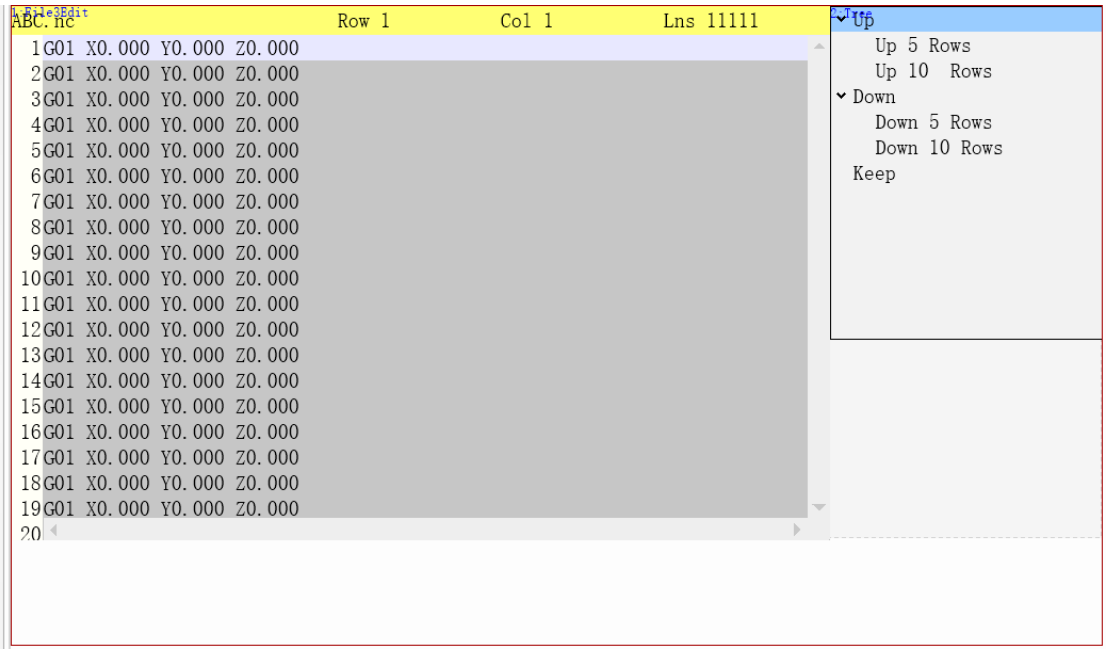


Corresponding BASIC Code:

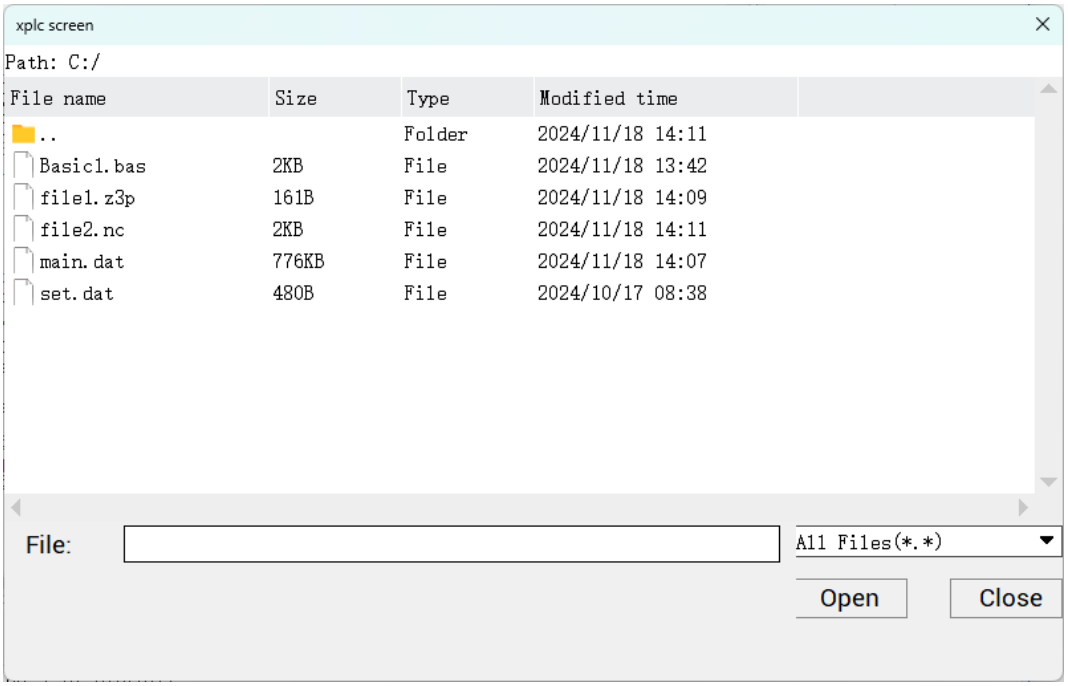
```
"tree & HMI_CONTROLATTR -- achieve "select multi-row" of file3 editor
global sub sub_tree ()
```

```
    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
```

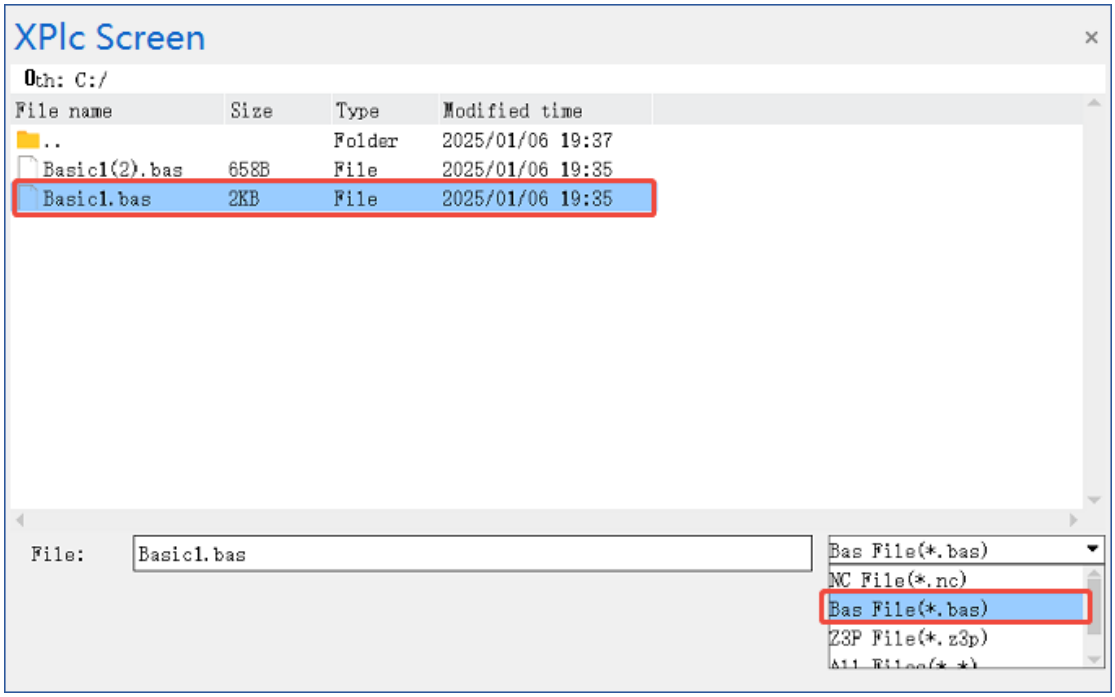
(8) The whole interface of base window 11:



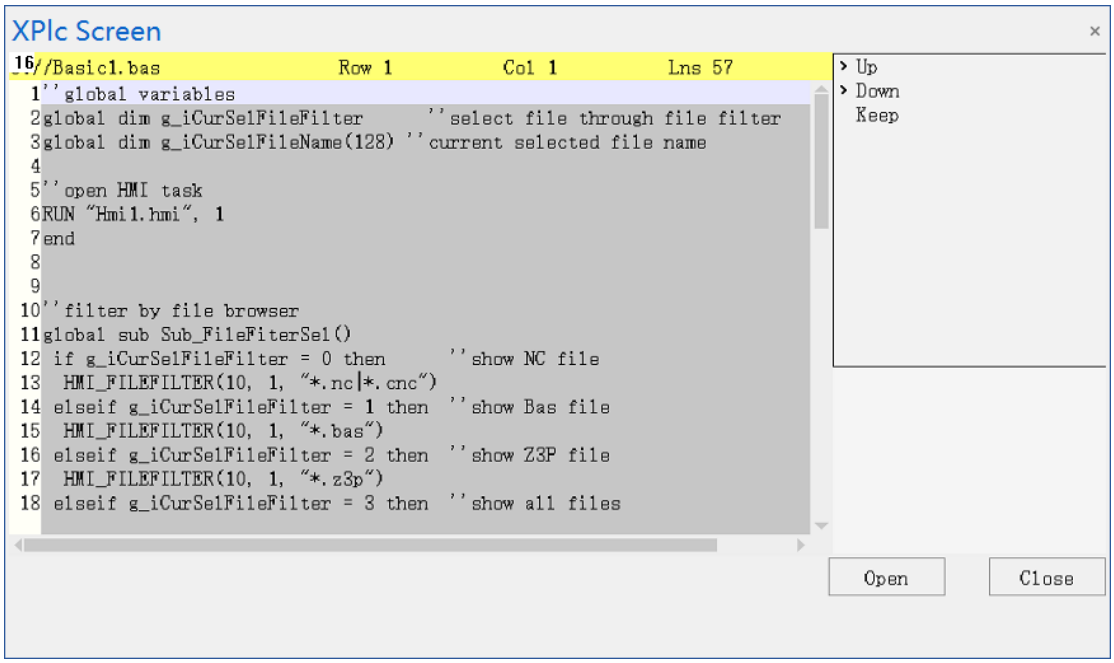
➢ Running effect:



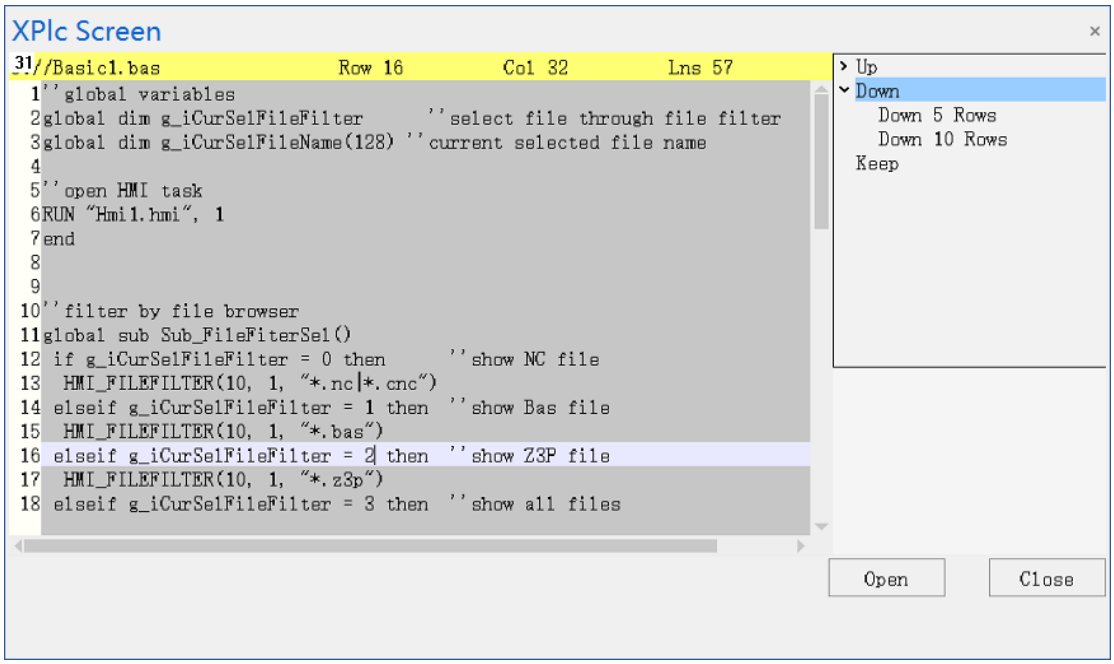
- Click “list” component, select needed file type, then in “file browser”, it will show corresponding file types’ files.

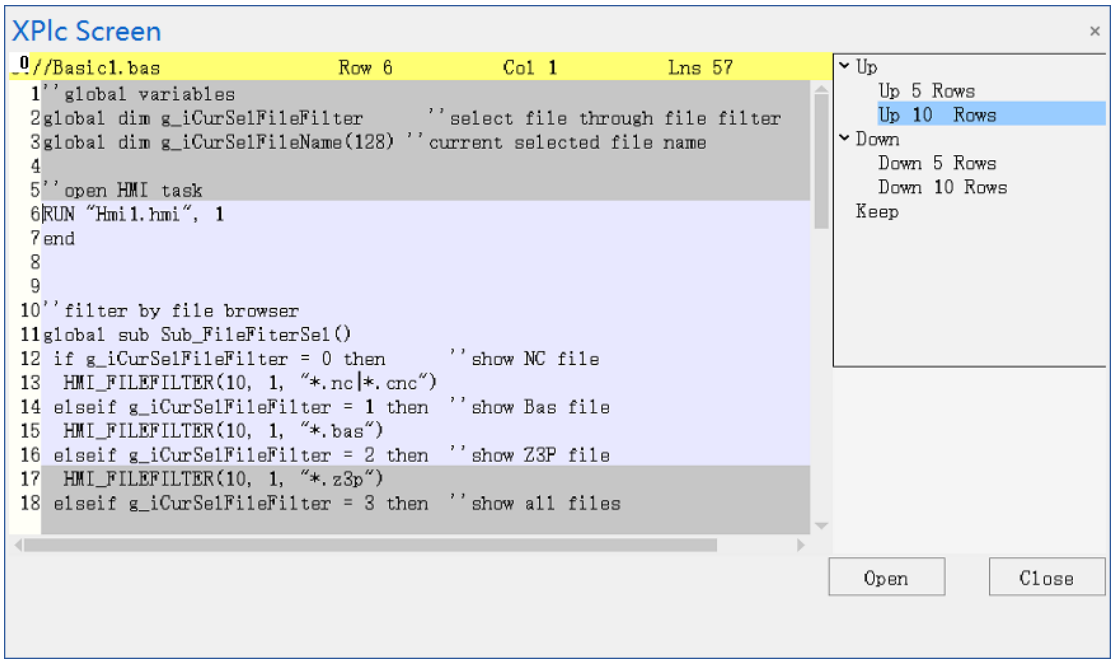


- Then, corresponding file3 editor interface appears, showing file content, you can edit it.

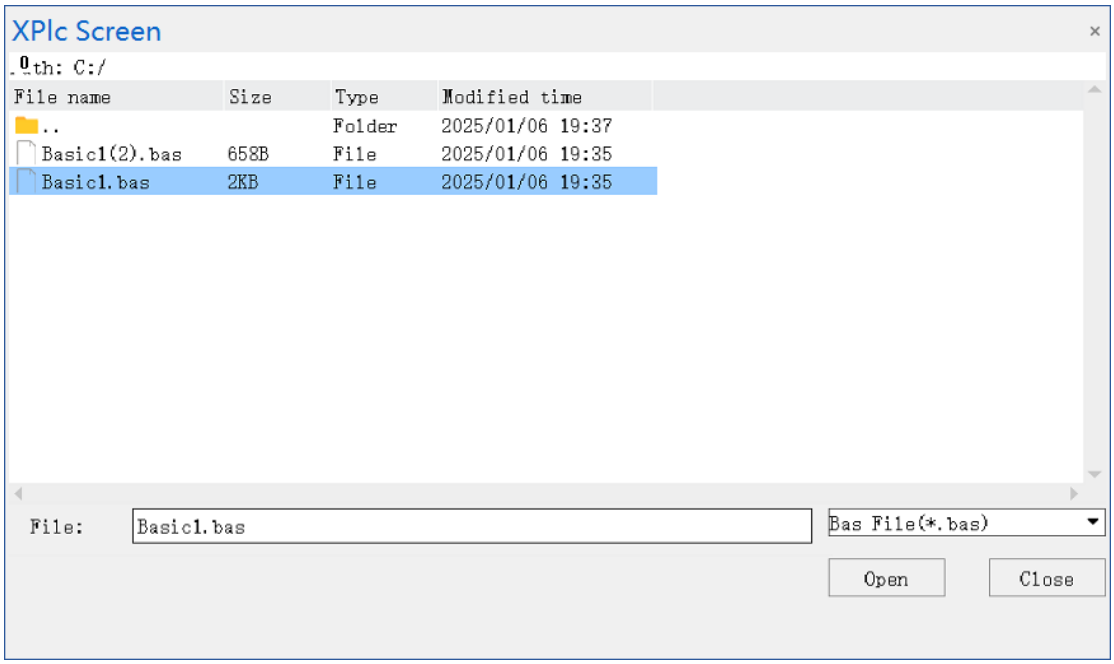


➢ Select any row, it will show tree all nodes. You can click to achieve corresponding actions





➢ Click “close”, file content will be closed, the interface will back to main file selection interface.



8.8. Get Routines

More HMI routines, please refer to Zmotion Website: www.zmotionglobal.com.

Exact Path: Zmotion – Download – 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		