

SD-TS04C/10C/10T
Programmable
Control Touch Panel
(UK 1-Gang)

SEADA
Showing the World

User Manual

VER 1.0

Thank you for purchasing this product

For optimum performance and safety, please read these instructions carefully before connecting, operating or adjusting this product. Please keep this manual for future reference.

Surge protection device recommended

This product contains sensitive electrical components that may be damaged by electrical spikes, surges, electric shock, lighting strikes, etc. Use of surge protection systems is highly recommended in order to protect and extend the life of your equipment.

Table of Contents

1. Introduction	1
2. IDE Functions Introduction	2
2.1 Menu	2
2.1.1 New Project	3
2.1.2 Open Project	3
2.1.3 Open Recently	3
2.1.4 Download Project	4
2.1.5 Export Project Package	4
2.1.6 Save As	5
2.1.7 Generate Authorization Code	5
2.1.8 System	7
2.1.9 Language	7
2.1.10 Exit	8
2.2 Toolbar	8
2.2.1 Save	8
2.2.2 Preview	8
2.2.3 Upload Project	9
2.2.4 Console	9
2.3 Project Configuration	9
2.4 Interface Design	10
2.4.1 Toolbar	11
2.4.2 Page	14
2.4.3 Component	17
2.4.4 Page Layout	38
2.4.5 Button Panel	46
3. Logic Configuration	49
3.1 Toolbar	49
3.2 Logic Module	51
3.2.1 UI	54
3.2.2 Panel	57
3.2.3 Command	60
3.2.4 Port	61
3.2.5 Network	63
3.2.6 Data Process	67
3.2.7 Extend	71
3.2.8 Timing	73
3.2.9 Advanced	74
3.2.10 System	79

Getting Started

Environment Setup

Software Environment

Control Studio Integrated Development Environment (hereinafter referred to as IDE) is an integrated tool for controlling project development. The software can be downloaded at:

<https://seada.co.uk/downloads/>

Tip: It is recommended to run the software on Win10 with 2G Memory or more. There are many function windows in the software. For a better experience, it is recommended to work on a computer with a screen resolution of 1920*1080 or above, and set the zoom ratio to 100%. If you adjust to other scaling, some windows of the software may not be displayed properly.

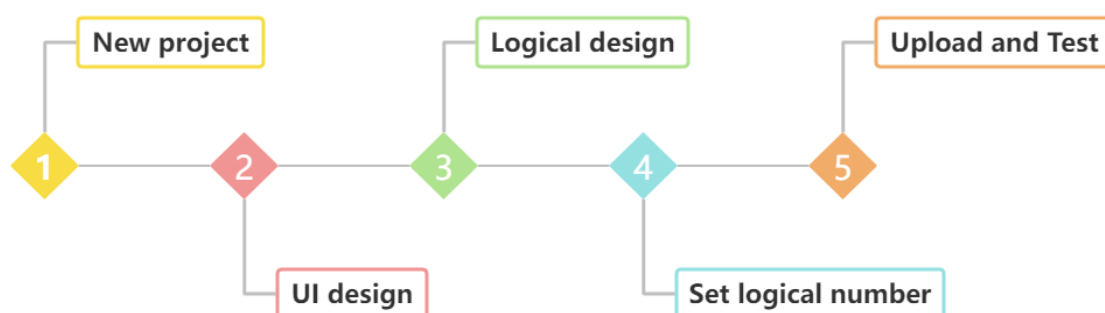
Central Controller and Connections

Taking the SD-TS04C central controller as an example, the default factory IP of the device is set to 192.168.0.101. When using it for the first time, it can be modified by entering the WEB configuration page through the central controller network settings or directly connecting the computer to the central controller.

After obtaining the IP address of the central controller, if you need to adjust the IP settings or other initialization settings of the central controller, you can visit the WEB configuration page of the central controller. For example, if the central controller IP is 192.168.0.101, the WEB access URL is <http://192.168.0.101/home>, and the initial password is admin.

Quick Start

Using Control Studio for control system development typically follows the following overall steps:



As shown in the figure, steps 2-5 may be modified repeatedly in the actual project development process, as follows:

1. **New Project:** Follow the instructions for [New Project](#) to establish your project.
2. **Interface Design:** **Use the interface designer to design the control interface. For details, please refer to [Interface Design](#).**
3. **Interface Design:** **Design logic modules on the logic canvas. For details, please refer to [Logic Configuration](#). Interface design and logic configuration can be adjusted by engineers.**
4. **Set Logical Number:** **Set action logical number and attribute logical number in the**

component or page attribute bar, and associate with logical modules such as UI interaction, setting UI attributes, and outputting UI attributes.

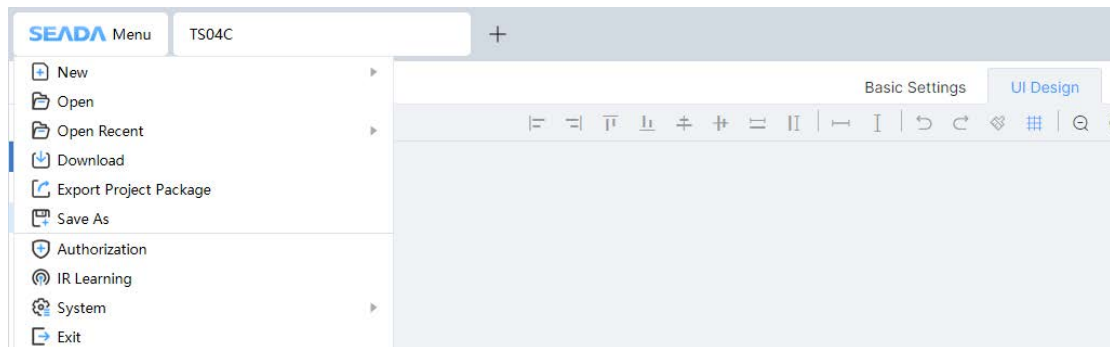
5. Upload and Test Project:** After the preliminary design of the project is completed, upload the project to the central controller (referred to as: controller) for testing and verification according to the instructions in [Upload Project](#).

IDE Functions Introduction

Engineers can quickly create user interfaces through the Control Studio integrated development environment (hereinafter referred to as IDE), without writing code, only through logic module wiring, you can connect controlled devices (referred to as peripherals). The IDE can also parse peripheral instructions through the relevant logic modules to ensure that the peripheral and component states are updated synchronously. In order to improve development efficiency, IDE supports copying and pasting UI components and logic modules across projects, greatly simplifying the project development process and achieving an efficient and convenient development experience.

Menu

The menu includes the basic management of the project, as shown in the figure below:



Specific functions include:

New Project;

Open Project;

Open Recently;

Download Project;

Export Project Packages;

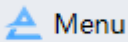
Save As;

Generate Authorization Code;

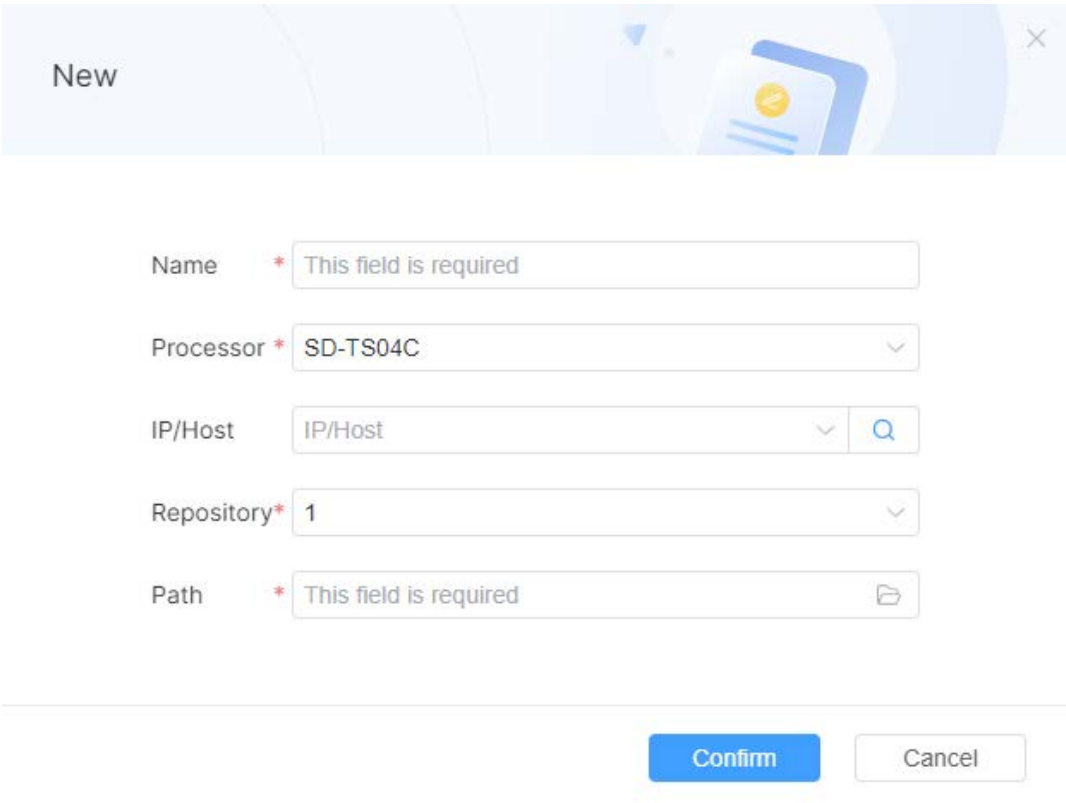
System;

Exit.

New Project

1. Click the icon , and select [New Project] from the drop-down menu.

2. Set the project-related information in the window opened by the IDE, as shown in the figure below:

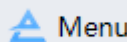


Attributes marked with red star are required. The path represents the local path where the new project will be stored.

Note: Up to 5 projects can be opened simultaneously. When the number of opened projects exceeds this limit, new projects cannot be created. You need to close one project before creating a new project.

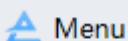
Open Project

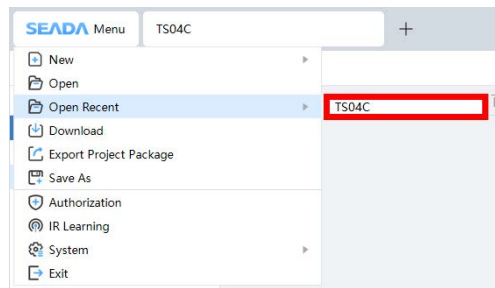
IDE has no default open projects. You can access recently opened projects by clicking on History in the initial screen. You can also select another project file by clicking on [Open Project].

1. Directly click [Open Project] in the initial interface, or you can click the icon , to select [Open Project] from the drop-down menu.

2. Find the file path of the saved project, select the file name in *.ea format in the project directory, and click Open.

Open Recently

Click the icon , and move the cursor to [Open Recently]. The IDE displays up to 5 recently used projects.



The red box shows a recently accessed project file. Clicking this line will open the corresponding project.

Download Project

Engineers can upload edited engineering files to the controller. If necessary, engineers can also download engineering files from the controller.

1. Click the icon  Menu, and select [Download Project] from the drop-down menu.

2. Click , IDE will search for accessible controllers within the current local area network.

After the search is complete, the drop-down box will display the IP address of the controller from which you can select. The IP address of the controller can also be entered manually.

3. Enter the controller access code.

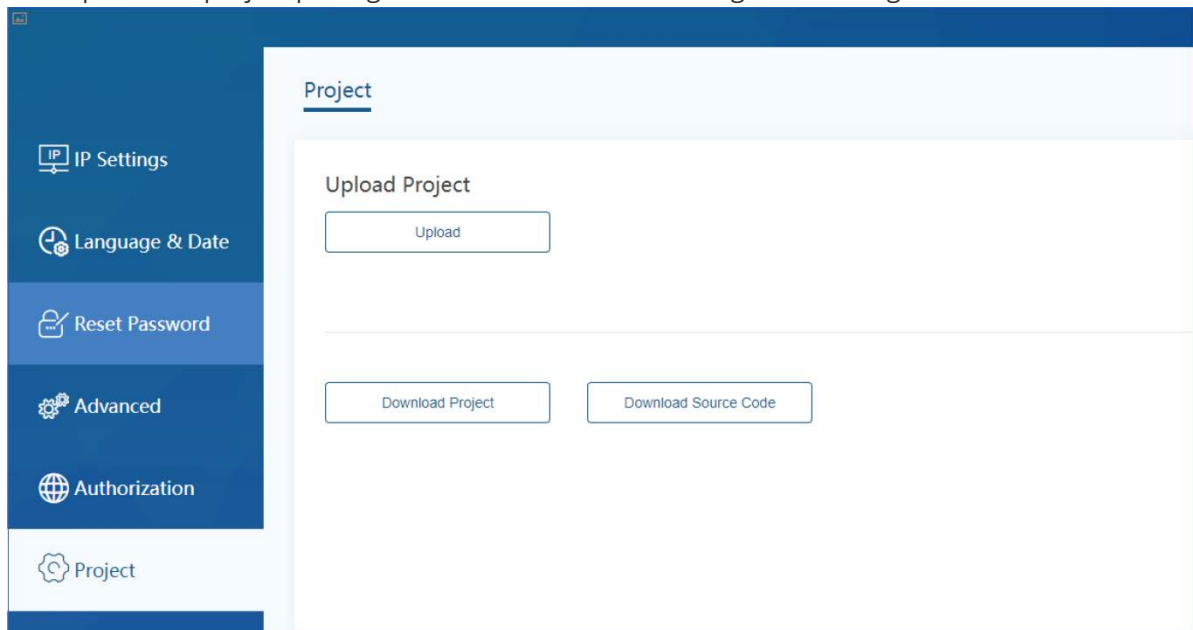
4. Specifies the location path of the project to be downloaded. Click [OK] to download. After downloading, the project file will be stored in the specified location in the form of .zip. compressed package. If [Auto Open] is checked, the compressed package will be decompressed to the selected directory and automatically opened.

 A screenshot of a 'Download' dialog box. The dialog has a title bar with the word 'Download' and a close button (X). Inside the dialog, there are three input fields: 'IP/Host' with a dropdown arrow and a search icon, containing the text '192.168.1.21'; 'Password' with a masked input (dots) and a toggle icon; and 'Path' with a folder icon, containing the text 'D:\Download'. Below these fields is an unchecked checkbox labeled 'Auto Open'. At the bottom right of the dialog are two buttons: 'Confirm' (blue) and 'Cancel' (white with a grey border).

Export Project Package

1. Click the icon  Menu, and select [Export] from the drop-down menu. Projects with incorrect entries will not be downloaded.

2. Select the path in the folder selection pop-up window, click[Select Folder]to export the project package, and the exported project package suffix is eap.
3. Upload the project package in the central control management background.



Save As

1. Click the icon  Menu, and select[Save As]from the drop-down menu.
2. Enter the project name to be saved as and the save path in the pop-up window, and click[OK]to save.

Generate Authorization Code

1. Click the icon  Menu, and select[Generate Authorization Code]from the drop-down menu.
2. Select Authorization Module in the pop-up window. You do not need to select an authorization expiration date when you select permanent authorization; you need to select an authorization expiration date when you select limited authorization.
3. Enter the MAC address of the controller to be authorized, and click[Start Generation]to generate the authorization code.

Authorization

Mode

☐ Perpetual ☒ Limited

MAC

*

Expiration

*

Generate

Cancel

4. When the authorization expires, enter: `http:// controller IP/unlock` in the APP interface or browser to enter the authorization code.

System Authorization

Please enter or paste your authorization code

Authorization has expired

XXXX

—

XXXX

—

XXXX

—

XXXX

XXXX

—

XXXX

—

XXXX

—

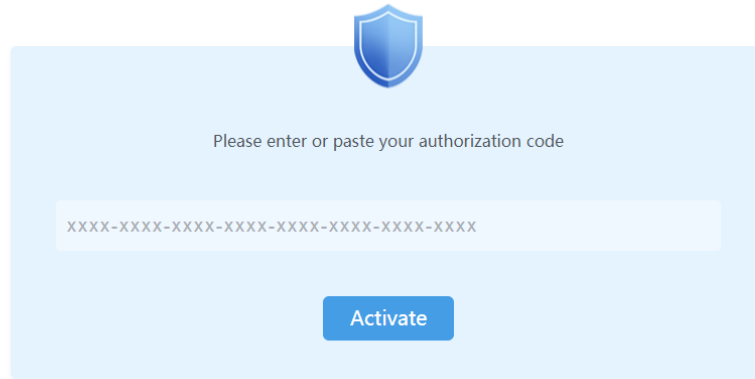
XXXX

If you don't have an authorization code, please contact your service engineer or customer support personnel.

Activate

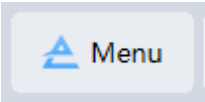
Authorization code

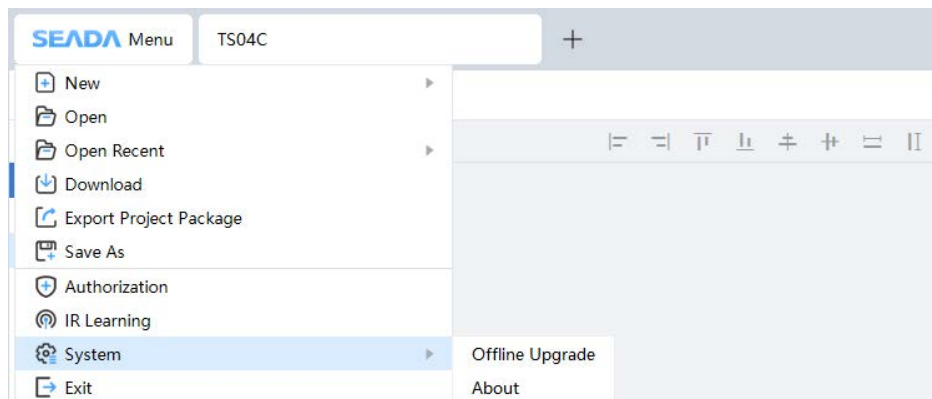
Please use the authorization code for activation. If you don't have an authorization code, please contact your designated service engineer or customer support representative for further assistance and guidance.



The form features a blue shield icon at the top center. Below it, the text "Please enter or paste your authorization code" is displayed. A text input field contains the placeholder "XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX". At the bottom center, there is a blue button labeled "Activate".

System

Click the icon , and move the cursor to [System] to display the content related to [System].



The system mainly includes three parts: [Offline Upgrade], [About Control Studio].

[Offline Upgrade] is used to select higher version offline package to upgrade IDE; [About Control Studio] displays IDE registration information

Exit

Click the icon  in the upper right corner of the project tab to save and close the corresponding project.

Click the icon  in the upper right corner of the IDE to save and exit the IDE.

Click the icon  **Menu**, and select [Exit] from the drop-down menu to save and exit the IDE.

Toolbar

It contains functions such as saving and uploading projects, as shown in the figure:



Specific functions include:

Save;

Preview;

Upload Project;

Console.

Save

Clicking the icon  in the toolbar will save all the contents of the project currently in the active state.

During saving, if there are warning items and error items, it can also be saved normally. You can view warning items and error items in the problem list.

Preview

Click the icon  in the toolbar to preview the design interface opened by the current project.

Preview the page or pop-up window opened by the current project. Subpages cannot be previewed directly, only through the Subpage Container component. It supports component interface operations, such as buttons, switches, sliders and other interface operations. Communication with peripherals is not supported.

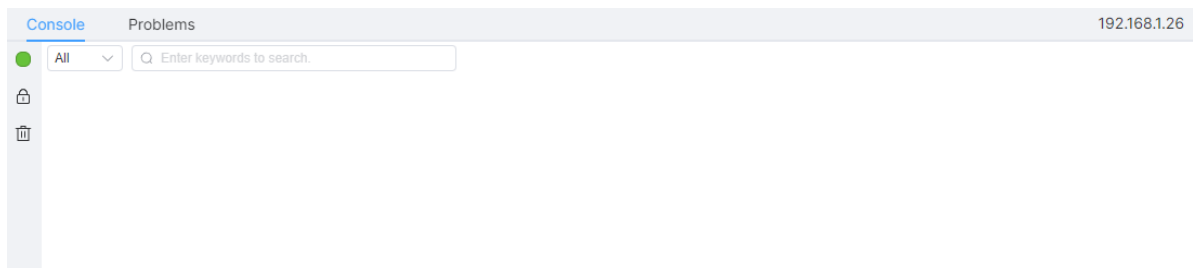
Upload Project

Click the icon  in the toolbar to upload the current open project. Before uploading, the IDE saves the project. Please refer to [Save](#). If there are no error items, IDE will upload the current compilation package to the controller according to the project configuration information, and the relevant printing information will be output to the [Console] window of IDE console when the project is running; users can connect to the central control screen or user APP to operate, and engineers can download the project code through the [Download Project] function of IDE.

Console

When uploading a project, the system automatically opens the console. You can also switch manually by clicking on the toolbar icon .

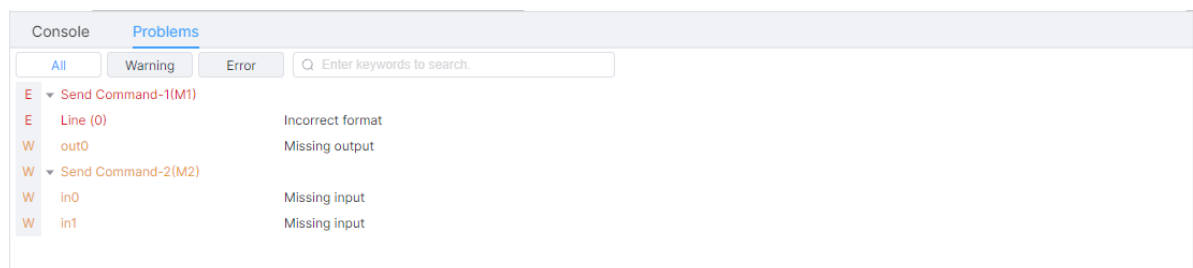
After opening the console, the console is displayed under the IDE and automatically connected to the controller configured by the project, as shown in the figure:



Log area outputs necessary information in real time, such as project upload, load log, and logic module sending and receiving data during operation. Error messages are shown in red, warning messages in yellow, and normal messages in black.

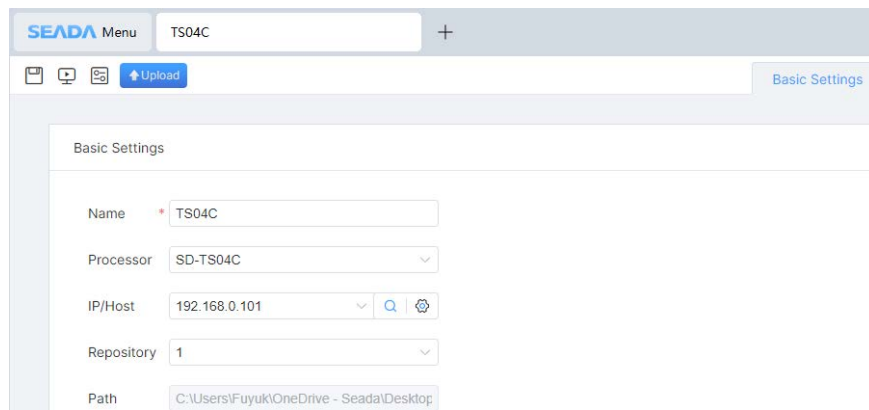
The problem list displays error items and warning items in logical configuration in turn. You cannot upload project and export project package when project has error item.

Double-click the error or warning prompt, and the IDE will jump to the relevant logic module to help engineers quickly locate the error.



Project Configuration

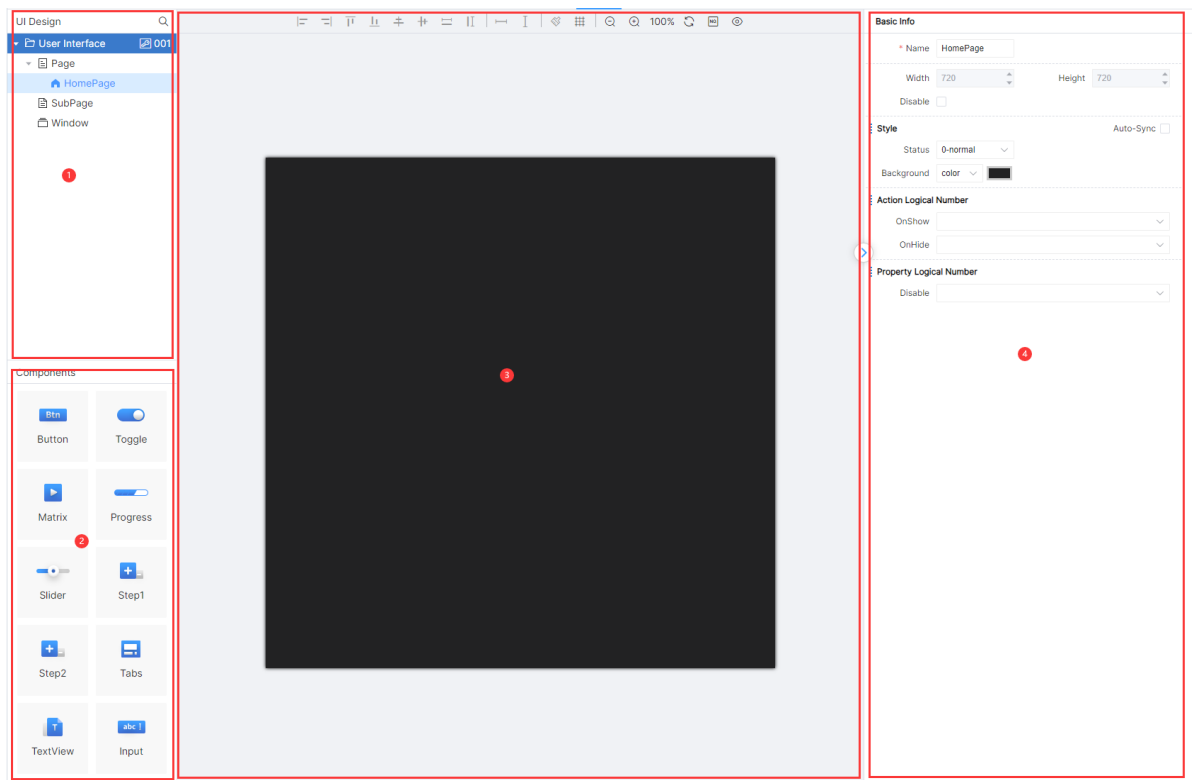
Project name, device model, IP/Host and path are displayed in project configuration. The project name, device model and path cannot be modified.



IP/Host can be entered, or you can select the IP searched in the drop-down box. Click  to enter the controller background management page corresponding to the currently selected IP/Host.

Interface Design

The interface design is divided into regions for functional description, as shown in the figure:



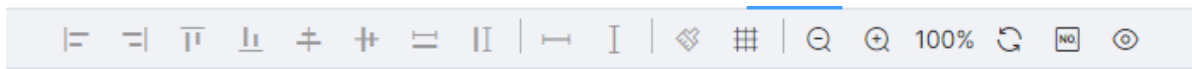
①: [User Interface]area; ②: [Component]area; ③: Interface design area; ④: Interface/Component attribute setting area.

UI Designer is the core module of interface design, including the addition of interfaces and components, style design, interface layout and other functions. This chapter will introduce them one by one.

The software provides WYSIWYG interface designer, real-time rendering, flexible property settings and layout functions. After the user creates a normal page, sub-page and pop-up window, he/she can drag the required UI component to the page, and adjust the display effect by modifying the parameters of the UI component. By setting action logic number and attribute logic number, it is associated with UI related logic module.

Toolbar

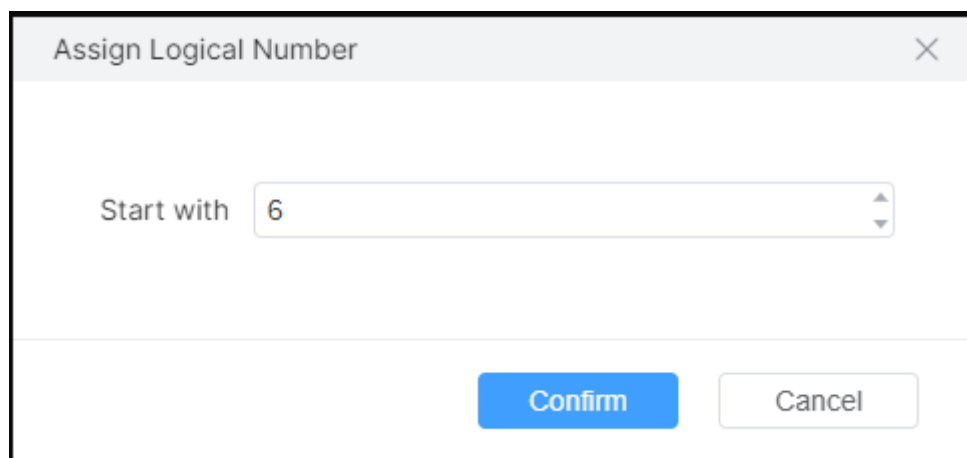
In the window of [Interface Design], the toolbar mainly contains function icons such as alignment, format brush, grid, etc., as shown in the figure:



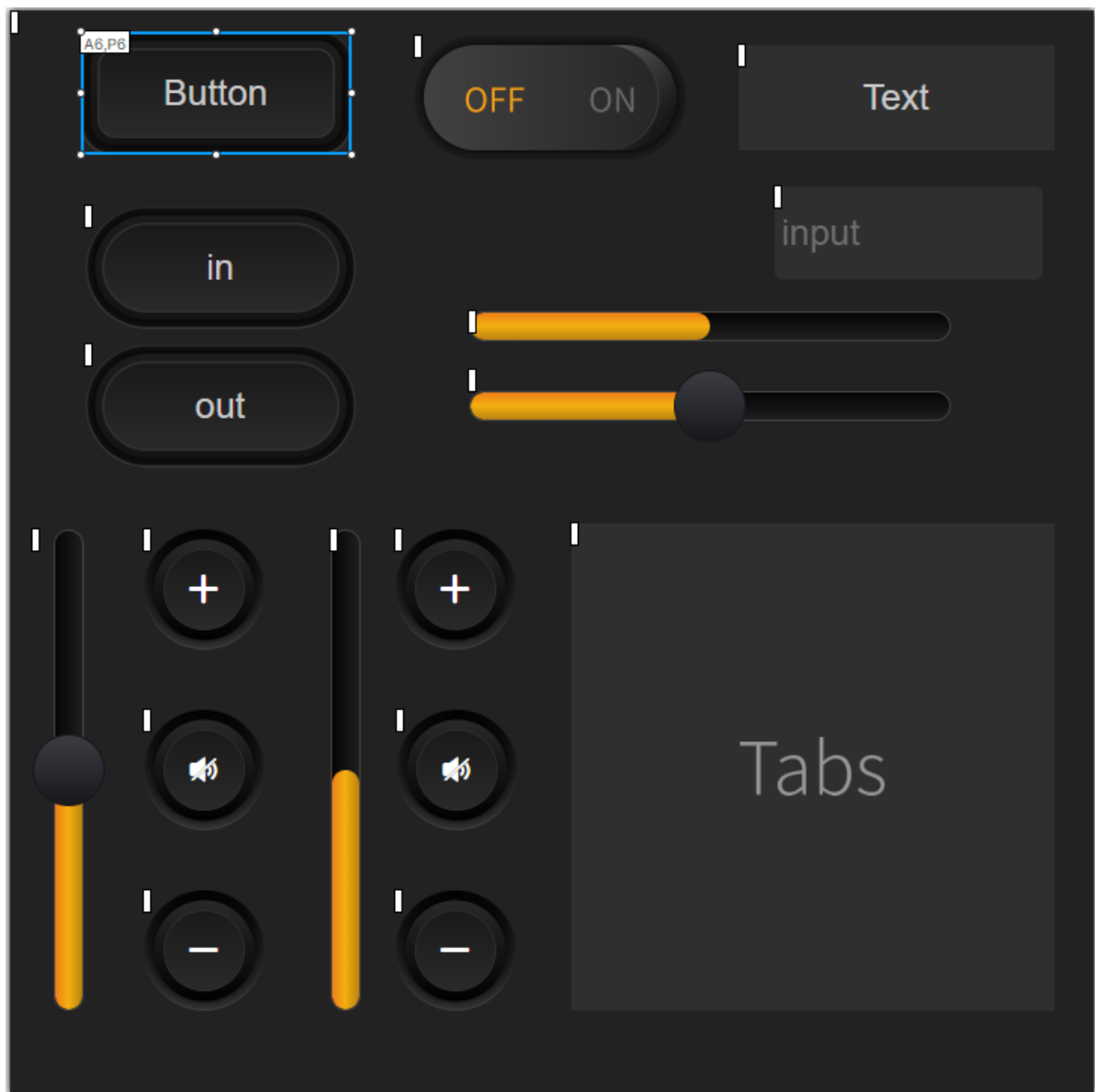
Component alignment (refer to [\[Quick Alignment\]](#)), spacing adjustment, formatting brush, grid and zoom in and out can assist engineers in rapid layout and typesetting of components in the interface.

Fast Allocation

Click the fast allocation icon  in the toolbar to enter the batch allocation logical number mode. In the allocation logical number pop-up window, the starting number defaults to the current maximum action logical number or attribute logical number +1. For example, if the current maximum number is 5, the starting number in the pop-up window is 6. As shown in the figure:



Click on the corresponding component to complete the logical number assignment.



Display Logical Number

Click the icon  in the toolbar to enter the display logic number mode. The upper left corner of the component will display the assigned action logic number and attribute logic number. As shown in the figure:

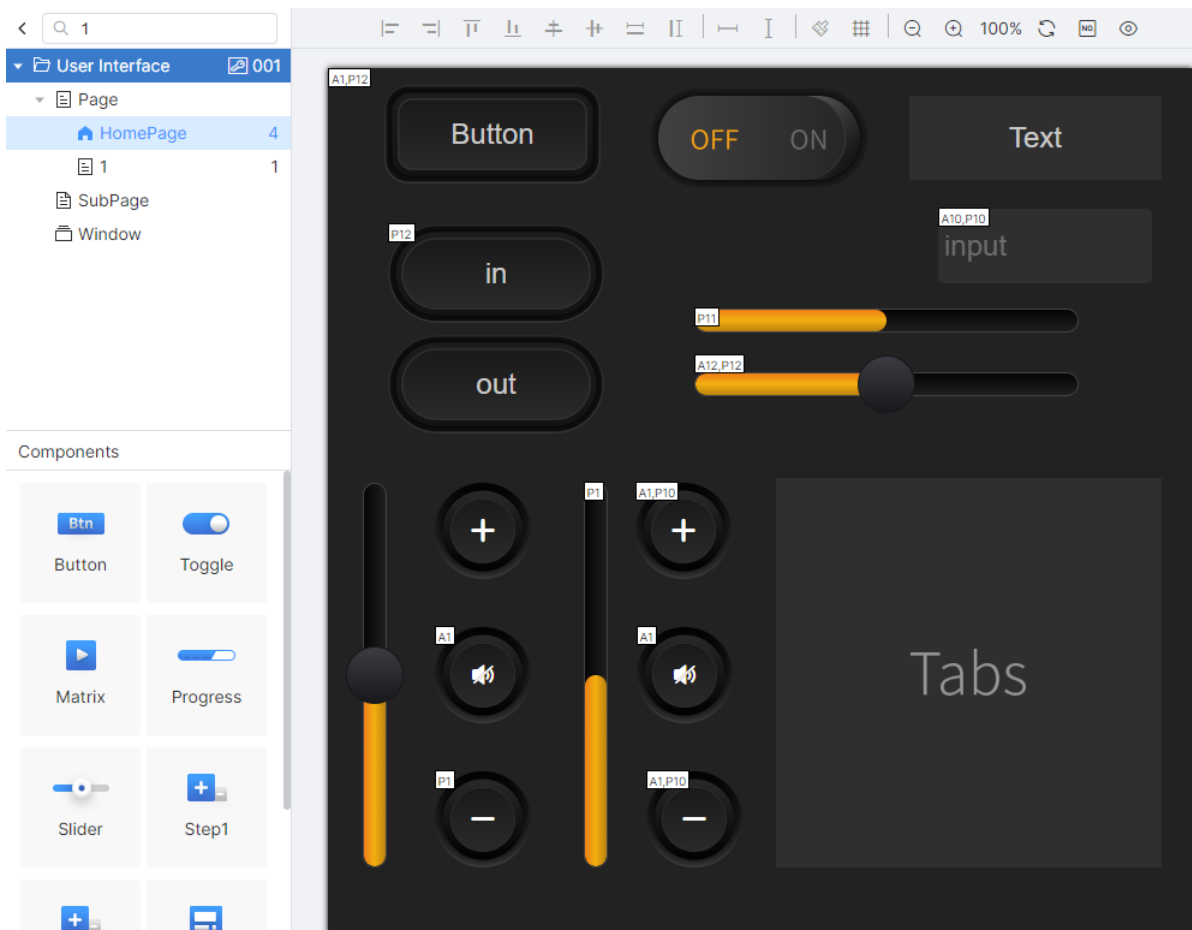


Click the icon  again to exit the display mode.

Search Logical Number

Engineers can search the logical number to find the page where the corresponding component is located. The specific operation steps are as follows:

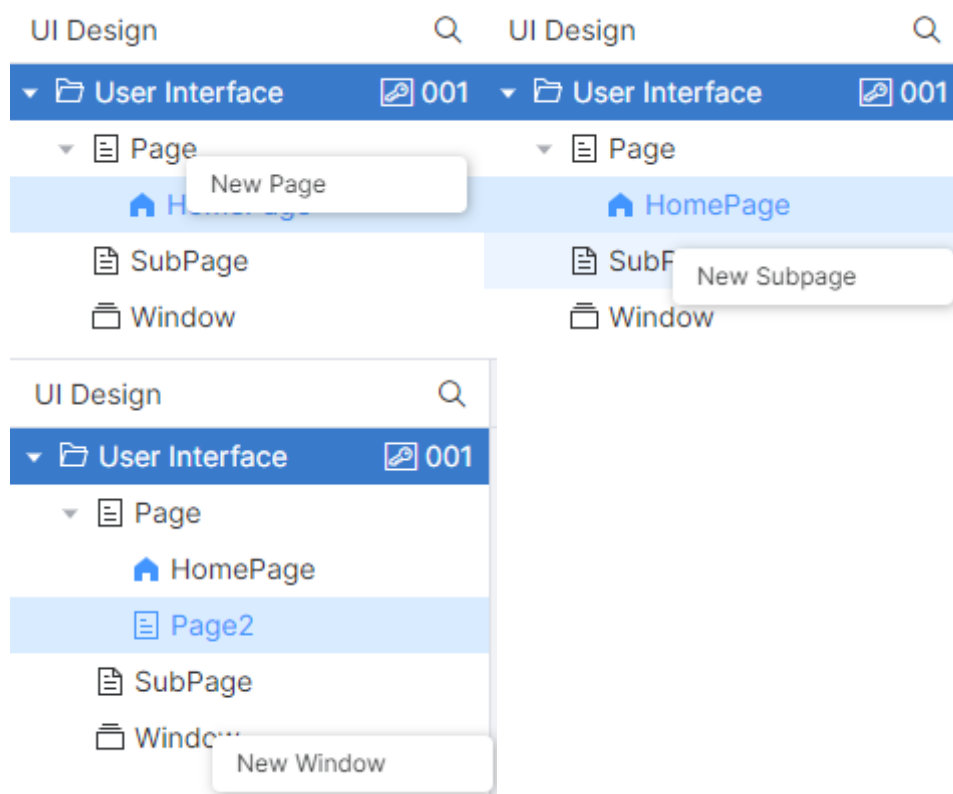
Click the search icon , and enter the logical number of the component or page you want to find in the search box to search. This search method uses fuzzy matching mechanism.



The page node will display the matching quantity of the corresponding page, and the upper left corner of the matched component in the page will display the corresponding action logical number and attribute logical number. Clear the search box, then automatically exit the search.

Page

Right-click the corresponding tab of [Interface Design] to open the menu. You can select [New Page],[New Subpage],[New Pop-up Window], as shown in the figures below:



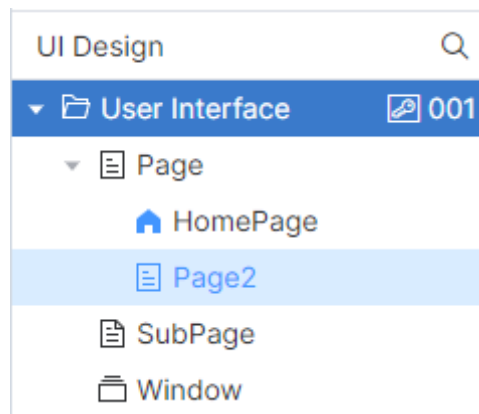
Pages are divided into three types: normal pages, sub-pages and pop-up windows. IDE will put different types of pages under the corresponding page nodes. Among them, the size of ordinary pages is determined by the model of controller configured by engineers. For example, after TS04C is selected as configuration in project, the size of all ordinary pages is 480*480; the size of sub-pages and pop-up windows needs to be defined by engineers.

Please note that subpages are different from normal pages/pop-up windows, and can only be displayed through subpage containers. Subpages can be embedded into two other types of pages through subpage containers.

The jump function of the page will be described in [Button](#).

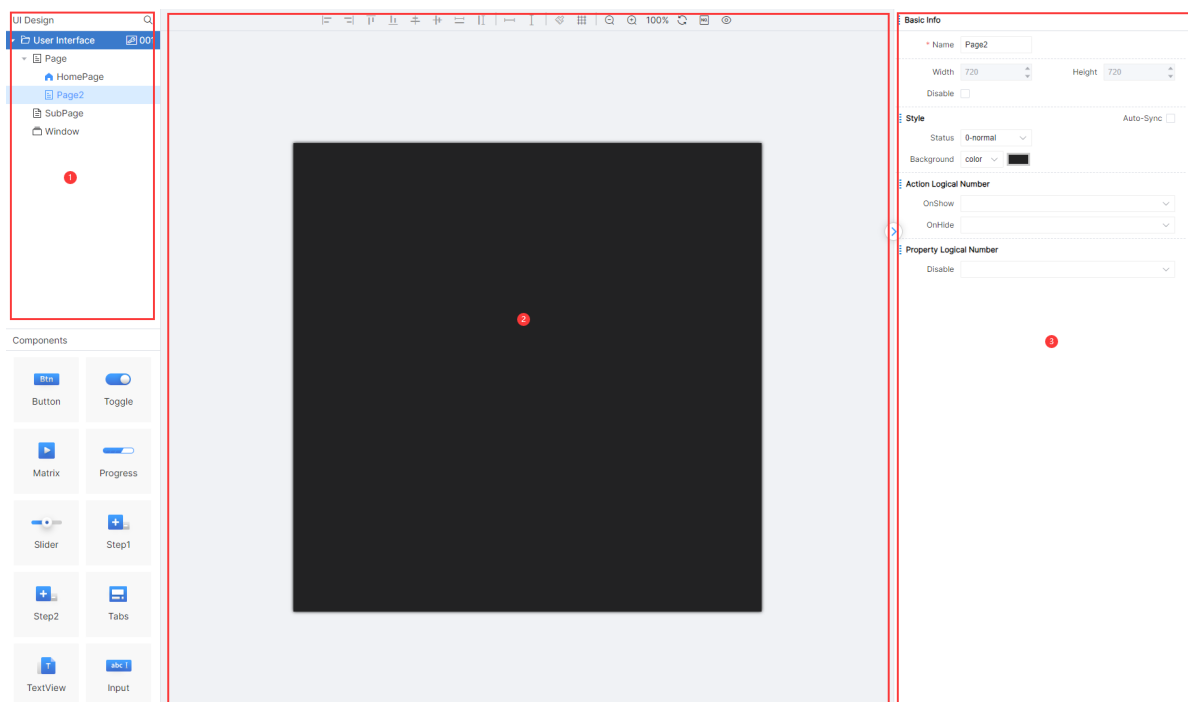
Page

Click[New Page], , enter the page name in the pop-up window of New Page, and click[OK]to create it. The new page is saved under the[Page]node of[User Interface]. As shown in the figure:



When creating a project, IDE will automatically create the home page. If you want to set other pages as the home page, right-click on the page and select [Set as Home Page]to set the current page as the home page. Only normal pages can be set as the home page, and unique.

Double-click a page, the IDE interface design area displays the corresponding page, and the right side displays the properties of the page. As shown in the figure:

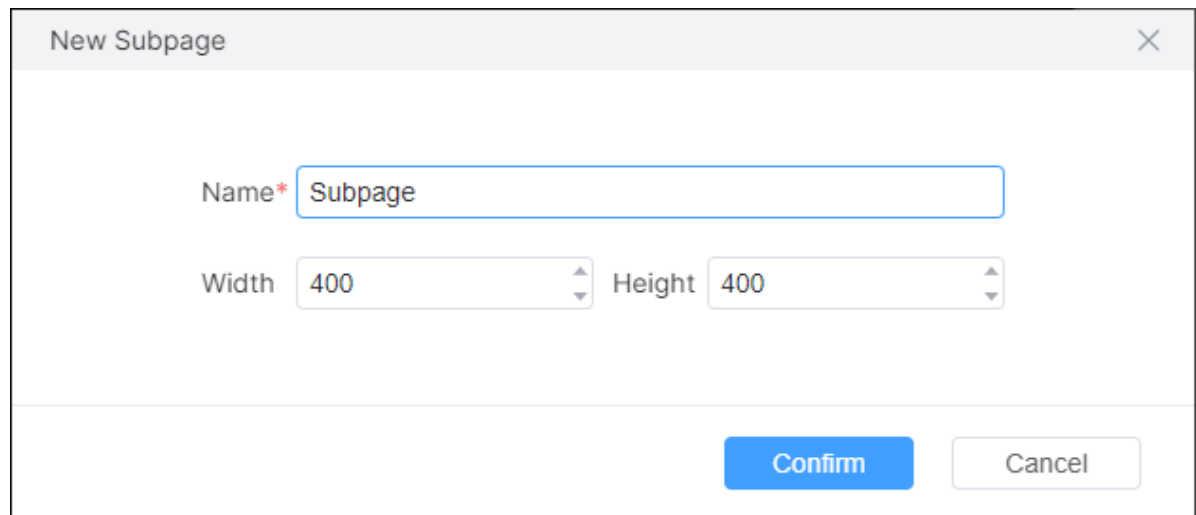


Double-click the page in ①, perform page layout operation in ② area, and perform relevant attribute setting in ③ area. Page attributes include name (required), width (read-only), height (read-only), disable, style, action logical number, attribute logical number. The names of normal pages, subpages and pop-up windows cannot be repeated, and the names of components can be repeated.

Note: If the engineer selects the component in the current page, the attributes of the selected component are displayed in the area ③. To edit the page attributes, click the page area or blank space.

Subpage

Right-click the page navigation area and click [New Subpage] in the menu. The IDE will pop up a new subpage pop-up window:

A screenshot of a 'New Subpage' dialog box. The dialog has a title bar with the text 'New Subpage' and a close button (X) on the right. Inside the dialog, there is a 'Name*' label followed by a text input field containing the word 'Subpage'. Below this, there are two labels: 'Width' and 'Height'. The 'Width' label is followed by a numeric input field containing '400'. The 'Height' label is followed by a numeric input field containing '400'. At the bottom right of the dialog, there are two buttons: 'Confirm' (in blue) and 'Cancel' (in light gray).

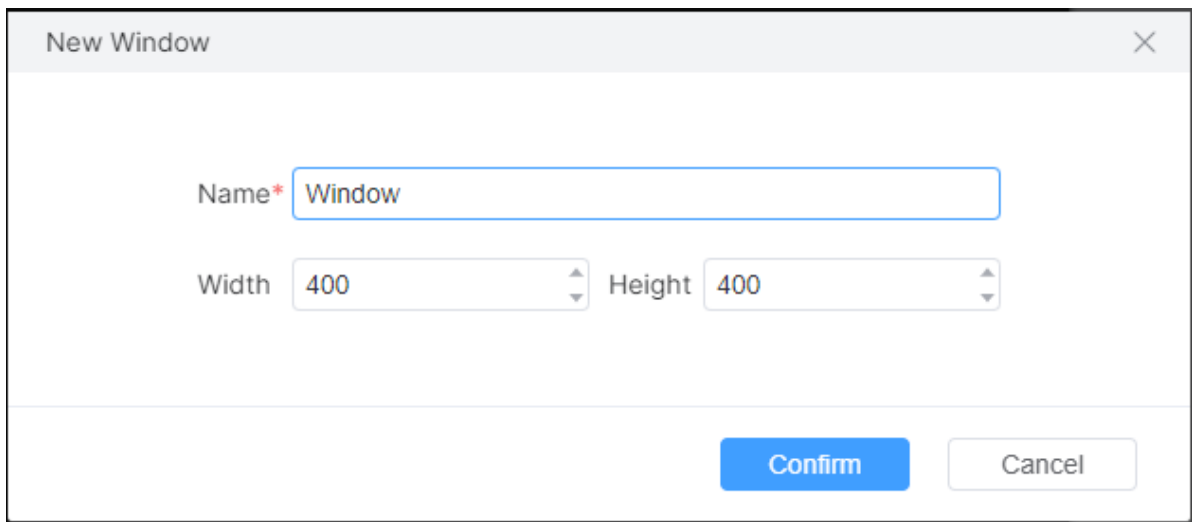
Enter the name, width and height and click [OK] to find the subpage under the [Subpage]node in the page navigation area. At this time, relevant configurations can be modified in the attribute editing area, and the attributes are editable, which is the same as that of ordinary pages.

During page preview and project operation, subpages need to be displayed through subpage containers, that is, different subpages can be switched and displayed by controlling the display interface properties of subpage containers.

Note: The size of the subpage needs to be consistent with the size of the container area in which it is displayed, otherwise scroll bars may appear to affect the visual experience or only part of the content may be displayed.

Pop-up Window

Pop-up windows are usually used primarily for informational purposes. Right-click the navigation area of the page and click [New Pop-up Window] in the menu. The IDE will pop up a new pop-up window:

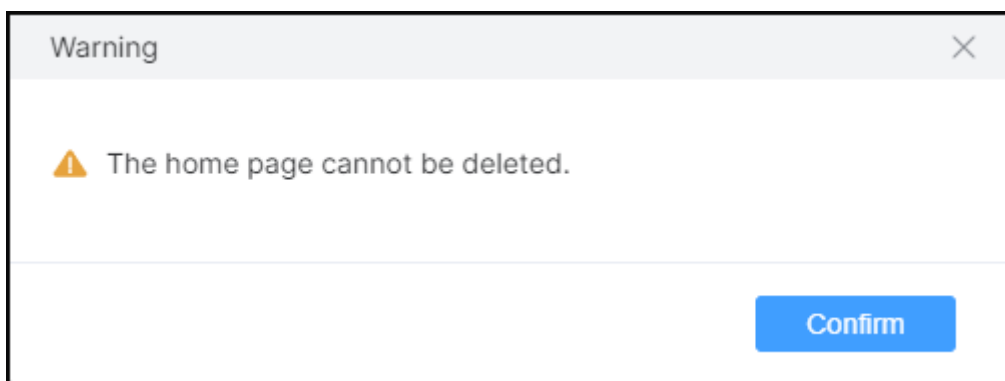
A dialog box titled "New Window" with a close button (X) in the top right corner. It contains two input fields: "Name*" with the value "Window" and "Width" with the value "400". Next to "Width" is a "Height" field with the value "400". At the bottom right, there are two buttons: "Confirm" (blue) and "Cancel" (white with a grey border).

Currently IDE does not support pop-up position settings, the default pop-up position in the center of the screen. Its properties are set the same as normal pages.

Delete pages

Click the right mouse button on the page in the page navigation area and select [Delete] from the menu to delete the page, and the deletion operation cannot be undone.

Note: The home page cannot be deleted.



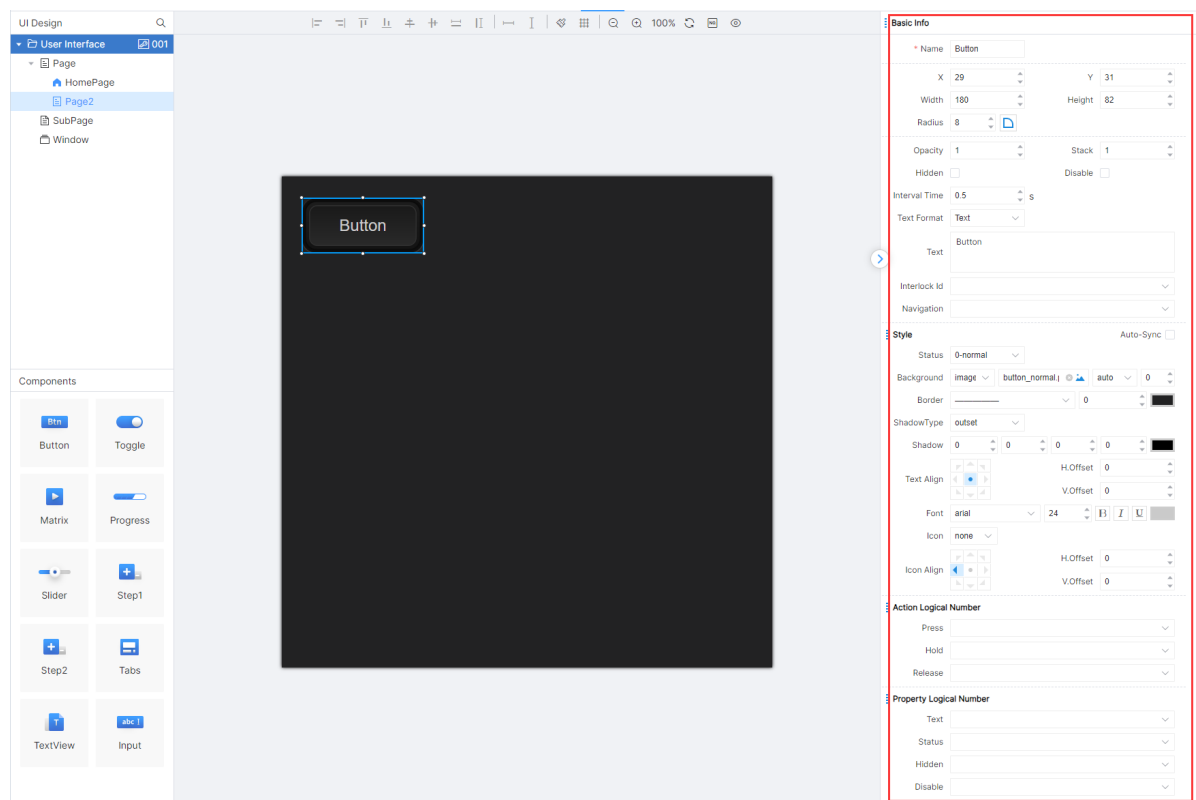
Component

UI component is an important part of user interface design. The attribute settings of pages and components determine the layout and style of UI interface. Combined with the action logic number and attribute logic number of components, engineers can design personalized user interface and interaction.

Component contains buttons, switches, matrix, progress bar, slider bar, step 1, step 2, subpage container, text, input box.

Component Property settings

Click any component (including the page itself) on the page, and the IDE will display the selected border on the corresponding component, and display the detailed properties of the component on the right side, as shown in the red box area below:



Basic information, style, action logic number and attribute logic number are displayed in the component attribute setting area.

The engineer can click  and  buttons to hide and restore the attribute setting area respectively when designing the page layout.

Common Property Settings

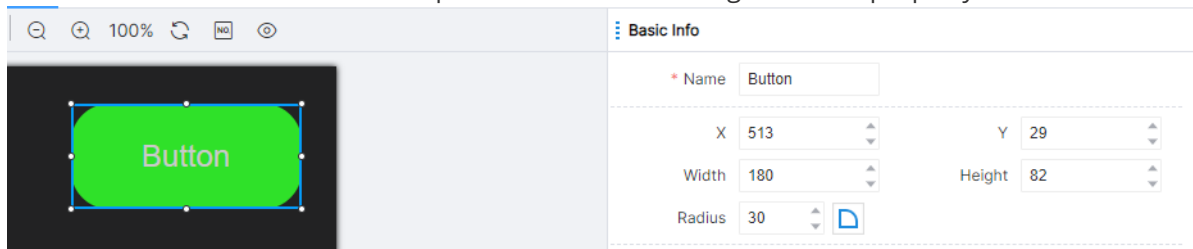
UI component properties vary according to component type. But each component has a set of common attributes, also known as common properties, whose names and meanings are exactly the same. Details of basic information are shown in the table below:

Property Name	Description
Name	Required, default to the name of the component type
X	Distance between the top left corner and the left edge of the component
Y	Distance between the top left corner and the upper edge of the component
Width	The width of the component occupying the interface
Height	The height of the component occupying the interface
Rounded Corner	Refer to rounded corner related settings for details
Transparency	Transparency of components in interface
Level	Display hierarchy of components in the interface
Hide	Component visible by default, check this attribute to be invisible
Forbid	The component takes effect by default, but checking this property will not take effect.
Interval Time	The time to trigger a long press action every interval after a button/step component is pressed for 1 second
Text Format	Default is text, there are text, time, date options
Text	Text displayed on component
Interlock Number	When multiple buttons are set to the same interlock number, only one button is active at the same time
Page Skip	When the button is pressed, the page jump is executed. You can choose to jump to the page, pop up the window or control the subpage container to switch subpages.
Text-On	Text displayed by switch component in On state
Text-Off	Text displayed by switch component in Off state
Group Number	The group number of the matrix group according to which the matrix group is controlled. Read-only
Encoding Type	Encoding type of matrix group, string or Hex, read-only
Type	Type of matrix, input or output, read-only
Communication Code	Set communication code according to code type, used for matrix control
Direction	Progress bar or slider direction settings
Default	Initial value of progress bar/slider/switch
Maximum	Maximum value of progress bar or slider

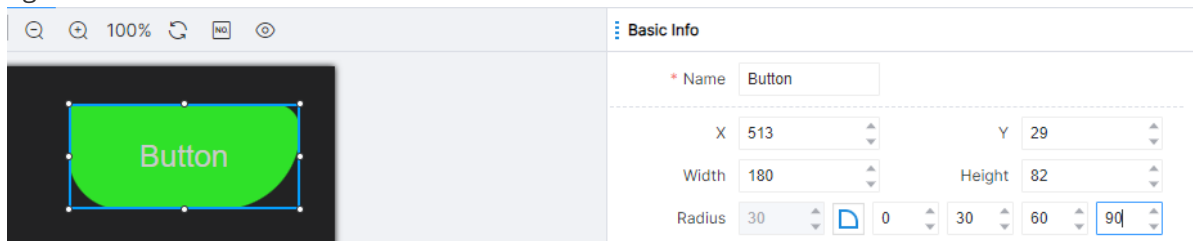
Property Name	Description
Minimum	Minimum value of progress bar or slider
Stepping	Stepping component changes value per click
Default Page	Subpage container default display subpage
Input Type	Input box default input type is text, there are text, password, text field options
Hint Text	Prompt when input box has no input content

Fillet Settings

The radius at the corner of the component can be set through the fillet property.



By default, when you set the value of this property, each corner of the component appears with a radius corresponding to the set value. Click on the sector icon to the right of this property, and four edit boxes will appear, indicating that you can set the value of one or several fillet radii individually. The four rounded radii of the button can be set independently, as shown in the figure:



Note: After setting the overall value of the fillet (left side of the sector icon), click on the sector icon to expand the radius value edit box set separately, and set the radius of each angle. At this time, the value set separately shall prevail, and the overall value shall not take effect.

Style Settings

A component can change the style it appears in the interface by adjusting the value of the style property. Components change styles logically in some scenarios. For example, the button component states are normal, activated and disabled, as shown in the figure:

Style

Auto-Sync ☐

Status

0-normal

Background

color

Border

0

ShadowType

outset

Shadow

0

0

0

0

Text Align

H.Offset

0

V.Offset

0

Font

arial

24

B

I

U

Icon

none

Icon Align

H.Offset

0

V.Offset

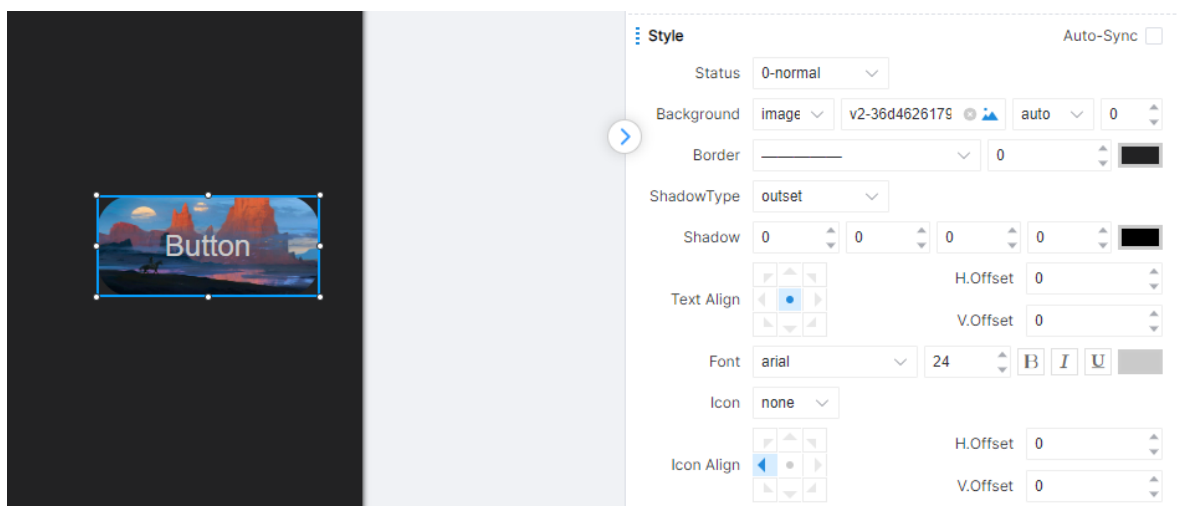
0

In each state, the component has a separate set of style settings. For different states, engineers need to set up each style separately.

Common style properties are as follows:

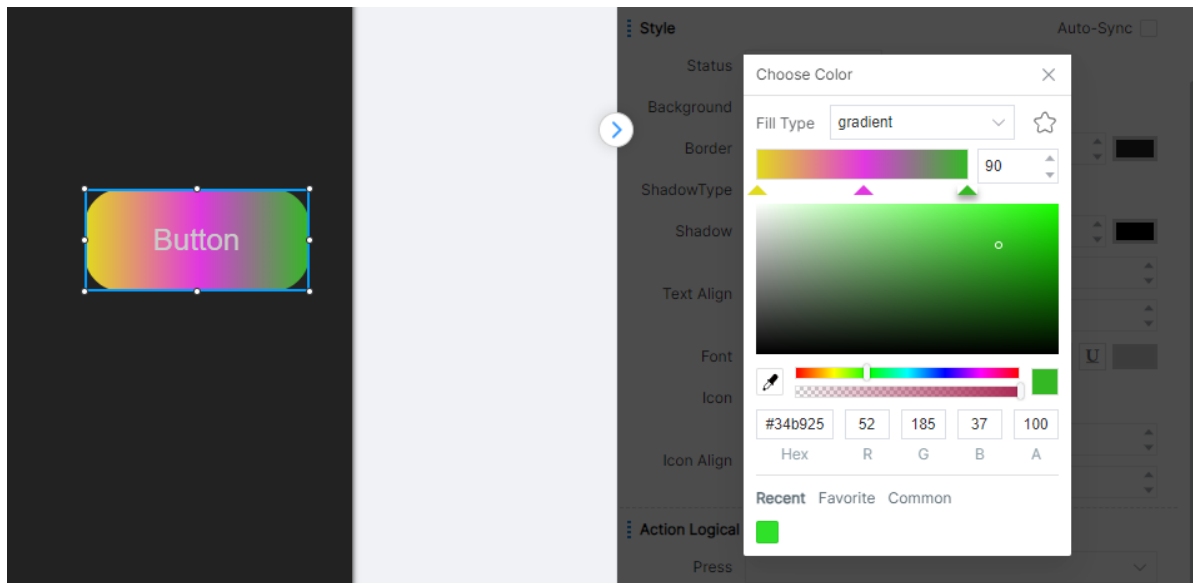
Background

The background property can be used to modify the background of a component. In this property, engineers can select a local or online image as the background image for the component:



In terms of fill rules for image property, the IDE offers three options: Auto, Hold, and Repeat. Auto means adaptive, that is, the image will be scaled according to the length and width of the component regardless of its size; Hold, that is, the image keeps its own size. If the size of the component does not match the size of the background image, only part of the image or blank space will be left on the background of the component; Repeat means that when the image is small and its size is not enough to occupy the background of the component, the purpose of occupying space is achieved by repeating its own image. At this time, the background of the component is formed by repeatedly splicing the image.

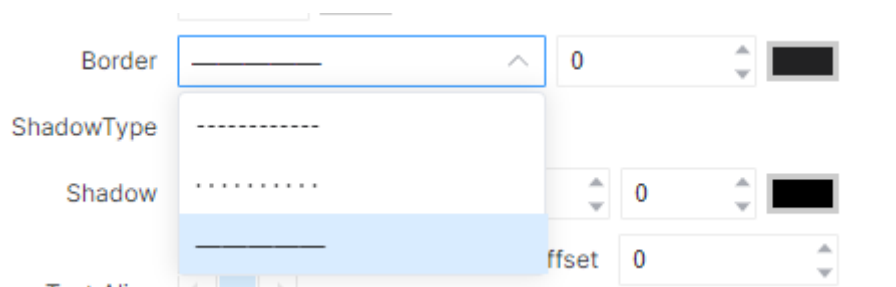
You can also set the background to color, click the color icon on the right, click the open palette to set the background color of the component:



The color palette allows you to set the background of components to solid or gradient colors, which can be found in the **Color** section of this section.

Border

Border properties can be used to modify the style of component border lines, including the types of border lines (dotted line, dot, solid line), width, and color of border lines.



Shadow

Shadow can simulate projection effect for components in actual scenes. Shadow types are divided into outer shadow and inner shadow. Shadow parameters are mainly set from four aspects: position, size, blur and color.

Style

Auto-Sync ☐

Status

0-normal

Background

color

Border

0

ShadowType

outset

Shadow

0

0

0

0

Text Align

H.Offset

0

V.Offset

0

Font

arial

24

B

I

U

Icon

none

Icon Align

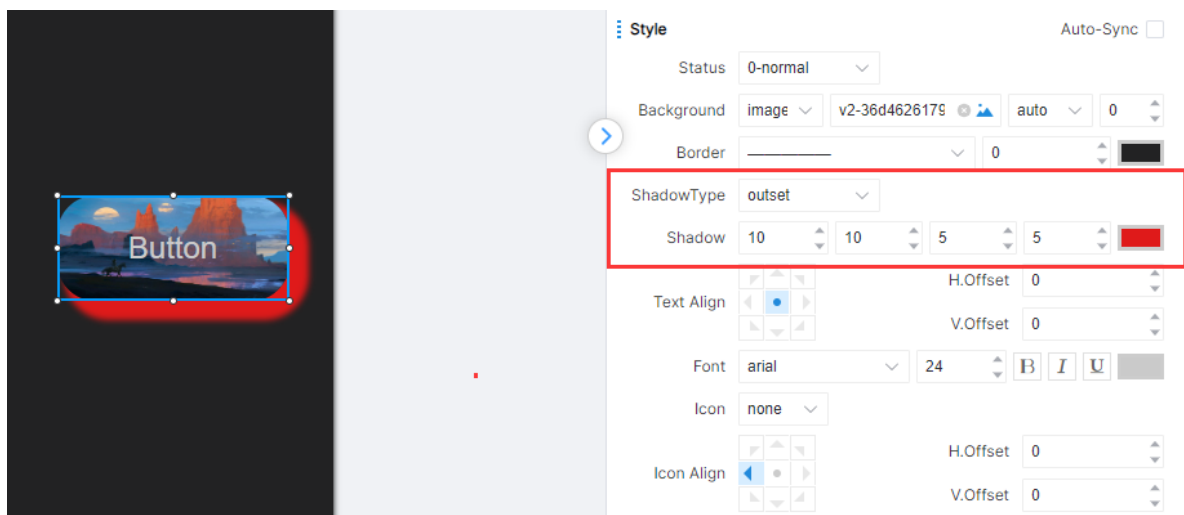
H.Offset

0

V.Offset

0

The first and second edit boxes represent the X and Y wheelbases, respectively. The X wheelbase indicates the horizontal distance of the shadow from the component, with the shadow being positive on the right side and negative on the left; the Y wheelbase indicates the vertical distance of the shadow from the component, with the shadow being positive on the lower side and negative on the upper side. The third edit box indicates the blur level, with lower values indicating sharper shadows and higher values indicating more blurred shadows. The fourth edit box represents the shadow size. A value of 0 means that the shadow is consistent with the original component width and height. If you want to increase the width and height of the shadow, you can increase this value. As shown in the figure:



Text Alignment

Text alignment adjusts the alignment of text within a component within the component area, as shown in the figure:

Style

Auto-Sync ☐

Status

0-normal

Background

image

v2-36d4626179

auto

0

Border

0

ShadowType

outset

Shadow

10

10

5

5

Text Align

H.Offset

0

V.Offset

0

Font

arial

24

B

I

U

Icon

none

Icon Align

H.Offset

0

V.Offset

0

Once the alignment properties are set, you can adjust the position of the text by adjusting the horizontal offset and vertical offset so that the content can be placed anywhere within the component.

Font

For text that appears on various components, you can style it through font properties, as shown in the figure:

Style

Auto-Sync ☐

Status

0-normal

Background

image

v2-36d4626179

auto

0

Border

0

ShadowType

outset

Shadow

10

10

5

5

Text Align

H.Offset

0

V.Offset

0

Font

arial

24

B

I

U

Icon

none

Icon Align

H.Offset

0

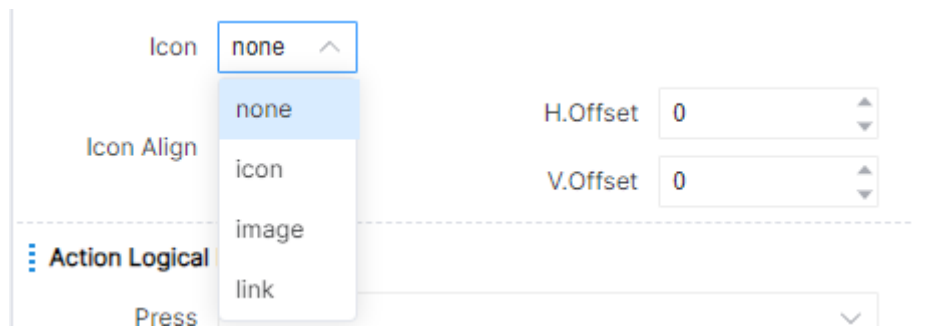
V.Offset

0

The first drop-down list can set the font, the second edit box can set the font size, B stands for bold, I stands for italics, U stands for underline, and the color block after can call up the palette to set the font color.

Icon

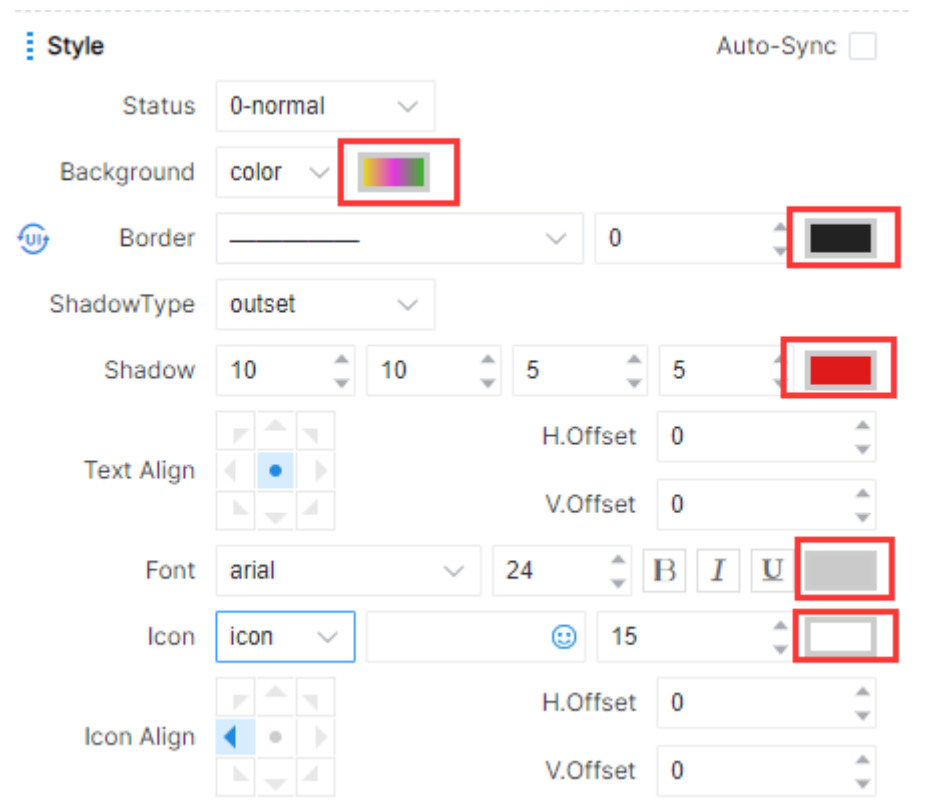
Icon attributes are provided for buttons, switches, matrices, steppings, and text, and none, icon, image, and link can be selected. As shown in the figure:



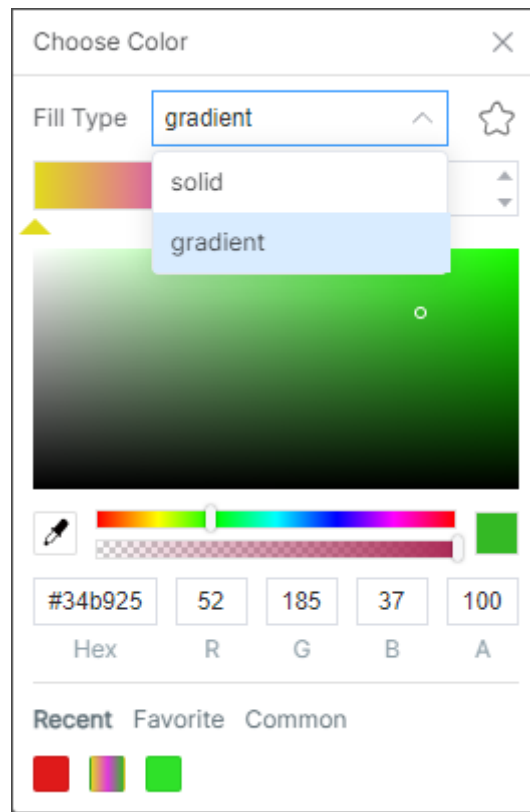
Icon alignment can adjust the alignment direction of icons within the component area, with specific settings referring to text alignment.

Color

During the interface design process, when color adjustment is required, you can open the color palette by clicking on the icon for setting colors. The icon for setting colors is usually located on the far right side of a property settings bar, as shown in the figure:



Click on the color setting icon and a color palette will pop up, as shown in the figure:

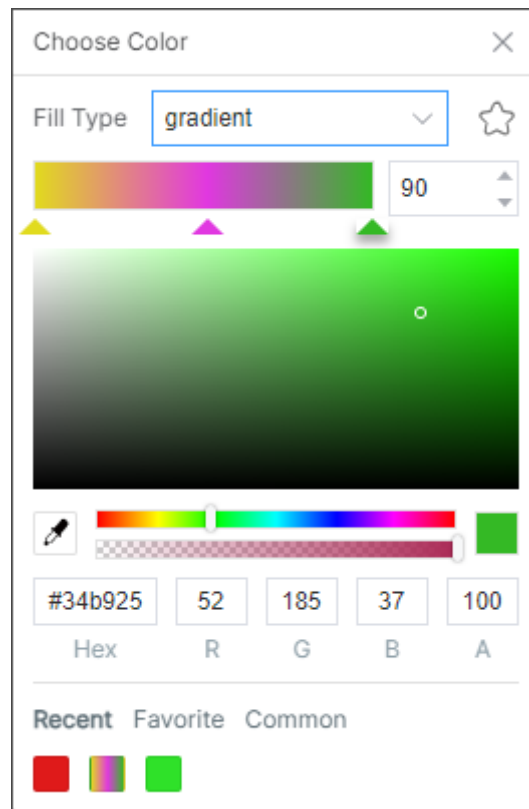


The fill type supports both solid and gradient colors.

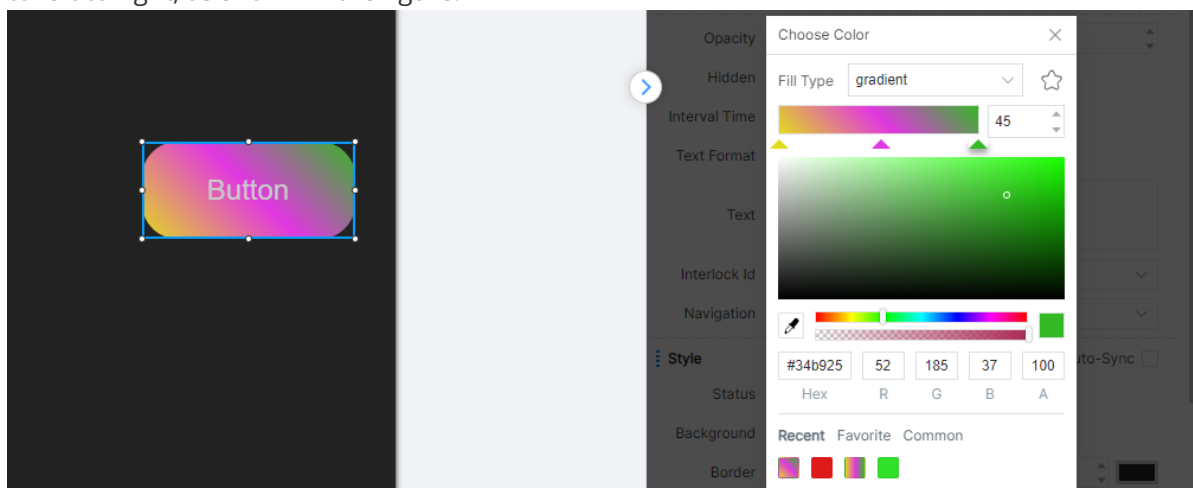
Click on solid color, and the entire filled area will be the color selected by the palette. The color range can be selected through the color selection bar under the color palette. Engineers can determine the approximate range of colors through the color selection bar, and then select the specific color they need through the color palette.

Under the color bar is the transparency selection bar. The leftmost end indicates complete transparency, in which states, the color of the component will be consistent with the interface. The far right end represents complete opacity, and the color of the component is completely determined by the color selected in the palette. Any range in the middle of the selection bar represents a certain degree of transparency, and the background color of the component will be determined by both the page's background color and the color selected in the palette.

Click on the gradient color, and a gradient color adjustment tool will appear below the fill type, as shown in the figure:



The leftmost position specified by the triangle represents the start, the rightmost represents the end, and at least two colors must be specified to represent the start and end of the color. The middle position can be set with a certain transition color by clicking to bring up (or remove) the triangle symbol, and multiple triangular symbols can be set to make the transition color more rich. The number on the right represents the rotation angle in that direction. By default, the gradient color changes from bottom to top. If set to 90, the direction from bottom to top changes to left to right, as shown in the figure:



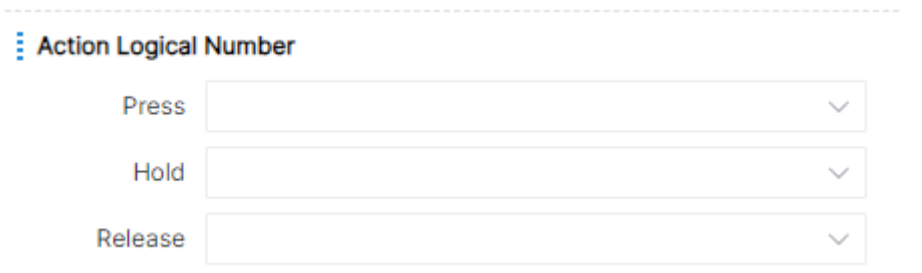
In addition, engineers can also modify colors by directly modifying the values of RGB, where A represents transparency and Hex is the hexadecimal representation of RGB. Each option can be manually edited, but when the edited content is not in compliance or exceeds the reasonable range of that option, this editing will not take effect and its value will be restored to its previous state.

The Recently Used, My Favorite, and Universal Colors below are the colors that engineers have recently used, the Favorite Colors, and the recommended colors provided by the IDE for engineers based on common scenarios, making it convenient for engineers to quickly choose.

Note: Only palettes of some properties can set color fill types, while border lines, fonts, etc. cannot set fill types and only support solid colors.

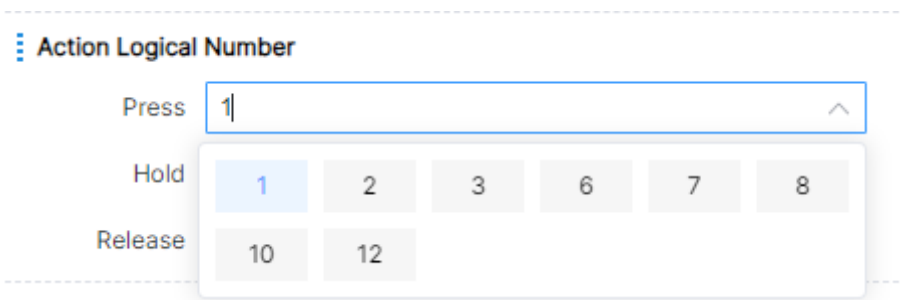
Action Logic Number

A component can perform subsequent logical operations by setting the logical number of the action to correspond to the logical number in the UI interaction logic module. It can be blank. When no action logic number is set, the component action will not trigger any subsequent logic. Different components may have different action logic numbers, for example, the button has three action logic numbers: press, long press and release, as shown in the figure:



The figure shows a UI interaction logic module interface. It has a title 'Action Logical Number' with a three-dot menu icon. Below the title, there are three rows, each with a label and a drop-down box: 'Press', 'Hold', and 'Release'. All three drop-down boxes are currently empty.

Select the used action logic number through the drop-down box, or manually input the new action logic number, as shown in the figure below for example:



The figure shows the same UI interaction logic module interface as before, but with the 'Press' drop-down box selected. The value '1' is entered in the box. Below the 'Press' box, a list of numbers is displayed: 1, 2, 3, 6, 7, 8, 10, and 12. The number '1' is highlighted in blue.

Attribute Logic Number

The component can change the component attribute value or output the current attribute value of the component by setting the attribute logic number corresponding to the logic number in the UI attribute setting and UI attribute output logic module. When the attribute logic number is not set, the attribute value of the component will not be changed by the UI attribute logic module, and will not be output through the UI attribute logic module. Different components may have different attribute logical numbers, for example, buttons have four attribute logical numbers: text, status, hidden and disabled, as shown in the figure:



The figure shows a UI attribute setting and output logic module interface. It has a title 'Property Logical Number' with a three-dot menu icon. Below the title, there are four rows, each with a label and a drop-down box: 'Text', 'Status', 'Hidden', and 'Disable'. All four drop-down boxes are currently empty.

Select the used attribute logical number through the drop-down box, or manually enter the new attribute logical number, as shown in the figure below for example:

Property Logical Number

Text

Status

1

2

3

4

5

6

Hidden

7

9

10

11

12

Disable

Component Details

Most attributes have been described in detail in Component Property Settings. Here, the unique attributes, action logical numbers and attribute logical numbers of each component will be described according to the classification of the component.

Button

All the properties that can be set for the button are shown in the figure:

100%

Button

Basic Info

Name

Button

X

525

Y

309

Width

180

Height

82

Radius

8

Opacity

1

Stack

2

Hidden

Disable

Interval Time

0.5

s

Text Format

Text

Text

Button

Interlock Id

Navigation

Style

Status

0-normal

Background

image

button_normal.i

auto

0

Border

0

ShadowType

outset

Shadow

0

0

0

0

Text Align

H.Offset

0

V.Offset

0

Font

arial

24

B

I

U

Icon

none

Icon Align

H.Offset

0

V.Offset

0

Action Logical Number

Press

Hold

Release

Property Logical Number

Text

Status

Hidden

Disable

Button supports interlock number setting, a group of buttons can be bound into an interlocking relationship to ensure that in any operation situation, only one button is active at the same time.

Page jump support: page, subpage, pop-up window. Trigger page jump when button is released.

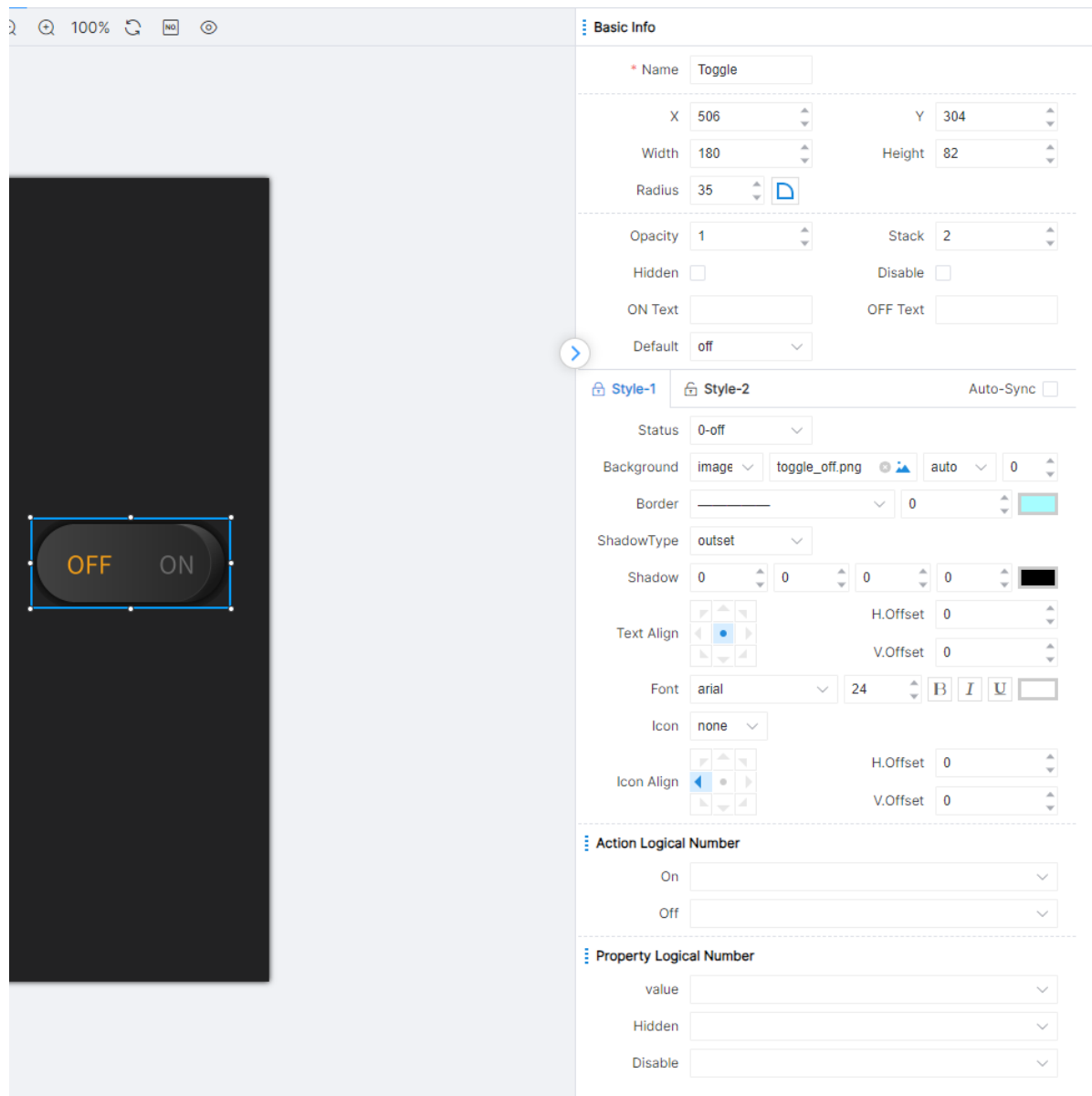
The IDE provides three state attributes for button components: 0-General, 1-Active, and 2-Disabled.

Logical numbering of button actions: press, hold and release. The first trigger long press time is fixed at 1 second, and the subsequent ones are executed according to the interval time set by the engineer.

Button attribute logical numbering: text, status, hidden, disabled.

Switch

All the properties that can be set by the switch are shown in the figure below:



Switch-specific attributes include text-on/text-off, which indicate the text displayed on the component when on/off; default indicates the default state of the component (on or off).

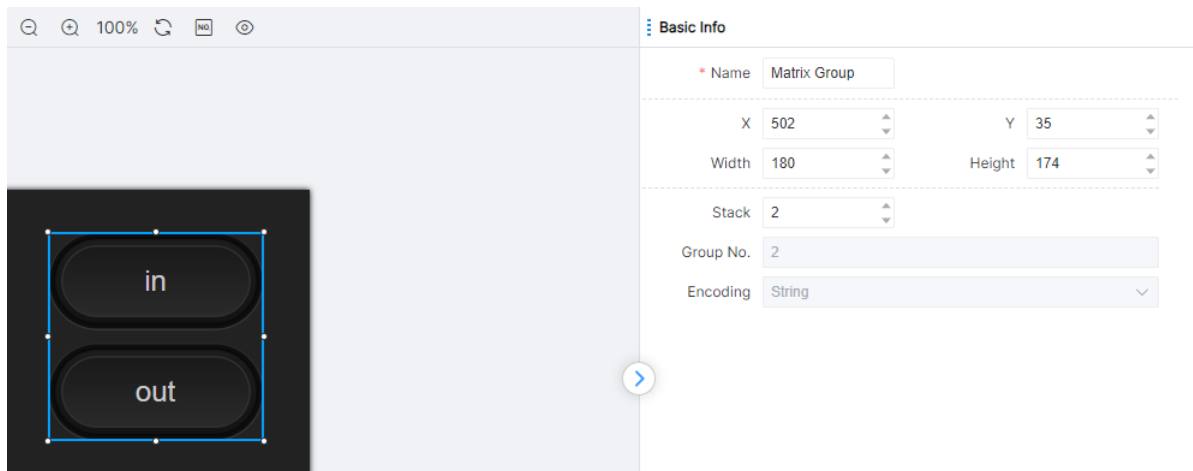
The IDE provides two styles for switch components, Style-1 and Style-2. Each style has three state attributes: 0-off, 1-on, and 2-disable. It should be noted that by setting the UI attribute module to modify the attribute value of the component (default attribute value is configured in the attribute bar), you can cause a state change. Switches cannot control their state directly through logic modules.

Switch action logic number: ON, OFF.

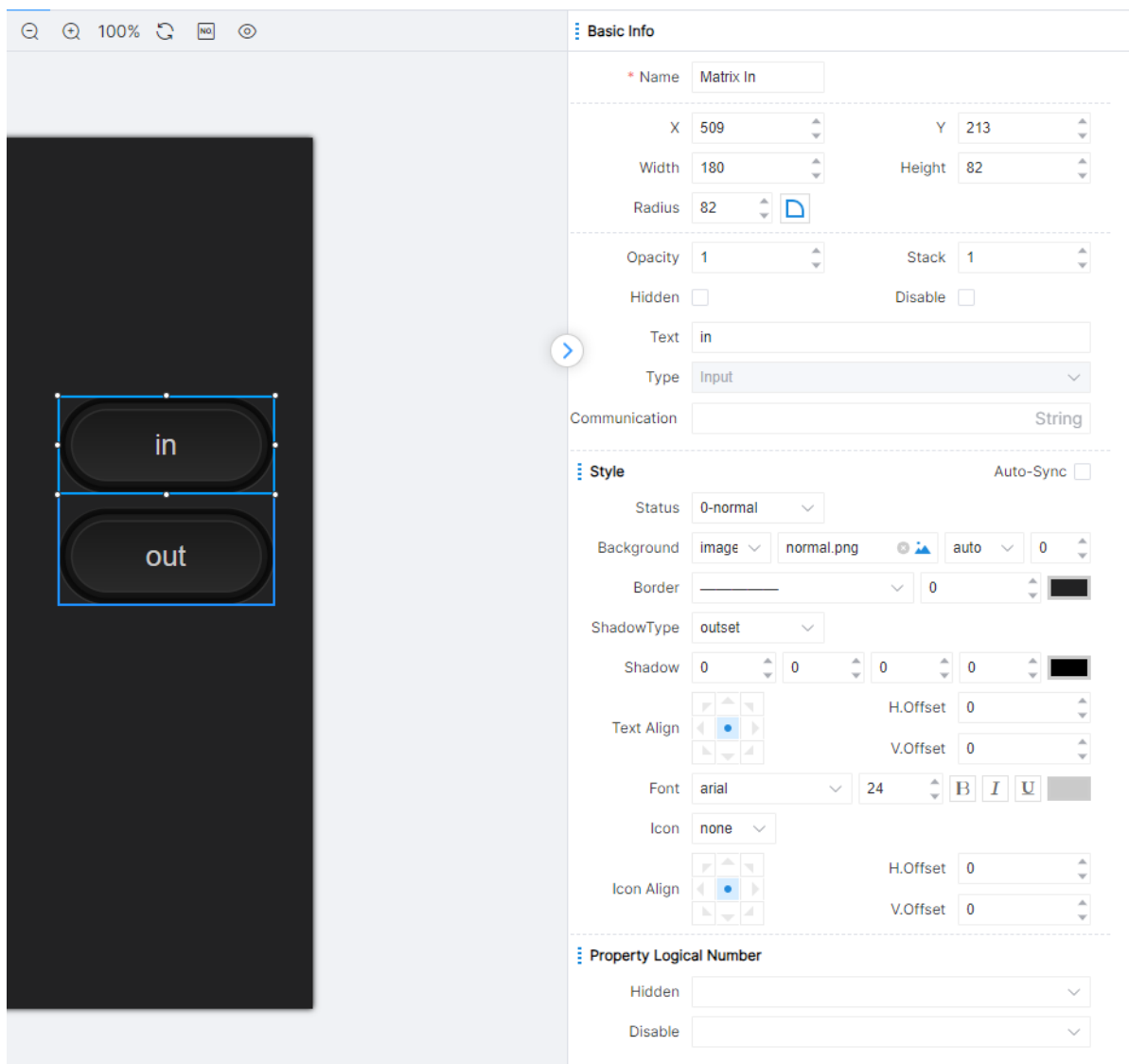
Switch attribute logical numbering: value, hidden, disabled.

Matrix

Matrix is a special component provided for matrix equipment. Only matrix components with the same matrix group number can be configured for corresponding control operations. All properties that can be set for a matrix group are shown in the figure below:



Enter the type matrix, and all the properties that can be set are as shown in the figure:



Matrix communication codes are used to control input and output matrices.

The IDE provides three state attributes for matrix components: 0-General, 1-Active, and 2-Disabled.

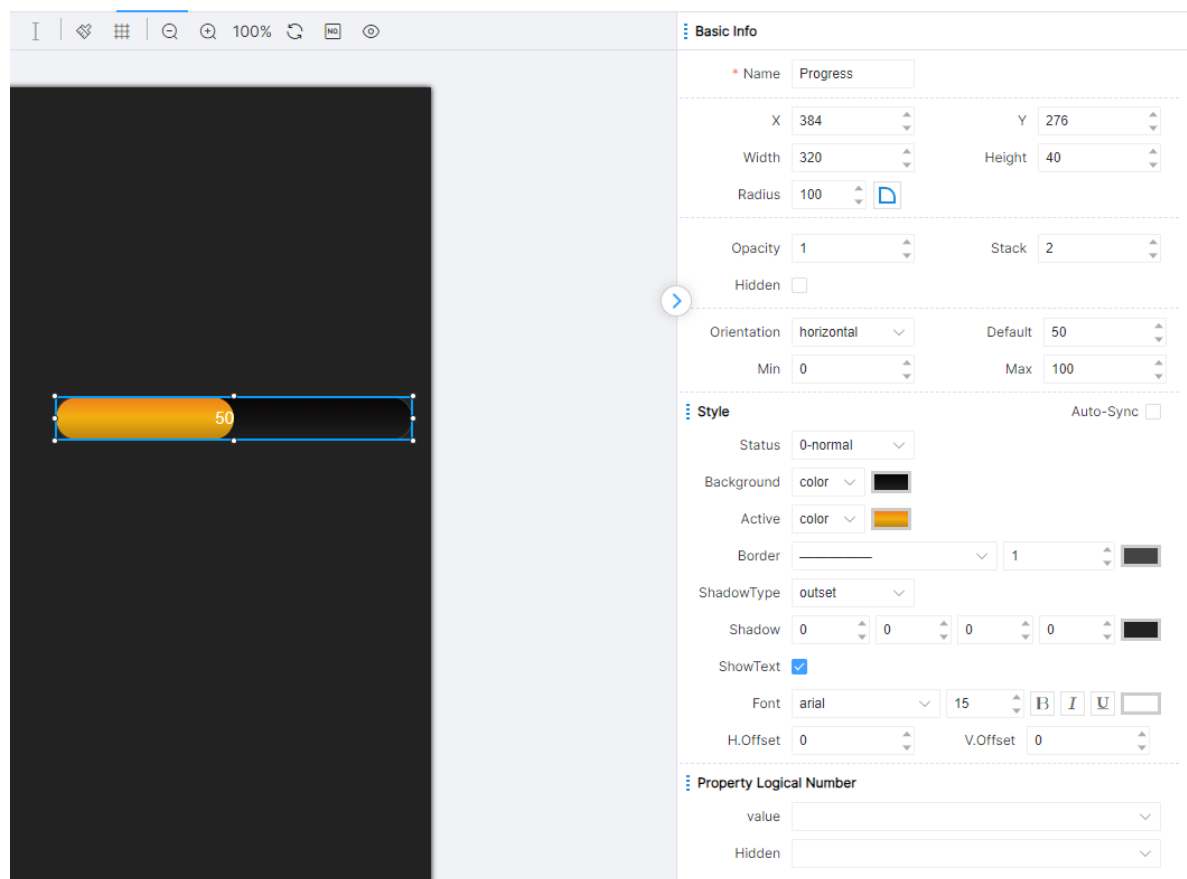
Matrix has no action logic number.

Matrix attribute logical numbering: hidden, disabled.

Output type matrix, all attributes that can be set refer to output type matrix.

Progress Bar

All the properties that can be set in the progress bar are shown in the figure below:



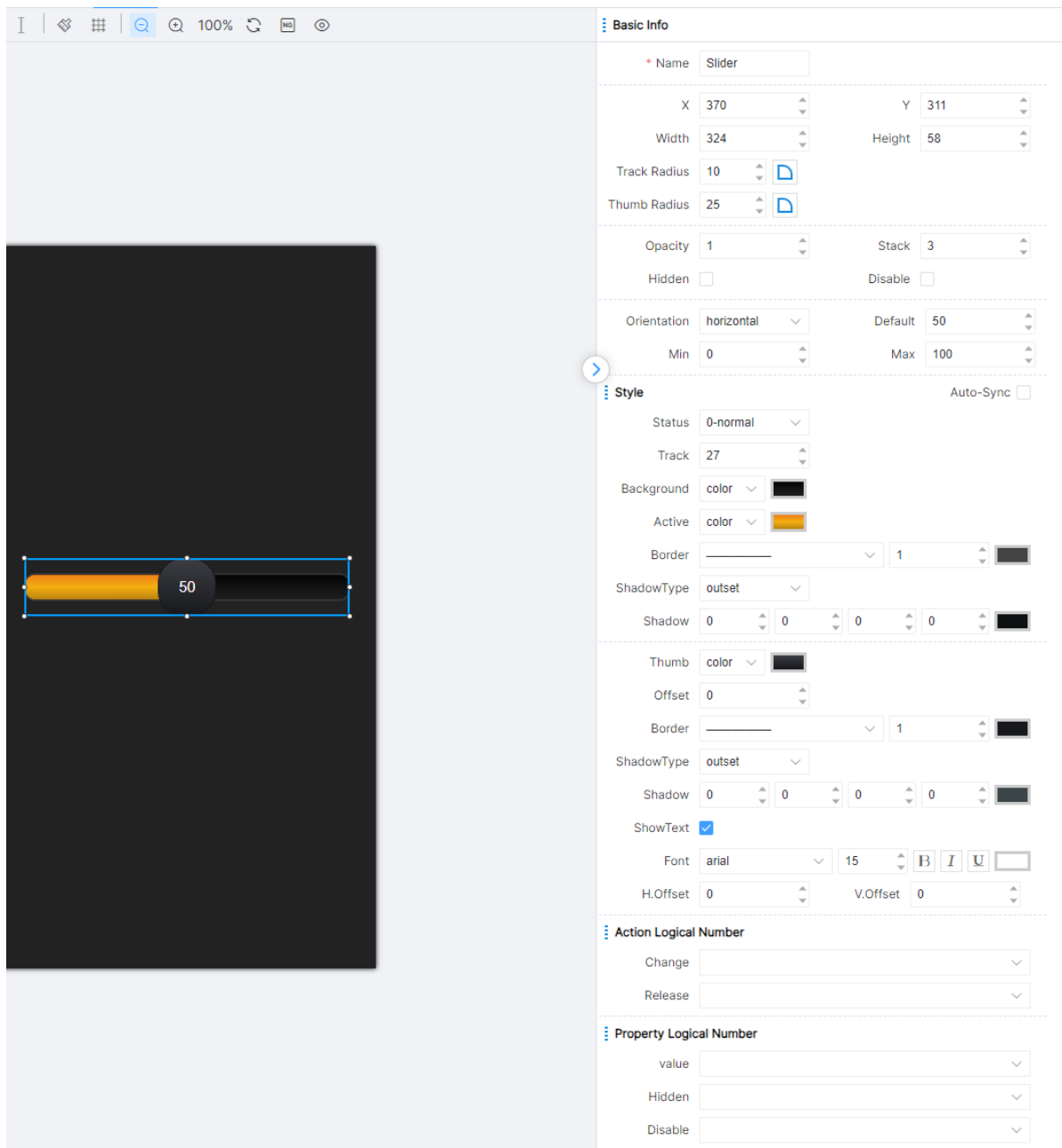
The direction values of the progress bar include horizontal, vertical, horizontal flip, and vertical flip, indicating the layout direction of the progress bar on the page. The default value represents the position of the active area on the progress bar in the default state. The minimum and maximum values represent the minimum and maximum values of the progress bar, respectively. The IDE provides 0-General, a status attribute for the progress bar component.

The progress bar has no action logic number.

Progress bar attribute logical numbering: value, hidden.

Sliding Bar

The sliding bar is divided into two parts structurally: the slide rail and the slider. The properties and styles of these two parts are separated to a certain extent in terms of configuration. All properties that the slider bar can set are as shown in the figure:



The slider bar has some of the same attributes as the progress bar. The difference between them is that the progress bar is passive. According to the UI attribute setting module, the value of the progress bar is modified, while the slider bar is the position where the user actively triggers the modification drag button.

When setting the fillet property, it is divided into two property configurations: slide fillet and slider fillet. The specific configuration method is the same as other fillet configurations. In the style properties, the track and drag button are also divided into two areas, respectively. Depending on where you drag the button, you can divide the track into two parts, the dragged part styled by the active area property and the undragged part styled by the background property. In addition, the direction, default value, minimum value, maximum value have the same meaning as these properties of the progress bar.

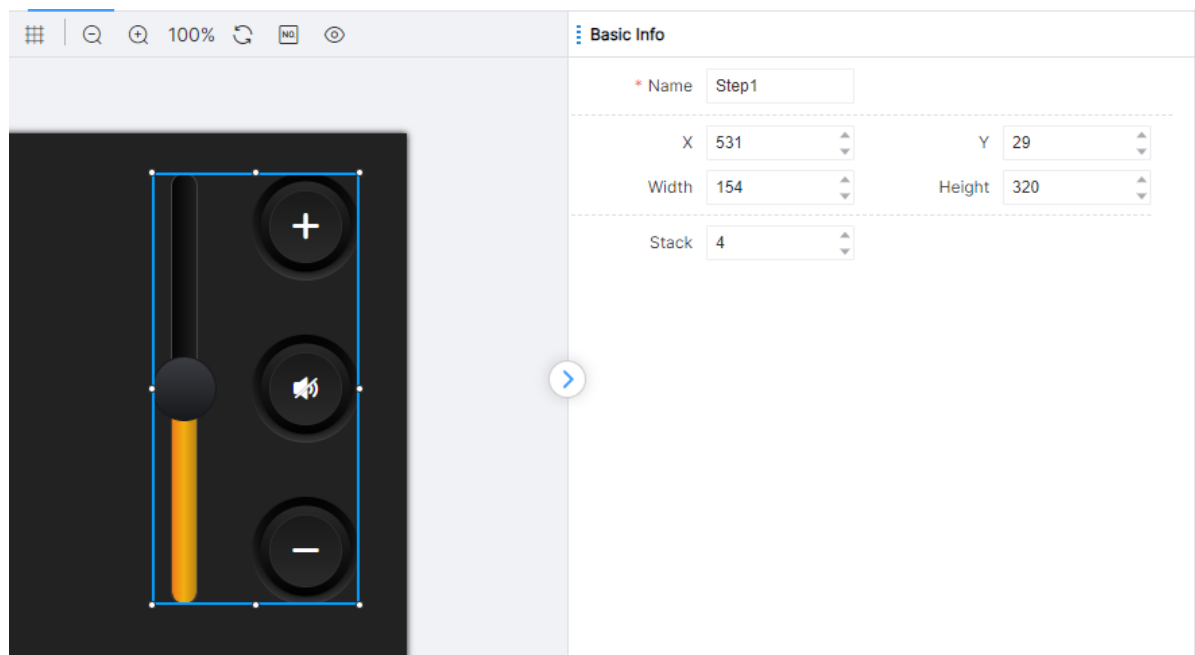
The IDE provides two status properties for slider components: 0-General and 1-Disabled.

Slide bar action logic number: change, release.

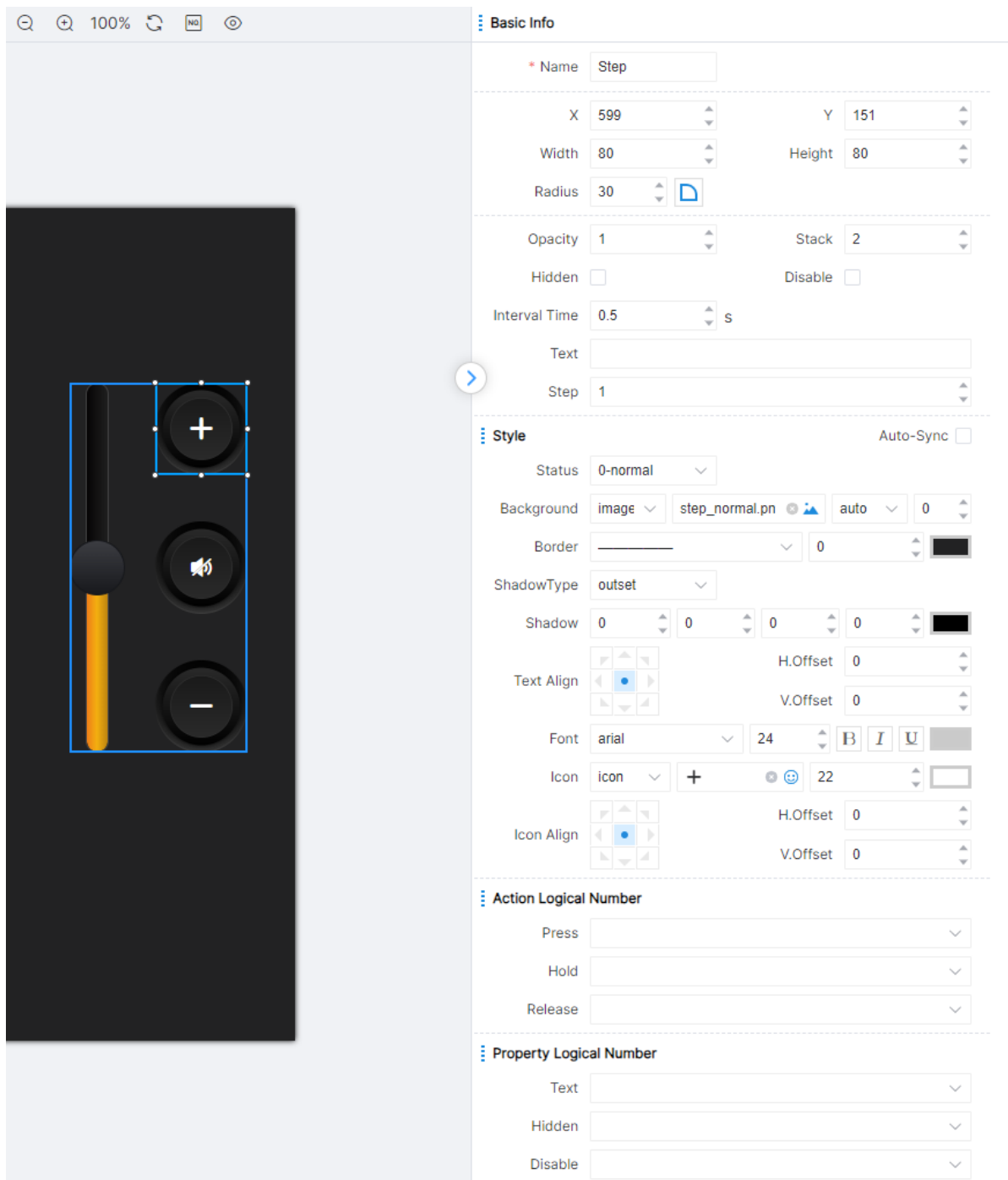
Slider attribute logical numbering: value, hidden, disabled.

Stepping 1

Stepping is used in conjunction with slider bar to increase or decrease the slider value by a fixed value. All the properties that can be set for Stepping 1 are shown in the figure below:



Stepping 1 is composed of 1 slide bar component, 2 stepping components and 1 switch component. The components in Stepping 1 group cannot be deleted or copied separately. All the properties that can be set for stepping are shown in the figure:



Basic Info

- Name: Step
- X: 599, Y: 151
- Width: 80, Height: 80
- Radius: 30
- Opacity: 1, Stack: 2
- Hidden: ☐, Disable: ☐
- Interval Time: 0.5 s
- Text:
- Step: 1

Style Auto-Sync ☐

- Status: 0-normal
- Background: image, step_normal.png, auto, 0
- Border: 0
- ShadowType: outset
- Shadow: 0, 0, 0, 0
- Text Align: H.Offset: 0, V.Offset: 0
- Font: arial, 24, B, I, U
- Icon: icon, +, 22
- Icon Align: H.Offset: 0, V.Offset: 0

Action Logical Number

- Press:
- Hold:
- Release:

Property Logical Number

- Text:
- Hidden:
- Disable:

The text content is displayed in the stepping. + can be configured to indicate that the stepping is an increasing value, and - can be configured to indicate that the stepping is a decreasing value. It should be noted that only the displayed text is indicated here. The specific increasing value or decreasing value needs to be entered in the "stepping" field (positive or negative). Stepping indicates how much the stepping increases or decreases with each click (by setting the stepping to a positive or negative number), or it can be triggered multiple times with a long press.

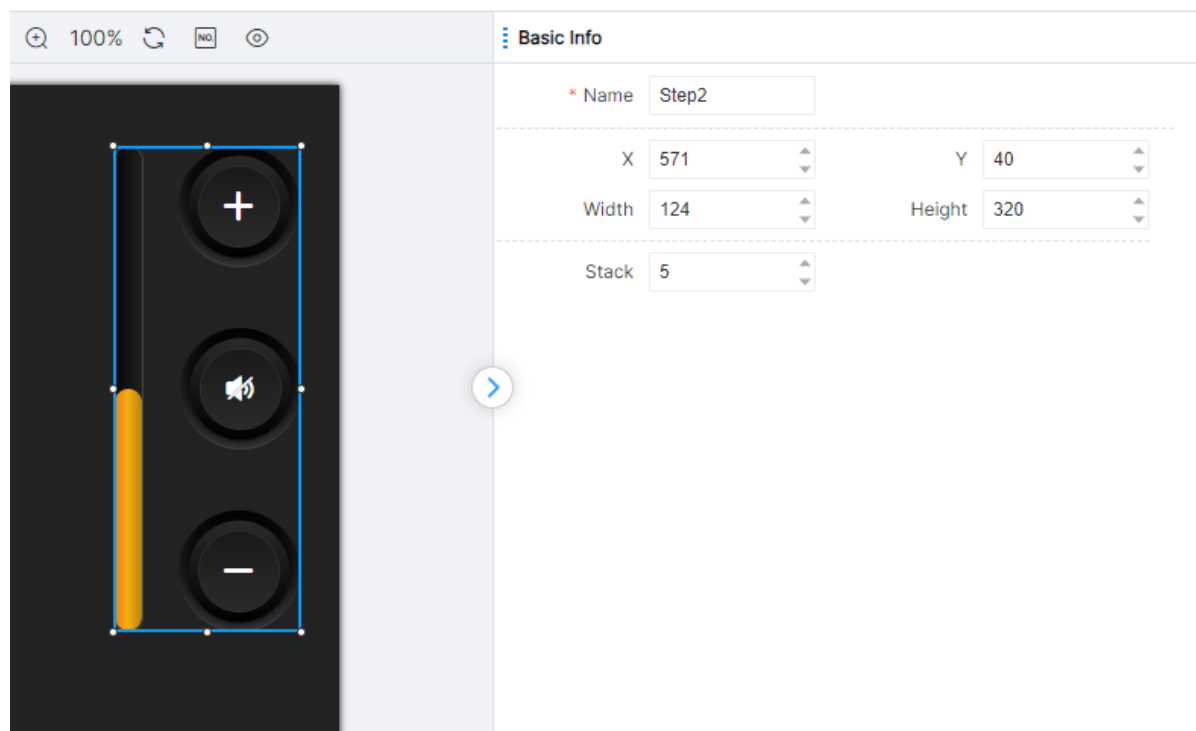
The IDE provides three state properties for the stepping component: 0-Normal, 1-Active, and 2-Disabled.

Stepping action logic numbering: press, hold and release. The first trigger long press time is fixed at 1 second, and the subsequent ones are executed according to the interval time set by the engineer.

Stepping attribute logical numbering: text, hidden, disabled.

Stepping 2

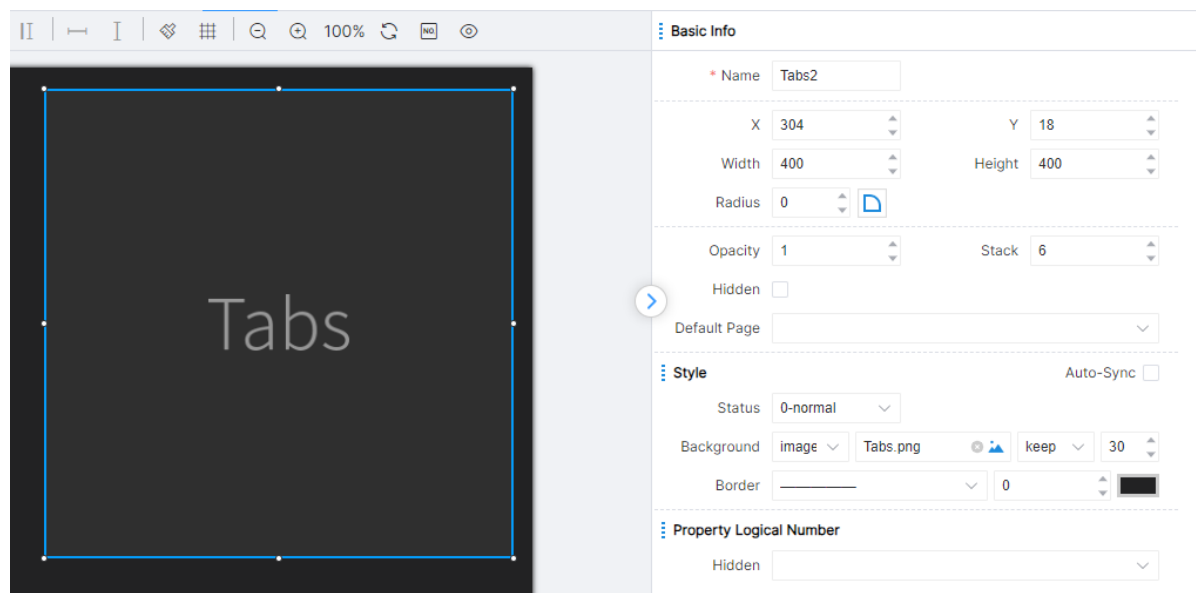
Stepping is used in conjunction with slider bar to increase or decrease the slider value by a fixed value. All the properties that can be set for Stepping 2 are shown in the figure below:



Stepping 2 is composed of 1 slide bar component, 2 stepping components and 1 switch component. The components in Stepping 2 group cannot be deleted or copied separately. For all the properties that can be set for stepping, please refer to the stepping in Stepping 1.

Subpage Container

All properties that can be set for the subpage container are shown in the figure below:



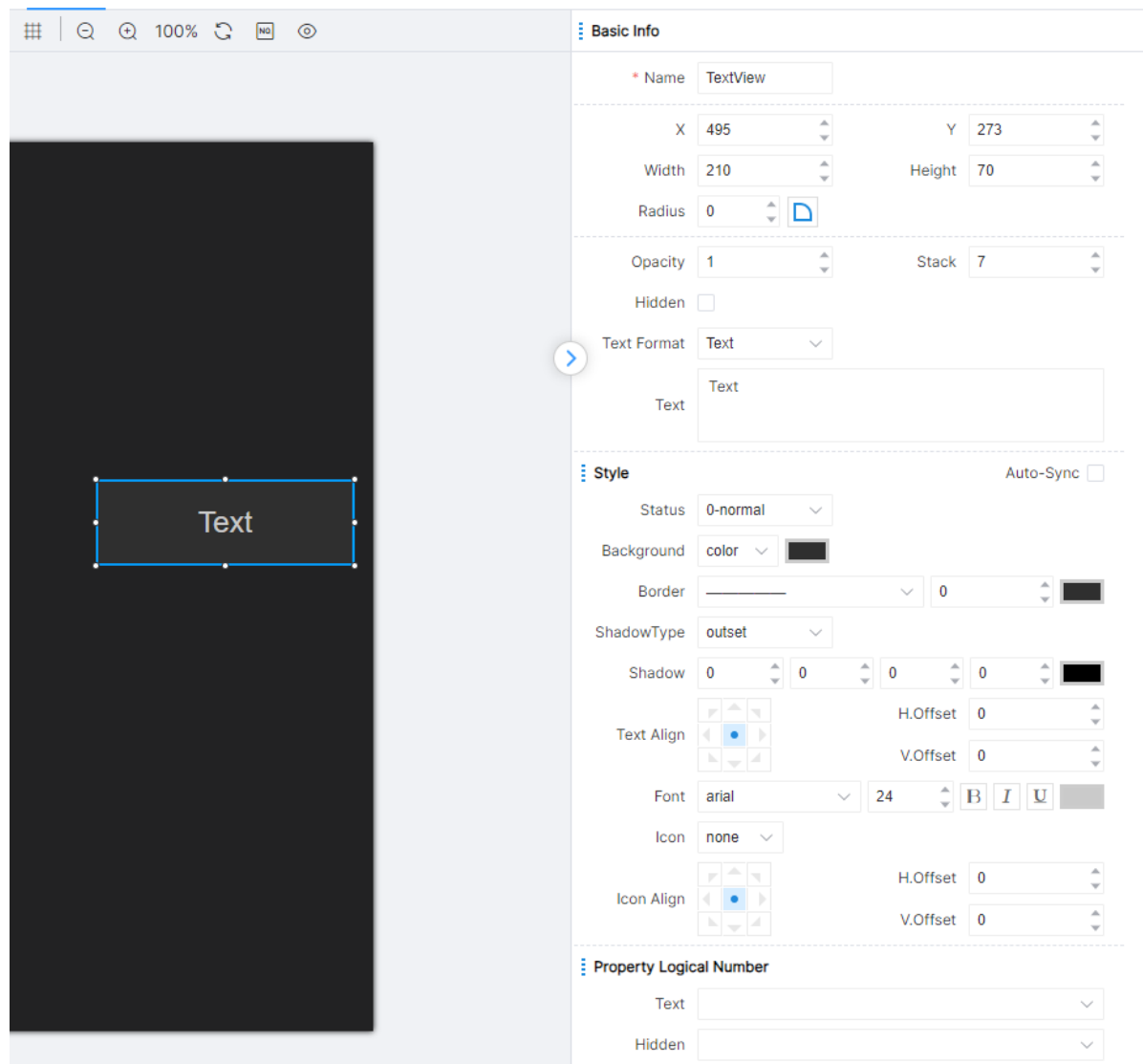
A subpage container loads the contents of a subpage, where the default page represents the subpage that the subpage container displays by default. According to the actual scene, modify the display subpage in the subpage container, so as to achieve the purpose of subpage replacement. The IDE provides 0-General, a status property for subpage container components.

Subpage container has no action logical number. Engineers can use the page jump property of the button, or use the page jump module to control the subpage container to complete the switching of the display content.

Subpage container attribute logical numbering: hidden.

Text

All the properties that text can set are shown in the figure:



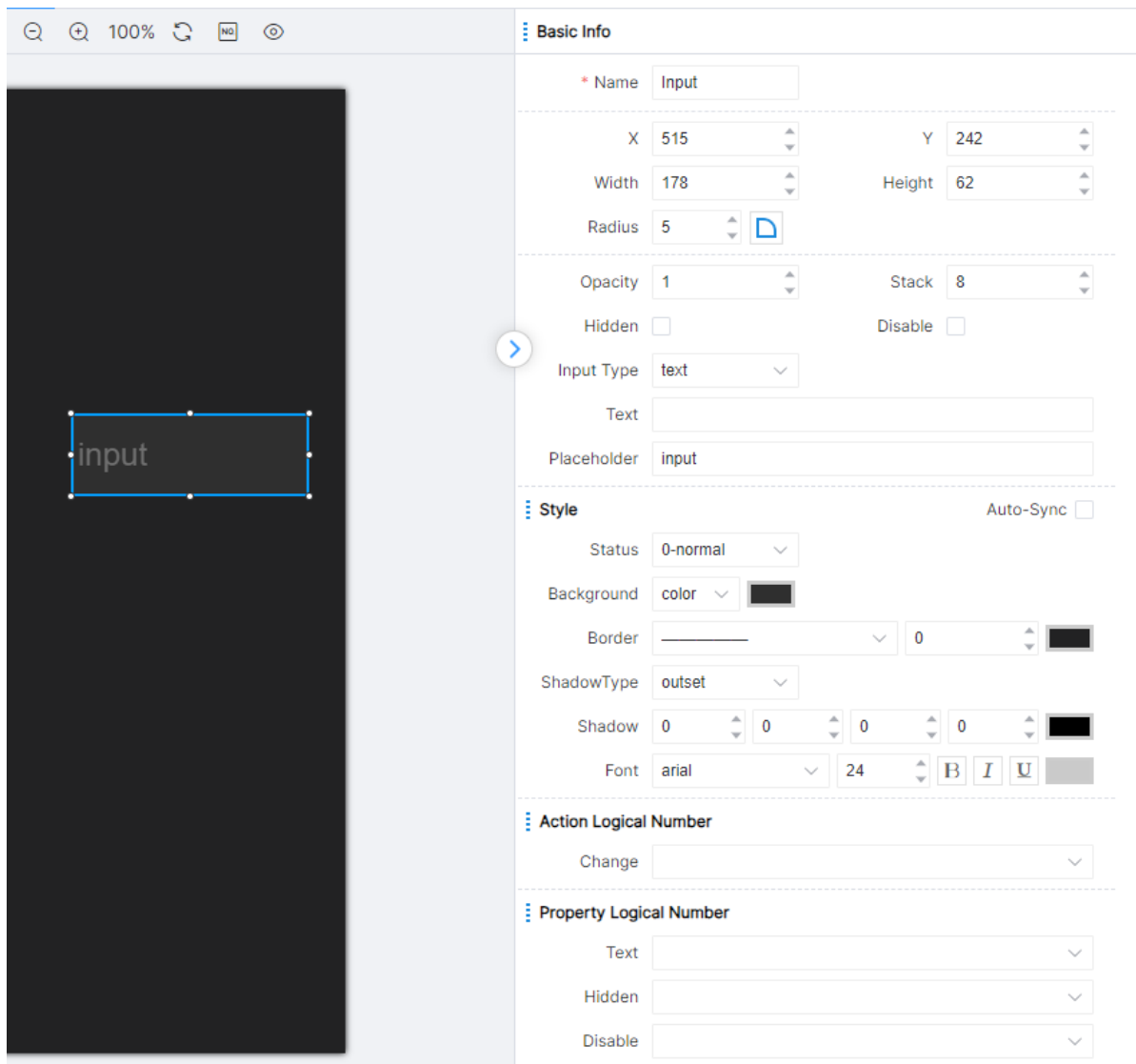
The content in the text is the content displayed by the text component, and it supports entering carriage returns for line feed. The IDE provides 0-General, a status property for text components.

Text has no action logic numbering.

Text attribute logical numbering: Text, Hidden.

Input Box

All the properties that can be set in the input box are shown in the figure:



Input type properties of input box component are divided into text, password and text field. Prompt text can be set for all three types, and default text can be set for all types. The difference is that the input content of text type is plain text display, and carriage return is not supported; the input content of password type is replaced by solid dots; the text field class supports carriage return for line feed.

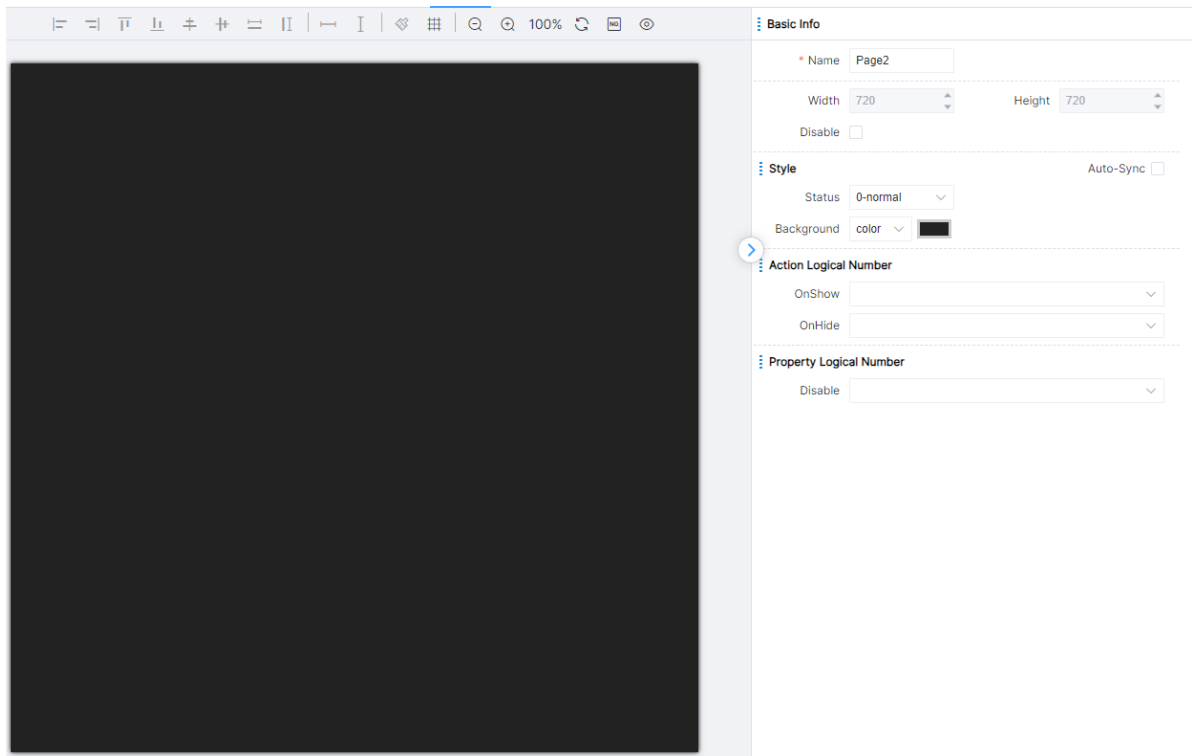
The IDE provides two status properties for Input Box components: 0-General and 1-Disabled.

Input box action logic number: change. When the user changes the text in the input box, both adding and deleting text triggers the change action.

Input box attributes logical numbering: text, hidden, disabled.

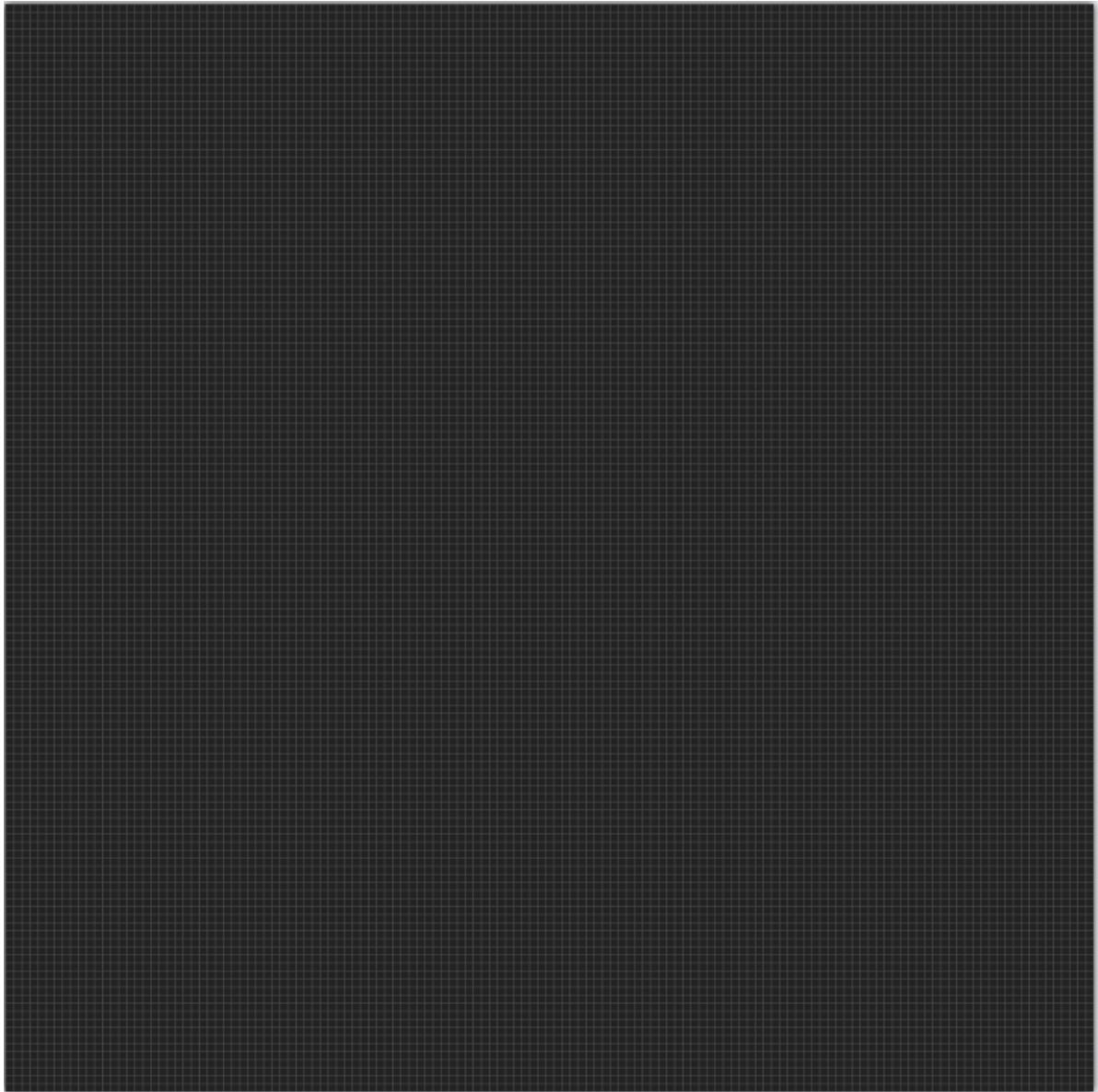
Page Layout

The width and height of the page are related to the central controller model set when creating the project. For example, the TS04C page size is 480*480. After setting, you can view the page layout rendered on the device through the preview icon  in the toolbar.



If you need to view partial details or view the layout of components as a whole, you can zoom in and out of the page by zooming in icon  and zoom out icon . The percentage on the right indicates the current zoom in and out ratio. You can quickly restore the page size to 100% by the Reset zoom icon .

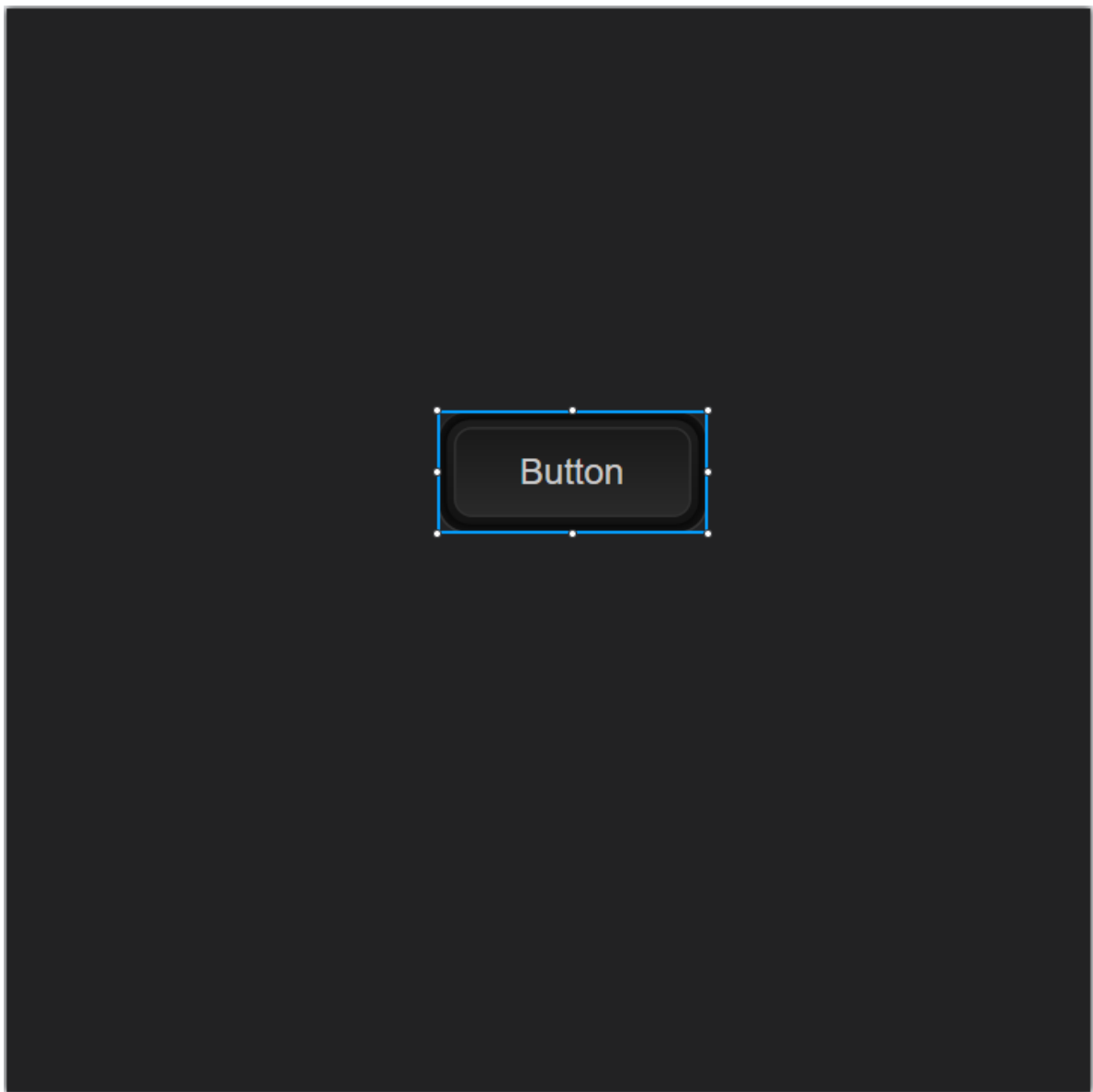
When the engineer drags each component in the page, the page will have a guide line for component alignment, and the engineer can also click on the page grid icon  in the toolbar for more precise operation. The grid effect is as shown in the figure:



Below are some basic operations for layout, and placement/addition/deletion of components on a page.

Add Component

Use the mouse to select a component in the component library, hold down the mouse, and then drag it onto the canvas. Once you find the appropriate location on the canvas, release the mouse to complete the component addition.



The addition of matrices is special. Drag the matrix to the canvas and a window will pop up to confirm the relevant parameters, as shown in the figure:

New Matrix

✕

Group No.

1

▼

Encoding

☒ String

☐ Hex

In

1

▲▼

Out

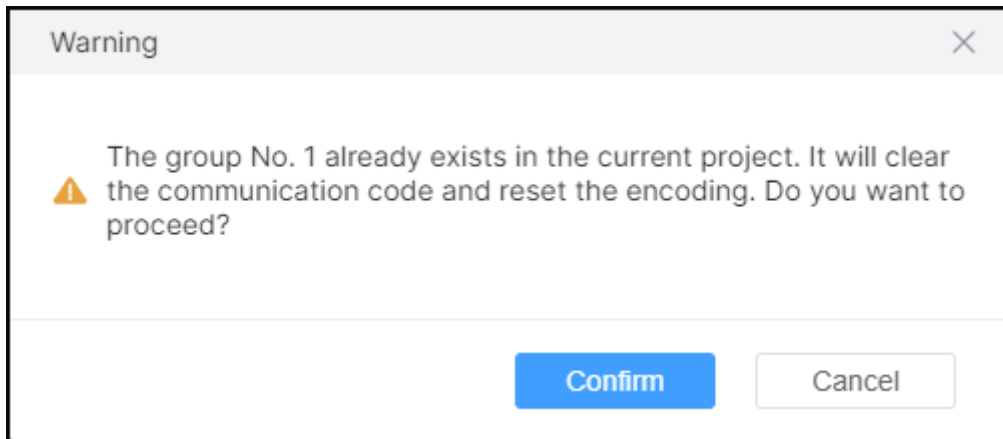
1

▲▼

Confirm

Cancel

[Group Number]: The matrix consists of multiple inputs and outputs, and this number indicates that they belong to the same matrix group. [Communication Code]: Format of communication code. [Input]: Number of inputs of the matrix. [Output]: Number of outputs of the matrix. Click [OK] to add a matrix component. When the selected group number already exists, a prompt will pop up and merge with the existing group number matrix. As shown in the figure:



The added component has a stretchable border as the selected identifier, and the current attribute values of the component are presented in the property window on the right. The style can be adjusted as needed.

Component Operations

Clicking on a component on the page allows engineers to adjust the size of the component by stretching its borders. Select the component and click the left button without releasing it. At this point, you can drag the component to any position on the page. Engineers can also select components and use keyboard directional keys to move them up, down, left, and right. Right click on the selected component and click on[Horizontal Center]and[Vertical Center]in the pop-up menu to quickly adjust the component to the horizontal and vertical center positions on the entire page.

During dragging, if multiple components overlap, it will be judged whether the component is displayed in the upper layer or the lower layer according to the attribute [Hierarchy]size. If the hierarchy value is large, it means that the component will be at the top layer of multiple components, and engineers can directly see the component through the interface. Otherwise, it is at the lowest level of multiple overlapping components. In addition, the IDE provides a way to quickly change this hierarchy. Right click on the selected component, and in the pop-up menu, you can quickly move the hierarchy among a group of overlapping components through the options of[Top]and [Bottom]. At this time, their attribute hierarchy values will automatically change accordingly.

Select the component, right-click on the menu, click [Delete], or use the shortcut keys <Ctrl+D>/ <delete > to delete the selected component.

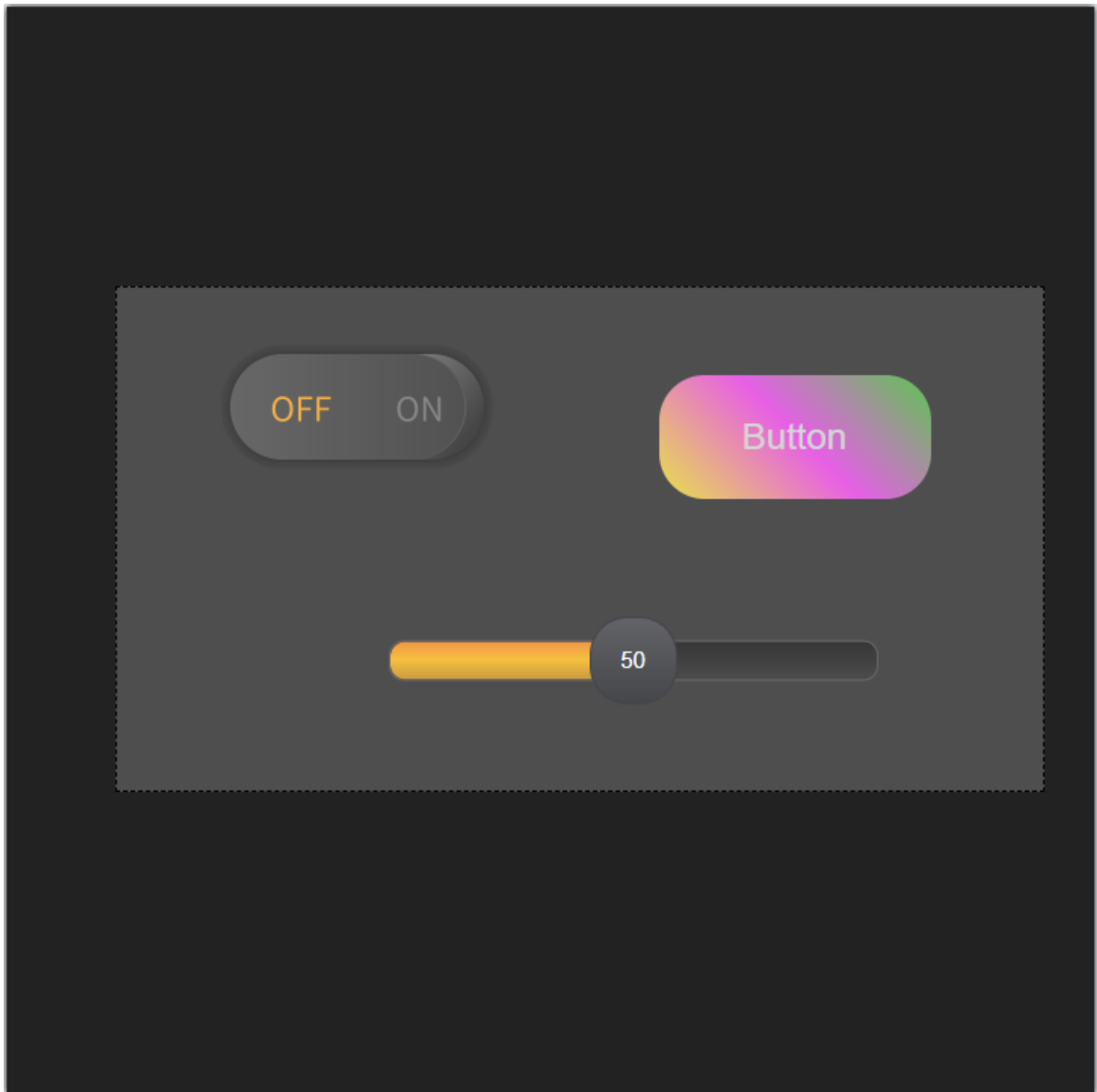
Select a component and hold down the shift key to achieve proportional scaling of multiple selected components.

In addition, the IDE provides commonly used shortcut keys. < Ctrl+C> represents copying a component,<Ctrl+X> represents cutting a component, <Ctrl+V> represents pasting a component, and <Ctrl+Z> represents undoing. If you undo multiple times, you can use <Ctrl+Y> to restore the previously undone operation.

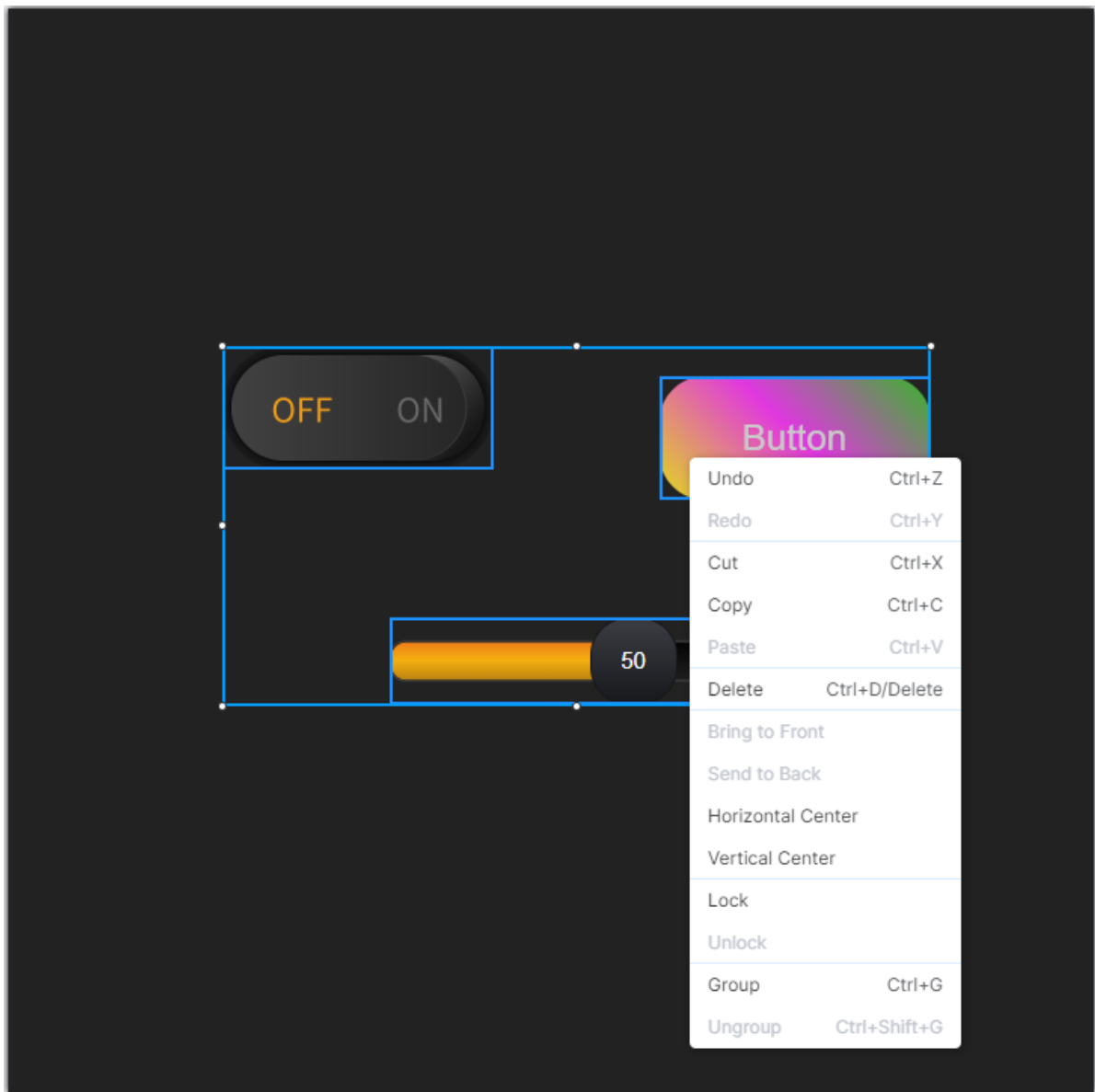
Group Operations

Group operations represent simultaneously manipulating multiple components.

Hold down the left mouse button in the blank area on the page, drag the mouse to the lower right will appear a light-colored rectangular box:



Find the appropriate location, fully incorporate multiple components into the rectangular box, release the left mouse button, and select the covered components simultaneously. At this time, the selected multiple components are within one operation box. You can also select the desired components one by one by holding down the keyboard Ctrl key and using the left mouse button. In the operation box, right-click the mouse and select [Combination] from the menu or use the shortcut <Ctrl+G> to fix the group components into a group and perform group operations:



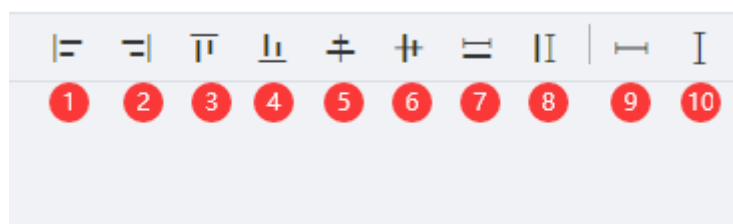
You can perform group operations as a whole, such as stretching, dragging, copying, cutting, pasting, level adjustment, centering, etc. These operations can be done either through the right-click menu or through the corresponding shortcut keys.

Under Group operation, hold down shift key to zoom in/out multiple selected groups/components equally.

If you need to cancel this group operation, you can right-click the operation box area and select [Remove Group] in the pop-up option menu or use Ctrl+Shift+G > to realize it.

Quick Alignment

The toolbar provides a set of buttons for quick position adjustment, as shown in the figure:



Note that quick alignment provided by the IDE is a group operation based on the last component added to the page or the last selected component.

Click the button ① in the toolbar as shown in the figure to align with the left border of the benchmark component.

Click the button ② in the toolbar as shown in the figure to align with the right border of the benchmark component.

Click the button ③ in the toolbar as shown in the figure to align with the top border of the benchmark component.

Click the button ④ in the toolbar as shown in the figure to align with the bottom border of the benchmark component.

Click the button ⑤ in the toolbar to center it vertically with the benchmark component.

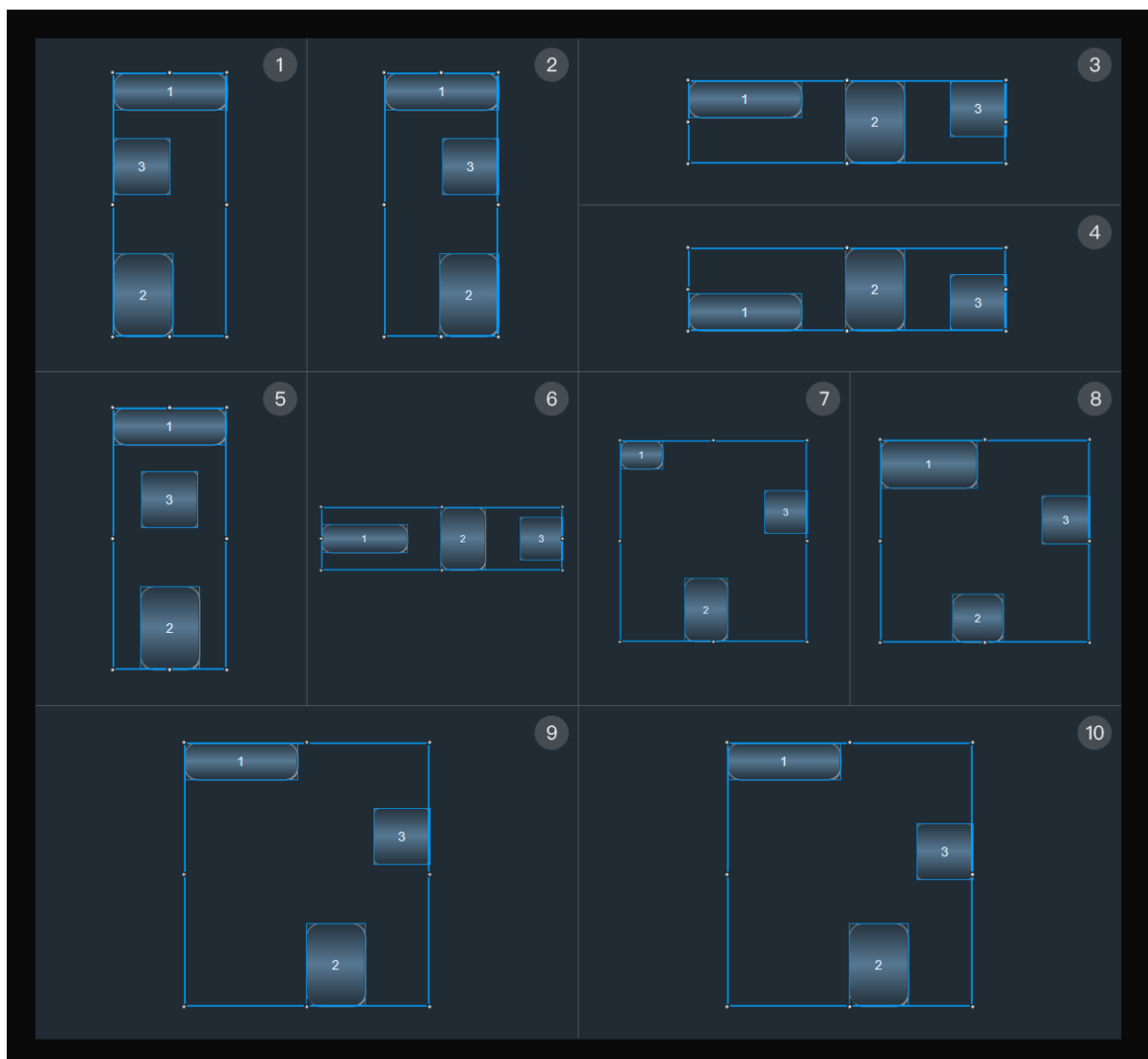
Click the button ⑥ in the toolbar to center it horizontally with the benchmark component.

Clicking the button ⑦ in the toolbar indicates that the width of the benchmark component will be used as the standard, and all other components in the group will be adjusted to that width.

Clicking the button ⑧ in the toolbar indicates that the height of the benchmark component will be used as the standard, and all other components in the group will be adjusted to that height.

Click the button ⑨ in the toolbar to adjust the horizontal spacing between components to be equal.

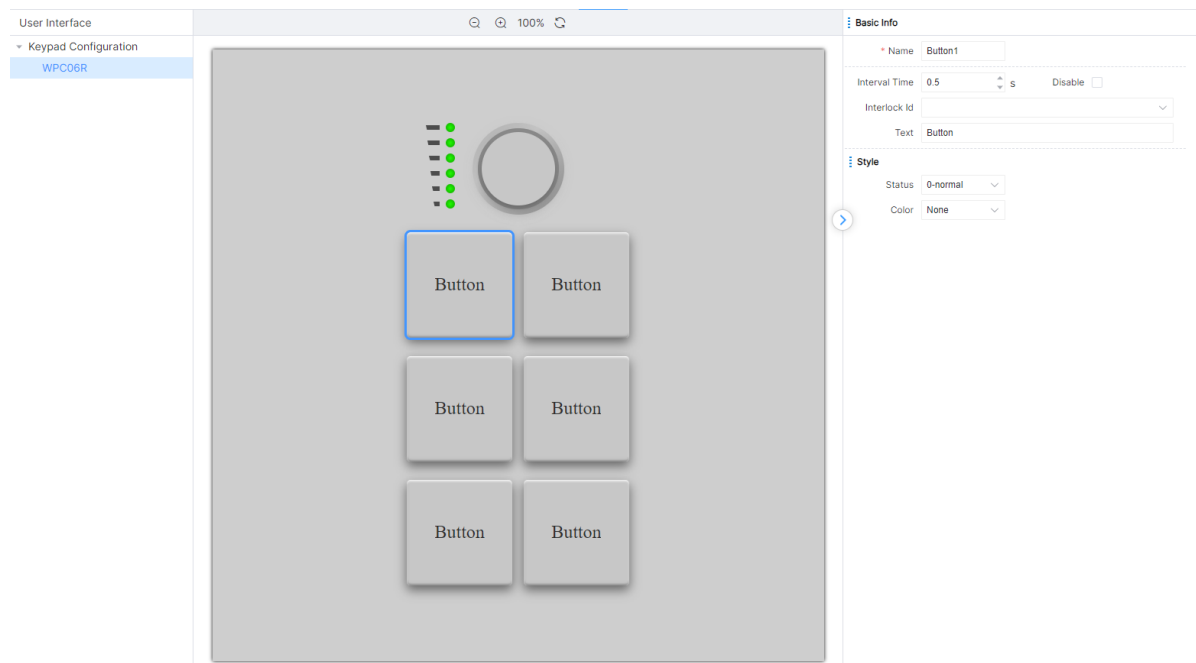
Click the button ⑩ in the toolbar to adjust the vertical spacing between components to be equal.



Button Panel

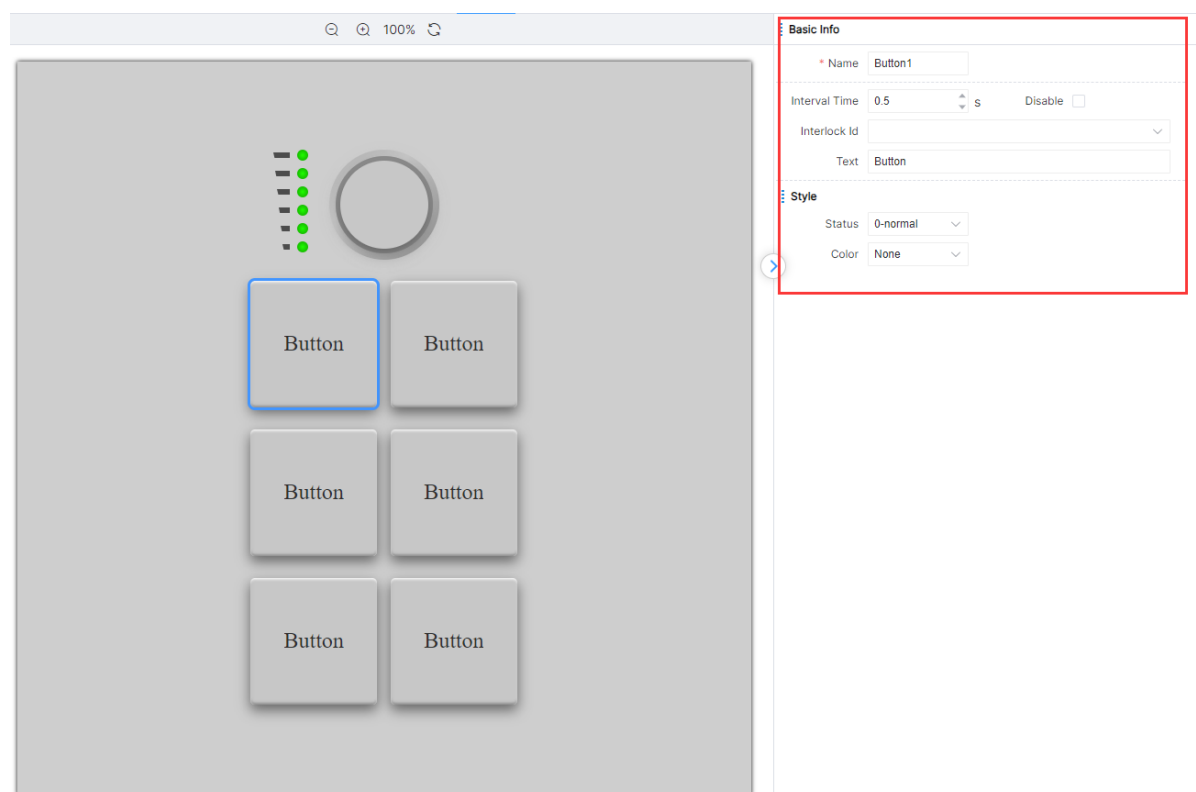
This section mainly describes panel buttons, including buttons, knob and knob switch.

The panel interface follows the controller model, and the following is the configuration interface:



Button Property Settings

Open the Settings page, the first button will be automatically selected, and the properties of the component can be seen on the right side, as shown in the red box area in the figure.



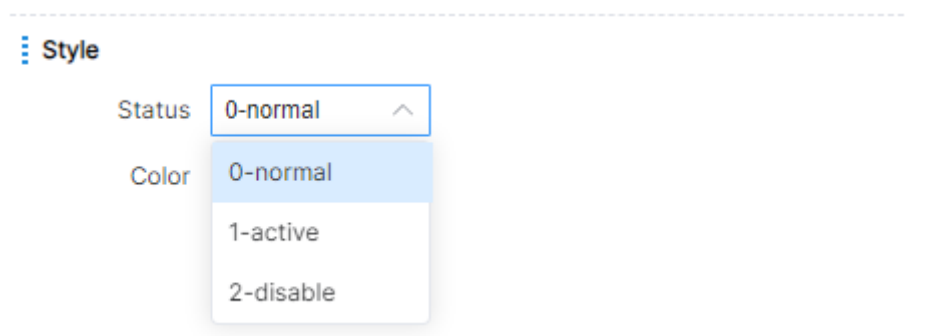
Basic Information Settings

Button properties are the same due to button type differences, but each button contains a batch of properties with the same name and meaning. This part of properties belongs to basic information. The basic information is shown in the table below:

Property Name	Description
Name	Required, default to the button type
Interval Time	The time interval for triggering the long press action after pressing and holding the button for 1 second
Disable	Component takes effect by default, if this property is checked, it will not take effect.
Interlock Number	The same interlock number button only supports one active state
Text	Identify buttons during design
Default Value	Initial value of knob and knob switch
Stepping Value	The change value of each rotation of the knob
Minimum	The minimum value of the knob
Maximum	The maximum value of the knob

Style Settings

The button can change its light color by adjusting the value of the style property. The buttons change the lighting according to the operation logic during the operation process, and these lights can be changed through the [Status] property. For example, the button mode component of the button has three states: [Normal], [Active], and [Disabled], as shown in the figure:



Button Type

There are three types of buttons on the button panel: buttons, knob, and knob switch. It contains 6 buttons, 1 knob and 1 knob switch.

Button

Button configurations in the button panel are similar to those in the Interface Designer. The configurable properties are shown in the figure:

The screenshot shows a configuration panel for a button. It is divided into two main sections: 'Basic Info' and 'Style'. The 'Basic Info' section contains fields for 'Name' (set to 'Button1'), 'Interval Time' (set to '0.5' with a unit 's'), 'Interlock Id' (a dropdown menu), 'Text' (set to 'Button'), and a 'Disable' checkbox. The 'Style' section contains 'Status' (set to '0-normal') and 'Color' (set to 'None').

Basic Info	
* Name	Button1
Interval Time	0.5 s
Interlock Id	
Text	Button
Disable	☐

Style	
Status	0-normal
Color	None

The interval time property is used to describe the interval time when the system triggers the long press action when the button is pressed for a long time. That is, when the user clicks the button and the system determines that it is a long press (pressing for more than 1 second, the long press event is triggered for the first time, and then the system executes a long press event every other time. The default interval given by the system is 0.5 seconds.

The IDE provides three state properties for button components: 0-General, 1-Active, and 2-Disabled, each corresponding to a different light.

Operating the buttons on the button panel can trigger the subsequent logic of pressing, long pressing and releasing the corresponding buttons in the logic module. It is also possible to set the state of the corresponding button through the logic module, or disable the property, so that it changes during operation.

Knob

The knob configuration in the button panel is similar to the slider bar in the Interface Designer. The properties that can be set are shown in the figure:

The screenshot shows a configuration panel for a knob. It is divided into two main sections: 'Basic Info' and 'Style'. The 'Basic Info' section contains fields for 'Name' (set to 'Knob'), 'Default' (set to '50'), 'Min' (set to '0'), 'Max' (set to '100'), 'Step' (set to '1'), and a 'Disable' checkbox.

Basic Info	
* Name	Knob
Disable	☐
Default	50
Step	1
Min	0
Max	100

Minimum and maximum values represent the minimum and maximum values of the knob respectively. Stepping indicates how much clockwise rotation increases or how much counterclockwise rotation decreases, and continuous rotation triggers multiple changes.

When the user rotates the knob to modify the value, the subsequent logic of the knob change action in the logic module will be triggered; when the user stops rotating 2S, the subsequent logic of the knob release action in the logic module will be triggered. The value of the knob can be set via the logic module, or the property can be disabled.

Knob Switch

The properties that can be set by the knob switch in the button panel are shown in the figure:

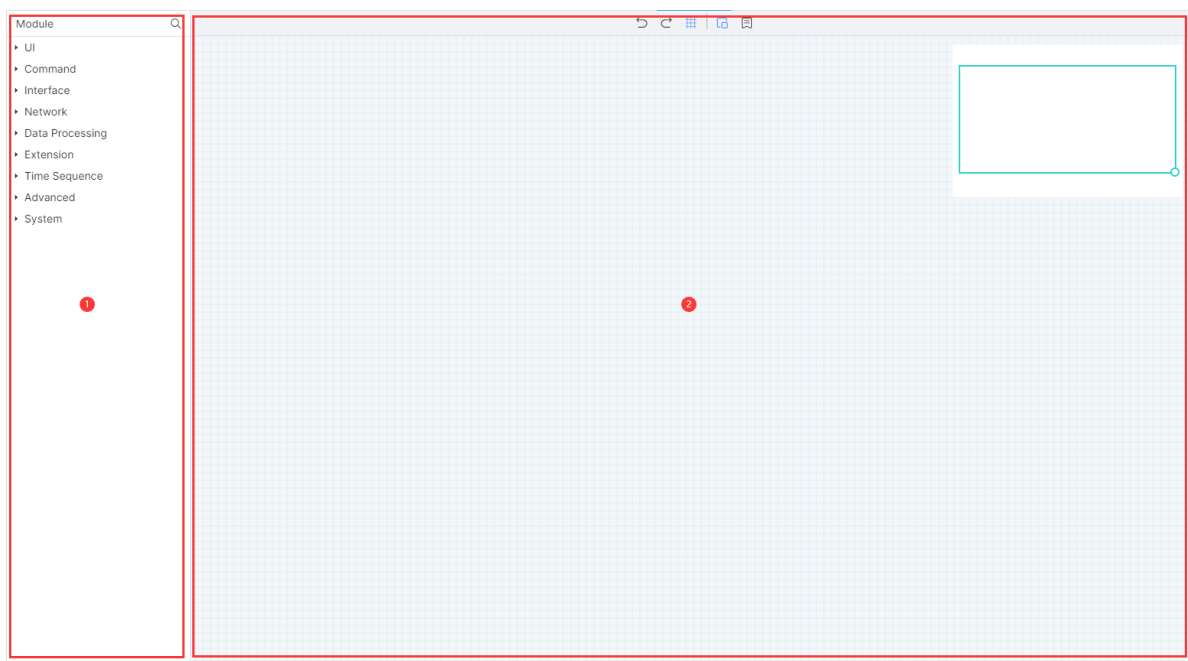
The image shows a configuration panel for a 'KnobToggle' module. It has a title bar with a blue icon and the text 'Basic Info'. Below the title bar, there is a field for '* Name' with the value 'KnobToggle'. A dashed horizontal line separates this from the 'Default' section, which contains a dropdown menu set to 'off'. To the right of the dropdown is a 'Disable' checkbox, which is currently unchecked.

Operating the knob switch on the button panel can trigger the subsequent logic of the ON/OFF action of the corresponding knob switch in the logic module. It is also possible to set the value of the knob switch and disable the attribute via the logic module.

Logic Configuration

Add different types of logic modules to the canvas, establish the calling relationship between modules by dragging lines, and modify the parameter configuration of logic modules according to them.

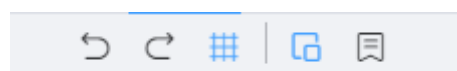
The logical configuration is described in different areas, as shown in the figure:



①: [Module]area; ②: Module design area.

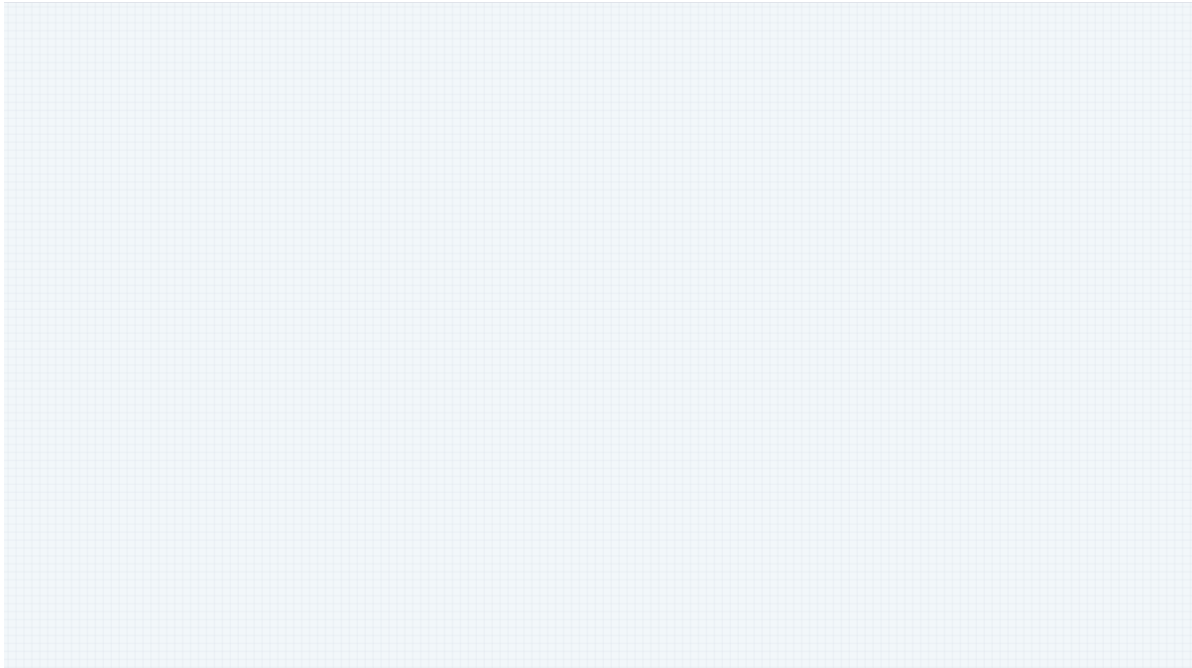
Tool Bar

In the [Logical Configuration] window, the toolbar is as shown in the figure:

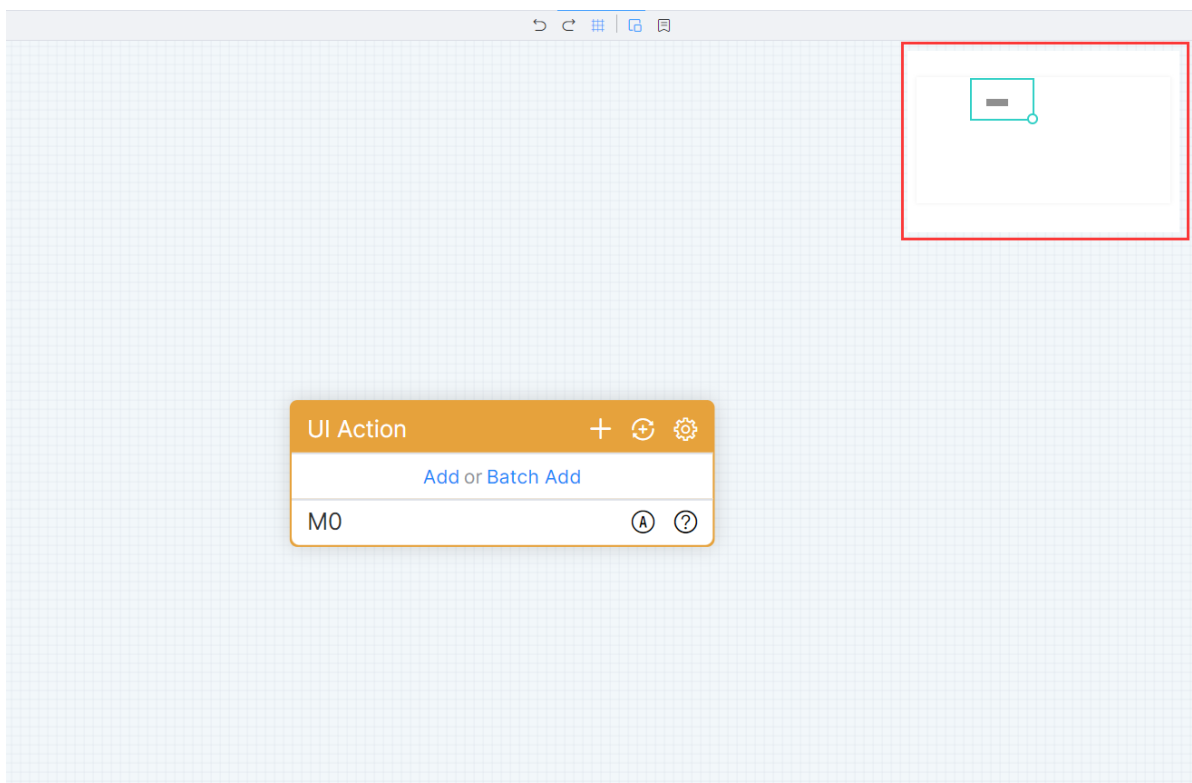


Click the Undo icon  or the shortcut keys < Ctrl+Z > to undo the operation of the module, click the Resume icon  or the shortcut keys < Ctrl+Y > to undo the operation just undone.

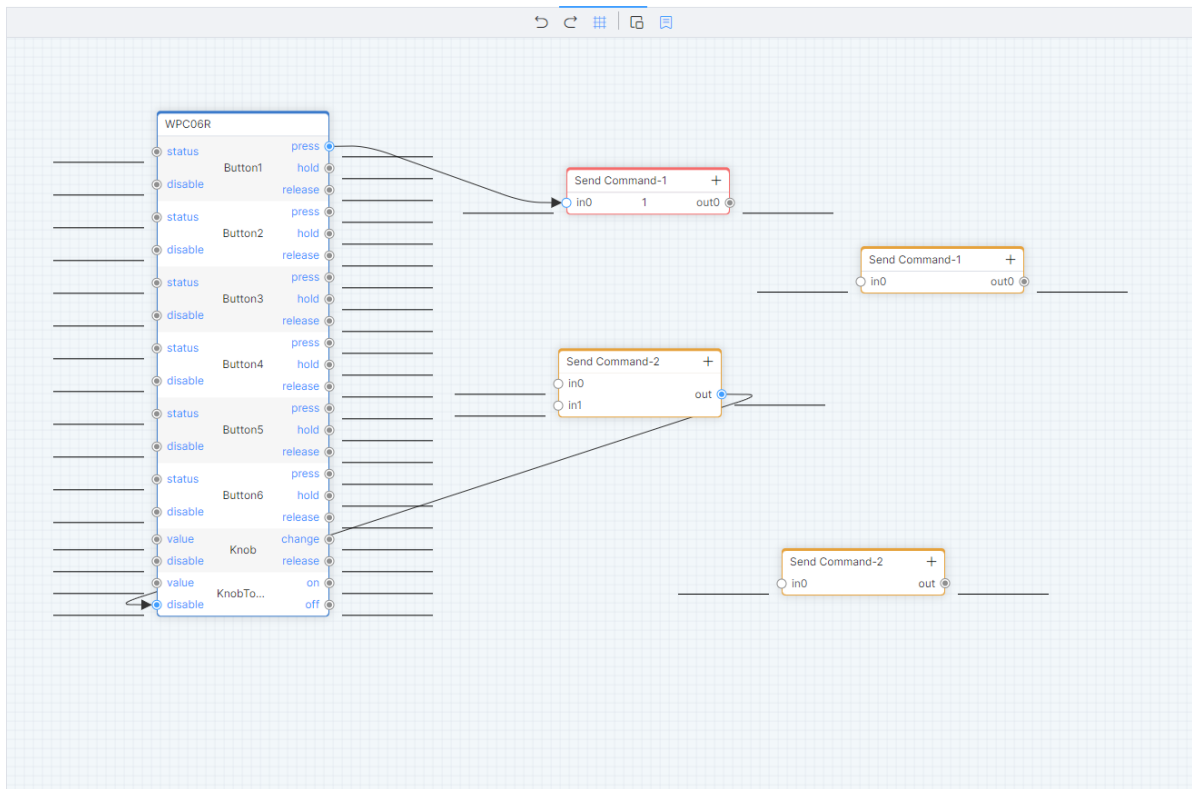
When the engineer drags each module in the page, the engineer can click the page grid icon icon  in the toolbar to perform more refined operations. After adding the grid, the canvas is as shown in the figure:



Click the icon , and a small map will be displayed in the upper right corner of the canvas. Engineers can quickly locate the required module range through the small map, as shown in the figure:



Click the global connection label switch icon , and all modules display the connection label edit box, as shown in the figure:

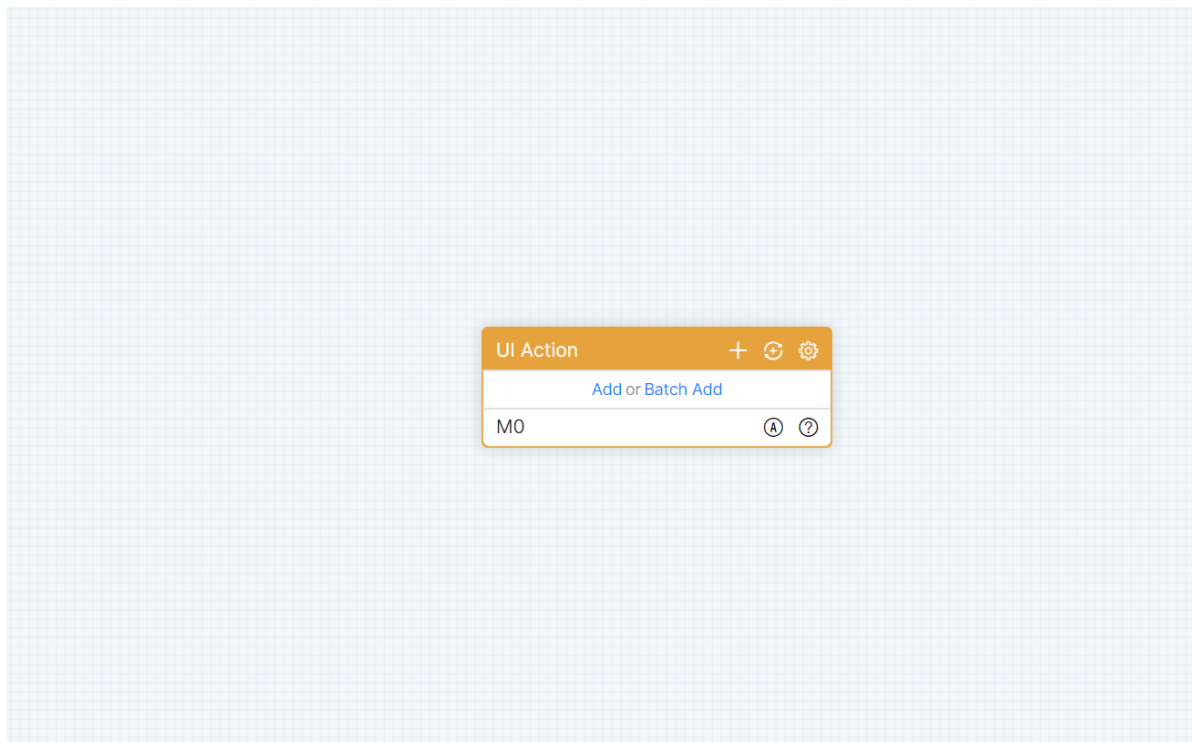


Logic Module

Set action logic number and attribute logic number in page and component properties, associate with UI logic module, and establish calling relationship between different logic modules by connecting lines or editing connection labels. Therefore, the subsequent logic of the related logic module can be triggered when the terminal is operated, and the properties of the page or component can be modified through the logic module when the project is running.

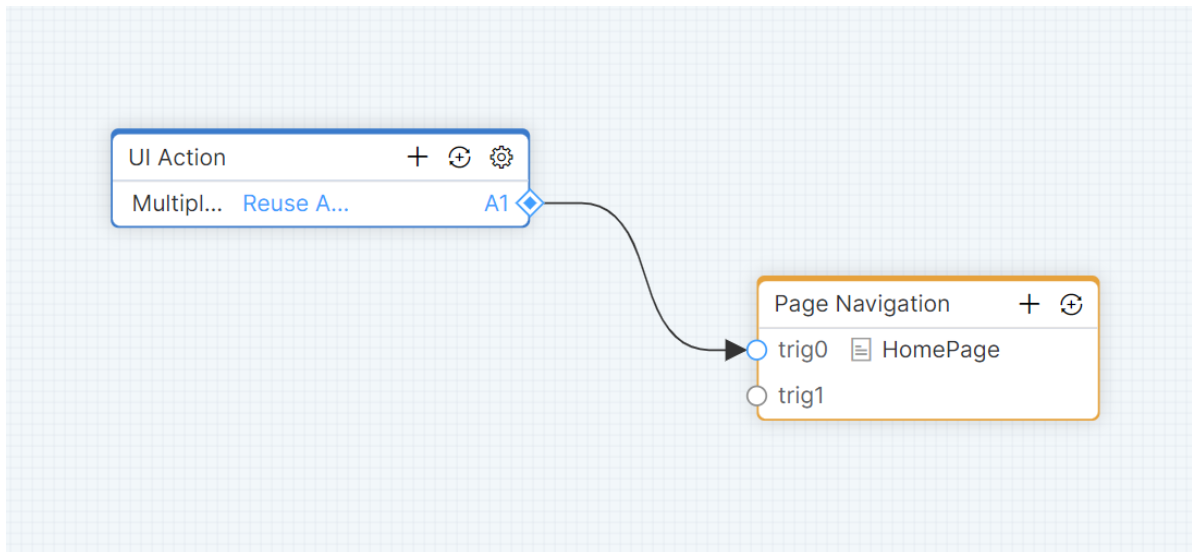
Add Module

Drag a component in the module library with the mouse, find a suitable position on the canvas, and release the mouse to complete the addition of the module.

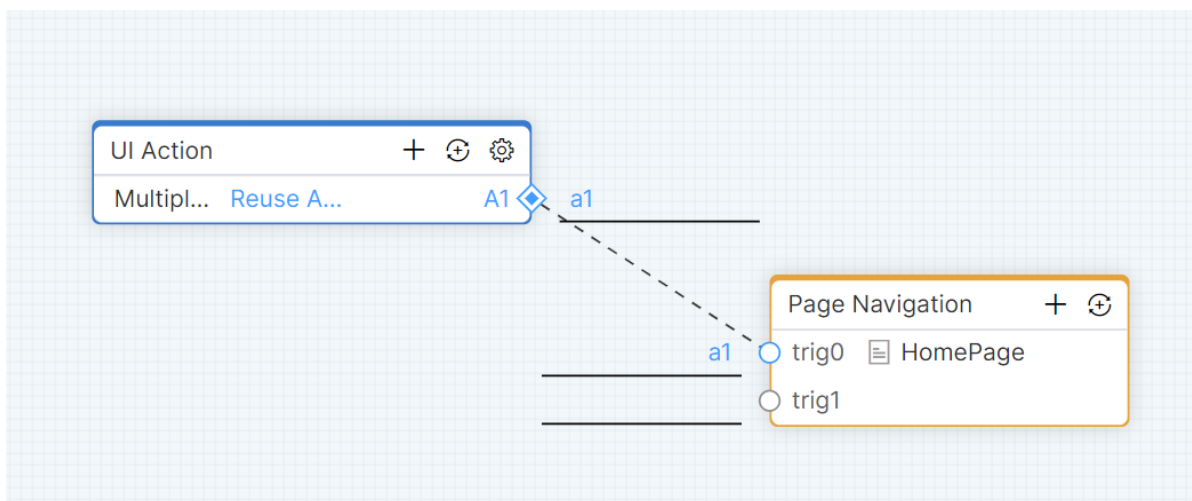


Connection Method

Call relationships between modules are established through wires. Move the cursor to the exit node of the module, press and drag for connection, press the space key for a long time to change the direction of the connection, move the arrow to the entrance node of the target module to end the connection, the connection succeeds, the node turns blue, and the call relationship is established.

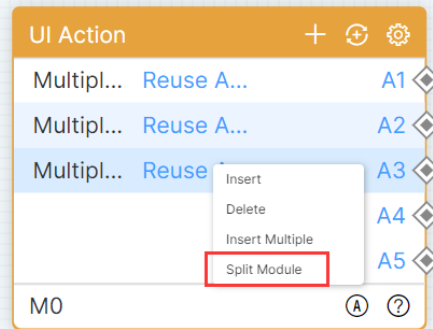


Call relationships between modules are established by connecting labels. When the module node connection labels are the same and the matching is successful, the node turns blue and the calling relationship is established.



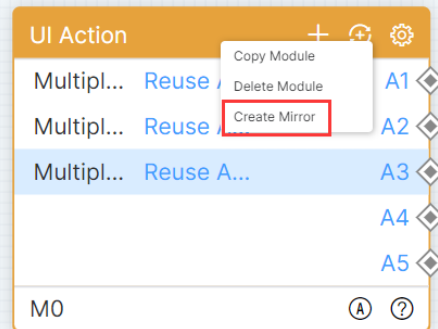
Split Module

Select a module row, right-click and select[Split Module]. Split the module after selecting the row. The connection and connection label in the split module need to be retained. Drag module and the same module overlap when merged, merged modules in the line and line labels need to be retained.



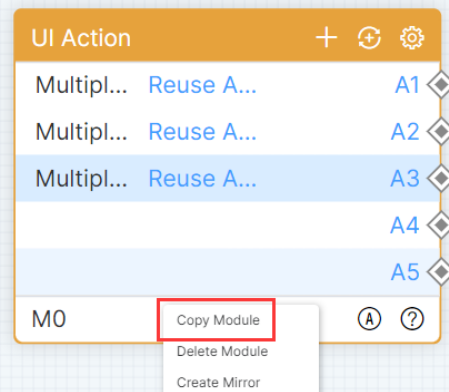
Create Mirror

Select the upper or lower area of the module, right-click and select [Create Mirror] to create a mirror module. The connection labels of each mirror module should be synchronized.



Cross-engineering Operation

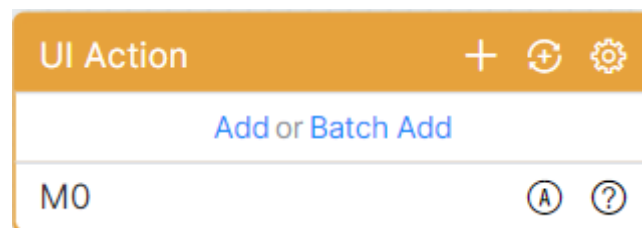
Copy and cut support cross-engineering operations. Select a single or multiple modules, right-click on [Copy Module] or Ctrl+C > to copy the module in the upper or lower area of the module. < Ctrl+X > means cut module, < Ctrl+V > means paste module.



Select a single module or multiple modules and hold down the left mouse button to drag the module anywhere on the canvas. Module details list floating window, configuration floating window, can be freely dragged position. A logical introduction to the module can be found in the help floating window. Select a module, right-click [Delete Module] or use < Ctrl+D >/delete > to delete the selected module. Move the cursor in the canvas, hold down Ctrl key, and scroll the mouse up and down to zoom in/out on the canvas and module.

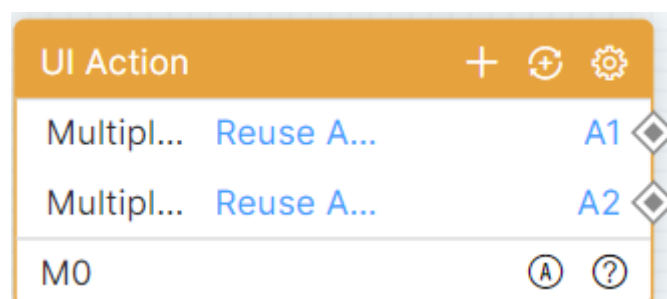
UI Interaction

Module name default display: UI interaction, can be modified. The initial state is as shown in the figure:

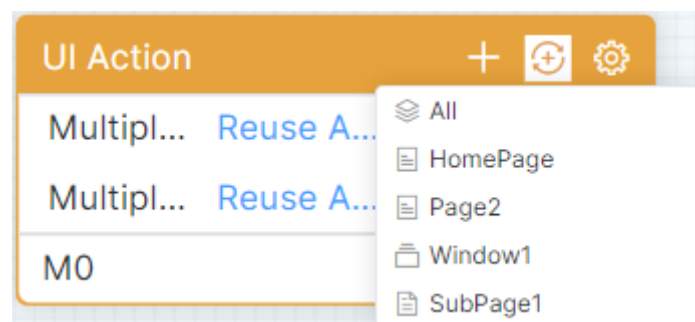


Click [Add] or [Batch Add] to add single or multiple pieces of data. Click on the icon to add a piece of data at the end of the module. Right-click the module row menu, click Insert or Insert Multiple Rows, and insert a single or multiple pieces of data below the selected row.

Click [Add] or [Batch Add] to add single or multiple pieces of data. Click on the icon  to add a piece of data at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple pieces of data below the selected row.



Click the synchronization icon  to synchronize all pages, or only one page. After selecting, the logical number of actions not in this module will be added. As shown in the figure:



Click the icon  to pop up the configuration floating window, fill in the start number and end number, click [OK] to refresh the module. As shown in the figure:

Settings

Start A

End A

OK

Double-click the row to pop up the details list floating window to modify the comment information. As shown in the figure:

Logical	Remark
A1	
A2	

This module only has exit nodes. When the action logic number set in the page or component exists in the UI interaction module, terminal operations will trigger the subsequent logic of the corresponding action logic number exit node.

UI Property Settings

Module name default display: Set UI properties. The initial state is as shown in the figure:

Property Setter

+

+

⚙

Add or Batch Add

M1

A

?

Click [Add] or [Batch Add] to add single or multiple pieces of data. Click the icon **+** to add a piece of data at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple pieces of data below the selected row.

Property Setter

+

+

⚙

P1 Multiple ... Reuse P...

P2 Multiple ... Reuse P...

M1

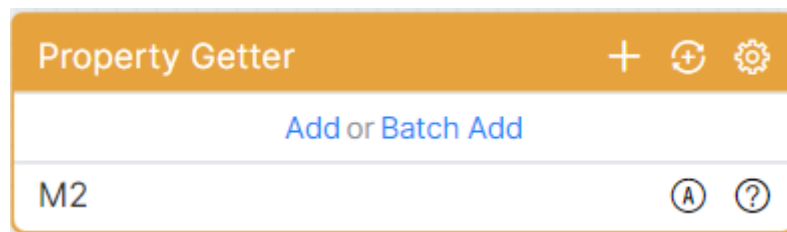
A

?

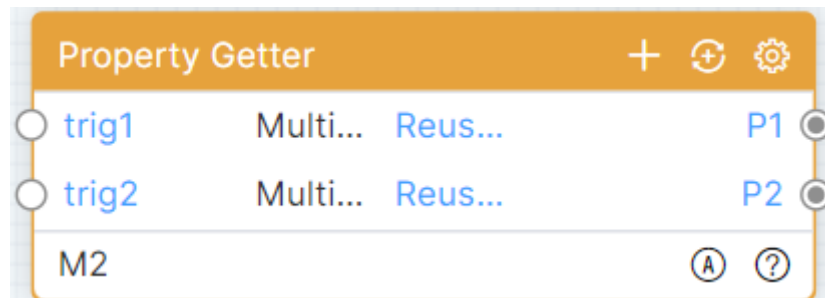
Please refer to the UI interaction module for synchronization, configuration, details and other operations of the UI property module. This module only has entry nodes. When the logical attribute number set in the page or component exists in this module, triggering the entry node through a connection or connection label will change the corresponding property value.

Output UI Property

Module name default display: Output UI properties. The initial state is as shown in the figure:



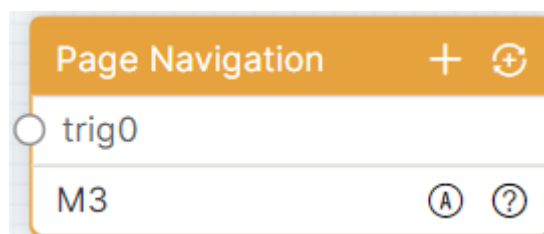
Click [Add] or [Batch Add] to add single or multiple pieces of data. Click the icon **+** to add a piece of data at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple pieces of data below the selected row.



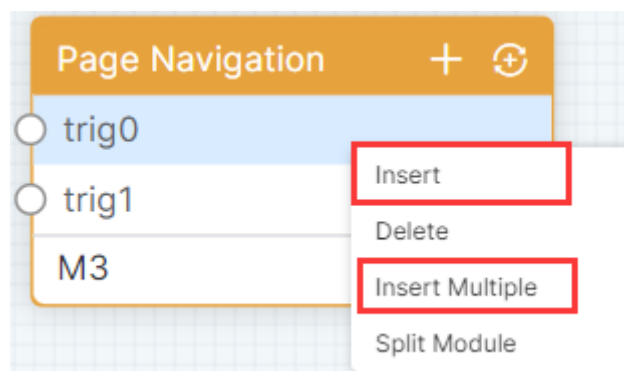
Please refer to the UI interaction module for synchronization, configuration, details and other operations of the output UI property module. An entry node of the module corresponds to an exit node. When the logical attribute number set in the page or component exists in this module, trigger the entry node through a connection or connection label, then the corresponding exit node will output the current property value.

Page Jump

Module name default display: page jump. The initial state will automatically open the list of details, as shown in the figure:

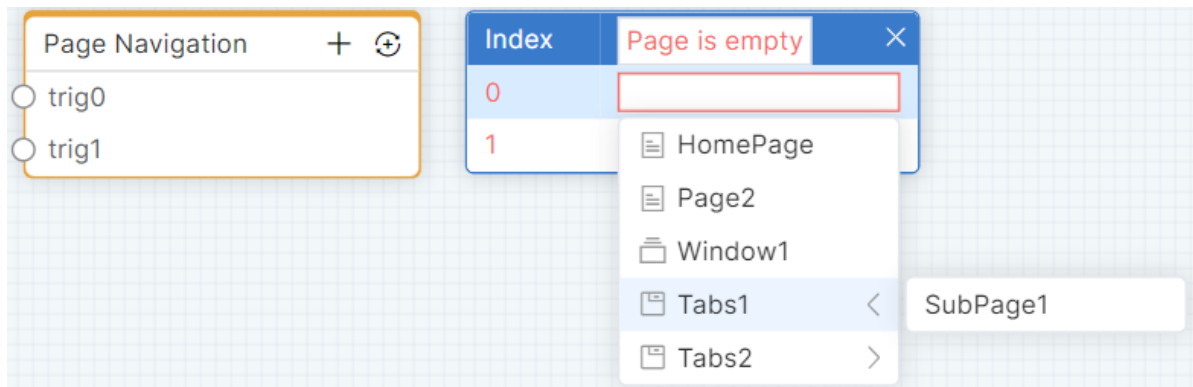


If there is no row data, click [Add] or [Batch Add] to add single or multiple data. Click the icon **+** to add a piece of data at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple pieces of data below the selected row.



Click the synchronization icon **+** to synchronize common pages and pop-ups in [Interface Design]. Only add pages and pop-ups that have not been synchronized yet. Subpages need to be added manually.

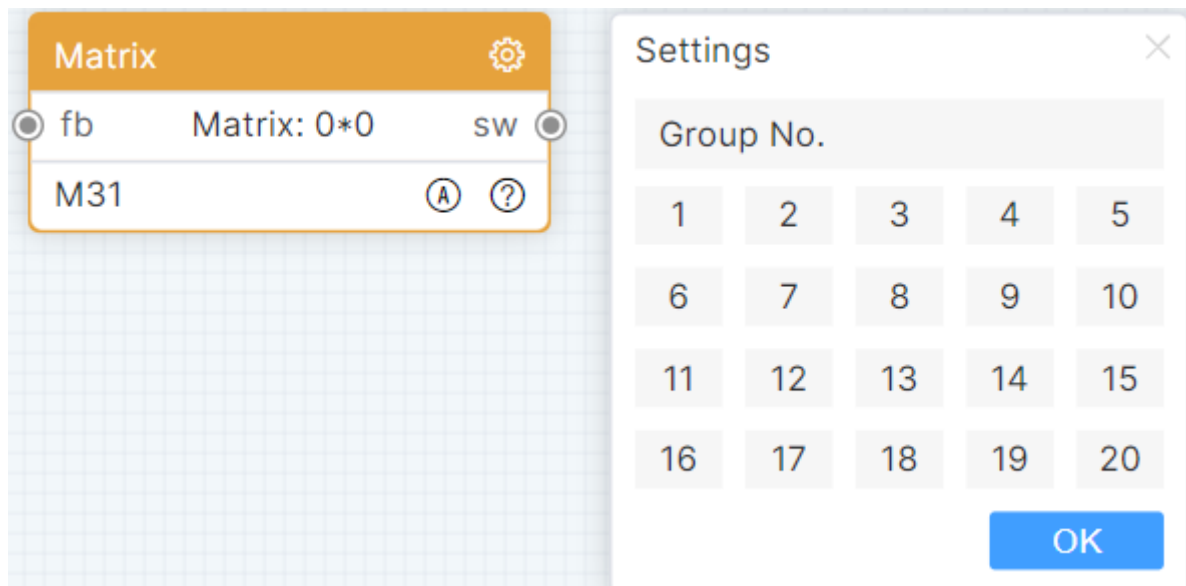
Double-click the row to pop up the details list floating window, you can configure the page, pop-up window, subpage. As shown in the figure:



This module only has an entry node, which is triggered by a connection or connection label. When the user interacts, it will jump to the configured page, pop-up window, or subpage container will switch to the configured subpage.

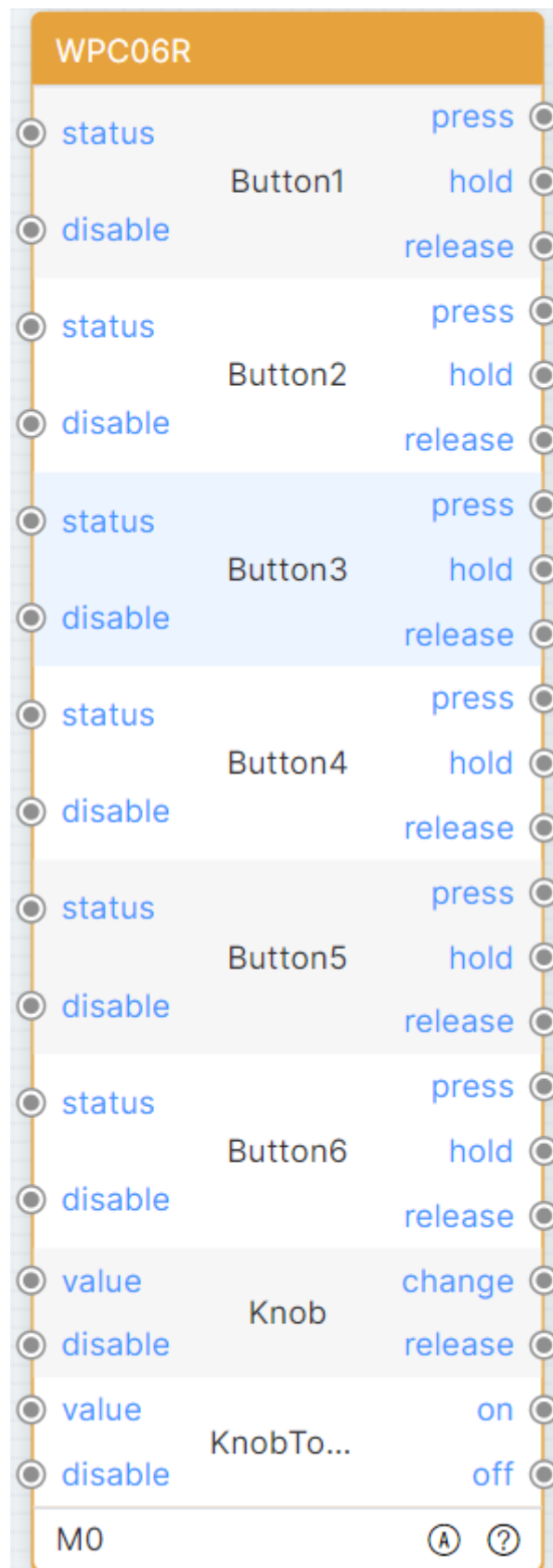
Matrix

Module name default display: Matrix. The initial state will automatically open the configuration floating window, as shown in the figure:



Click the icon  to pop up the configuration floating window, select the group number, and click [OK].

The feedback entry is triggered by a connection or a connection label, the data set is received (the first data is the input matrix communication code, and the second data is the output matrix communication code), and the corresponding relationship between input and output in the matrix group with the same group number is updated. When the user operates the matrix group at the terminal, the switching exit of the matrix module with the same group number is triggered, and the data set is output (the first data is the communication code of the current operation input matrix, and the second data is the communication code of the current operation output matrix).



The module structure is fixed, including 6 buttons information, 1 knob information, 1 knob switch information. Each button has 2 entry nodes: status, disable, and 3 exit nodes: press, hold, release; The knob has 2 entry nodes: value, disable, and 2 exit nodes: change, release; The knob switch has 2 entry nodes: value, disable, and 2 exit nodes: on, off. Trigger the entry node through the connection or connection label to change the property value of the corresponding button. The

user operates on the button panel to trigger the corresponding exit node of the module, thereby executing the subsequent logic.



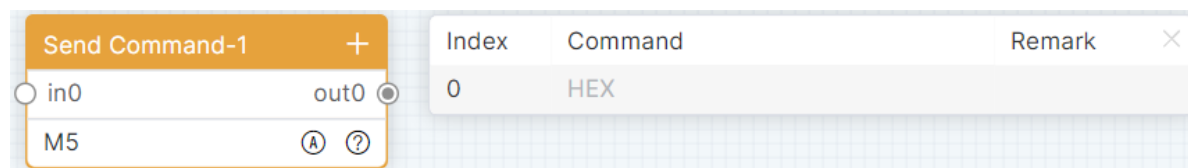
The module structure is fixed, with 8 buttons information. Each button message has 2 entry nodes: status, disabled, and 3 exit nodes: press, hold, and release. Trigger the entry node via a connection or connection label to change the property value of the corresponding button. The user operates on the button panel to trigger the corresponding exit node of the module, thereby executing the subsequent logic.

Command

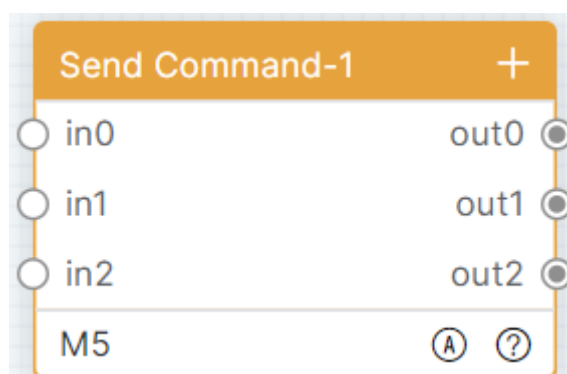
Mainly used to receive data, construct instructions and issue them.

Command Send-1

Module name default display: Command Send-1. The initial state will open the detailed list as shown in the figure:



If there is no row data, click [Add] or [Batch Add] to add single or multiple data. Click the icon **+** to add a piece of data at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple pieces of data below the selected row.



Double-click the row to pop up the details list floating window, you can modify the commands and comments. As shown in the figure:

Index	Command	Remark	×
0	HEX 6162		
1	HEX hello		
2	HEX how{0}you		
3	HEX hello{0}world{1}!		

An entry node corresponds to an exit node for this module. The command can be configured with fixed command (for example, hello) or dynamic command (for example, how{0}you, {0} represents the received data). Use as a constant when a directive contains wildcards {*}. Hold down Ctrl and click the left mouse button to set the matching symbol {N} as a constant. When the command content is incorrect, it can be saved successfully, but the problem list will prompt an error or warning message.

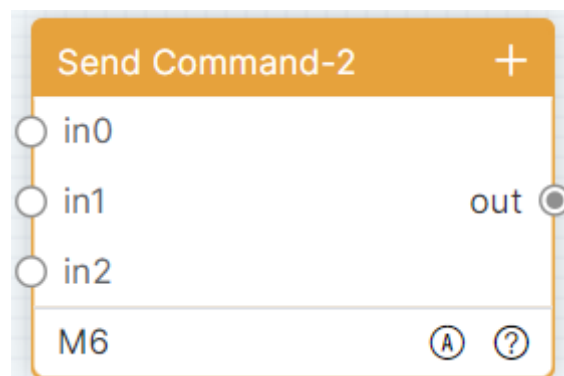
Command Send-2

Module name default display: Command Send-2. The initial state will open the detailed list as shown in the figure:

Send Command-2		+
in0	out	
M6	(A) (?)	

Index	Command	Remark	×
0	HEX		

For Command Send-2 Module adding rows and details list operation, please refer to Command Send-1. The module has multiple entry nodes and an exit node from which all commands are output.



Send Commands Sequentially

Module name default display: Send commands sequentially. The initial state will open the detailed list as shown in the figure:

Serial Send Command		+
in	out0	
M7	(A) (?)	

Index	Command	Delay(ms)	Remark	×
0	HEX	0		

For Sequential Send Command Module adding rows and command configuration operation, please refer to Command Send-1. Double-click the line to pop up the details list floating window, you can modify the commands, delay, comments. As shown in the figure:

Serial Send Command		+
616263	out0	
in {1}hello{0}	out1	
{1}61{0}	out2	
M7	(A) (?)	

Index	Command	Delay(ms)	Remark	×
0	HEX 616263	500		
1	HEX {1}hello{0}	1000		
2	HEX {1}61{0}	0		

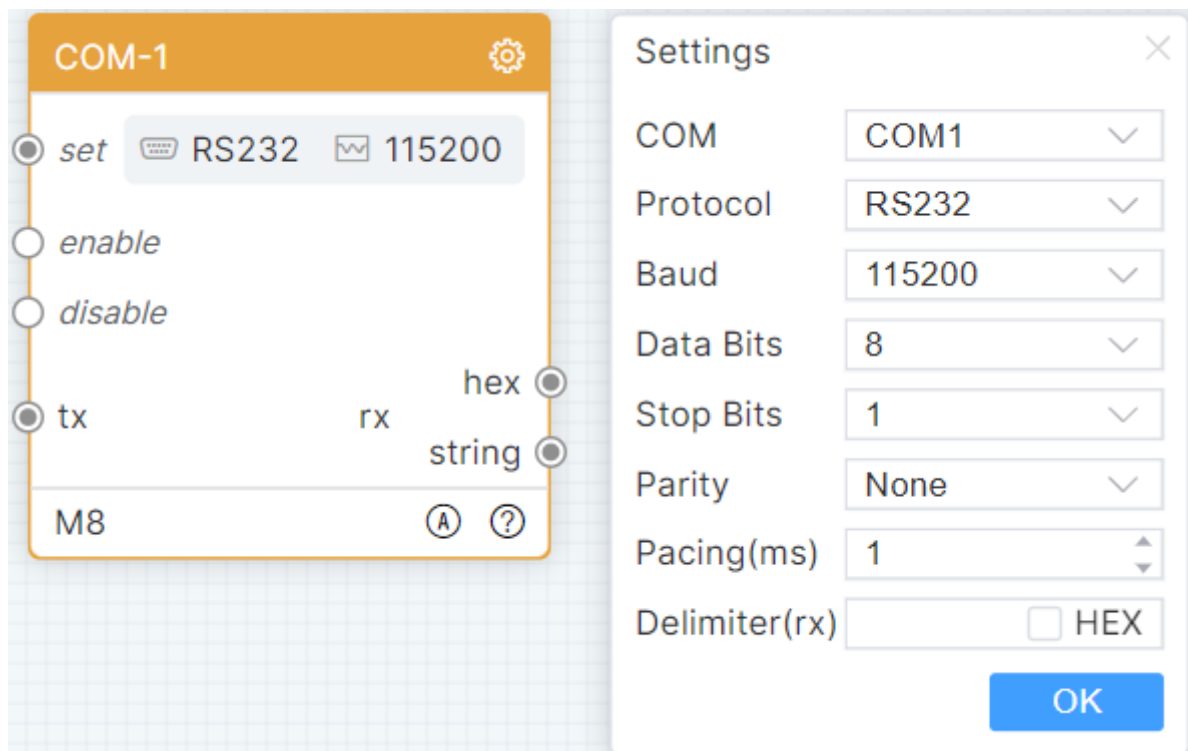
The module has one entry node and multiple exit nodes. After receiving the data, construct the command and output the command in sequence from top to bottom. If the delay time is configured, the corresponding delay time needs to be executed, and then continue to execute. When an command fails, the call is not interrupted and subsequent commands continue to be executed.

Port

For different models of controller, the type and number of port modules are also different. For example, TS04C has only one serial port-1 module.

Serial Port

Module name default display: Serial Port-N. The initial state will automatically open the configuration floating window, as shown in the figure:

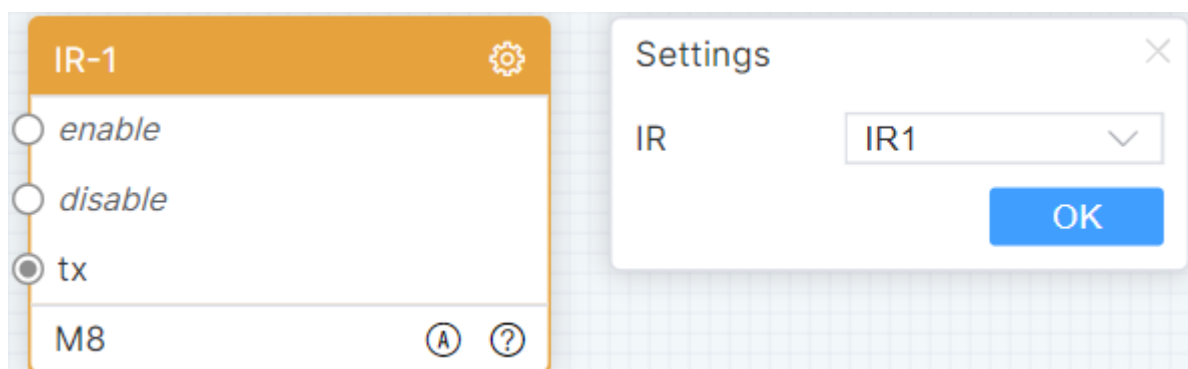


Click the icon  to pop up the configuration floating window, select port assignment, serial protocol, baud rate, data bit, stop bit, parity bit, fill in interval and separator, and click [OK].

Trigger the setting entry and dynamically set serial port parameters according to the received data. This module is enabled by default, triggering disabled entry, unable to receive and send data. Trigger the send entry to send data to the controlled device. When the module receives data, it can output it in hexadecimal or string format.

IR

Module name default display: IR-N. The initial state will automatically open the configuration floating window, as shown in the figure:

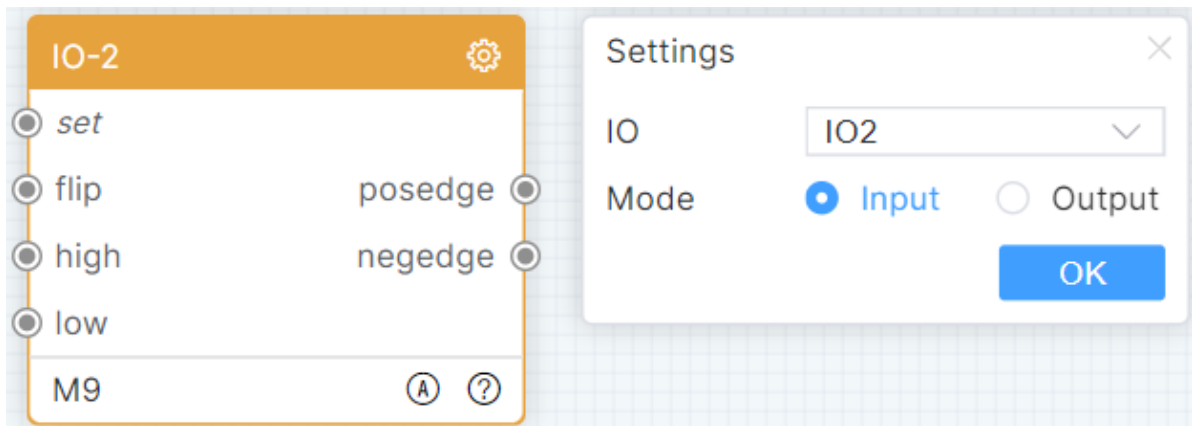


Click the icon  to pop up the configuration floating window, select port assignment, and click [OK].

This module is enabled by default, triggering disabled entry and unable to send data. Triggering the sending entry and sending infrared data in CCF code format to the controlled device.

IO

Module name default display: IO-N. The initial state will automatically open the configuration floating window as shown in the figure:

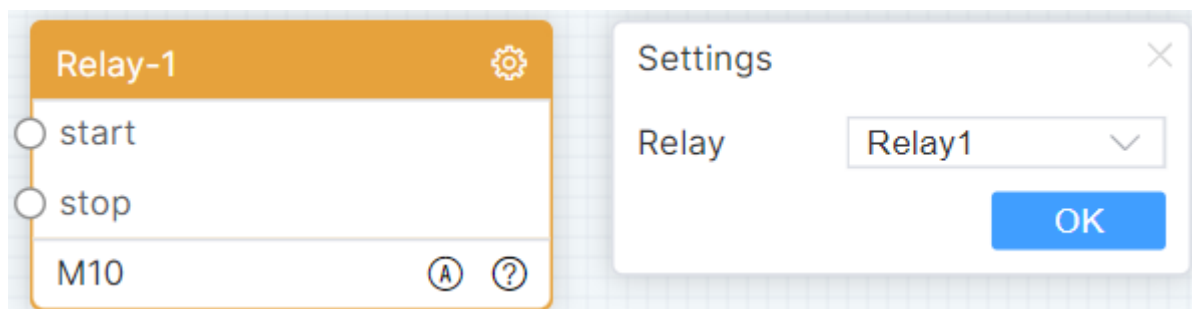


Click the icon  to pop up the configuration floating window, select port assignment and module, then click [OK].

Trigger the setting entry to dynamically set IO mode according to the received data. Trigger the flip entry, and change level. Trigger the high entry, set to high. Trigger the low entry, set to low. The module receives a rising edge message output 1 and a falling edge message output 0.

Relay

Module name default display: Relay-N. The initial state will automatically open the configuration floating window as shown in the figure:



Click the icon  to pop up the configuration floating window, select port assignment, and click [OK].

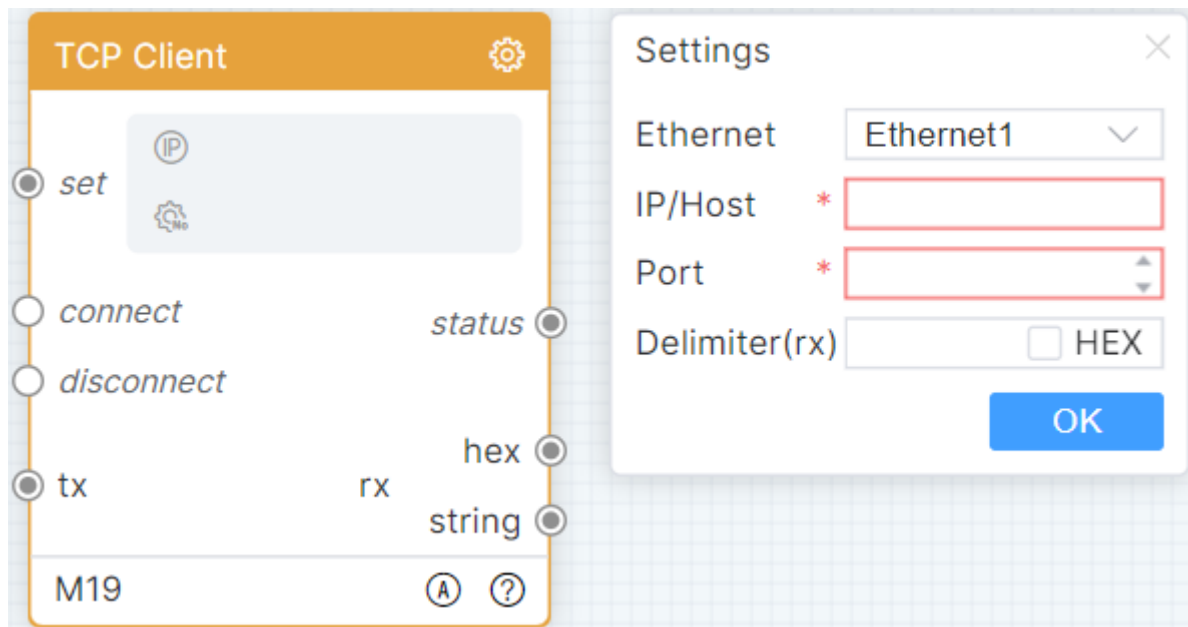
Trigger opens the inlet, control relay opens. Trigger closes inlet, control relay closes.

Trigger to open the entrance and control the relay to turn on. Trigger to close the entrance and control the relay to turn off.

Network

TCP Client

Module name default display: TCP Client. The initial state will open the configuration floating window as shown in the figure:

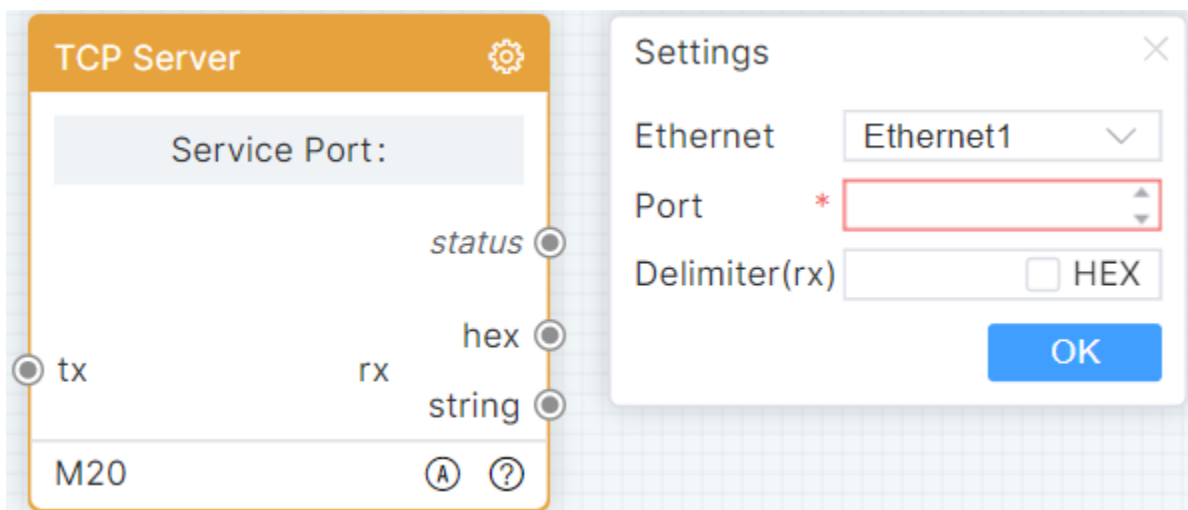


Click the icon  to pop up the configuration floating window, select the network port, fill in the IP/Host, port and separator, then click [OK].

Trigger the setting entry to dynamically set TCP client communication parameters according to the received data. The module automatically establishes a connection by default, trigger a disconnection entry, and disconnects. Trigger the sending entry to send data to the controlled device. Output status information when connection changes occur. When the module receives data, it can output it in hexadecimal or string format.

TCP Server

Module name default display: TCP Server. The initial state will automatically open the configuration floating window, as shown in the figure:

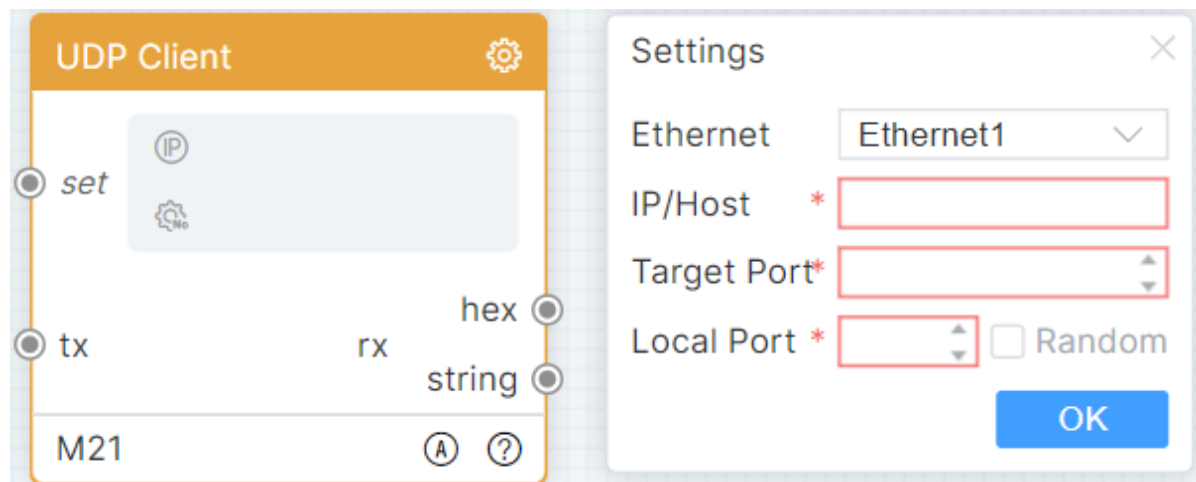


Click the icon  to pop up the configuration floating window, select the network port, fill in the port and separator, then click [OK].

Trigger the send entry to send data to the controlled device. Output status information when connection changes. When the module receives data, it can output it in hexadecimal or string format.

UDP Client

Module name default display: UDP Client. The initial state will automatically open the configuration floating window, as shown in the figure:

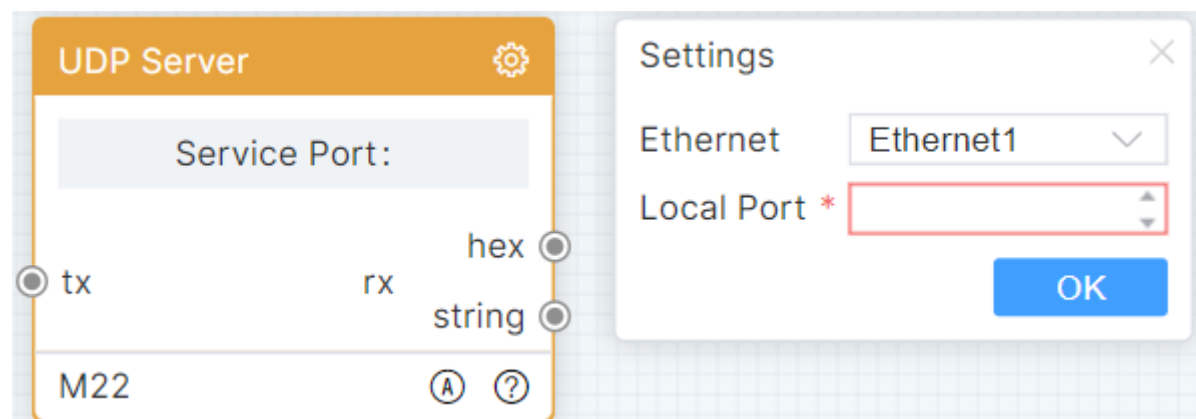


Click the icon  to pop up the configuration floating window, select the network port, fill in the IP/Host, target port and local port, then click [OK].

Trigger the setting entry to dynamically set UDP client communication parameters according to the received data. Trigger the sending entry to send data to the controlled device. When the module receives data, it can output it in hexadecimal or string format.

UDP Server

Module name default display: UDP Server. The initial state will automatically open the configuration floating window, as shown in the figure:

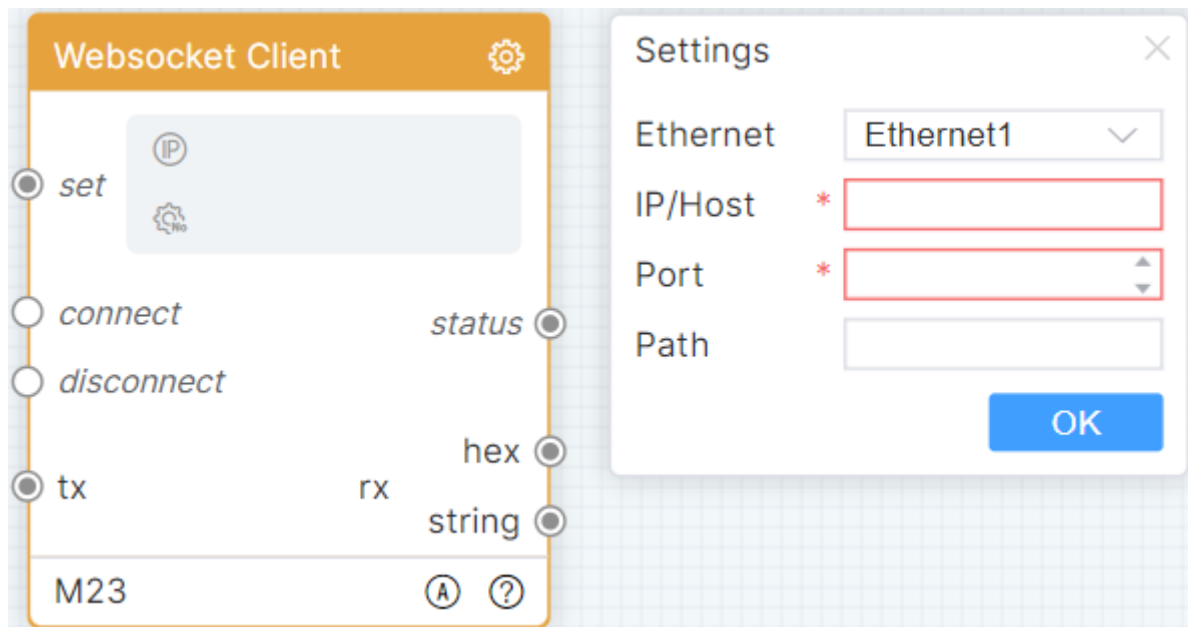


Click the icon  to pop up the configuration floating window, select the network port, fill in the local port, then click [OK].

Before sending data for the first time, the client needs to send data once before triggering the sending entry to send data to the controlled device. When the module receives data, it can output it in hexadecimal or string format.

Websocket Client

Module name default display: Websocket Client. The initial state will automatically open the configuration floating window, as shown in the figure:

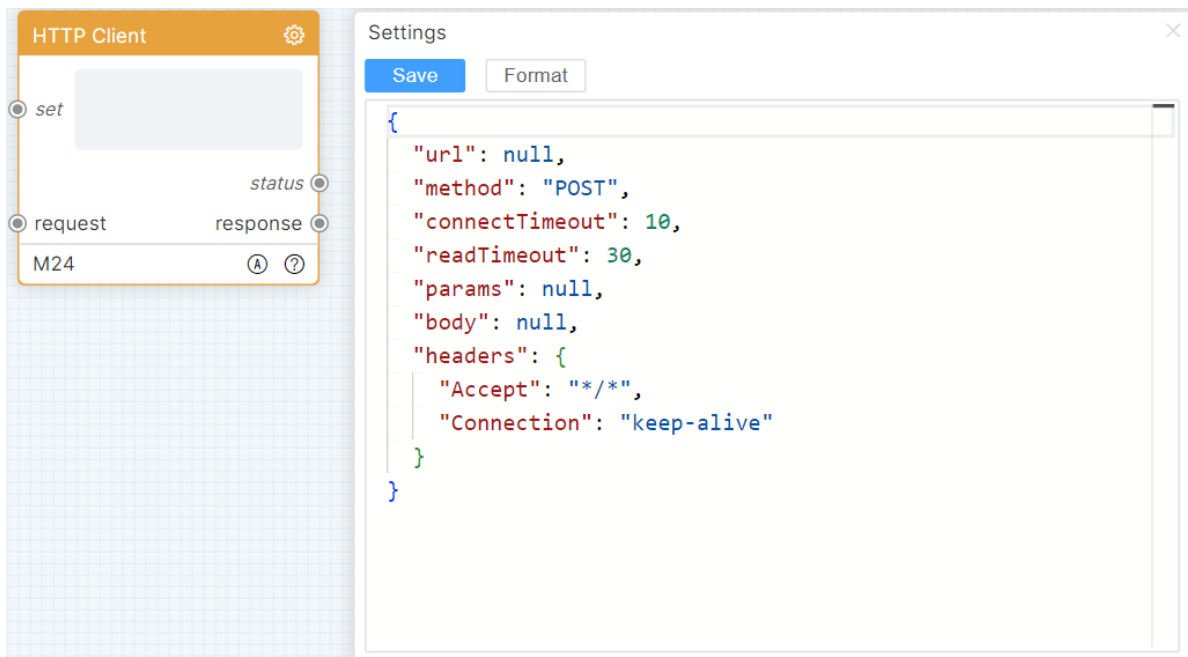


Click the icon  to pop up the configuration floating window, select the network port, fill in the IP/Host, port and local path, then click [OK].

Trigger the setting entry, dynamically set the Websocket client communication parameters according to the received data. The module automatically establishes a connection by default, triggers a disconnection entry, and disconnects. Automatic reconnection when connection is abnormally disconnected. Trigger the sending entry to send data to the controlled device. Output status information when connection changes. When the module receives data, it can output it in hexadecimal or string format.

Http Client

Module name default display: Http Client. The initial state will automatically open the configuration floating window, as shown in the figure:

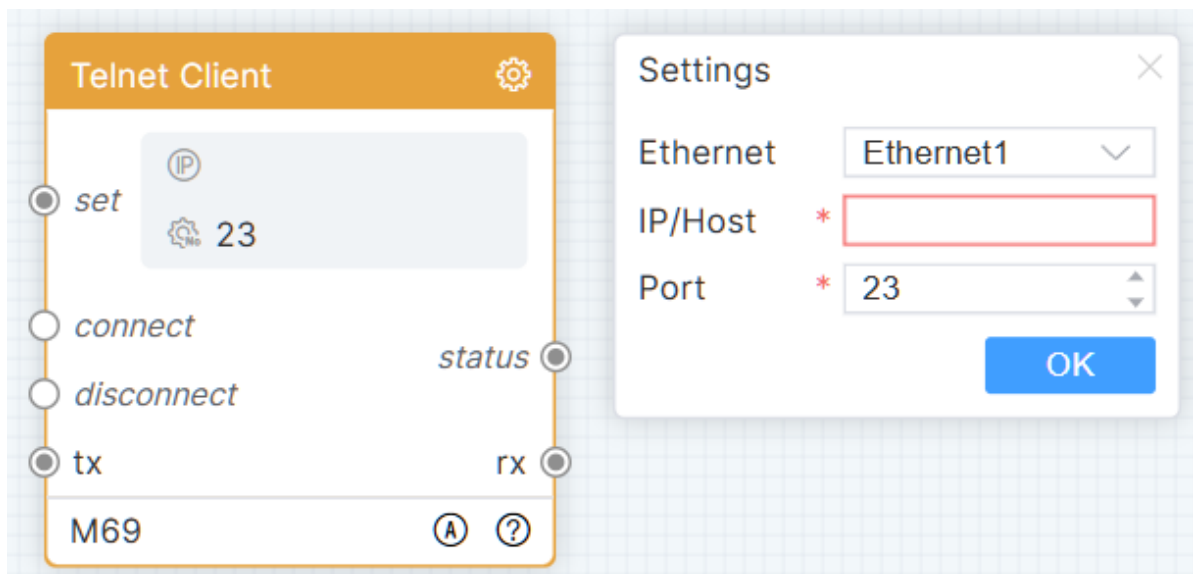


Click the icon  to pop up the configuration floating window, fill in the parameter values, then click [Save].

Trigger the setting entry to dynamically set Http client communication parameters based on the received data. Trigger the request entry to send data to the HTTP server. Output status code when communicating. Output response data as a string.

Telnet Client

Module name default display: Telnet Client. The initial state will open the configuration floating window as shown in the figure:



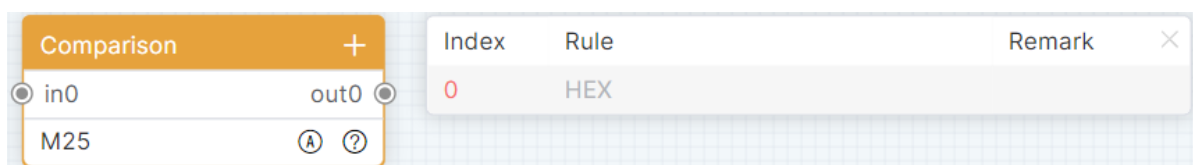
Click the icon  to pop up the configuration floating window, select the network port, fill in the IP/Host and port, then click [OK].

Trigger the setting entry to dynamically set Telnet client communication parameters according to the received data. The module automatically establishes a connection by default, trigger a disconnection entry, and disconnects. Trigger the sending entry to send data to the controlled device. Output status information when connection changes occur. When the module receives data, it can output it in string format.

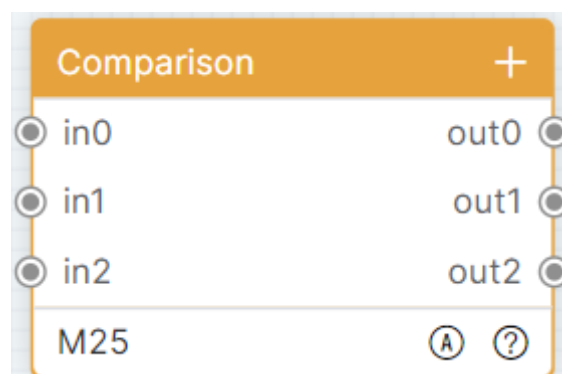
Data Process

Data Matching

Module name default display: Data Matching. The initial state will automatically open the configuration floating window, as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row of data at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



Double-click the row to pop up the details list floating window, and you can modify the matching rules and comments. As shown in the figure:

Index	Rule	Remark	×
0	HEX 6262		
1	HEX {*}test{}		
2	HEX {1}aaa		

An entry node of this module corresponds to an exit node. The matching rule can be a fixed value (for example: hello) or an expression (for example: hello{*}). Only wildcards can be used in the expression. Using {N} will only be regarded as a constant. Hold down Ctrl and click the left mouse button to set wildcard characters as constants. Match the received data with the matching rule, and if the match succeeds, the received data will be output, and if the match fails, no output will be made.

String Concatenation

Module name default display: String Concatenation. The initial state will automatically open the configuration floating window, as shown in the figure:

Index	Constant Value	Param	Constant
0	HEX	☐	☒

This module will have at least one row of data. Click the icon **+** to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.

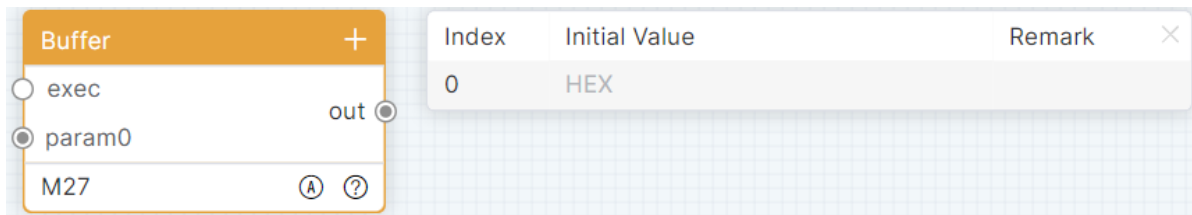
Index	Constant Value	Param	Constant
0	HEX	☐	☒
1	HEX 6162	☒	☒
2	HEX {0}{*test123	☒	☒
3	HEX	☐	☒

Double-click the line to pop up the details list floating window, you can set the parameter type, when it is a parameter, the constant value is grayed out, and the line has an entry node; when it is a constant, the constant value can be modified, and the line has no entry node.

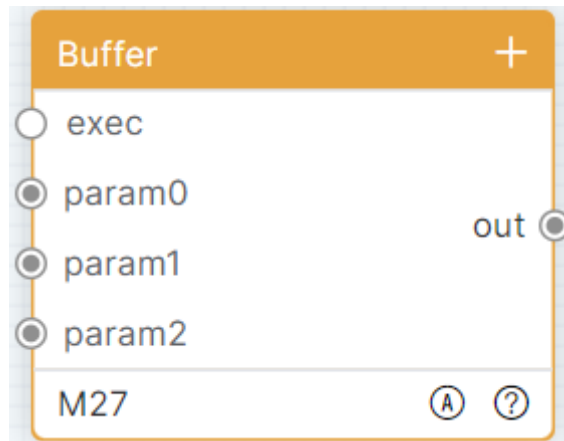
Constant values are used only as constants when there are matching characters in them. When the execution entry is triggered, the module assembles parameters from top to bottom and outputs them in hexadecimal or string format.

Data Cache

Module name default display: Data Cache. The initial state will automatically open the configuration floating window, as shown in the figure:



This module will have at least one row of data. Click the icon to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



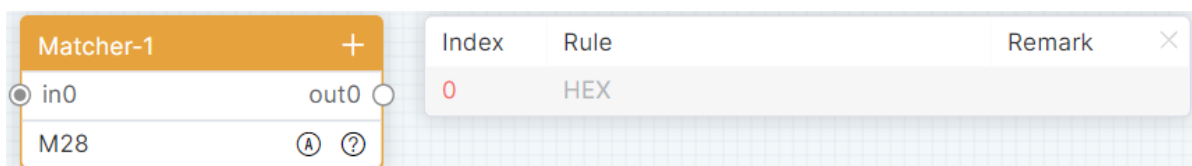
Double-click the row to pop up the details list floating window, you can modify the initial value and comments.

Index	Initial Value	Remark
0	HEX 6162	
1	HEX {0}test{*}	
2	HEX	

When there is a match in the initial value, it is only used as a constant. When triggering parameter entry, receive and update corresponding cache data. When an execution entry is triggered, the module assembles a data set from top to bottom and outputs the data set.

Data Extraction-1

Module name default display: Data Extraction-1. The initial state will automatically open the configuration floating window, as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.

Index	Rule	Remark
0	HEX {0}test{1}{*}	
1	HEX 61{0}62	
2	HEX hello{1}	
3	HEX 6161	

Double-click the row to pop up the details list floating window, and you can modify the matching rules and comments.

An entry node of this module corresponds to an exit node. The matching rule can be a fixed value (for example: hello) or an expression (for example: hello{0}).{N} in the expression must start from 0 and cannot be repeated.{*} can be repeated. Trigger an entry to extract data from received data according to a matching rule, and output no data if no data is extracted; Extract one piece of data and output the data; Extract multiple pieces of data and output a data set.

Data Extraction-2

Module name default display: Data Extraction-2. The initial state will automatically open the configuration floating window, as shown in the figure:

Index	Rule	Remark
0	HEX	

Refer to Data Extraction-1 for adding rows and detail lists in the Data Extraction-2 module.

Index	Rule	Remark
0	HEX 123	
1	HEX {0}{1}	
2	HEX 61{0}62	

The module has one entry node and several exit nodes. Trigger an entry port to extract the received data with multiple matching rules, thereby extract different data.

Data Fractionation

Module name default display: Data Fractionation. The initial state is as shown in the figure:

Refer to Data Extraction-1 for adding rows to this module. Double-click the row to pop up the details list floating window, you can modify the remarks.

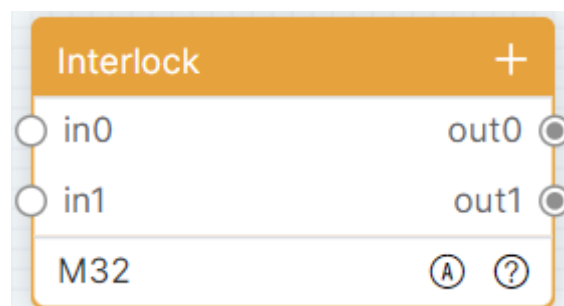
Index	Remark	×
0		
1		
2		
3		

The module has one entry node and several exit nodes. Trigger the entry port to split the received data set into multiple parameters and output.

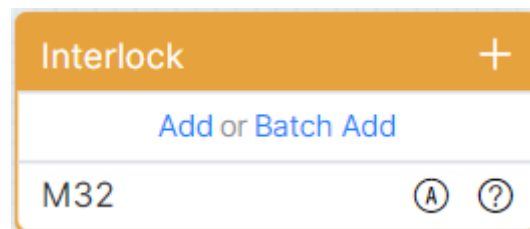
Extend

Interlock

Module name default display: Interlock. The initial state is as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



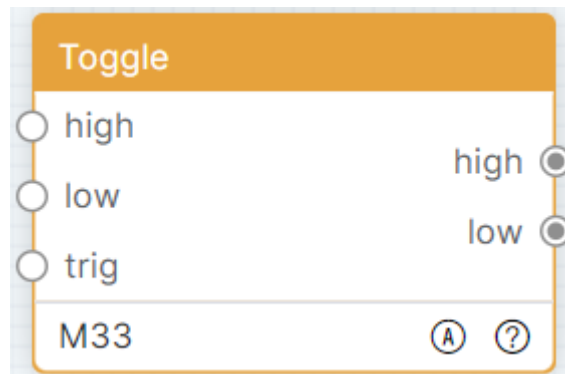
Double-click the row to pop up the details list floating window, you can modify the comments.

Index	Remark	×
0		
1		
2		

One entry node corresponds to one exit node in this module. Trigger any entry node, corresponding to output 1 at the exit, and output 0 at the other exits.

Trigger

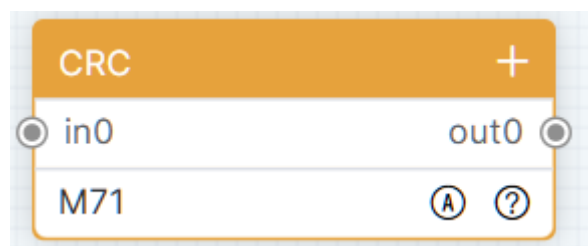
Module name default display: Trigger. The initial state is as shown in the figure:



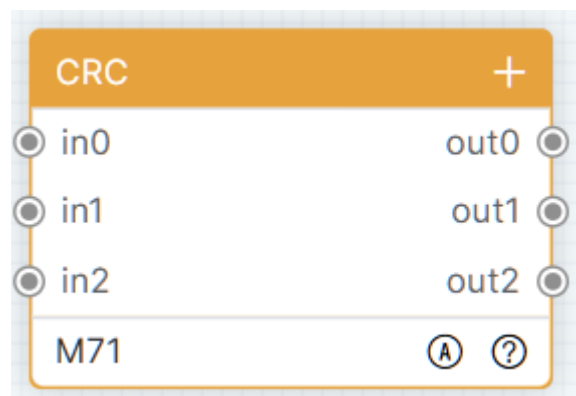
Trigger the high entry, set the state to high and outputs 1. Trigger the low entry, set the state to low and output 0. Trigger the [Trigger] entry, change the current state and output. For example: the module default state is low, directly trigger the [trigger] entry, set the state to high and output 1.

CRC

Module name default display: CRC. The initial state will open the detailed list as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



Double-click the row to pop up the details list floating window, you can modify the comments.

Index	Rules
0	This field is required
1	
2	CRC-16/MODBUS Checksum

Timing

Delayer

Module name default display: Delayer. The initial state will open the detailed list as shown in the figure:

Delay

+

in0

500ms

out0

M34

A

?

Index

Delay(ms)

×

0

500

Index	Delay(ms)
0	500

If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.

Delay		+
in0	500ms	out0
in1	500ms	out1
in2	500ms	out2
M34		A ?

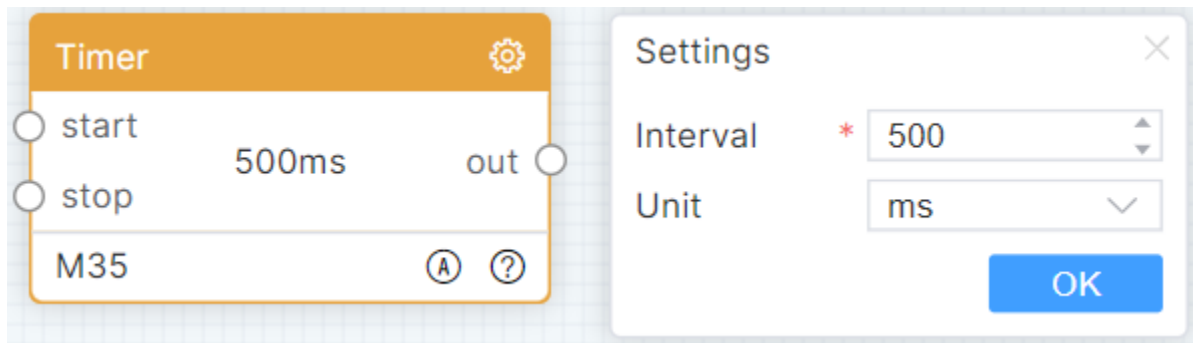
Double-click the row to pop up the details list floating window, you can modify the comments.

Index	Delay(ms)
0	1
1	500
2	1000

The module does not do any processing to the data, triggers the input port, executes the delay of the corresponding row, and outputs the original data from the corresponding row outlet.

Timer

Module name default display: Timer. The initial state will automatically open the configuration floating window, as shown in the figure:



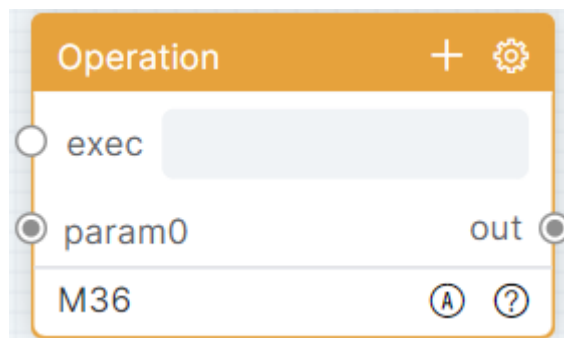
Click the icon  to pop up the configuration floating window, fill in the interval time, select the time unit, and click [OK].

The module is closed by default, trigger to open the entry, periodically execute the subsequent logic of the outlet, trigger to close the entry, and close the timer.

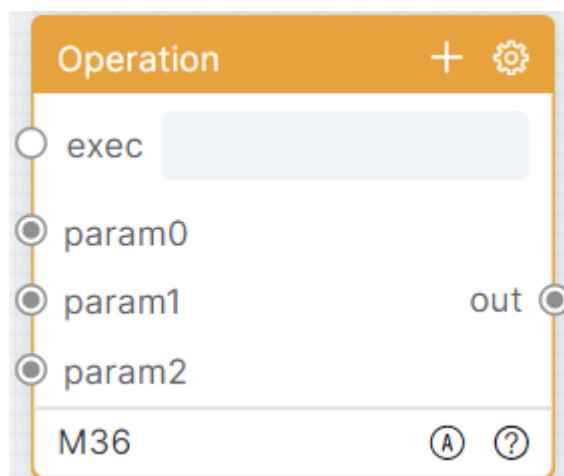
Advanced

Operation

Module name default display: Operation. The initial state is as shown in the figure:



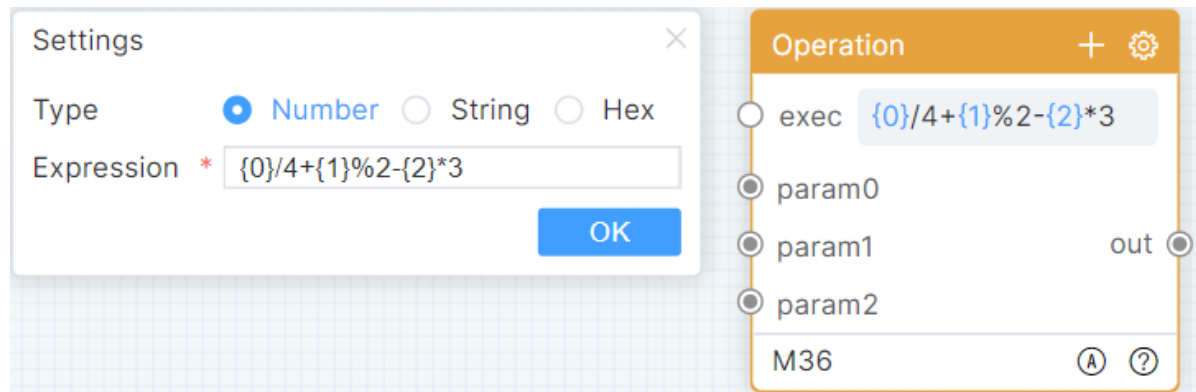
This module will have at least one row of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



Double-click the row to pop up the details list floating window, you can modify the comments.

Index	Remark
0	
1	
2	

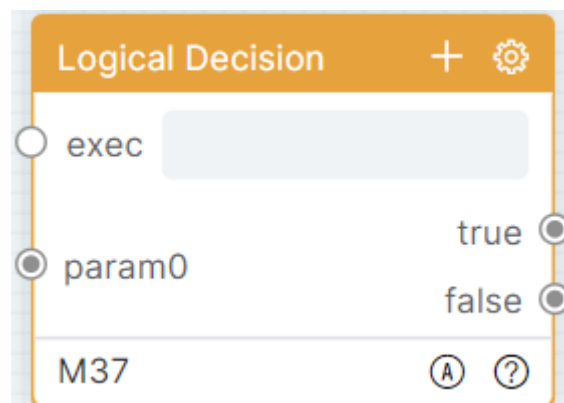
Click the icon  to pop up the configuration floating window, select the operation mode, fill in the operation logic, and click [OK].



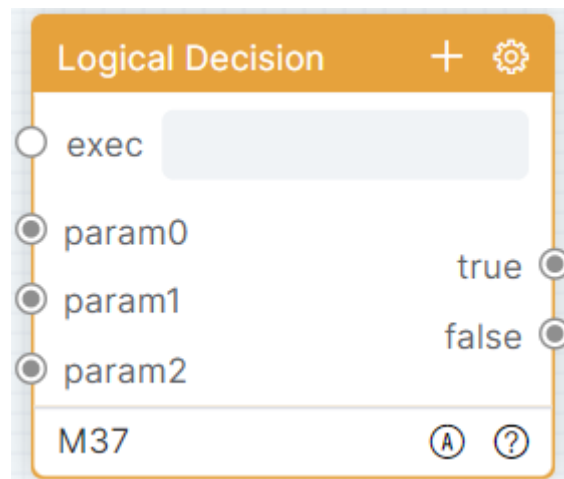
The module has multiple entry nodes and one exit node. Trigger parameter entry to receive and update cache data. Trigger execution entry, perform operation according to operation logic expression, and output operation result.

Logic Judgment

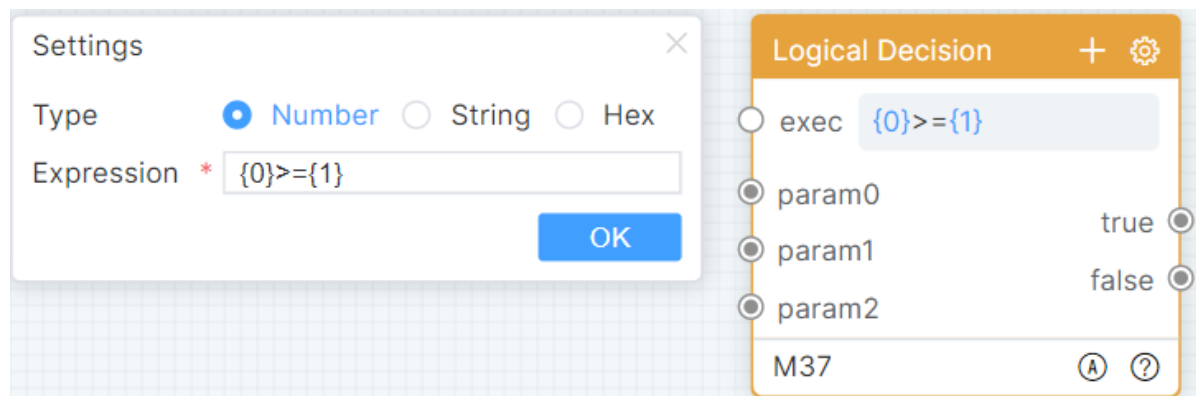
Module name default display: Logical Judgment. The initial state is as shown in the figure:



This module will have at least one row of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



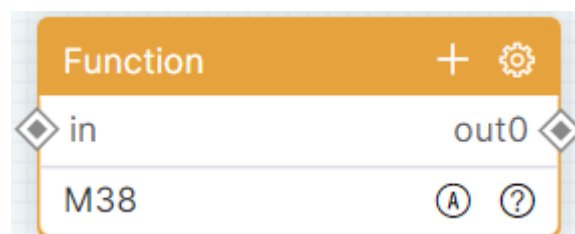
Refer to Operation for the module details list operations. Click the icon  to pop up the configuration floating window, select the operation mode, fill in the operation logic, and click [OK].



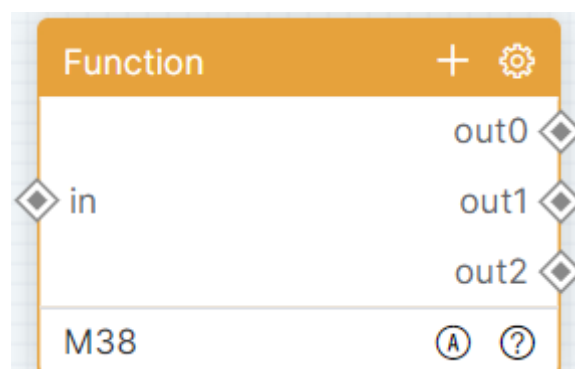
Trigger parameter entry to receive and update cache data. Trigger the execution entry, judge the operation according to the judgment logic expression, and output the data set (the first data is the expression result, 1 is true, 0 is false. Then parameter 0-parameter N received data).

Functions

Module name default display: Functions. The initial state is as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



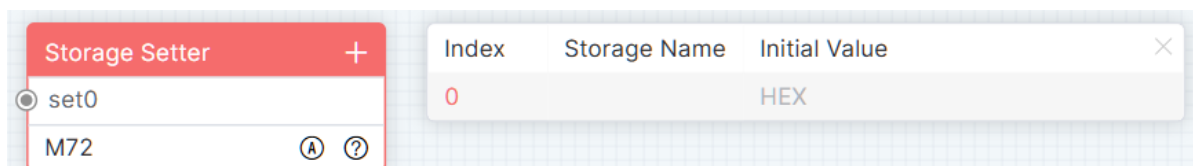
Refer to Operation for the module details list operations. Click the icon  to pop up the configuration floating window, write custom functions in JavaScript script language, and click the [Save] button.



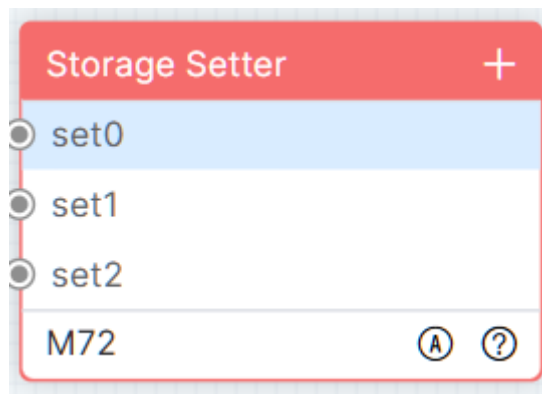
The module can use built-in functions. params is a string array of input parameters, params[0] indicates the first parameter; \$Log(message) prints the contents to the console; \$TriggerOutString(int, string) outputs data in string format, int indicates the [out] sequence number; \$TriggerOutHex(int, string) outputs data in hexadecimal format. The module has one entry node and multiple exit nodes. Trigger an entry port, receive data and execute a function, with the output specified by the function logic. The module exit is wired, but does not use \$TriggerOutString or \$TriggerOutHex to output data, so the exit will not output data and will not execute subsequent logic.

Storage Setter

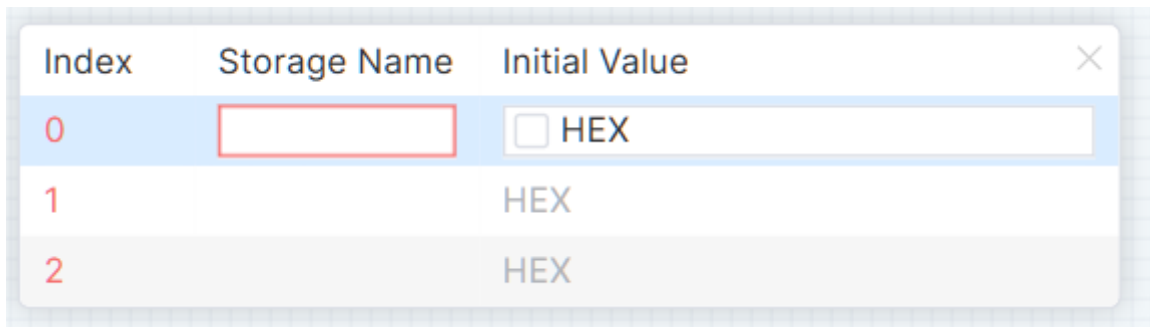
Module name default display: Delayer. The initial state will open the detailed list as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.

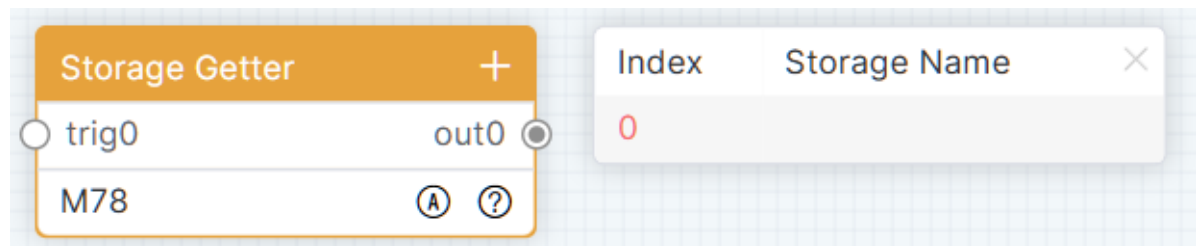


Double-click the row to pop up the details list floating window, you can modify the comments.

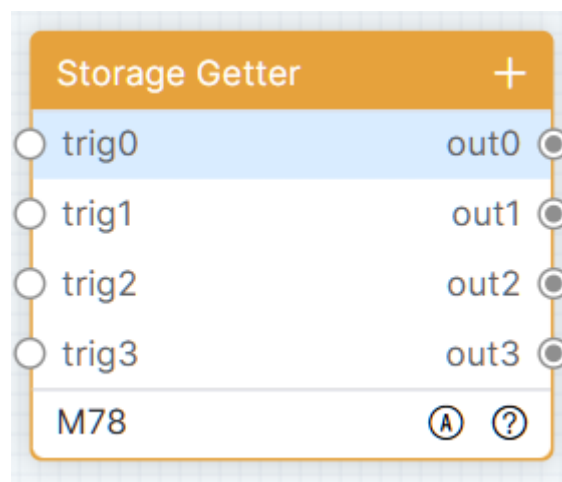


Delayer

Module name default display: Delayer. The initial state will open the detailed list as shown in the figure:



If there is no data, click [Add] or [Batch Add] to add single or multiple rows of data. Click the icon  to add a row at the end of the module. Right-click the module row menu, click [Insert] or [Insert Multiple Rows] to insert a single or multiple rows of data below the selected row.



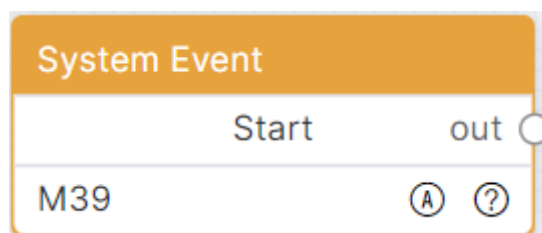
Double-click the row to pop up the details list floating window, you can modify the comments.

Index	Storage Name	✕
0		
1		
2		
3		

System

System Events

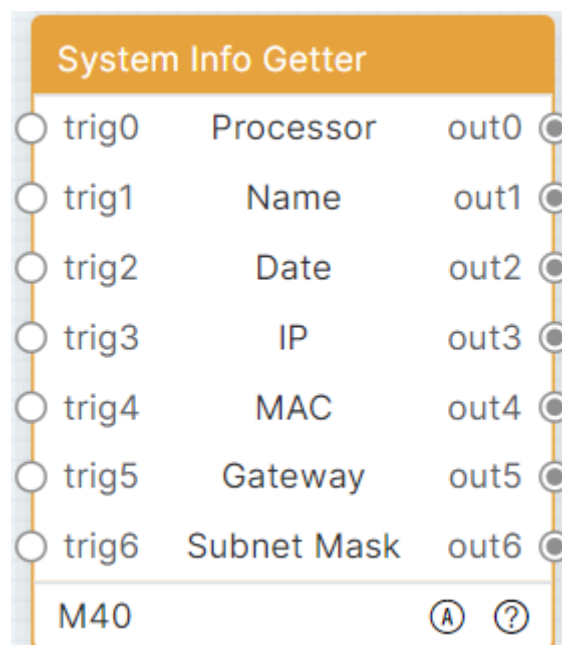
Module name default display: System Events. The initial state is as shown in the figure:



After the system starts successfully, immediately execute the subsequent logic of the module exit.

Output System Information

Module name default display: Output System Information. The initial state is as shown in the figure:



The module has 6 entry nodes and 6 exit nodes. Trigger [Trigger] 0-6 entries and [out] 0-6 entries to output the current system model, device alias, date, IP, MAC address, gateway and subnet mask.