Using AppWizard to Design Industrial Smart Thermostat GUI

Example Code Introduction for 32-bit NuMicro® Family

Document Information

Application	This example code uses SEGGER emWin AppWizard, combined with M467 series development platform and development resources, to complete the graphical use interface (GUI) of the industrial smart thermostat.
BSP Version	M460_Series_BSP_CMSIS_V3.00.002 emWin Software Package for M460 V3.0
Hardware	NuMaker-M467HJ V1.0 NuMaker-TFT-LCD43 V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. Overview

This example code uses SEGGER emWin AppWizard tool and NuMaker-HMI-M467 development platform to complete the screen design and screen interaction of an industrial smart thermostat project.

1.1 Principle

1.1.1 Block Diagram

This example code directly uses Nuvoton's GUI development platform NuMaker-HMI-M467 evaluation board. The components used during development include a TFT-LCD with a resolution of 480x272 and a screen size of 4.3 inches, a capacitive touch panel, a M467HJHAN microcontroller (MCU) on-board, and an SPI Flash with a capacity of 4 Mbytes. The system block diagram is shown in Figure 1-2.

Figure 1-1 NuMaker-HMI-M467 Evaluation Board

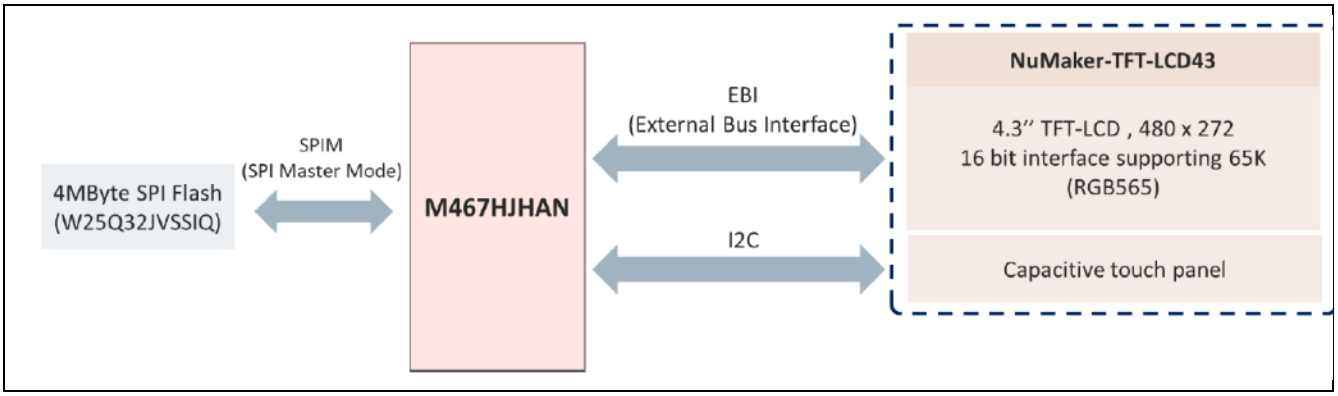
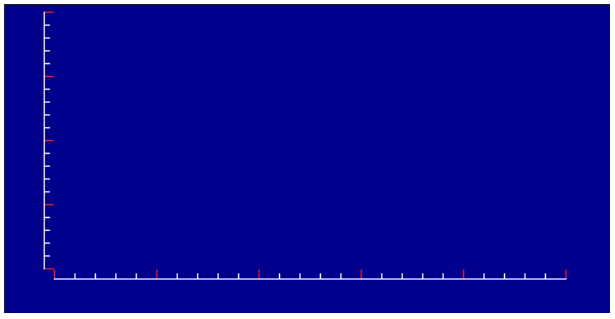
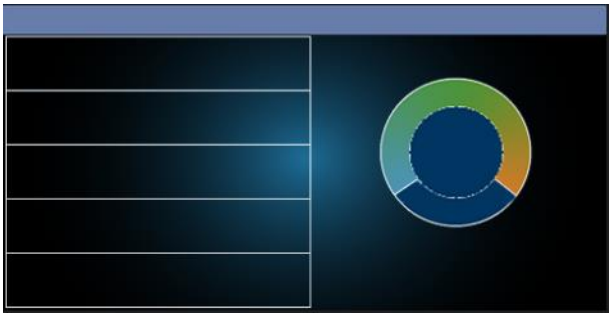
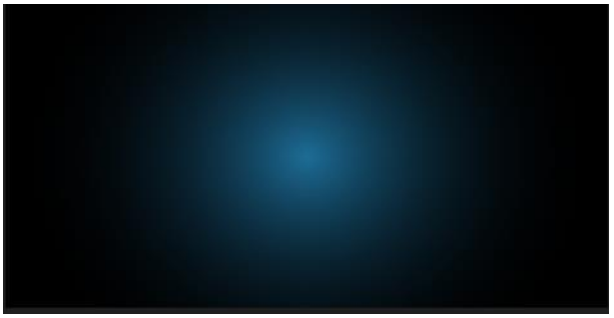
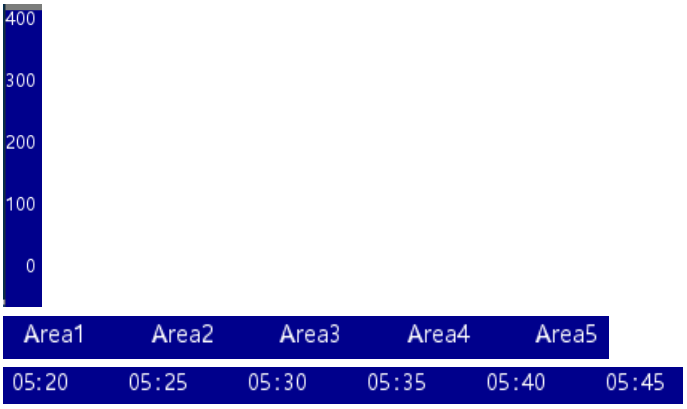
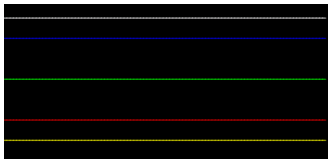
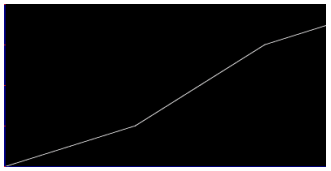


Figure 1-2 System Block Diagram

1.1.2 Overview of Elements Used in Industrial Smart Thermostat Screen

Application	GUI	Objects
Main Menu		Button
OK		
Back		
Temperature area selection	Transparent, no display	
View menu		Listbox
Temperature setting menu		
Time setting menu		
Language setting menu		
Coordinate background image, size 480x242		Image
Temperature list background image, size 480x242		

Application	GUI	Objects
Temperature setting background image, size 480x242		
Time and date set screen background image, size 480x242		
Date icon		
Time icon		
Text on coordinates: Temperature, area, time		TEXT
Area name, set temperature, actual temperature, temperature unit		

Application	GUI	Objects
Year, month, day, hour, minute, second	Day Month Year 日 月 年 Hour Minute Second 小时 分钟 秒	
Area color		Drawings
Time-Temperature waveform		
Area-Temperature waveform		
Temperature setting button		Rotary
Date selection	30 November 2121 30 十一月 2121 31 December 2122 31 十二月 2122 01 January 2023 01 一月 2023 02 February 2024 02 二月 2024 03 March 2025 03 三月 2025	Wheel
Time selection	03 18 08 04 19 09 05 : 20 : 10 06 21 11 07 22 12	
Date and time selection background		Box

Application	GUI	Objects
The background of the selected area		

Table 1-1 Elements Overview

1.1.3 Supporting Resources

- SEGGER emWin official forum: Segger emWin Forum
- Nuvoton emWin Forum: Nuvoton emWin Forum
- Download SEGGER emWin AppWizard tool user manual: AppWizard User Manual

1.2 Execution Steps

The execution steps of using AppWizard tool to complete a GUI project can be divided into four major stages:

Prepare hardware and download software.

Complete screen design, object settings, interaction settings, and Drawings settings in AppWizard.

Use AppWizard to generate code and add user code to complete more application functions.

Program to the evaluation board and run.

The execution steps for each stage are explained in detail below.

1.2.1 Preparing Hardware and Software Resources

1. The hardware uses NuMaker-HMI-M467. For relevant information, please refer to Schematic Diagram,Evaluation board User Manual.
2. Software resources need to be downloaded and completed one by one:
   1) M460 BSP download link M460 BSP; unzip it after downloading.
   2) M460 emWin HMI Library download link M467 emWin package; unzip M460_emWin_NonOS folder after downloading.
   3) Please follow the path in Table 1-2, and click *AppWizard_V134a_630a_Install.exe* to install AppWizard tool. Please note that if there is an old AppWizard version, the previous version must be completely removed before the new version can be installed.

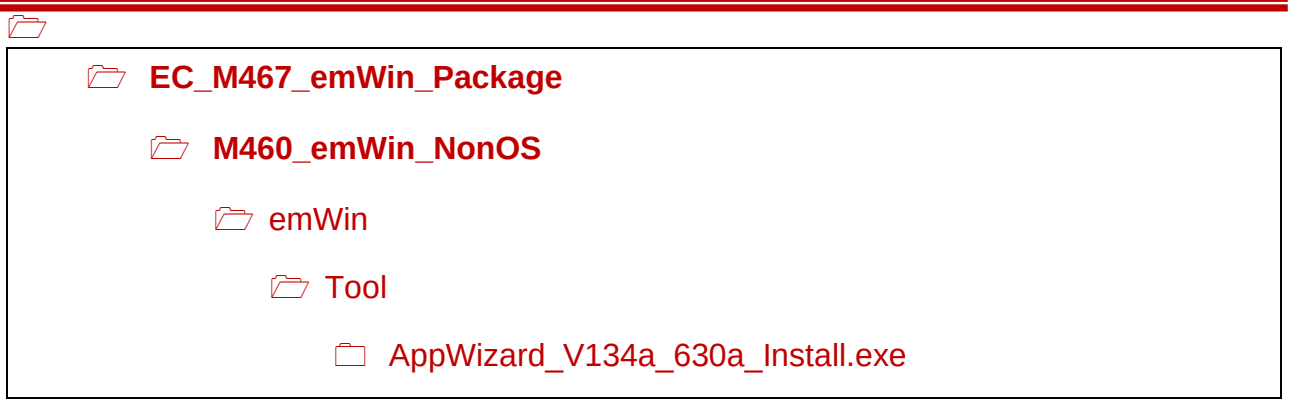


Table 1-2 AppWizard Installation File Path

3. Install M467_emWin_package to M460 BSP folder:

   1) Copy and paste M467_emWin_Package\M460_emWin_NonOS\emWin to M460_Series_BSP_CMSIS_V3.00.002\ThirdParty folder.

   2) Copy and paste M467_emWin_Package\M460_emWin_NonOS\emWin_SimpleDemoAppWizard to M460_Series_BSP_CMSIS_V3.00.002\SampleCode folder.

1.2.2 Editing and Configuring the GUI in AppWizard Tool

1.2.2.1 Create emWin Project

1. Open AppWizard tool from the desktop shortcut.

Figure 1-3 AppWizard Desktop Shortcut Icon

2. Click “Create new project” to create a new project.

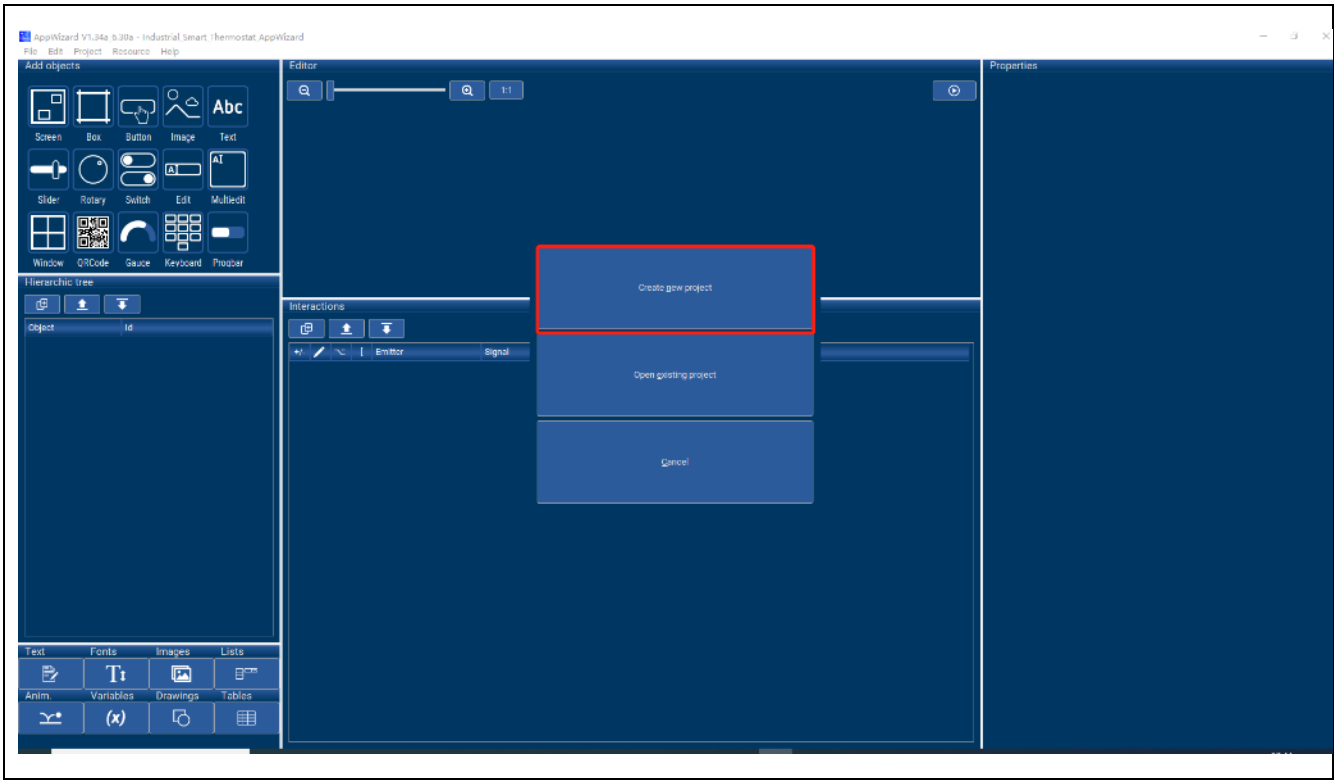


Figure 1-4 Create a New Project

3. Set “Project path” and “Project name” as shown in Figure 1-5, consistent with “color format” of the screen being used: 16 Bit, GUICC_M565, check “Enable Multibuffering”, and click “Ok”.

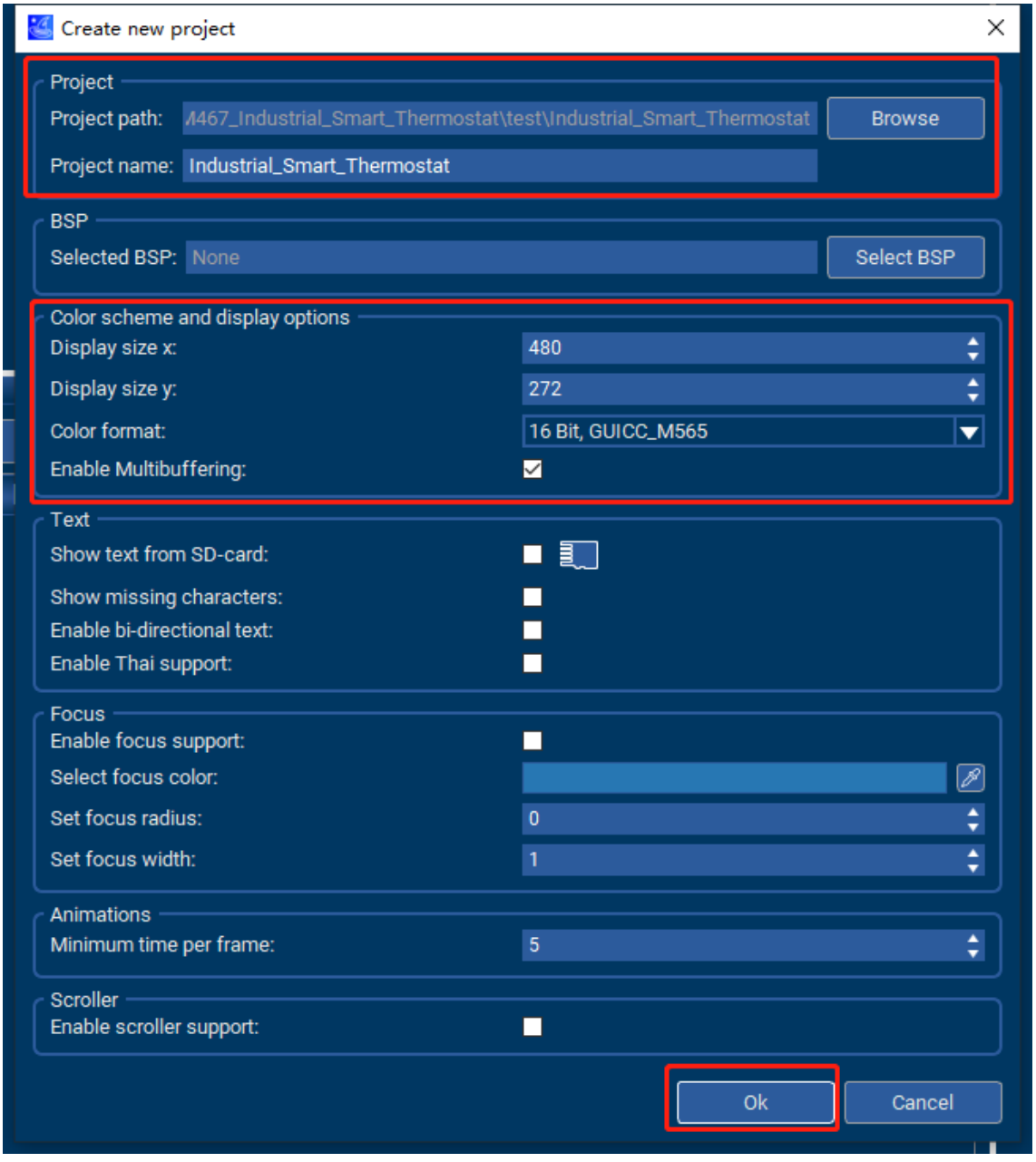


Figure 1-5 New Project Settings

1.2.2.2 Add Text

Define all Text contents used in the project, which can be added at any time during the development. The Texts are shown in Table 1-3. Due to the display of both Chinese and English languages in GUI, Chinese characters appear in the table.

ID	en	cn
ID_RTEXT_VIEW	View	显示
ID_RTEXT_SETUP	Setup	设置
ID_RTEXT_TIME	Time	时间
ID_RTEXT_LANG	Lang	语言
ID_RTEXT_HELP	Help	帮助
ID_RTEXT_AREA	Area:	区域：
ID_RTEXT_SET_TEMP	Set:	设定：
ID_RTEXT_ACTUAL_TEMP	Actual:	实际：
ID_RTEXT_AREA	Area1	区域 1
ID_RTEXT_AREA	Area2	区域 2
ID_RTEXT_AREA	Area3	区域 3
ID_RTEXT_AREA	Area4	区域 4
ID_RTEXT_AREA	Area5	区域 5
ID_RTEXT_DEG	℃	℃
ID_RTEXT_YEAR	Year	年
ID_RTEXT_MONTH	Month	月
ID_RTEXT_DATE	Day	日
ID_RTEXT_HR	Hour	小时
ID_RTEXT_MIN	Minute	分钟

ID	en	cn
ID_RTEXT_SEC	Second	秒
ID_RTEXT_JAN	January	一月
ID_RTEXT_FEB	February	二月
ID_RTEXT_MAR	March	三月
ID_RTEXT_APR	April	四月
ID_RTEXT_MAY	May	五月
ID_RTEXT_JUN	June	六月
ID_RTEXT_JUL	July	七月
ID_RTEXT_AUG	August	八月
ID_RTEXT_SEP	September	九月
ID_RTEXT_OCT	October	十月
ID_RTEXT_NOV	November	十一月
ID_RTEXT_DEC	December	十二月
ID_RTEXT_ENG	EN	英文
ID_RTEXT_CH	CN	中文
ID_RTEXT_AREA_TEMP_L	Area Temperature List	区域温度列表
ID_RTEXT_AREA_TEMP_W	Area Temperature Wave	区域温度曲线
ID_RTEXT_TEMP_TIME_W	Time Temperature Wave	时间温度曲线
ID_RTEXT_TEMP_SETUP	Temperature Setup	温度设置
ID_RTEXT_TIME_SETUP	Time Setup	时间设置
ID_RTEXT_DATE_SETUP	Date Setup	日历设置
ID_RTEXT_COLON	:	:

Table 1-3 Texts in Project

The steps for adding Texts are shown in Figure 1-6:

1. In step 1, Click “Text” to pop up “Choose/Edit Text” dialog box.
2. Repeat steps 2~4, click “New language” to add English and Chinese.
3. In steps 5~8, click “Add Text” to add text, modify ID, define English and Chinese text contents.
4. Repeat steps 5~8 to complete the definition of all texts in Table 1-3.
5. In step 9, click “Ok” to complete the definition of Text.

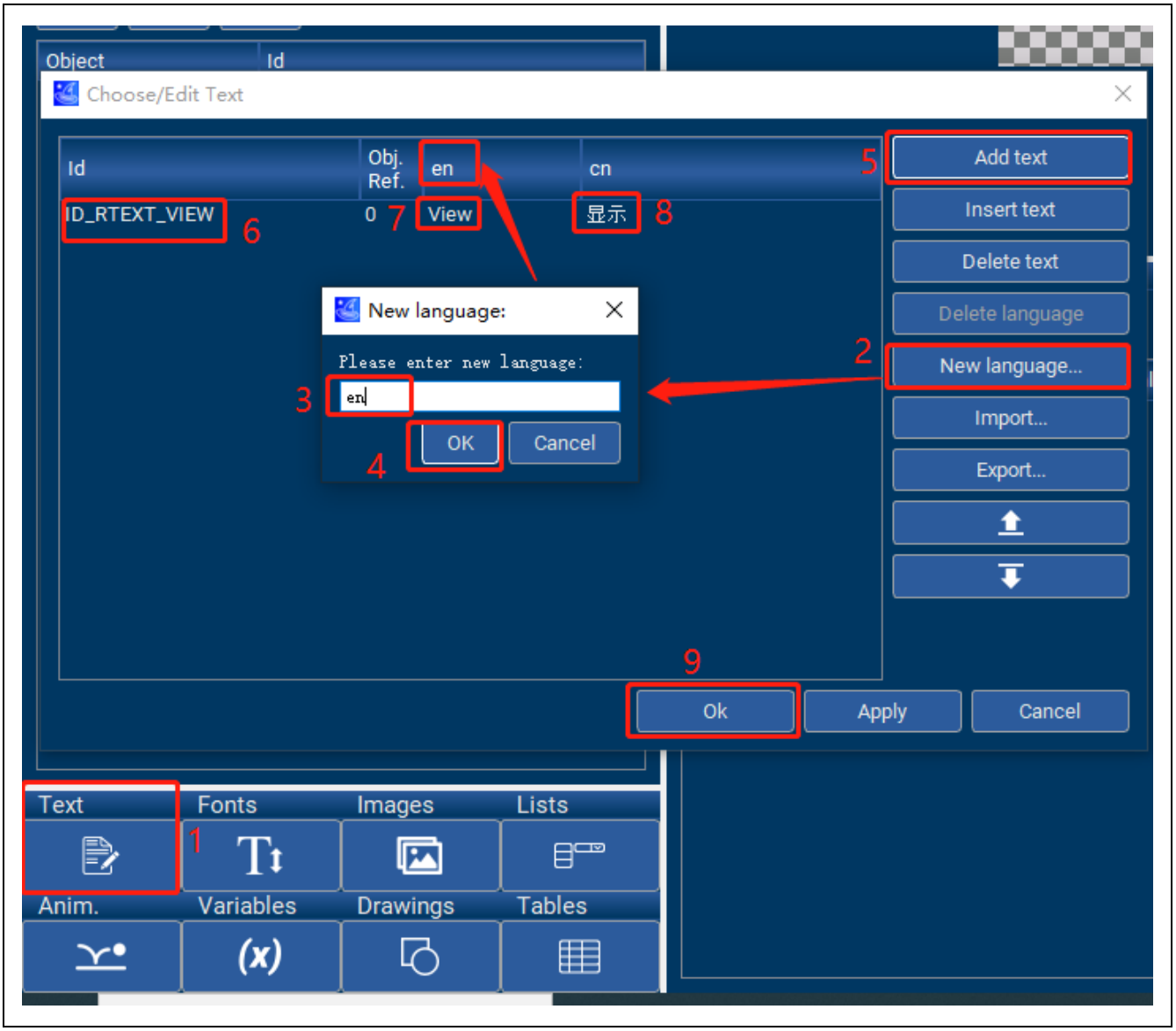


Figure 1-6 Add Text

1.2.2.3 Add Lists

Define all Lists used in the project, which can be added at any time during the development. The Lists are shown in Table 1-4.

ID	Content	Description
ID_CONTENT_VIEW	ID_RTEXT_AREA_TEMP_L ID_RTEXT_AREA_TEMP_W ID_RTEXT_TEMP_TIME_W	View drop-down menu displays list
ID_CONTENT_SETUP	ID_RTEXT_TEMP_SETUP	Setup drop-down menu displays list
ID_CONTENT_TIME	ID_RTEXT_TIME_SETUP ID_RTEXT_DATE_SETUP	Time drop-down menu displays list
ID_CONTENT_LANG	ID_RTEXT_ENG ID_RTEXT_CH	Language drop-down menu displays list
ID_CONTENT_MONTH	D_RTEXT_JAN ID_RTEXT_FEB ID_RTEXT_MAR ID_RTEXT_APR ID_RTEXT_MAY ID_RTEXT_JUN ID_RTEXT_JUL ID_RTEXT_AUG ID_RTEXT_SEP ID_RTEXT_OCT ID_RTEXT_NOV ID_RTEXT_DEC	Contents of month setting options
ID_CONTENT_HOUR	00~23	Contents of hour setting options
ID_CONTENT_MIN_SEC	00~59	Contents of minute and second setting options
ID_CONTENT_DAY	01~31	Contents of date setting options
ID_CONTENT_YEAR	2023~2122	Contents of year setting options

Table 1-4 Lists in Project

The steps for adding List are shown in Figure 1-7:

1. In step 1, click “List” to pop up “Content edit” dialog box.
2. In steps 2~3, define a List and rename it.
3. In step 4, click to add contents of the List.
4. Repeat steps 5~6 to add required content to the List. The content can be defined in “Text”, or it can be written directly into a string.
5. Repeat steps 2~7 to complete the definition of all Lists in Table 1-4.
6. In step 8, click “Ok” to complete the definition of List.

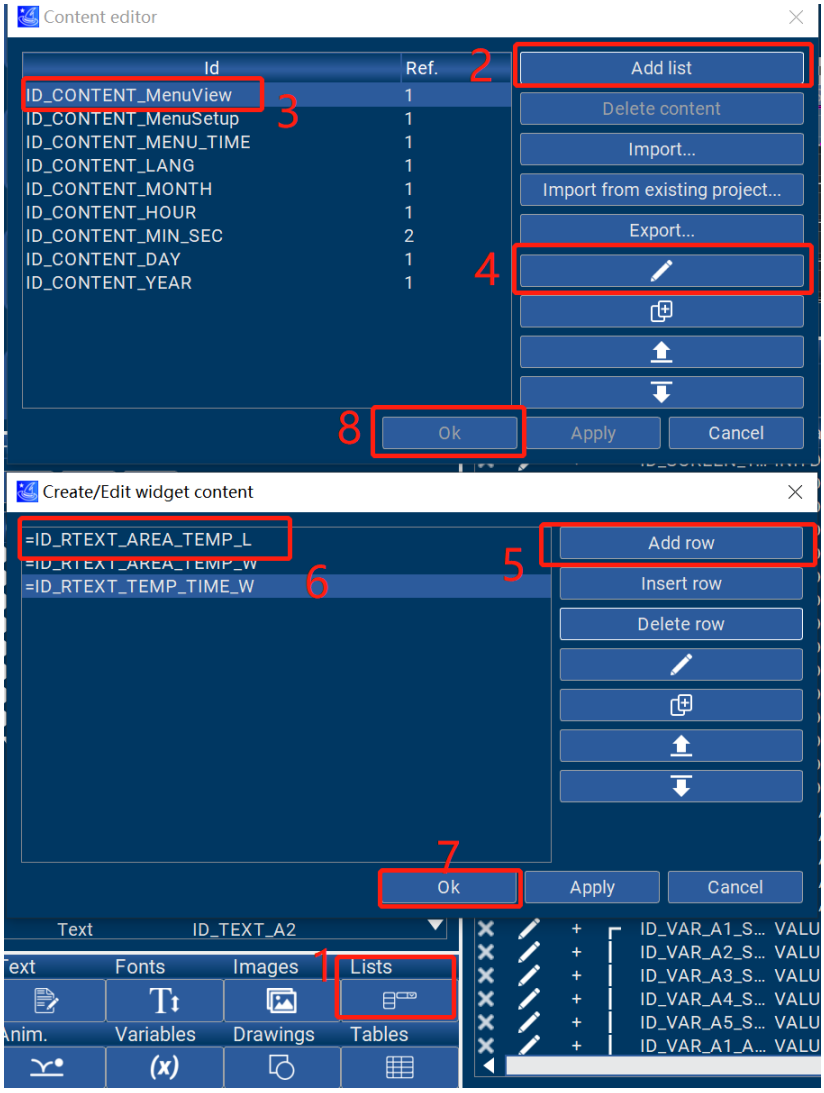


Figure 1-7 Add Lists

1.2.2.4 Add Fonts

Create 2 fonts, MicrosoftYaHeiUI_24_Normal_EXT_AA4 and MicrosoftYaHeiUI_20_Normal_EXT_AA4, as shown in Figure 1-8. Note that the font selected should be able to display both Chinese and English.

1. In step 1, click “Fonts” to pop up “Choose Font” dialog box.
2. In step 2, click “Create new” to pop up “Create Font” dialog box.
3. In steps 3~7, select MicrosoftYaHeiUI.
4. In steps 8~10, set the range of codepoint, select "Project text only" to select only the character range used by the project.
5. Repeat steps 2~10 to complete the definition of MicrosoftYaHeiUI_20_Normal_EXT_AA4 font.

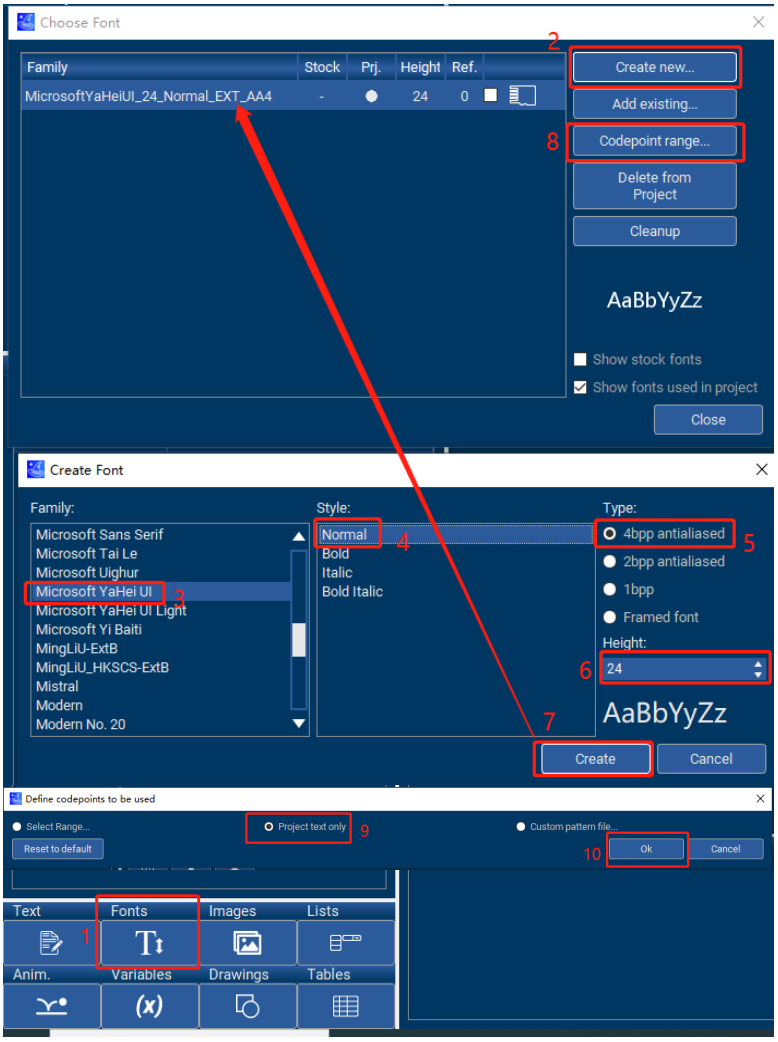


Figure 1-8 Create Fonts

1.2.2.5 Add Images

Load images used in the project.

1. In step 1, click “Images” to pop up “Images” dialog box.
2. In step 2, click “Import to project” to pop up “Add Image” dialog box.
3. In steps 3~4, select images for the project and import them.

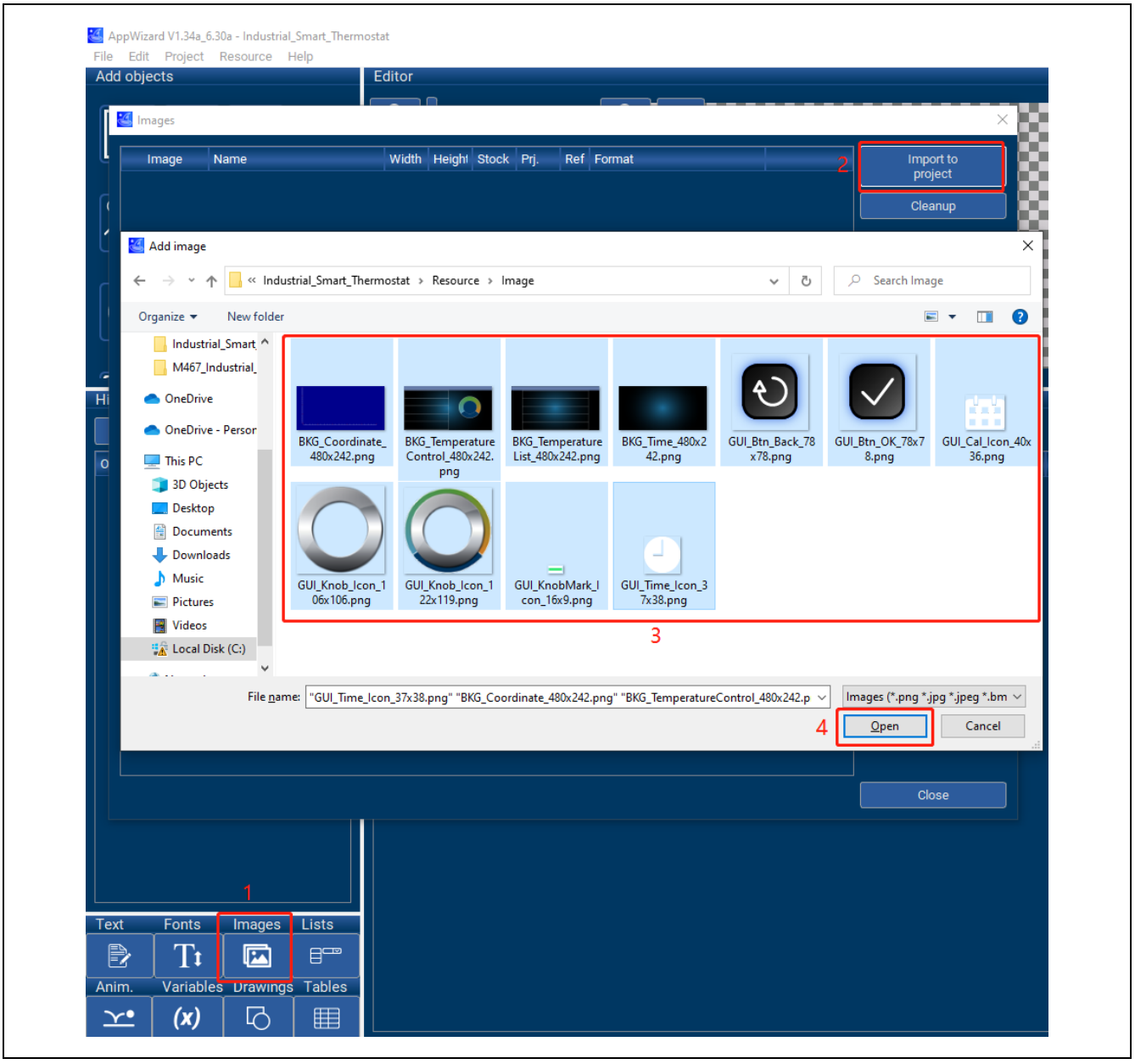


Figure 1-9 Load Images

1.2.2.6 Add Variables

Define all variables used in the project and can be added at any time during the development process. All variables are shown in Table 1-5.

ID	Initial value	Calculation formula	Description
ID_VAR_View_index	0		Index number of View drop-down menu
ID_VAR_A1_SetTemp	110		Set temperature of area (5 Areas)
ID_VAR_A2_SetTemp	120		
ID_VAR_A3_SetTemp	130		
ID_VAR_A4_SetTemp	140		
ID_VAR_A5_SetTemp	150		
ID_VAR_A1_ActualTemp	0		Actual temperature of area (5 Areas)
ID_VAR_A2_ActualTemp	0		
ID_VAR_A3_ActualTemp	0		
ID_VAR_A4_ActualTemp	0		
ID_VAR_A5_ActualTemp	0		
ID_VAR_Area_Index	0	ID_VAR_Area_Index*43+25	selected area
ID_VAR_Bar_Y0	0		The coordinate Y0 value of the selected area
ID_VAR_Temp	0		Temporarily saved value of temperature setting value
ID_VAR_Color_Yellow	-2698496		yellow
ID_VAR_Color_White	-1		White
ID_VAR_A1_Color~ ID_VAR_A5_Color	-1		The font color of the area
ID_VAR_Y1~ ID_VAR_Y5	-1	200-(ID_VAR_An_ActualTemp*200/400) where n = 1~5	The y coordinate of Area- Temperature waveform

ID	Initial value	Calculation formula	Description
ID_VAR_X0~ ID_VAR_X99	0	ID_VAR_X99 = ID_VAR_X99 +ID_VAR_X_Pixel_inc	The x coordinate of Time-Temperature waveform
ID_VAR_A1_Y0~ ID_VAR_A1_Y99	0	ID_VAR_An_Y99 = 200-(ID_VAR_An_ActualTemp*200/400) where n = 1~5	The y coordinate of Area-Temperature waveform (5)
ID_VAR_A2_Y0~ ID_VAR_A2_Y99	0		
ID_VAR_A3_Y0~ ID_VAR_A3_Y99	0		
ID_VAR_A4_Y0~ ID_VAR_A4_Y99	0		
ID_VAR_A5_Y0~ ID_VAR_A5_Y99	0		
ID_VAR_Trigger	0		Time-Temperature waveform refresh flag
ID_VAR_X_Index	0		The x-axis currently displays the position index number
ID_VAR_Point	99		The total number of points displayed in the waveform = ID_VAR_Point+1
ID_VAR_X_Pixel_inc	4	400/(ID_VAR_Point+1)	The number of pixels between two points in the time-temperature waveform
ID_VAR_Timer_Cnt	0		1 s timer count
ID_VAR_Refresh_Interval	60	ID_VAR_Refresh_Interval = ID_VAR_Refresh_inc	Waveform refresh interval variable
ID_VAR_Refresh_inc	60		Waveform refresh interval constant
ID_VAR_Year	0		Time and date actual value variables
ID_VAR_Month	0		
ID_VAR_Day	0		

ID	Initial value	Calculation formula	Description
ID_VAR_Hour	0		
ID_VAR_Min	0		
ID_VAR_Sec	0		
ID_VAR_Year_Tmp	0		Variables used for conversion for time and date
ID_VAR_Month_Tmp	0		
ID_VAR_Day_Tmp	0		
ID_VAR_Hour_Tmp	0	ID_VAR_Time_In_Sec_Display/3600	
ID_VAR_Min_Tmp	0	ID_VAR_Time_In_Sec_Display%3600/60	
ID_VAR_Sec_Tmp	0	ID_VAR_Time_In_Sec_Display%3600%60	
ID_VAR_Time_In_Sec	0	(ID_VAR_Hour*3600) + (ID_VAR_Min*60) + ID_VAR_Sec	Current time (converted to seconds)
ID_VAR_Time_In_Sec_Tmp	0	86400+ID_VAR_Time_In_Sec-(5*ID_VAR_X_Time_inc)	Calculate the middle amount of the coordinate scale based on the current time
ID_VAR_Time_In_Sec_Display	0	ID_VAR_Time_In_Sec_Tmp%86400	
ID_VAR_X_Time_inc	0	80*(ID_VAR_Point+1)/400*ID_VAR_Refresh_Interval	The time interval between two large ticks on the x-axis
ID_VAR_Current_Screen	0		Currently displayed screen

Table 1-5 Variables in Project

The steps for adding variables are shown in Figure 1-10:

1. In step 1, click “Variables” to pop up “Variables” dialog box.
2. In steps 2~4, click “Add Variable” to add a variable, set the ID and initial value.
3. In step 5, click in “Term” to pop up “Create calculated” dialog box and add calculation formula to the variable. Note that only variables that require calculation formulas need to be added.
4. In steps 6~9, select the parameters of the calculation formula in “Operand definition”, which can be object values, variables, fixed values, etc. Click “Add Operand” to add the selected parameters. The parameters that have already been added will be displayed in step 9. Repeat steps 6~9 to add multiple parameters.
5. In step 10, edit the formula in “Term to be calculated”.
6. Repeat steps 2~11 to add all variables. If variable does not need calculate formula, ignore steps 5~11.
7. In step 12, click “Ok” to save the added variables.

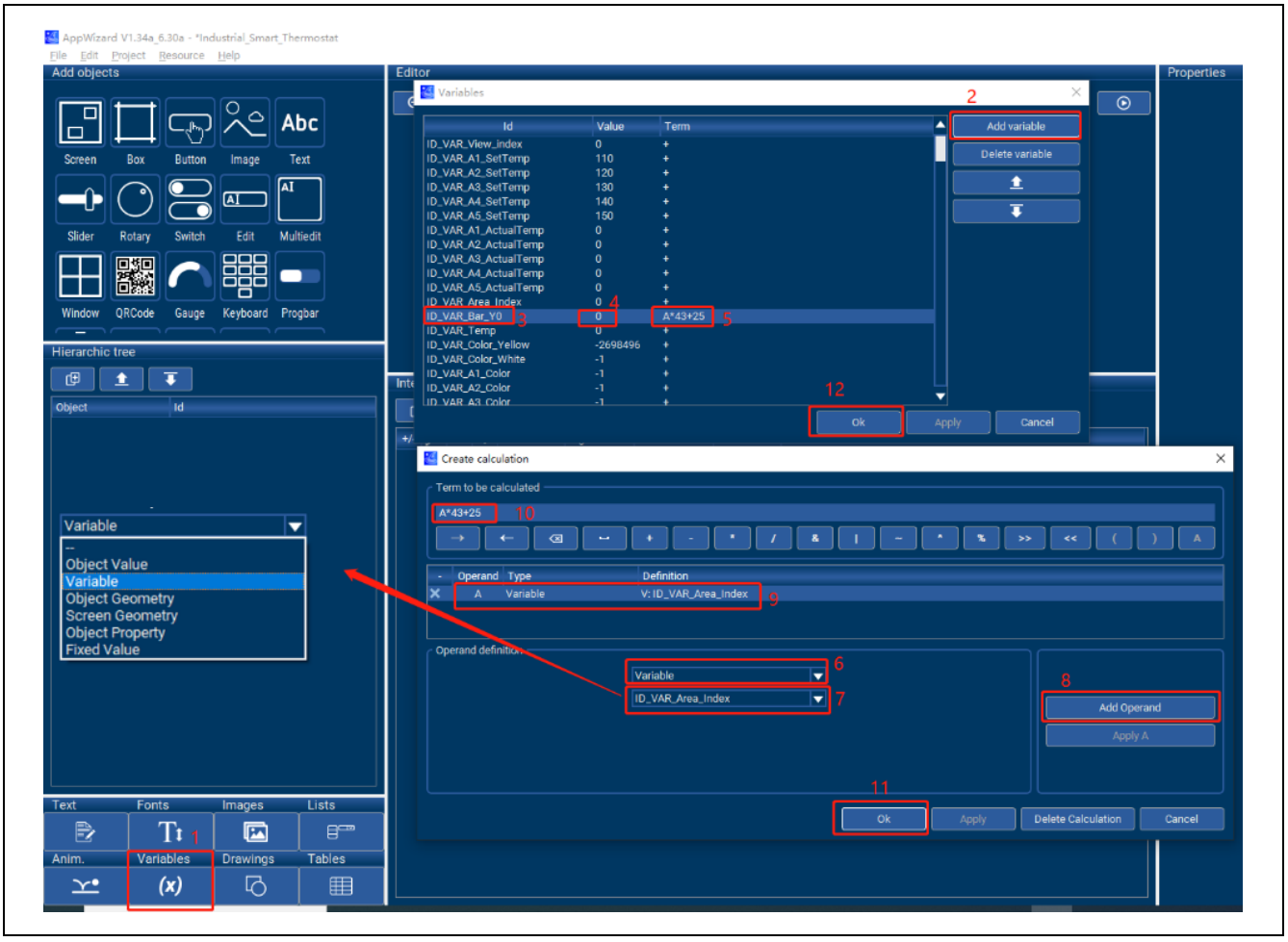


Figure 1-10 Add Variables

1.2.2.7 Add Drawings

Define all Drawings used in the project, which can also be defined during the development process. All actions are shown in Table 1-6.

Action	Description
ID_DRAW_BLK	Clear waveform drawing area
ID_DRAW_TempWave	Draw Area-Temperature waveform
ID_DRAW_TimeWave	Draw Time-Temperature waveform
ID_DRAW_COLOR	Draw waveform color indication

Table 1-6 Description of Drawings

1. Draw a black background, as shown in Figure 1-11:

   1) In step 1, click “Drawings” to pop up “Drawing objects” dialog box.

   2) In steps 2~3, click “Add drawing” to add a drawing action, and set the ID to ID_DRAW_BK.

   3) In step 4, click to pop up “Edit drawing” dialog box.

   4) In steps 5~7, select functions. Select function in “Available functions”, click “Add function” to add the selected function to “Used functions”, select “SET_BKCOLOR” to set background color first.

   5) In steps 8~9, select parameters that need to be set in “Parameters of selected function” and set them. “SET_BKCOLOR” has only one parameter “BkColor” (set background color), which is set to 0xff000000 (black).

   6) Repeat steps 5~6 to add “CLEAR”. The purpose is to clear the display area with background color just set. Note that “CLEAR” has no parameters to set, so steps 8~9 is not needed.

   7) In steps 10~11, click “Ok” to complete the action settings.

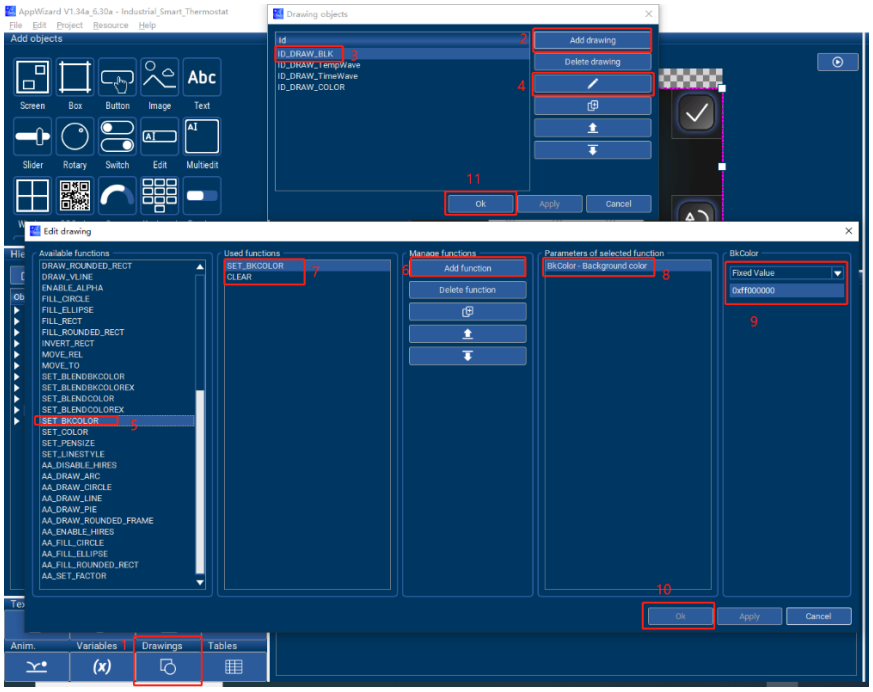


Figure 1-11 Draw Black Background

2. Draw Area-Temperature Waveform

The size of the waveform drawing area set in Area-Temperature waveform screen is 400x200. 400 pixels on the X-axis represent 5 areas. 200 pixels on the Y-axis represent 0 °C~400 °C, so 1 pixel corresponds to 2 °C. It should be noted that the coordinates in AppWizard start from the upper left corner of the object and extend to the right and downward. The waveforms to be drawn needs to start from the lower left corner and extend to the right and upward, so coordinate conversion is required during the drawing process. As shown in Figure 1-12, red is the coordinate of the Object, and green is the coordinate of the waveform.

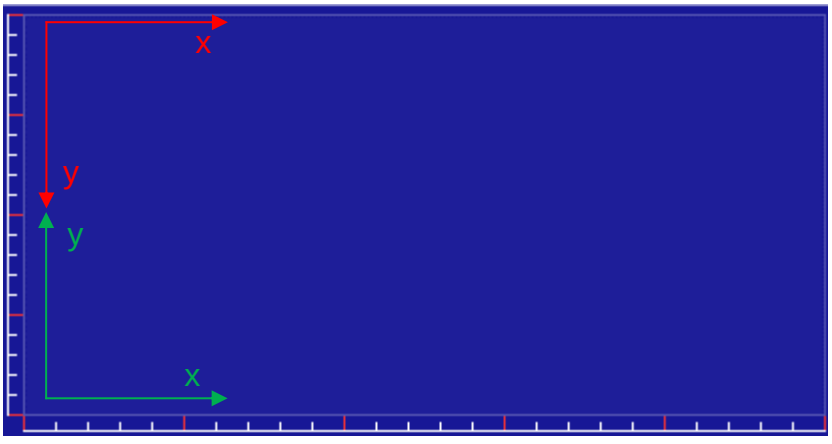


Figure 1-12 Object Coordinates and Waveform Coordinates

Area-Temperature waveform consists of the coordinate origin and the temperatures of 5 areas. Draw five line-segments from six points. The drawing action is shown in Table 1-7.

Function	Parameter	Description
SET_BKCOLOR	**BkColor = (Fixed value, 0xff000000)**	**Set background color to black**
CLEAR	**none**	**Clear screen display**
SET_COLOR	Color = (Fixed value, 0xffffffff)	**Set drawing color to white**
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 200, 80, ID_VAR_Y1)	**Draw a line segment from the origin to the Area1 temperature.**
AA_DRAW_LINE	(x0, y0, x1, y1) = (80, ID_VAR_Y1, 160, ID_VAR_Y2)	**Draw a line segment from the Area1 temperature to Area2 temperature.**
AA_DRAW_LINE	(x0, y0, x1, y1) = (160, ID_VAR_Y2, 240, ID_VAR_Y3)	**Draw a line segment from the Area2 temperature to Area3 temperature.**
AA_DRAW_LINE	(x0**,** y0, x1, y1) = (240, ID_VAR_Y3, 320, ID_VAR_Y4)	**Draw a line segment from the Area3 temperature to Area4 temperature.**
AA_DRAW_LINE	(x0, y0, x1, y1) = (320, ID_VAR_Y4, 400, ID_VAR_Y5)	**Draw a line segment from the Area4 temperature to Area5 temperature.**

Table 1-7 Parameters for Area-Temperature Waveform Drawing

First clear display, and then draw five line-segments, as shown in Figure 1-13.

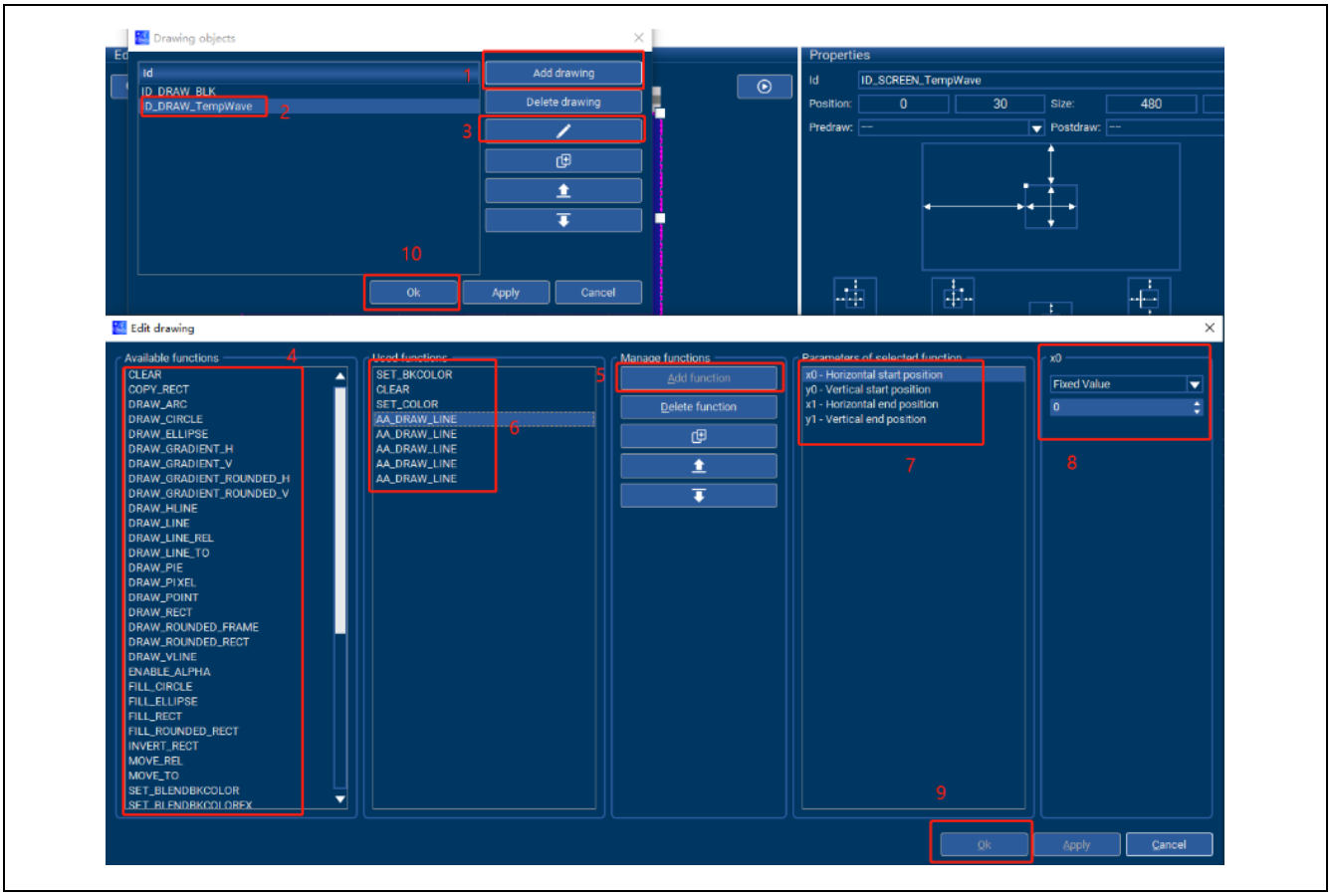


Figure 1-13 Draw Area-Temperature Waveform

3. Draw Color Indication of Time-Temperature Waveform.

When there are multiple waveforms on a screen, they are usually distinguished by different colors. Therefore, it is necessary to explain the corresponding relationship between color and waveform. Line segments of different colors are drawn to indicate the corresponding waveforms. The drawing actions are shown in Table 1-8, and the operation steps are shown in Figure 1-14.

Function	Parameter	Description
SET_COLOR	Color = (Fixed value, 0xffffff00)	Set drawing color to yellow
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 24, 40, 24)	Draw a yellow line segment to indicate that the yellow waveform is the Area1 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xffff0000)	Set drawing color to red
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 54, 40, 54)	Draw a red line segment to indicate that the red waveform is the Area2 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xff00ff00)	Set drawing color to green
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 84, 40, 84)	Draw a green line segment to indicate that the green waveform is the Area3 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xff0000ff)	Set drawing color to blue
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 114, 40, 114)	Draw a blue line segment to indicate that the blue waveform is the Area4 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xff000000)	Set drawing color to white
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 144, 40, 144)	Draw a white line segment to indicate that the white waveform is the Area5 Time-Temperature waveform

Table 1-8 Parameters for Waveform Color Indicates Drawing

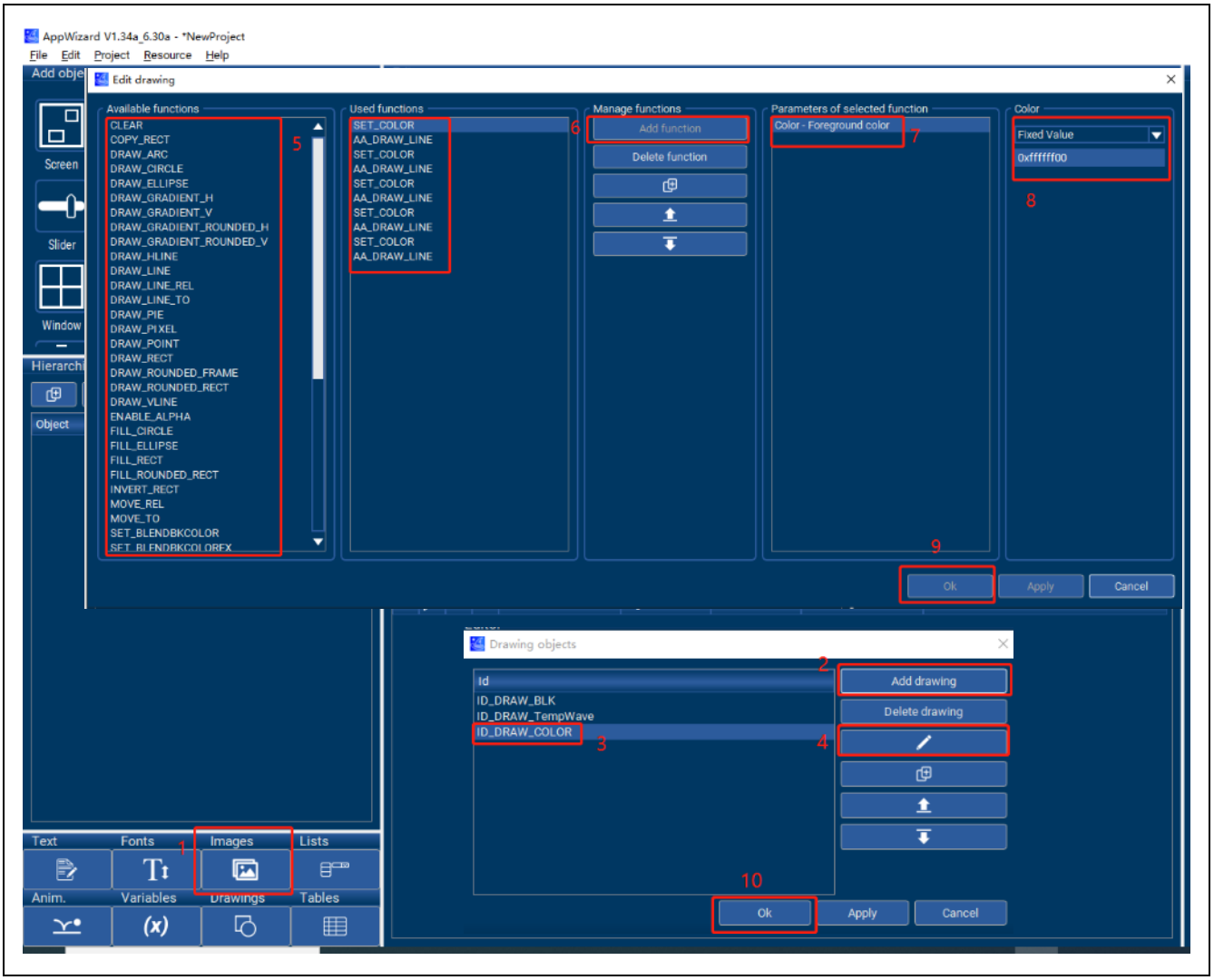


Figure 1-14 Draw Waveform Color Indication

4. Draw Time-Temperature Waveform

The size of the waveform drawing area set in Time-Temperature waveform screen is 400x200. The 400 pixels on the X-axis are divided into 100 points. 200 pixels on the Y-axis represent 0 ℃ ~ 400 ℃, so 1 pixel corresponds to 2 ℃. So, there are 100 variables for x-axis and 500 variables for y-axis. As shown in Table 1-9, the variables have been defined in Chapter 1.2.2.6, and a line segment is drawn between two adjacent points to realize the drawing of the waveform.

Variable	Function
ID_VAR_X0~ID_X99	**X coordinate of Time-Temperature waveform**
ID_VAR_A1_Y0~ ID_VAR_A1_Y99	**Y coordinate of Area1 Time-Temperature waveform**
ID_VAR_A2_Y0~ ID_VAR_A2_Y99	**Y coordinate of Area2 Time-Temperature waveform**
ID_VAR_A3_Y0~ ID_VAR_A3_Y99	**Y coordinate of Area3 Time-Temperature waveform**
ID_VAR_A4_Y0~ ID_VAR_A4_Y99	**Y coordinate of Area4 Time-Temperature waveform**
ID_VAR_A5_Y0~ ID_VAR_A5_Y99	**Y coordinate of Area5 Time-Temperature waveform**

Table 1-9 Coordinate Variables for Time-Temperature Waveforms

Draw 5 waveforms of different colors. The drawing action is shown in Table 1-10. Rename the drawing action to ID_TimeWave. The operation steps refer to the drawing of Area- Temperature waveform, as shown in Figure 1-13.

Function	Parameter	Description
SET_BKCOLOR	BkColor = (Fixed value, 0xff000000)	Set background color to black
CLEAR	None	Clear display
SET_COLOR	Color = (Fixed value, 0xffffff00)	Set drawing color to yellow
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A1_Yn, ID_VAR_Xn+1, ID_VAR_A1_Yn+1) Where n = 0~99	Draw Area1 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xffff0000)	Set drawing color to red
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A2_Yn, ID_VAR_Xn+1, ID_VAR_A2_Yn+1) Where n = 0~99	Draw Area2 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xff00ff00)	Set drawing color to green
AA_DRAW_LINE	(x0, y0, x1, y1) =	Draw Area3 Time-Temperature waveform

Function	Parameter	Description
	(ID_VAR_Xn, ID_VAR_A3_Yn, ID_VAR_Xn+1, ID_VAR_A3_Yn+1) Where n = 0~99	
SET_COLOR	Color = (Fixed value, 0xff0000ff)	Set drawing color to blue
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A4_Yn, ID_VAR_Xn+1, ID_VAR_A4_Yn+1) Where n = 0~99	Draw Area4 Time-Temperature waveform
SET_COLOR	Color = (Fixed value, 0xff000000)	Set drawing color to white
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A5_Yn, ID_VAR_Xn+1, ID_VAR_A5_Yn+1) Where n = 0~99	Draw Area5 Time-Temperature waveform

Table 1-10 Parameters for Time-Temperature Waveform Drawing

1.2.2.8 Create Main Menu Screen

1. Add Screen

Click “Screen” in “Add Objects” to create a screen, rename it to ID_SCREEN_Main in “Hierarchic tree”, and set the coordinates of the screen to (0, 0, 480, 30) in “Properties”.

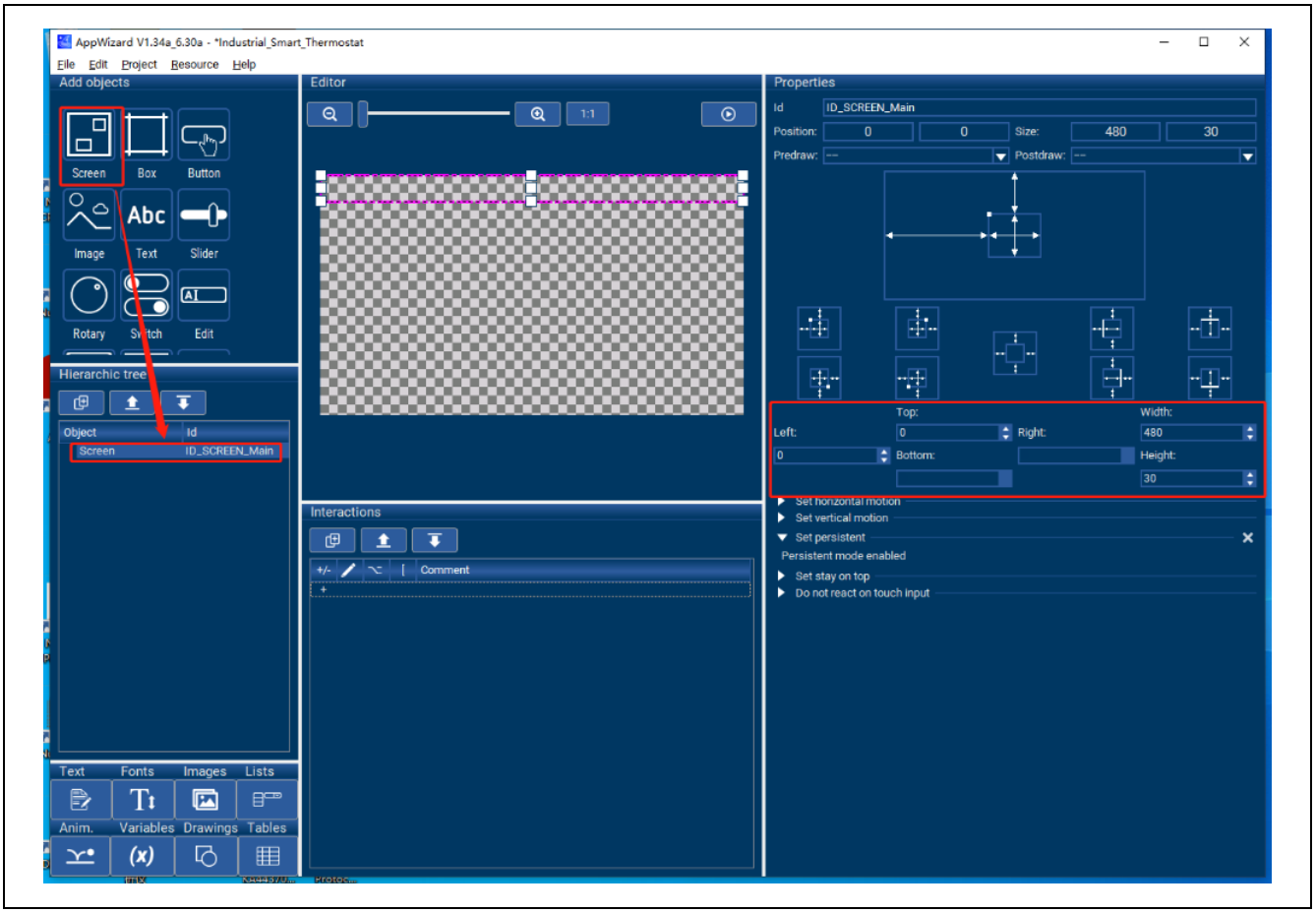


Figure 1-15 Create Main Menu Screen

2. Add Color Background

Select ID_SCREEN_Main and start editing the first screen.

1) Add Box

Use Box to display a solid color background. Click “Box” in “Add Objects” to add a Box and rename it to ID_BOX_BG in “Hierarchic tree”.

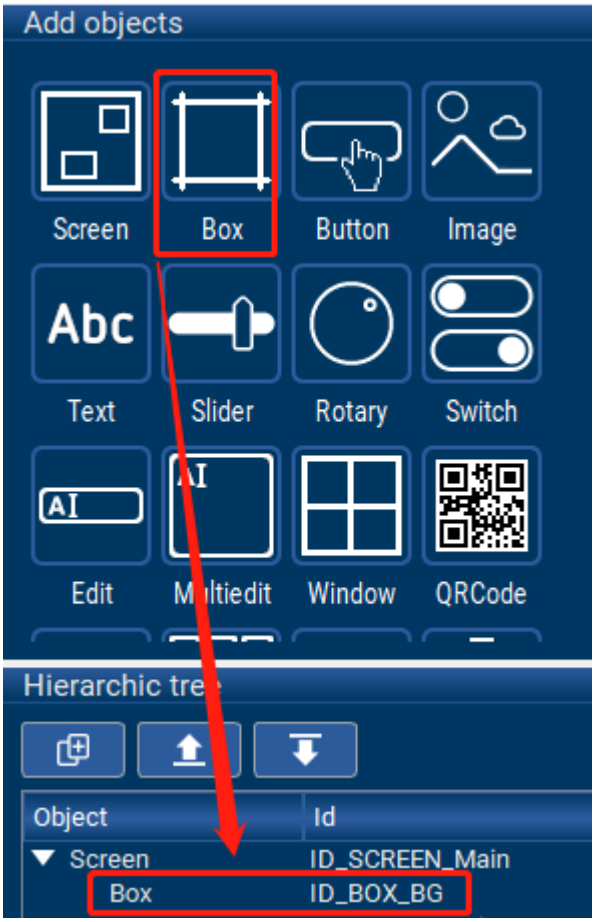


Figure 1-16 Add a Box - Display Solid Color Background

2) Set Box Properties

In “Properties”, set the coordinates to (0, 0, 480, 30) and “color” to RGB (123, 125, 123), where color is used as background color of the main menu.

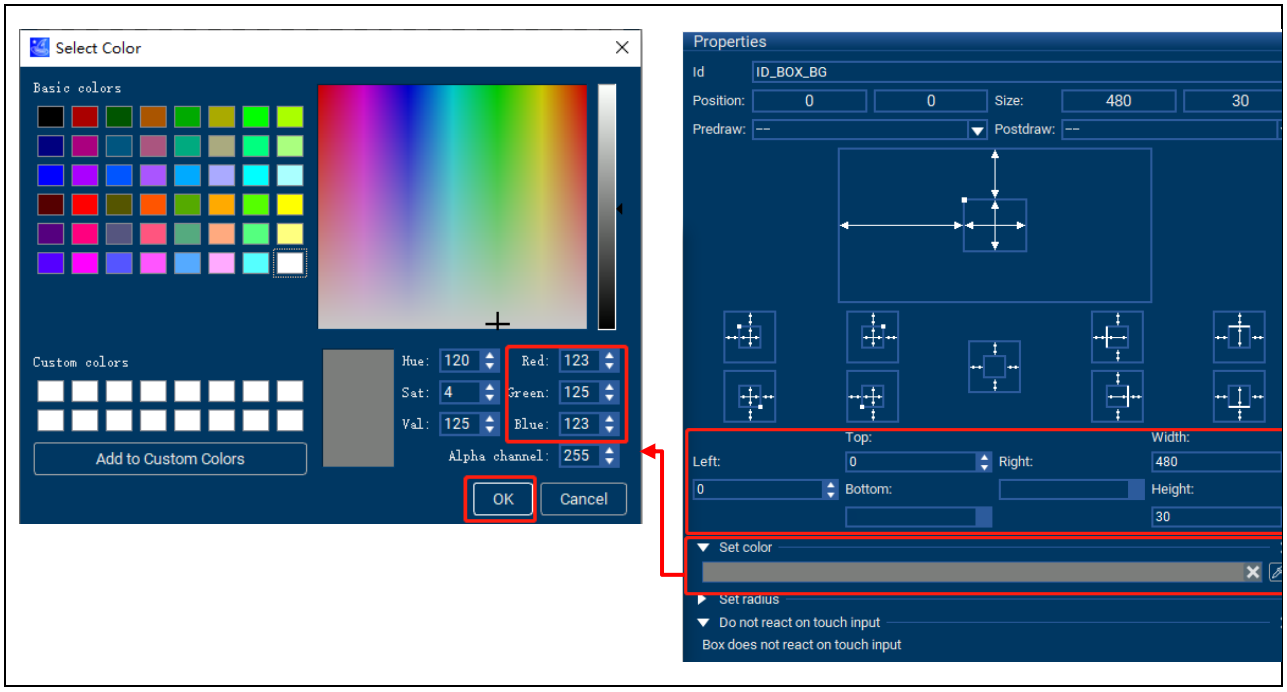


Figure 1-17 Set Main Menu Background Color

3. Add Button

1) Add Button

Click “Button” in “Add Objects” to add 5 buttons to the screen. Rename the Buttons in “Hierarchic tree” as shown in Figure 1-18.

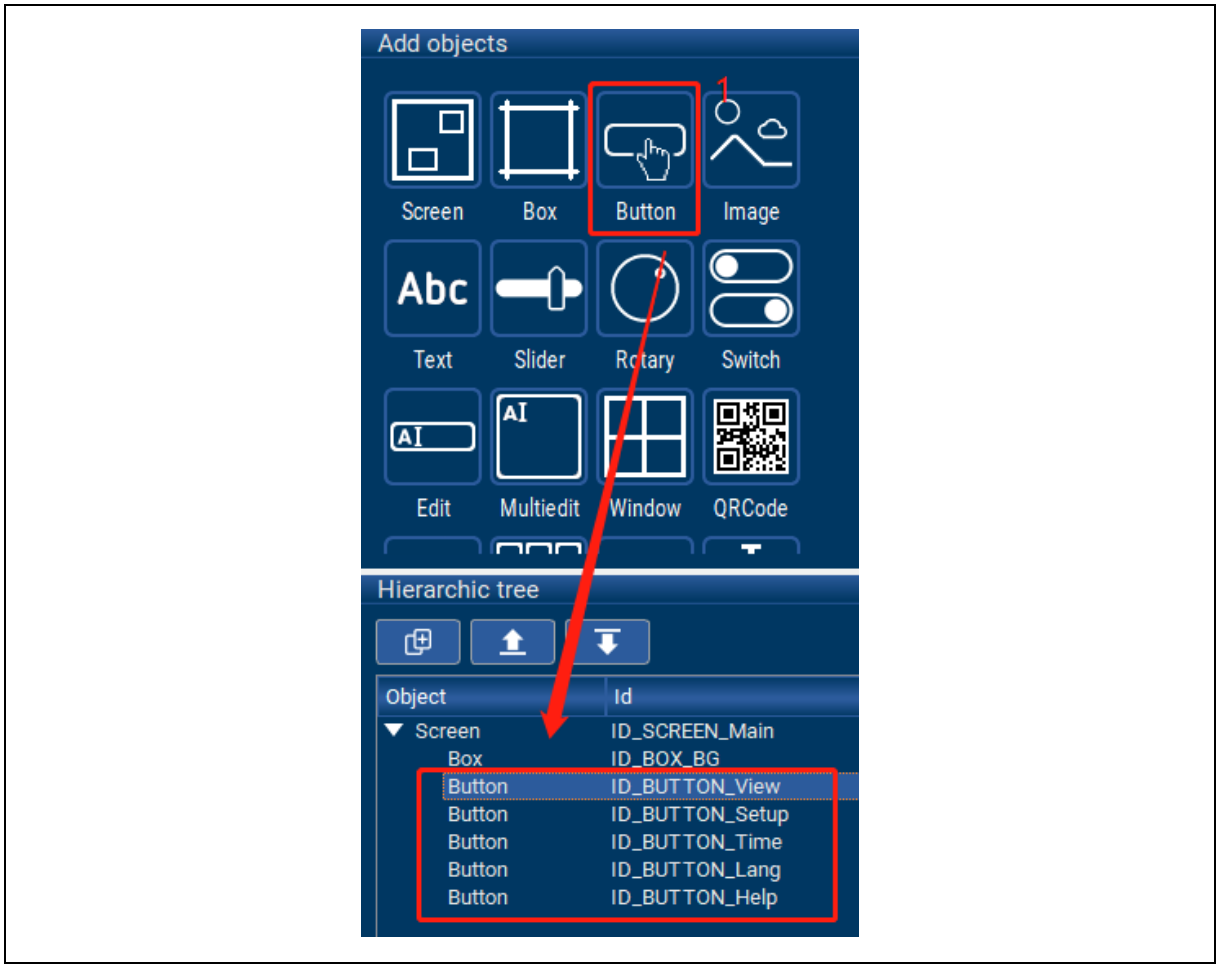


Figure 1-18 Add Buttons - Main Menu

2) Set Button Properties

In “Properties”, delete “bitmaps”, set “text color” to RGB (0, 0, 0), “text alignment” to “offset x” = “offset y” = 0, “Alignment” to left, “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4. The Coordinates and text of the button are set according to Table 1-11. The actual setting items are shown in Figure 1-19.

Button	**Coordinates (Left, Top, Width, Height)**	**text**
ID_BUTTON_View	(0, 0, 80, 30)	ID_RTEXT_VIEW
ID_BUTTON_Setup	(90, 0, 80, 30)	ID_RTEXT_SETUP
ID_BUTTON_Time	(182, 0, 80, 30)	ID_RTEXT_TIME
ID_BUTTON_Lang	(262, 0, 80, 30)	ID_RTEXT_LANG
ID_BUTTON_Help	(342, 0, 80, 30)	ID_RTEXT_HELP

Table 1-11 Description of Button Properties - Main Menu

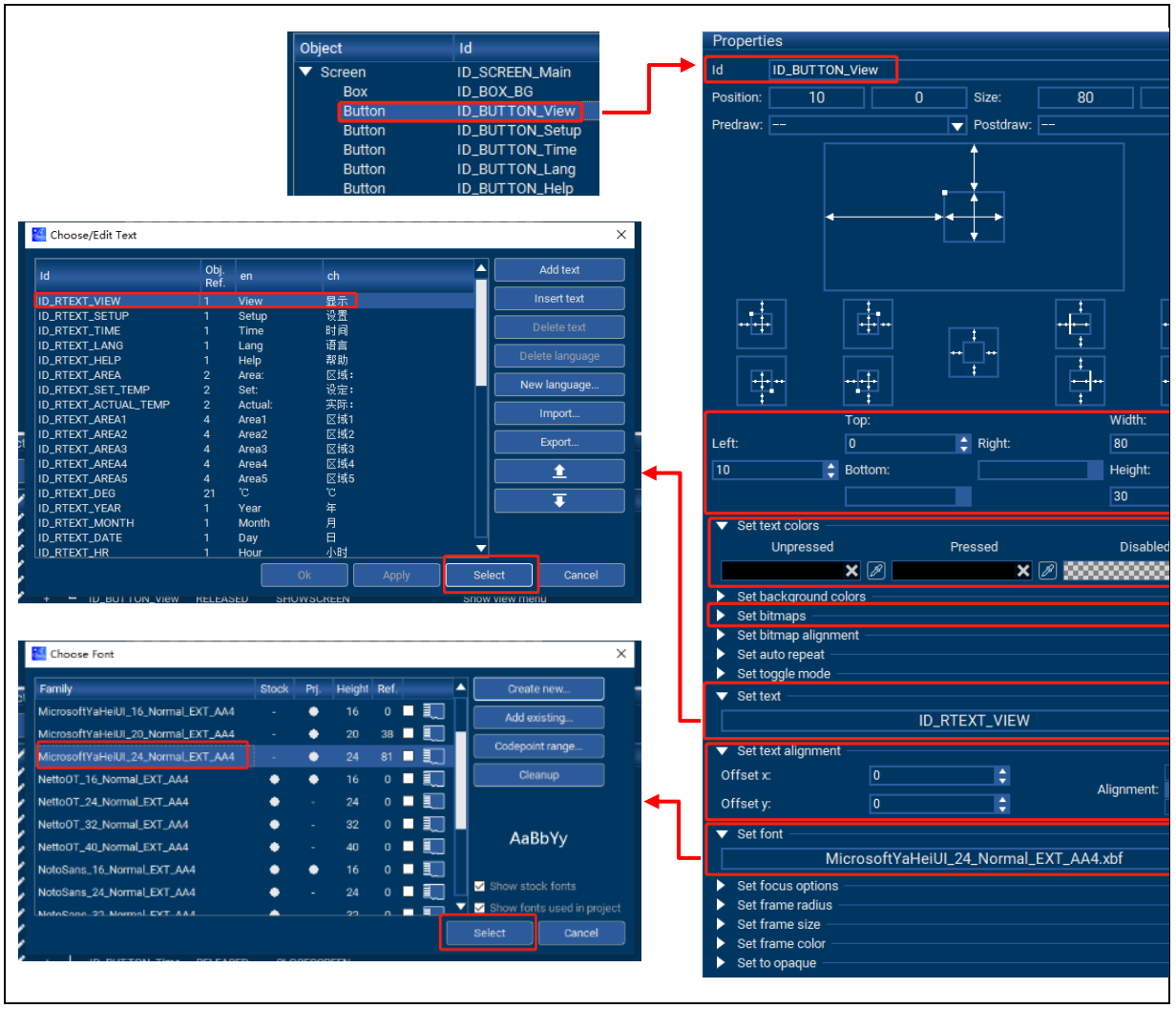


Figure 1-19 Button Properties Settings - Main Menu

4. Main Menu After Completion of Design

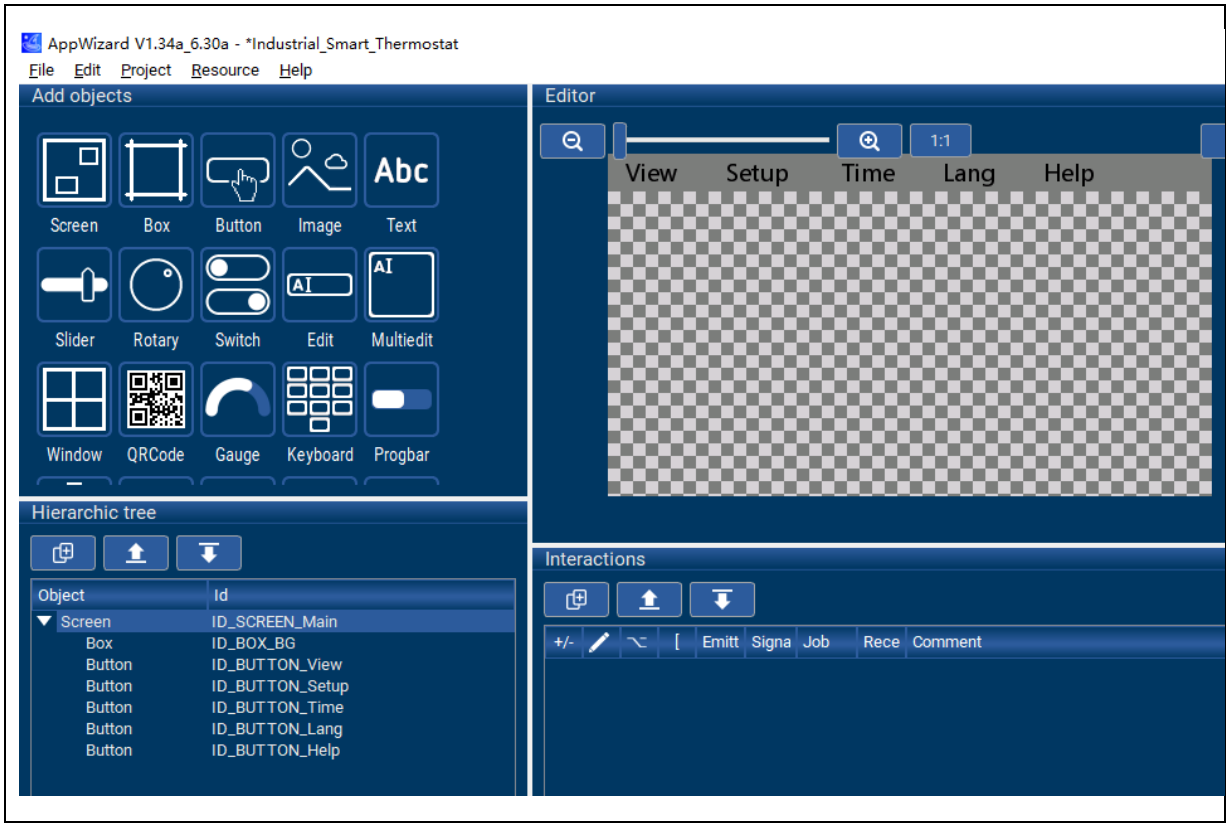


Figure 1-20 Main Menu Screen

1.2.2.9 Create Drop-Down Menu Screen

1. Add Screen

Add a drop-down menu for each menu (View, Setup, Time, Language, except Help). Click “Screen” in “Add Objects” to create 4 screens. As shown in Table 1-12, rename them in “Hierarchic tree” and set the coordinates in “Properties”. The actual steps are shown in Figure 1-21.

ID	Coordinates (Left, Top, Width, Height)	Description
ID_SCREEN_MenuView	(10, 30, 250, 76)	Drop-down menu: display mode setting
ID_SCREEN_MenuSetup	(90, 30, 200, 30)	Drop-down menu: area temperature setting
ID_SCREEN_MenuTime	(182, 30, 150, 52)	Drop-down menu: time setting
ID_SCREEN_MenuLang	(262, 30, 150, 30)	Drop-down menu: language setting

Table 1-12 Description of Screen Properties - Drop-down Menu Screen

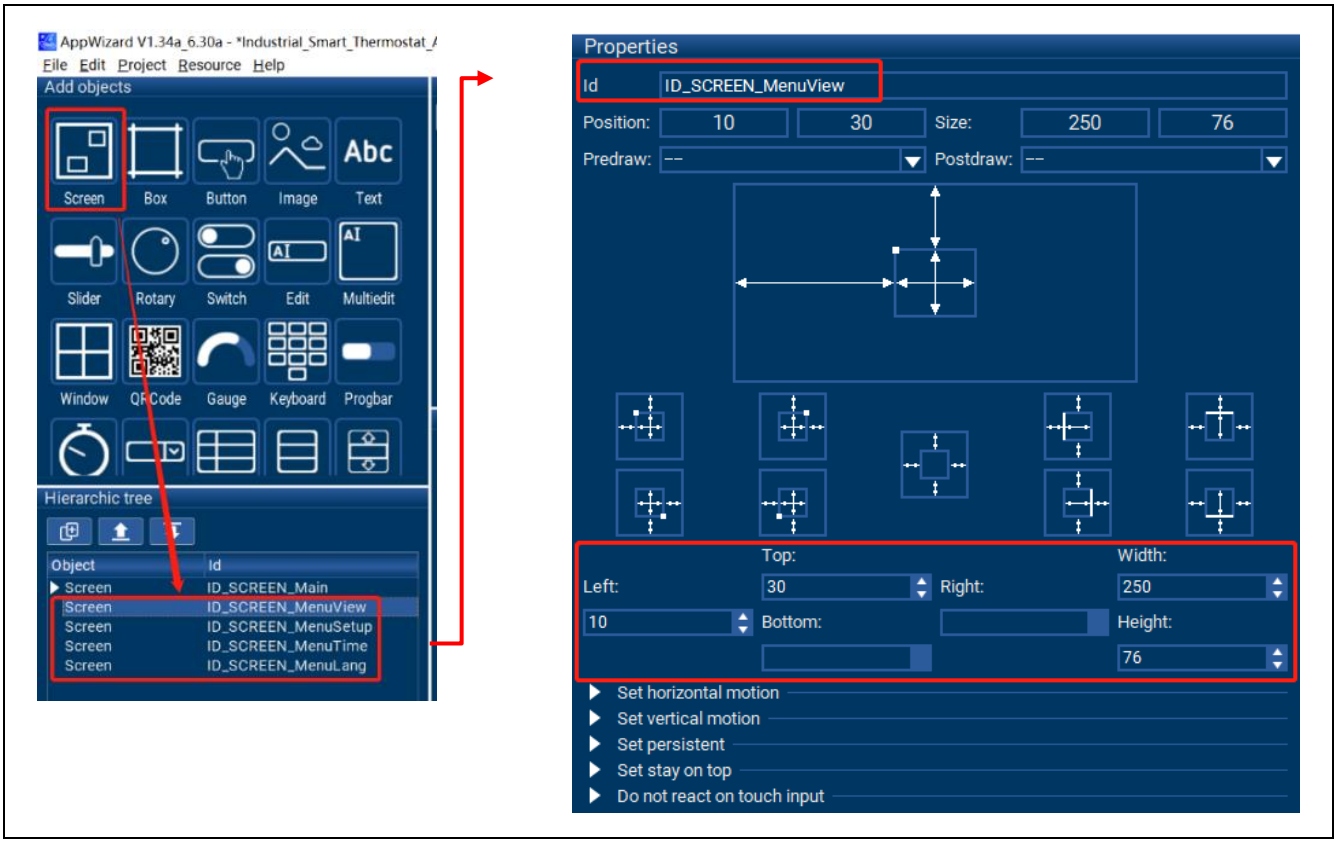


Figure 1-21 Create Drop-down Menu Screen

2. Add Window

Add a window to each drop-down menu screen. The window name and coordinates are shown in Table 1-13.

ID	Coordinates (Left, Top, Width, Height)
ID_WINDOW_MenuView	(0, 0, 250, 76)
D_WINDOW_MenuSetup	(0, 0, 200, 30)
D_WINDOW_MenuTime	(0, 0, 150, 52)
D_WINDOW_MenuLang	(0, 0, 150, 30)

Table 1-13 Description of Window Properties - Drop-down Menu Window

Select ID_SCREEN_MenuView in “Hierarchic tree”. Click “Window” in “Add Objects” to add a window. As shown in Table 1-13, rename it in “Hierarchic tree” and set the coordinates in “Properties”. The actual operation is shown in Figure 1-22. Follow the same method to add the other 3 windows.

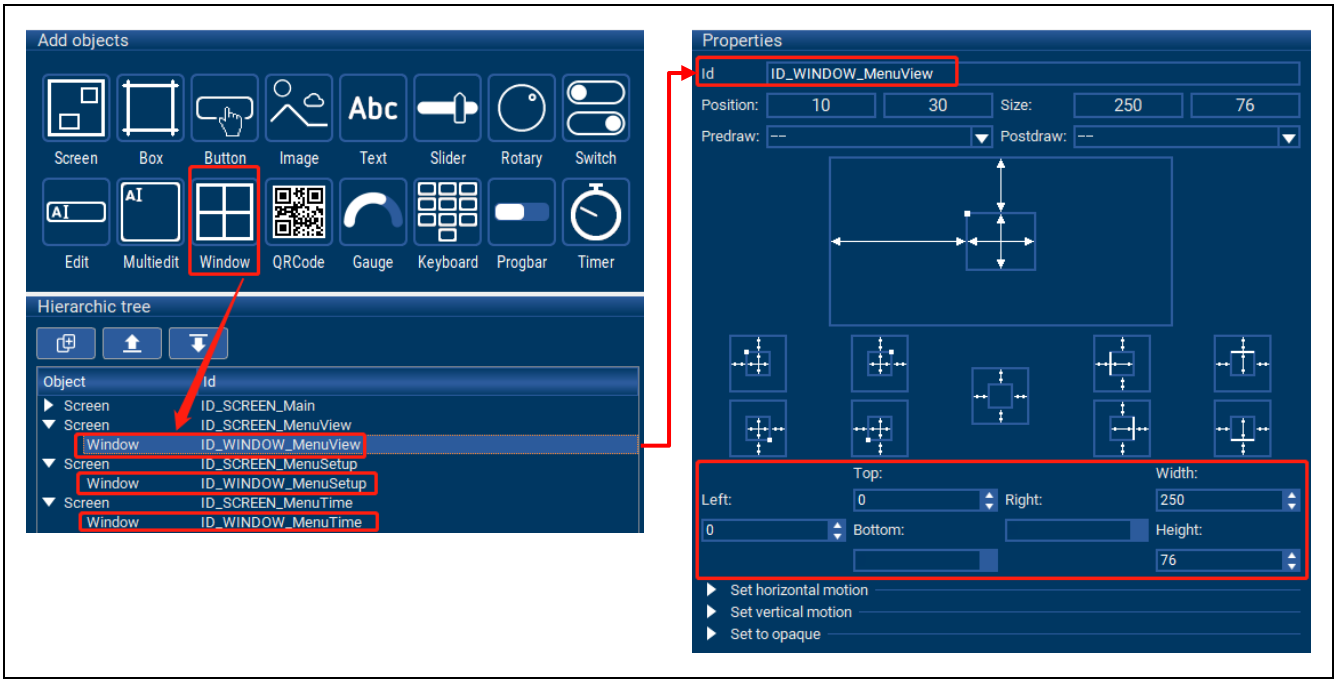


Figure 1-22 Add Drop-down Menu Window

3. Add ListBox

ListBox is used to display the contents of the drop-down menu.

1) Add ListBox

Select ID_WINDOW_MenuView in “Hierarchic tree”. Click “ListBox” in “Add Objects”, add a ListBox, and rename it to ID_LISTBOX_MenuView in “Hierarchic tree”. Follow the same method to add other 3 ListBoxes.

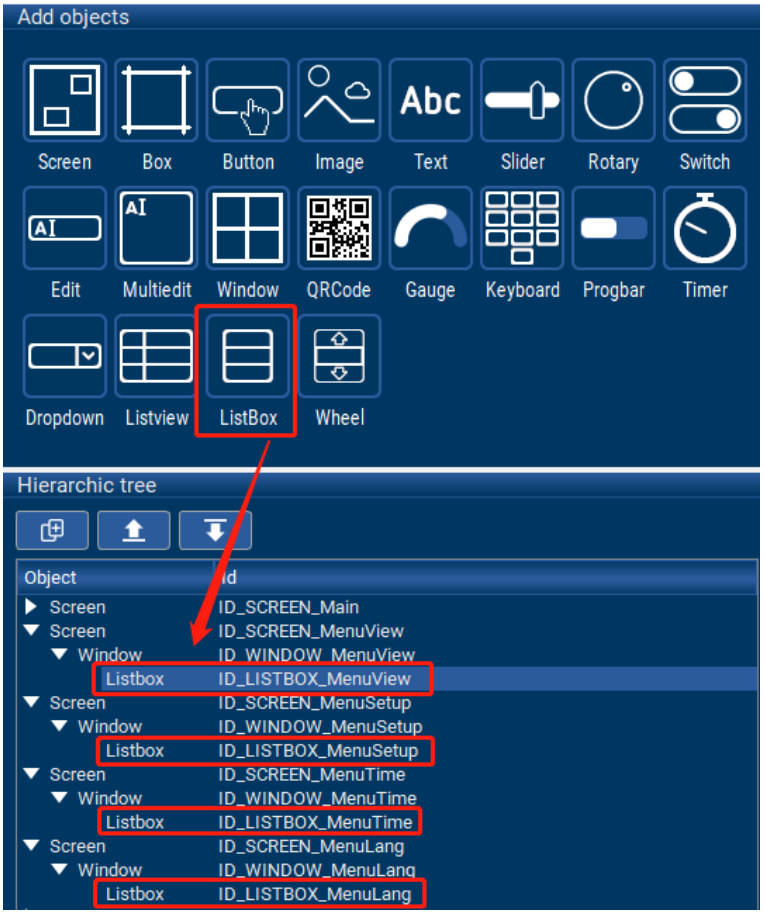


Figure 1-23 Add ListBox - Display Drop-down Menu

2) Set ListBox Properties

In “Properties”, set the coordinates and “text” according to Table 1-14, and set font to MicrosoftYaHeiUI_24_Normal_EXT_AA4. The actual setting items are shown in Figure 1-24. Set properties of the other 3 ListBoxes in the same way.

ID	Coordinates (Left, Top, Width, Height)	Display Content
ID_LISTBOX_MenuView	(0, 0, 250, 76)	ID_CONTEXT_VIEW
D_ LISTBOX_MenuSetup	(0, 0, 200, 30)	ID_CONTEXT_SETUP
D_ LISTBOX_MenuTime	(0, 0, 150, 52)	ID_CONTEXT_TIME
D_ LISTBOX_MenuLang	(0, 0, 150, 30)	ID_CONTEXT_LANG

Table 1-14 Description of ListBox Properties

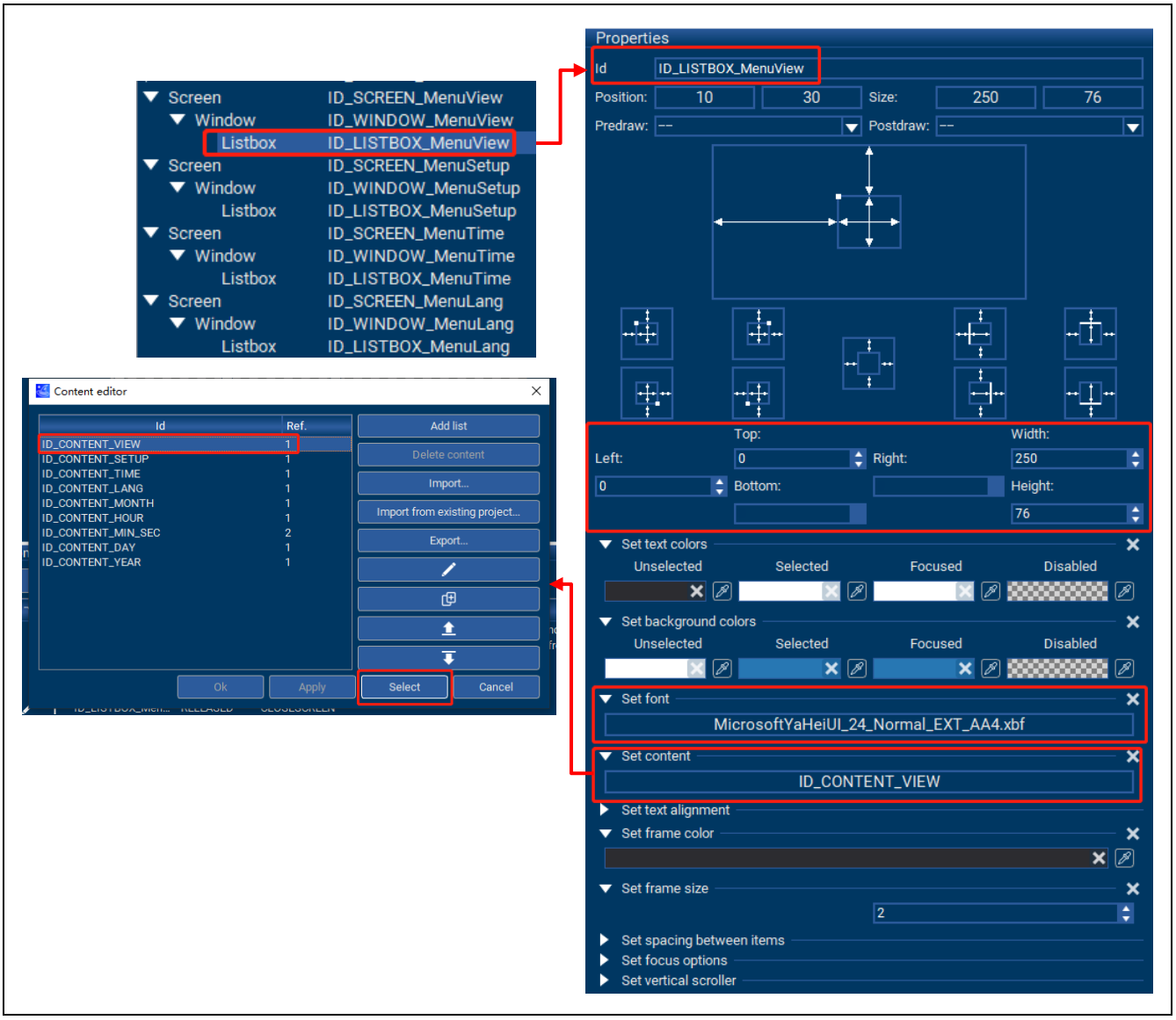


Figure 1-24 Set ListBox Properties

4. Drop-down Menu After Completion of Design

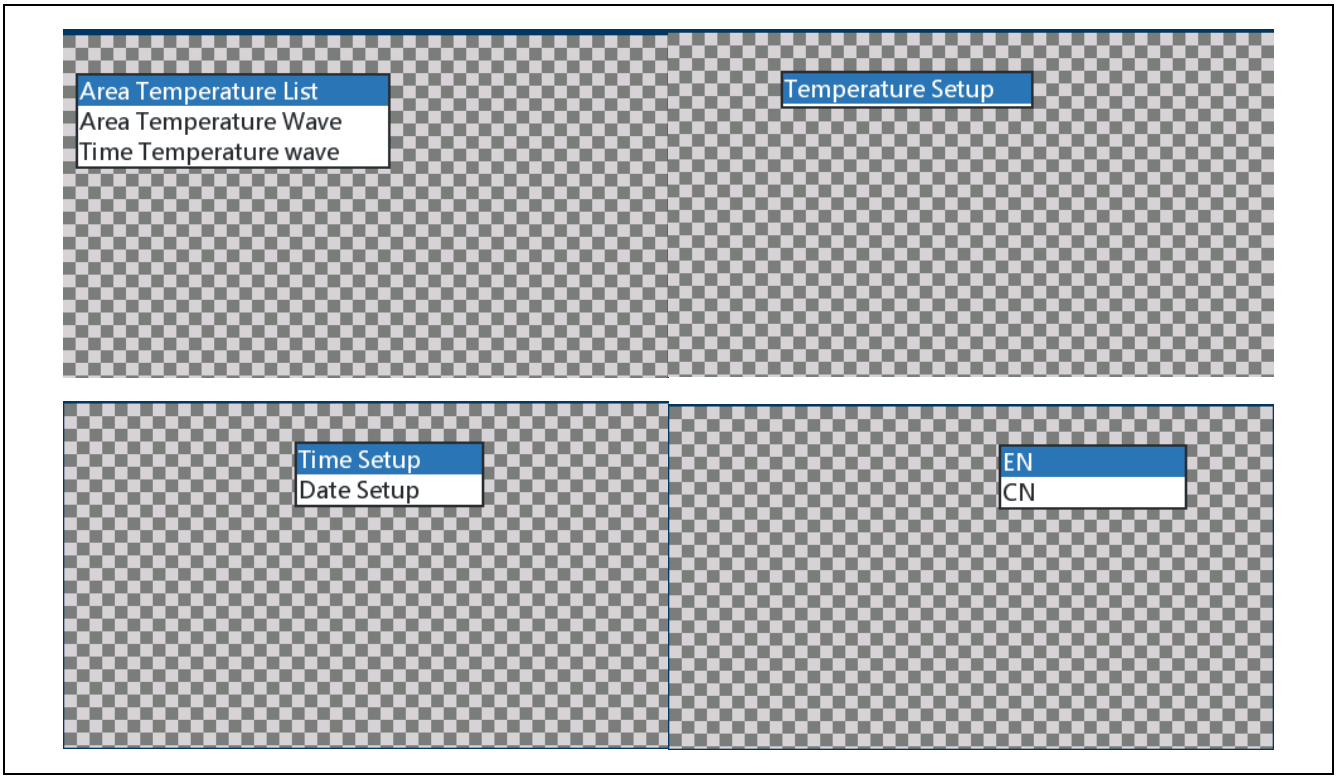


Figure 1-25 Drop-down Menu Screen

1.2.2.10 Create Time-Temperature Waveform Screen

1. Add Screen

Click “Screen” in “Add Objects” to add a screen, rename it to ID_SCREEN_TimeWave in “Hierarchic tree”, and set the coordinates to (0, 30, 480, 242) in “Properties”.

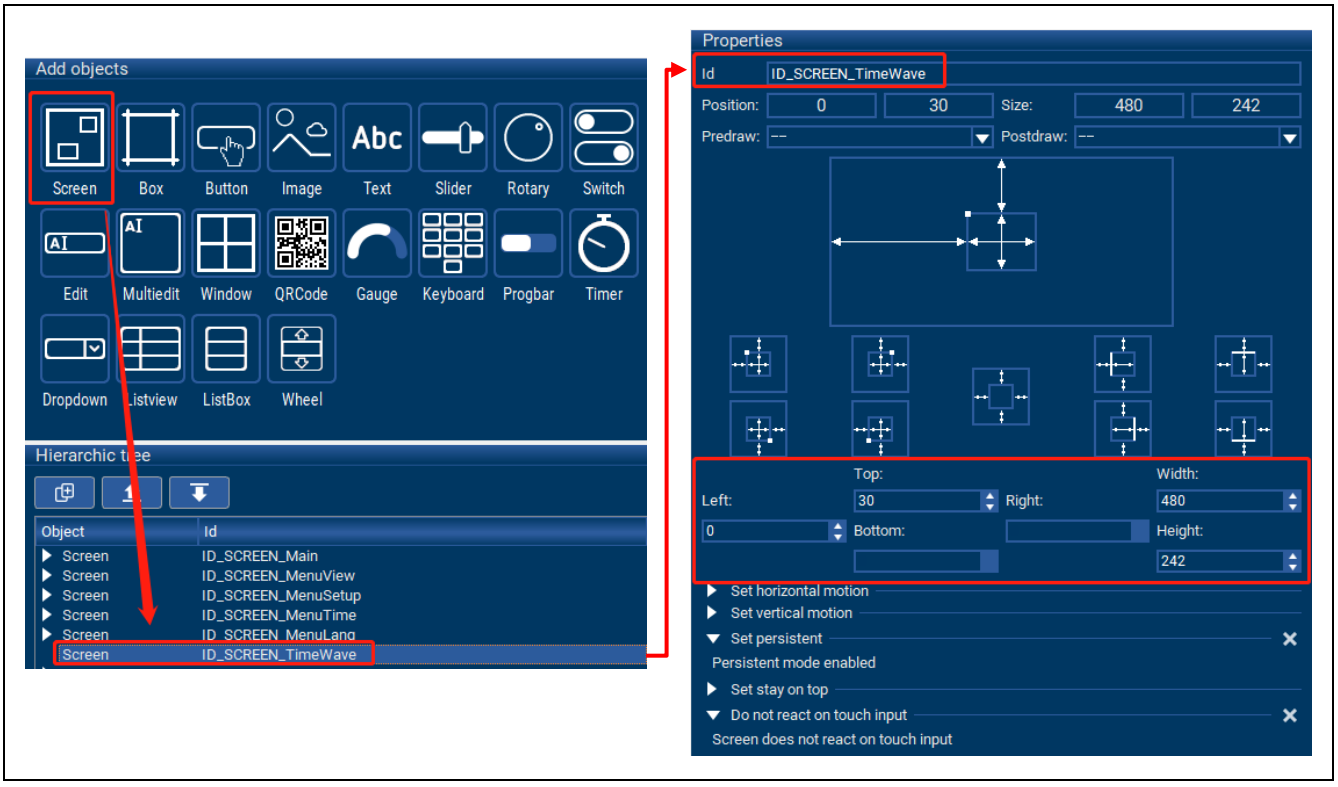


Figure 1-26 Add Time-Temperature Waveform Screen

2. Add Coordinate Display

Display coordinates of Time-Temperature waveform and color used to indicate area waveform.

1) Add Window

Click “Window” in “Add Objects” to add a window, rename it to ID_WINDOW_XY in “Hierarchic tree”, and set the coordinates to (0, 0, 480, 242) in “Properties”.

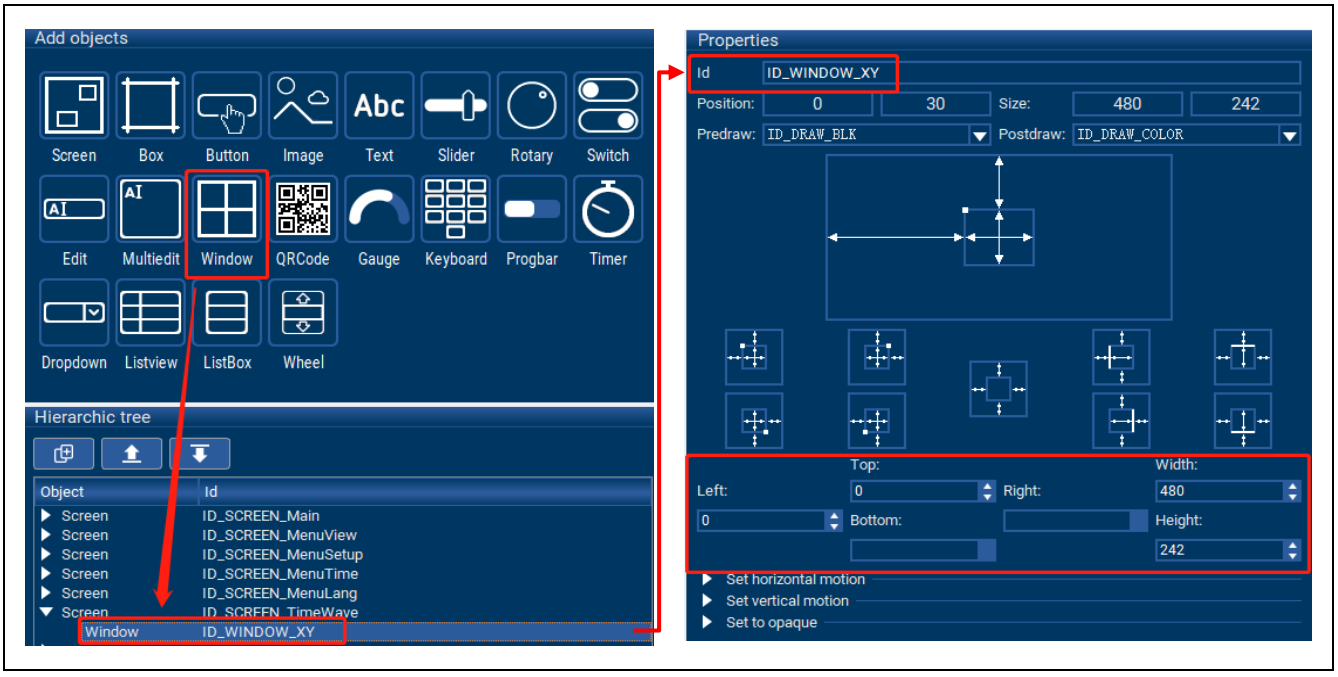


Figure 1-27 Create a Window - Coordinate Display of Time-Temperature Waveform

2) Add Image

Click “Image” in “Add Objects” to add an Image. Rename it to ID_IMAGE_BG in “Hierarchic tree”, as shown in Figure 1-28. In “Properties”, set the coordinates to (0, 0, 400, 242) and select BKG_Coordinate_480x242.png in “bitmap” as Time-Temperature waveform background image, which is mainly used to display the coordinate scale. The actual setting items are shown in Figure 1-29.

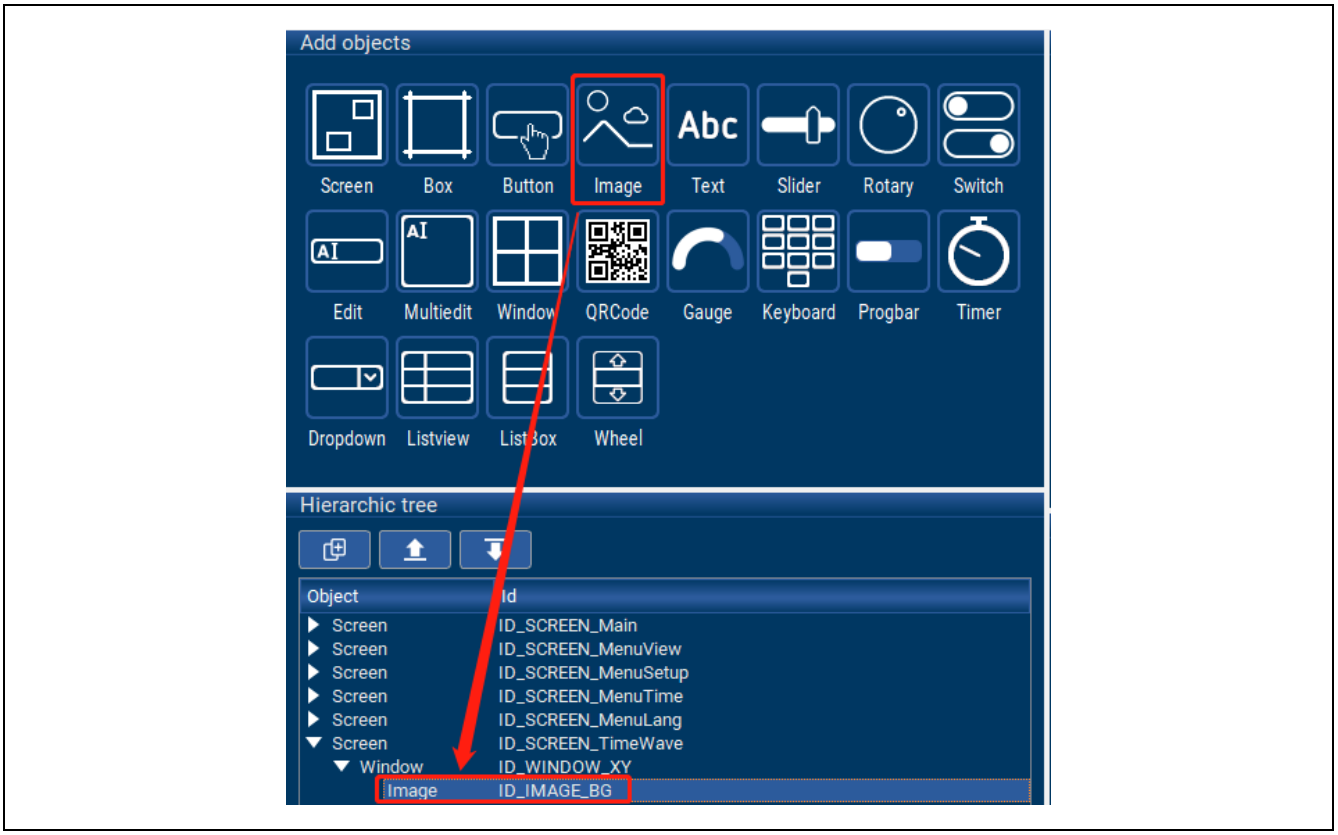


Figure 1-28 Add Image - Coordinate Background Display

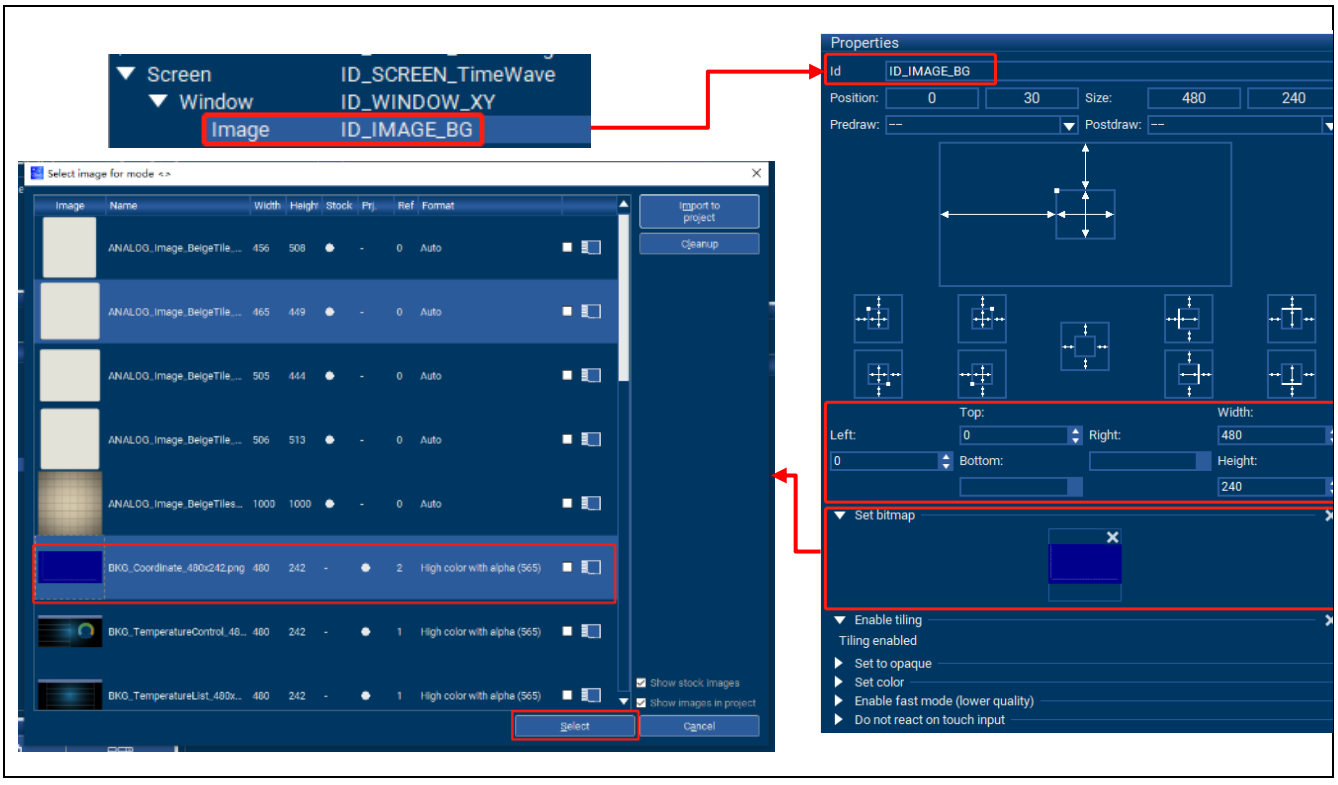


Figure 1-29 Set Image Properties

3) Add Text

Add Text to ID_WINDOW_XY window to display x coordinate (time) and y coordinate (temperature) of the waveform.

Click “Text” in “Add Objects”, add 23 Texts, and rename them as Figure 1-30 in “Hierarchic tree”.

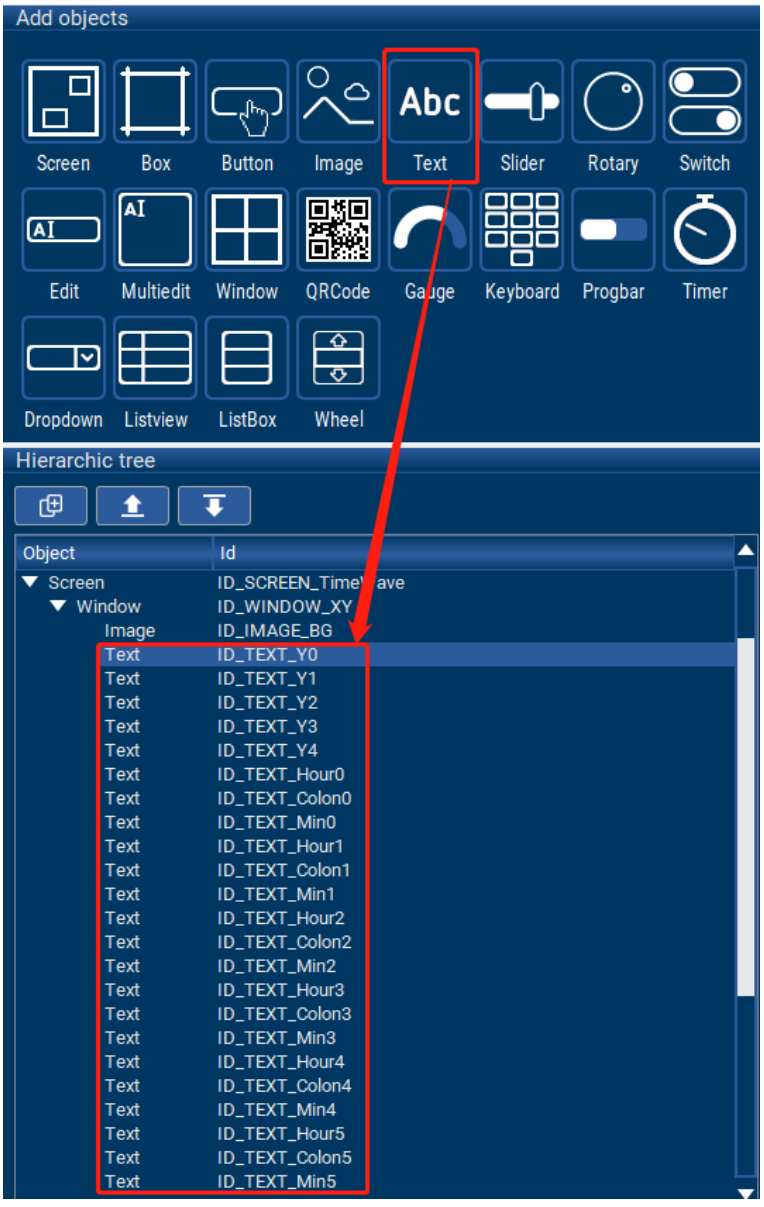


Figure 1-30 Add Text - Coordinate Value Display

Among them, 6 Texts are used to display text. In “Properties”, set “text color” to RGB (255, 255, 255), “text alignment” to “offset x” = “offset y” = 0 and “Alignment” to center, and “font” to MicrosoftYaHeiUI_20_Normal_EXT_AA4. Then set the coordinates and “text” according to Table 1-15. The actual setting items are shown in Figure 1-31.

ID	Coordinates (left, top, width, height)	text
ID_TEXT_Colon0	(34, 216, 8, 16)	ID_RTEXT_COLON
ID_TEXT_Colon1	(114, 216, 8, 16)	ID_RTEXT_COLON
ID_TEXT_Colon2	(194, 216, 8, 16)	ID_RTEXT_COLON
ID_TEXT_Colon3	(274, 216, 8, 16)	ID_RTEXT_COLON
ID_TEXT_Colon4	(354, 216, 8, 16)	ID_RTEXT_COLON
ID_TEXT_Colon5	(434, 216, 8, 16)	ID_RTEXT_COLON

Table 1-15 Description of Text Properties - Colon Display

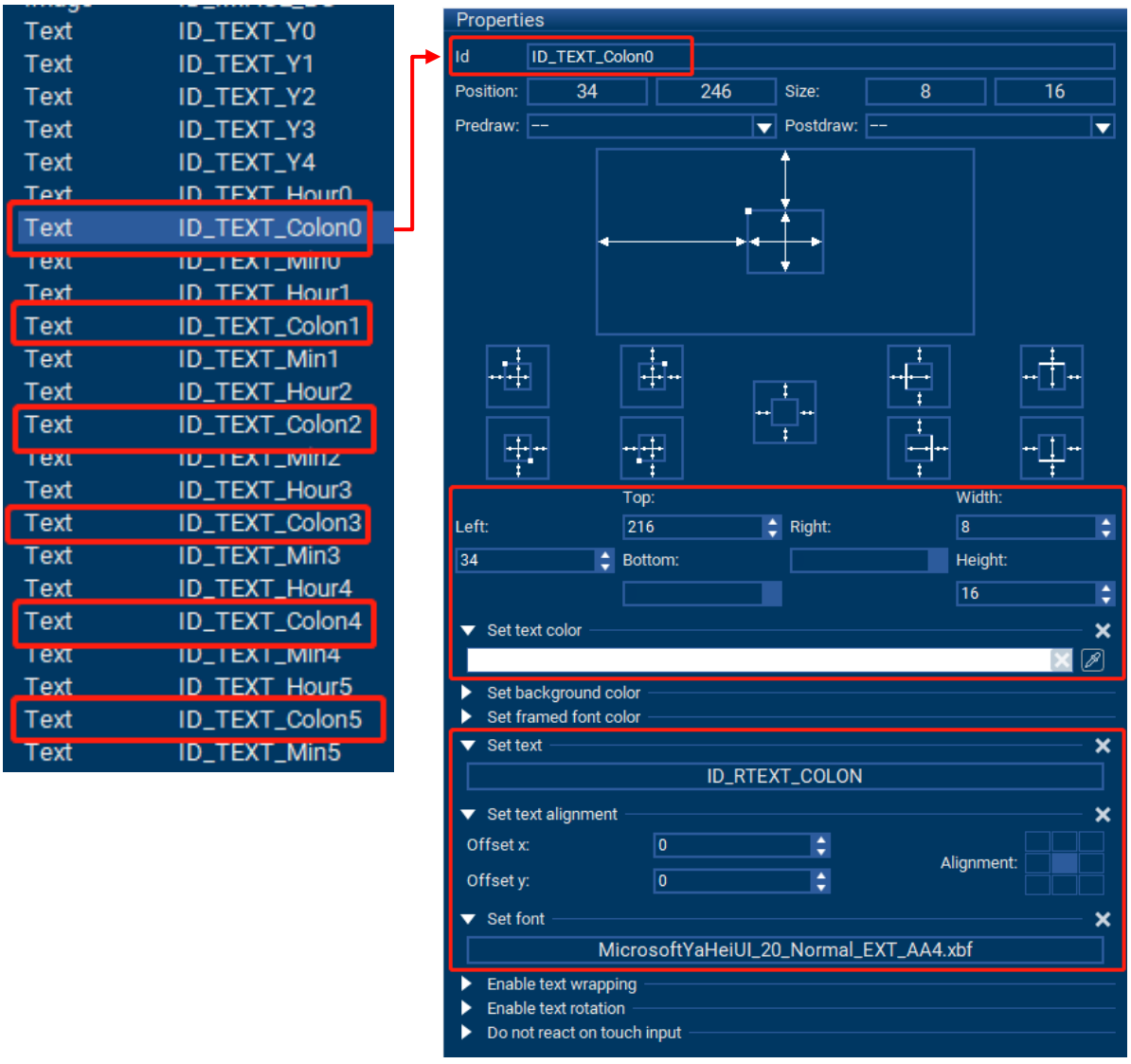


Figure 1-31 Set Text Properties - Colon Display

Among them, 17 Texts are used to display numbers, including temperature and time. In “Properties”, set “text color” to RGB (255, 255, 255) and “font” to MicrosoftYaHeiUI_20_Normal_EXT_AA4. Then set the coordinates, “decimal mode”, “rang”, “text alignment” according to Table 1-16. The actual setting items are shown in Figure 1-32.

ID	Coordinates (left, top, width, height)	Decimal mode	rang (min, max)	text alignment (offset x, offset y, Alignment)
ID_TEXT_Y0	(2, 196, 24, 20)	Mask:	(0, 0)	(0, 0, Right)
ID_TEXT_Y1	(2, 146, 24, 20)	Mask:	(100, 100)	(0, 0, Right)
ID_TEXT_Y2	(2, 96, 24, 20)	Mask:	(200, 200)	(0, 0, Right)
ID_TEXT_Y3	(2, 46, 24, 20)	Mask:	(300, 300)	(0, 0, Right)
ID_TEXT_Y4	(2, 0, 24, 20)	Mask:	(400, 400)	(0, 0, Right)
ID_TEXT_Hour0	(18, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, Right)
ID_TEXT_Min0	(42, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, Right)
ID_TEXT_Hour1	(98, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, Right)
ID_TEXT_Min1	(122, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, Right)
ID_TEXT_Hour2	(178, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, Right)
ID_TEXT_Min2	(202, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, Right)
ID_TEXT_Hour3	(258, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, Right)
ID_TEXT_Min3	(282, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, Right)
ID_TEXT_Hour4	(338, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, Right)
ID_TEXT_Min4	(362, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, Right)
ID_TEXT_Hour5	(418, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, Right)
ID_TEXT_Min5	(442, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, Right)

Table 1-16 Description of Text Properties - Digital Display

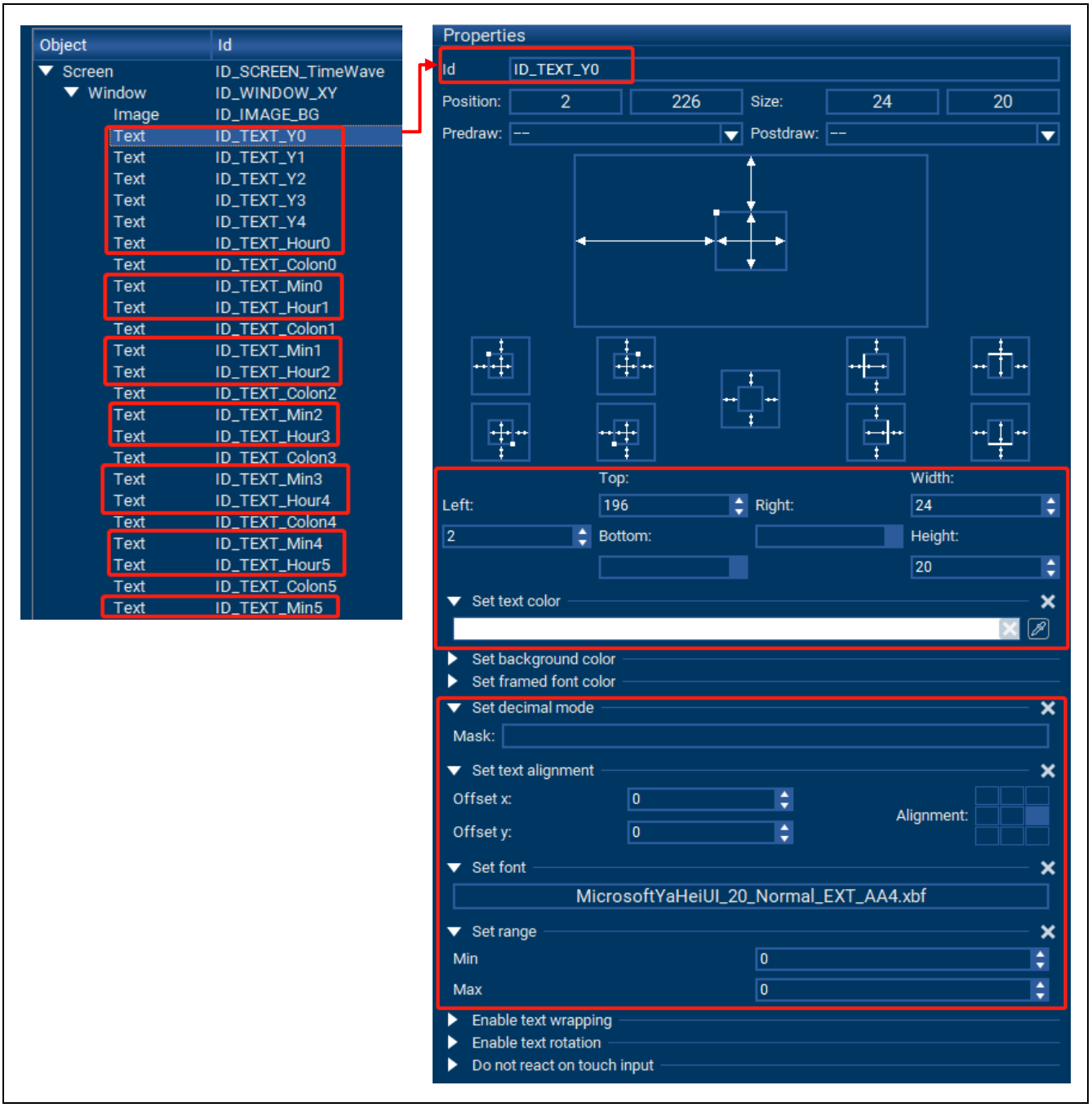


Figure 1-32 Set Text Properties - Digital Display

4) Add Box

Draw 5 line-segments of different colors to indicate the color used for each waveform. Click “Box” in “Add Objects” to add a Box to the window and rename it to ID_BOX_00 in “Hierarchic tree”. In “Properties”, set the coordinates to (440, 0, 40, 200), “color” to RGB (0, 0, 140), and select ID_DRAW_COLOR in the drop-down menu of “Postdraw” to draw 5 line-segments of different colors. The actual setting items are shown in Figure 1-33.

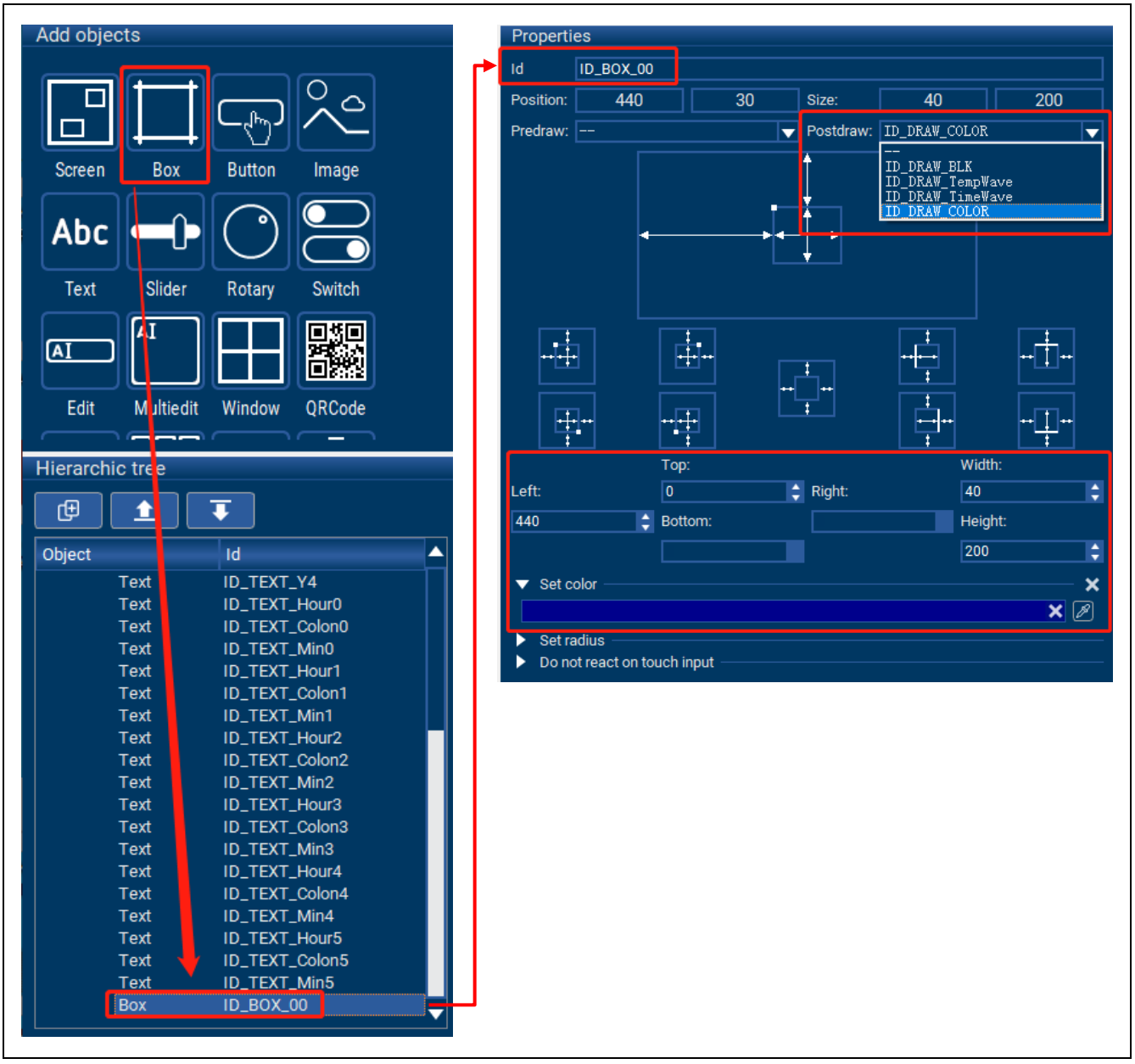


Figure 1-33 Draw 5 Line-Segments of Different Colors

5) Add Text

Add Texts to display area names, combined with the 5 line-segments drawn in the previous step, to indicate the color of each waveform.

Click “Text” in “Add Objects” to add 5 Texts. Rename them in “Hierarchic tree” as shown in Table 1-17. In “Properties”, set “text color” to RGB (255, 255, 255), “text alignment” to “offset x” = “offset y” = 0 and “Alignment” to center, “font” to MicrosoftYaHeiUI_20_Normal_EXT_AA4. Set the coordinates and “text” according to Table 1-17. The actual setting items are shown in Figure 1-34.

ID	Coordinates (left, top, width, height)	text
ID_TEXT_Area1	(440, 0, 40, 40)	ID_RTEXT_AREA1
ID_TEXT_Area2	(440, 30, 40, 40)	ID_RTEXT_AREA2
ID_TEXT_Area3	(440, 60, 40, 40)	ID_RTEXT_AREA3
ID_TEXT_Area4	(440, 90, 40, 40)	ID_RTEXT_AREA4
ID_TEXT_Area5	(440, 120, 40, 40)	ID_RTEXT_AREA5

Table 1-17 Description of Text Properties - Area Name Display

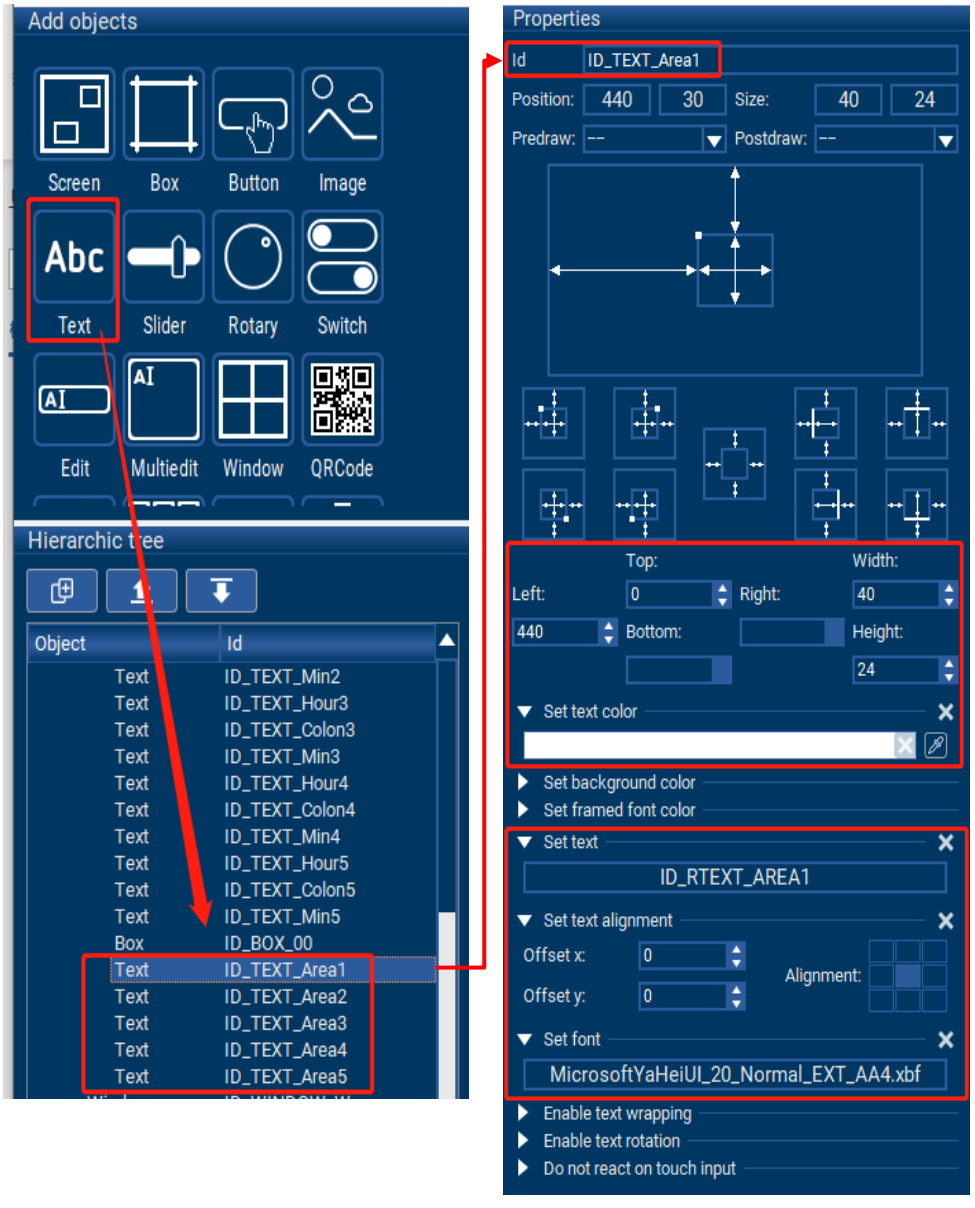


Figure 1-34 Add Text - Area Name Display

3. Add Window

Add a window to draw Time-Temperature waveform. First select ID_SCREEN_TimeWave, click “Window” in “Add Objects” to add a window, and rename it to ID_WINDOW_Wave in “Hierarchic tree”. In “Properties”, set the coordinates to (38, 5, 400, 200), and select Pre-draw action as ID_DRAW_BLK in the drop-down menu of “Predraw”, Select the drawing action as ID_Draw_TimeWave in the drop-down menu of “Postdraw”.

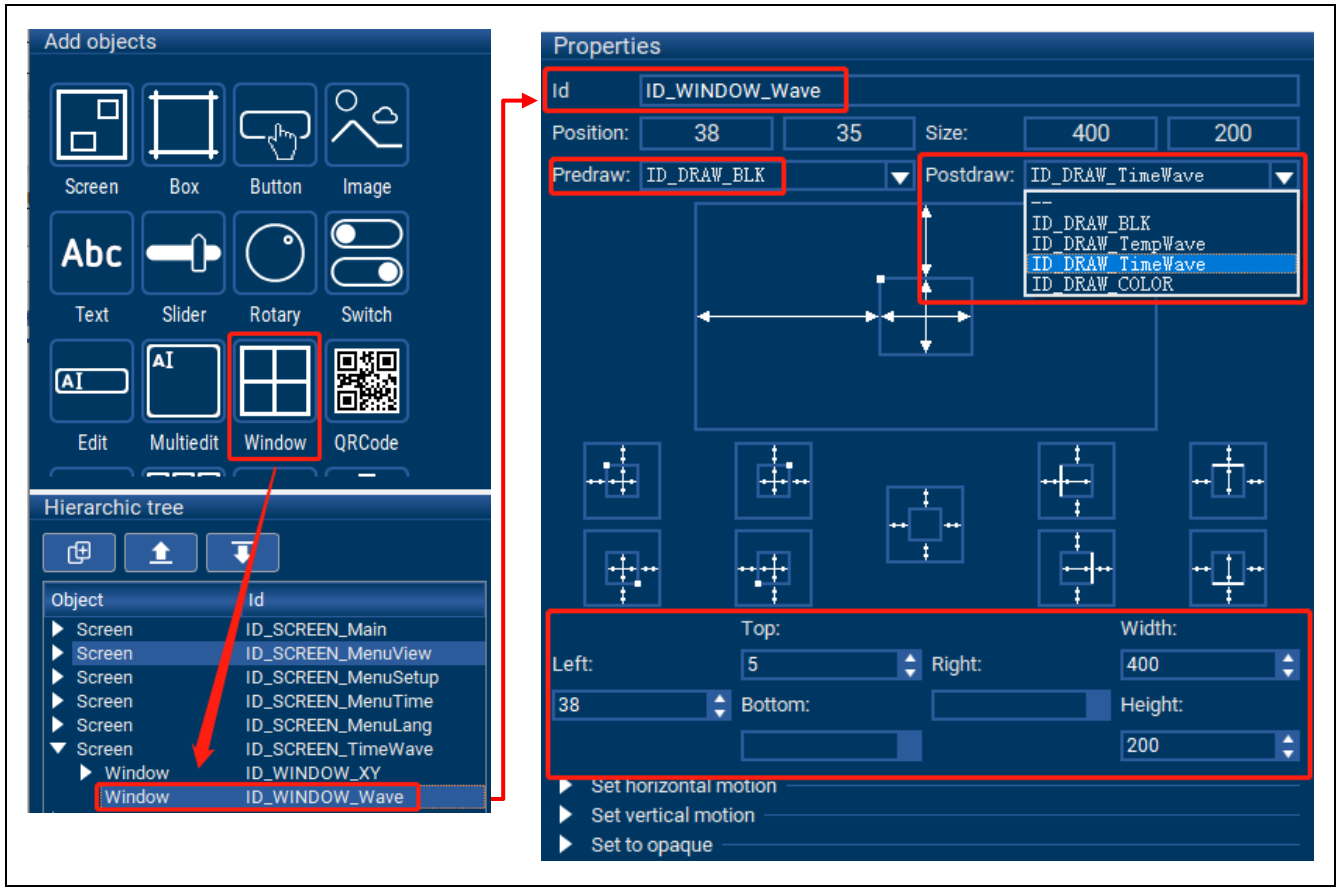


Figure 1-35 Add Time-Temperature Waveform Drawing Window

4. Add Timer

Add a Timer to refresh waveform regularly. Click “Timer” in “Add Objects” to add a timer. In “Properties” set “timer period” to 1000 ms. Timer does not have any display and only runs in the background.

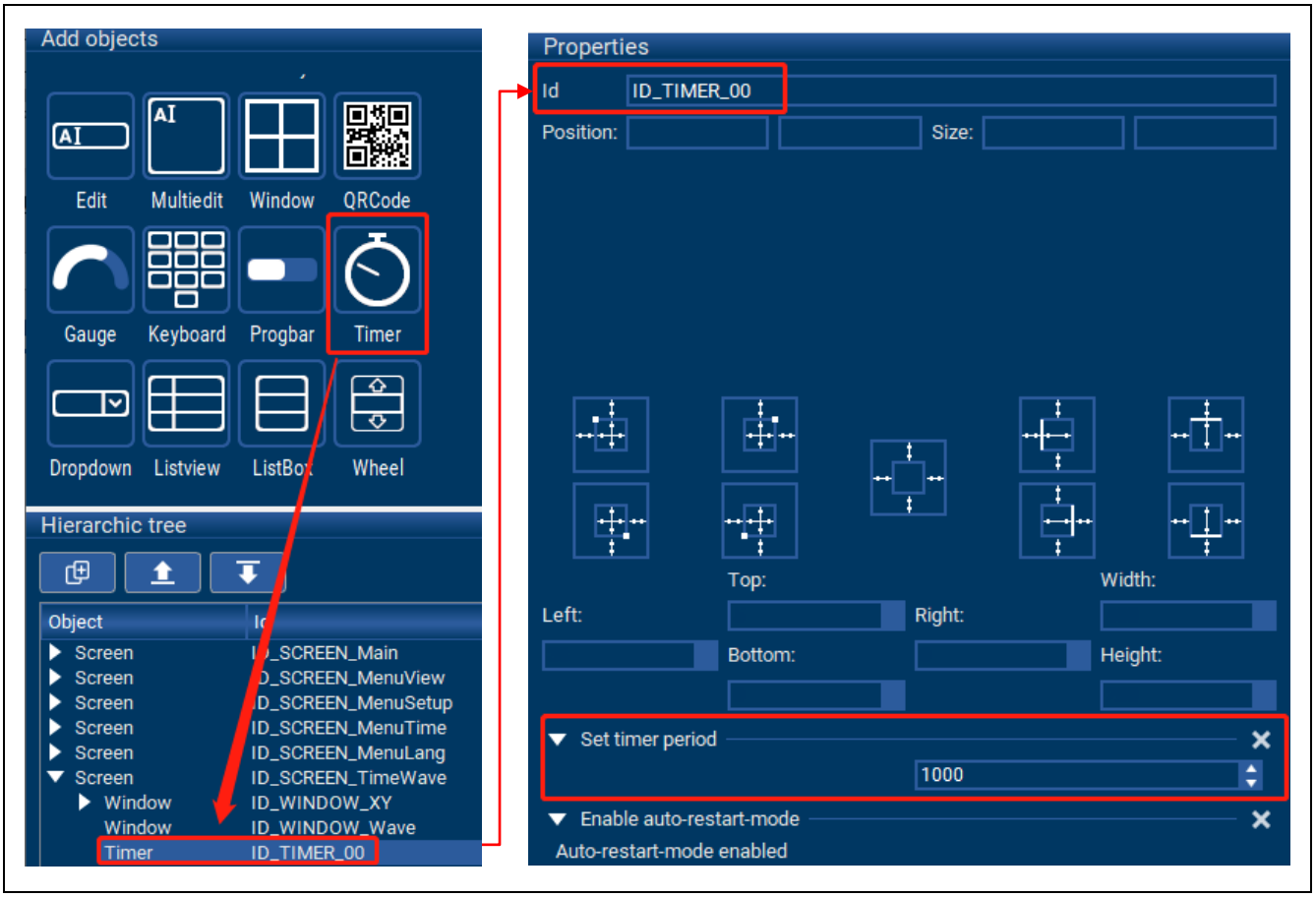


Figure 1-36 Add Timer

5. Time-Temperature Waveform After Completion of Design

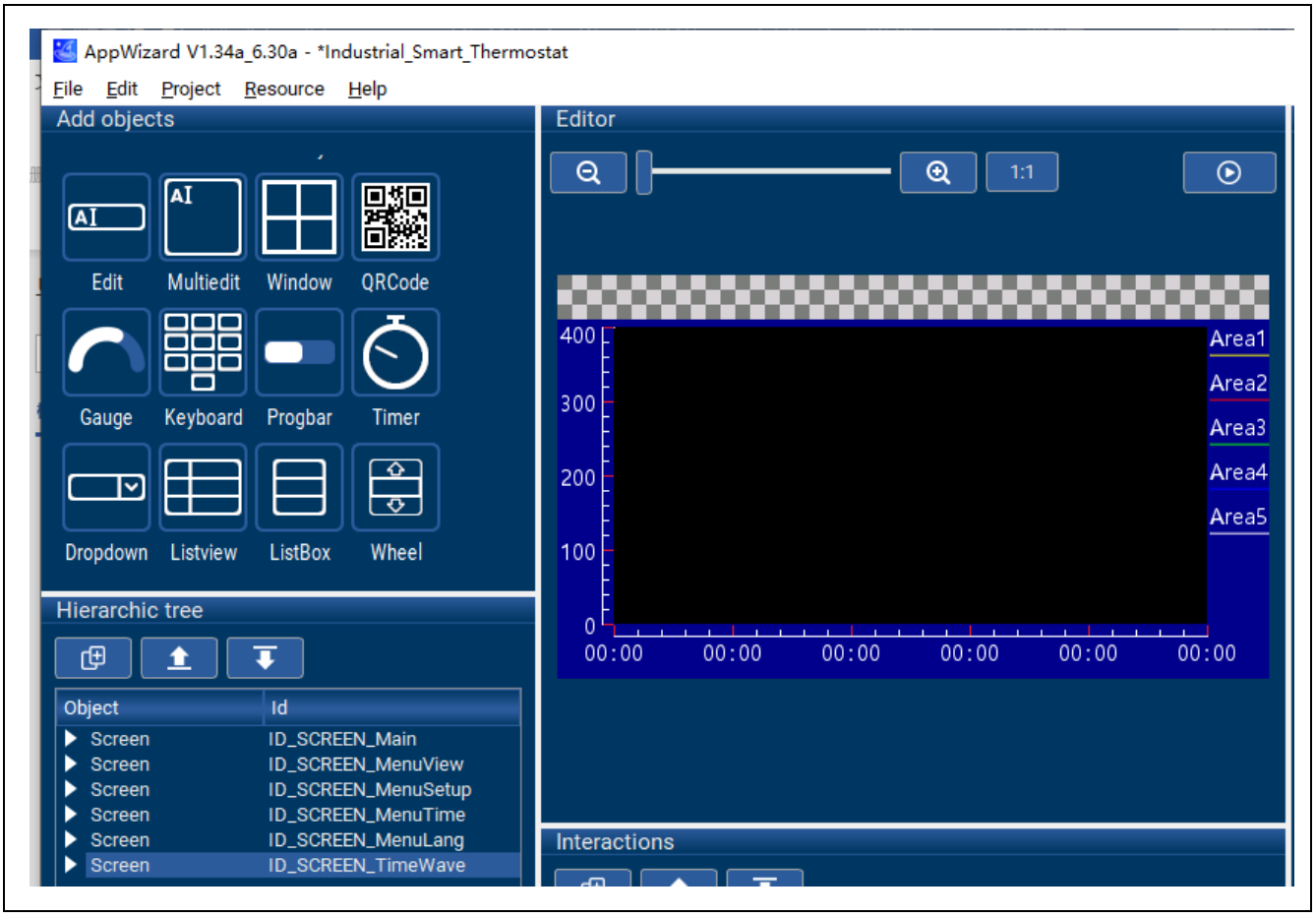


Figure 1-37 Time-Temperature Waveform Screen

1.2.2.11 Create Area-Temperature Waveform Screen

1. Add Screen

Click “Screen” in “Add Objects” to add a screen, rename it to ID_SCREEN_TempWave in “Hierarchic tree”, and set the coordinates to (0, 30, 480, 242) in “Properties”.

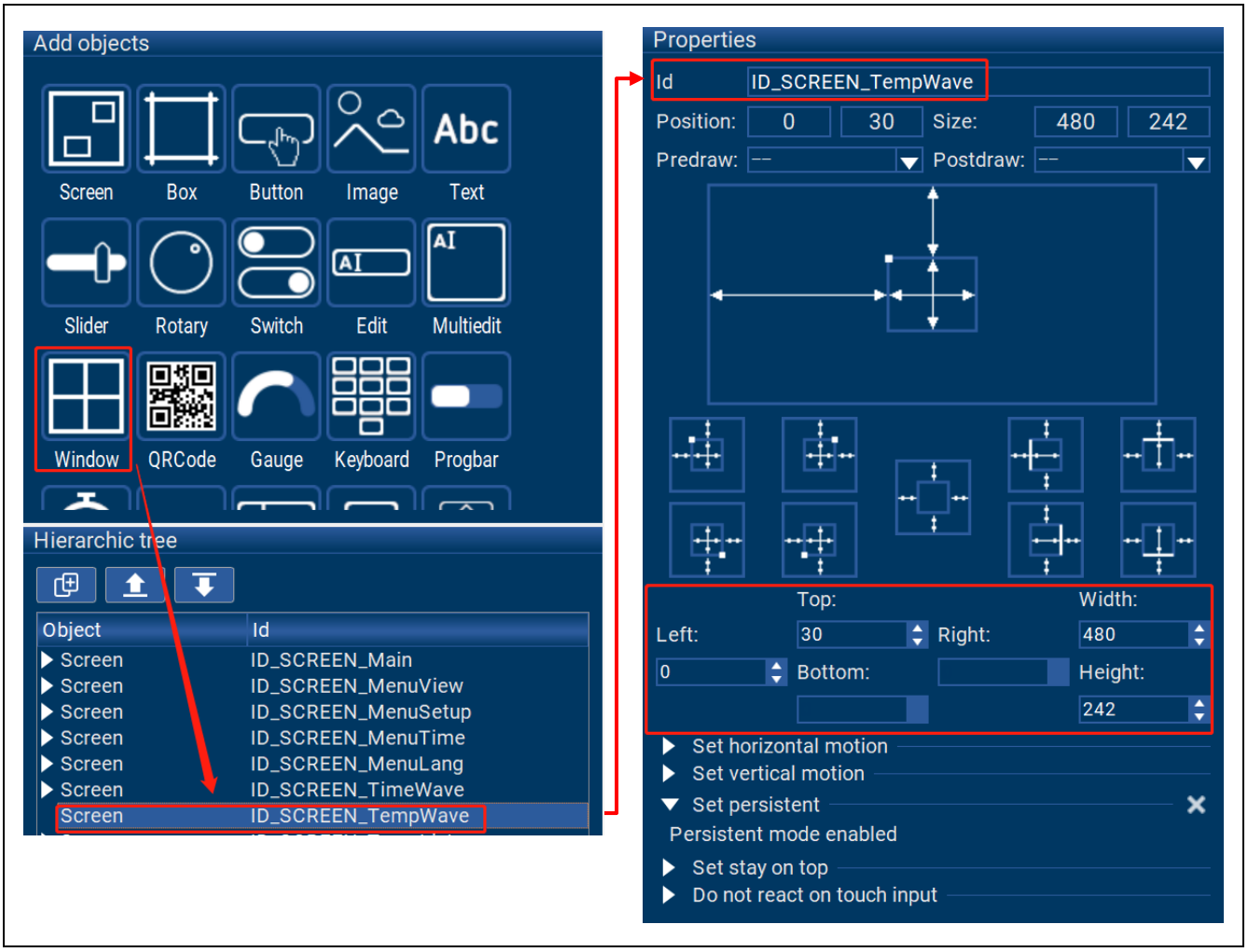


Figure 1-38 Add Area-Temperature Waveform Screen

2. Add Coordinate Display

Display the coordinates of Area-Temperature waveform.

1) Add Window

Click “Window” in “Add Objects” to add a window, rename it to ID_WINDOW_XY in “Hierarchic tree”, and set the coordinates to (0, 0, 480, 242) in “Properties”.

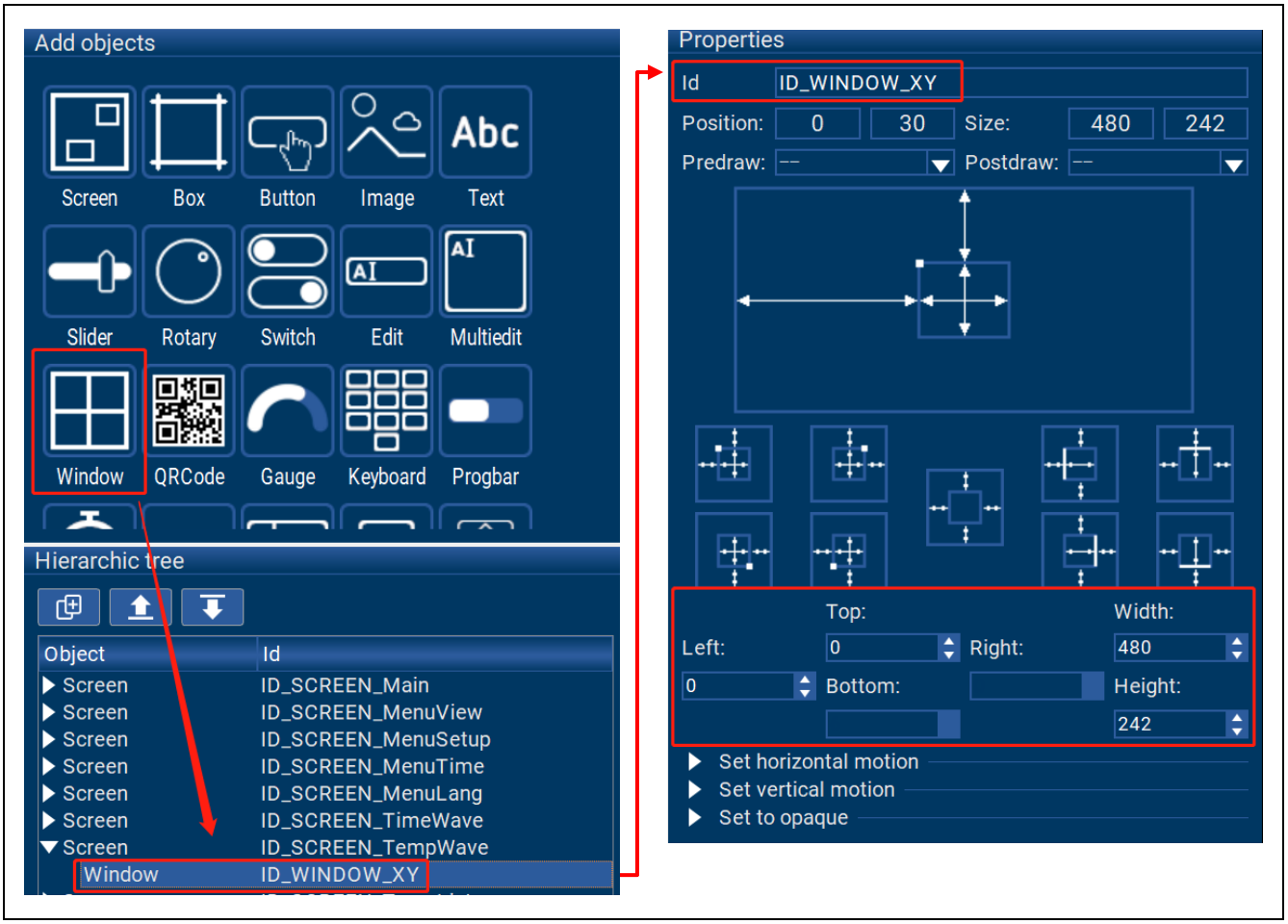


Figure 1-39 Create a Window - Coordinate Display of Area-Temperature Waveform

2) Add Image

Click “Image” in “Add Objects” to add an Image and rename it to ID_IMAGE_BG in “Hierarchic tree”. In “Properties”, set the coordinates to (0, 0, 400, 242) and select BKG_Coordinate_480x242.png in “bitmap” as background image of Area-Temperature waveform screen. The actual setting items are shown in Figure 1-40 and Figure 1-41.

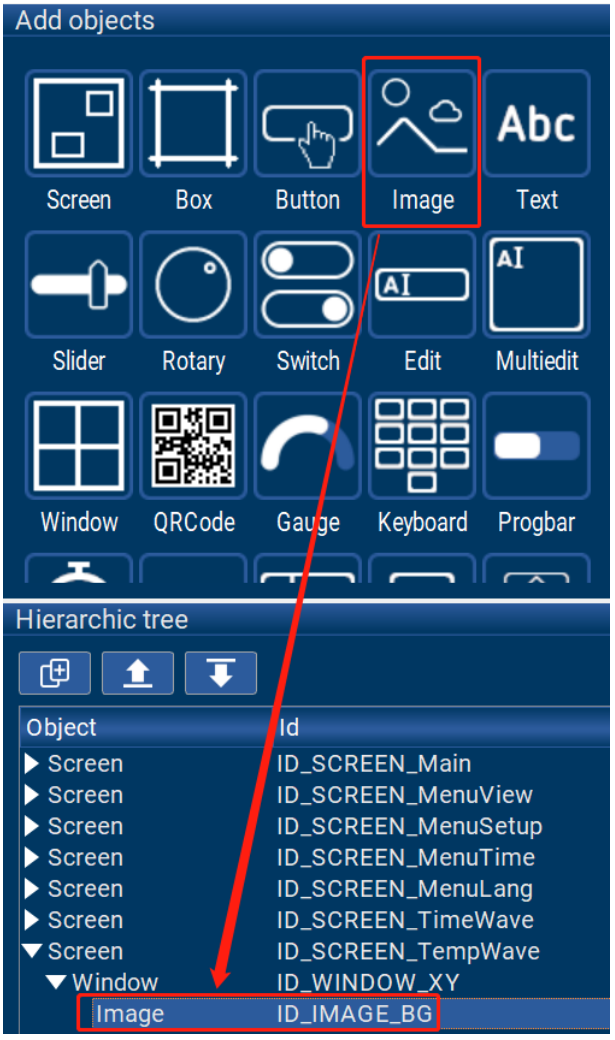


Figure 1-40 Add Image - Coordinate Background Display

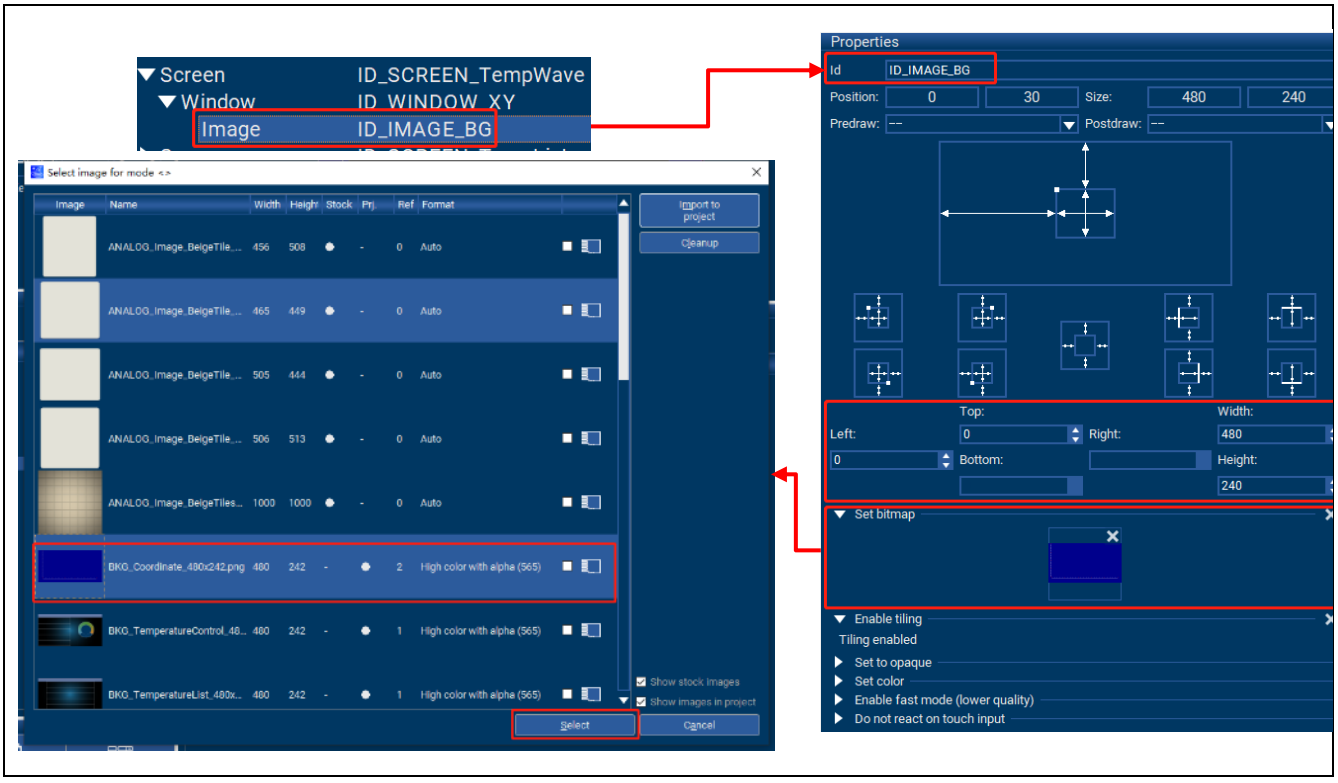


Figure 1-41 Set Image Properties

3) Add Text

Add Texts to ID_WINDOW_XY window to display x coordinate (area name) and y coordinate (temperature) of the waveform. Click “Text” in “Add Objects” to add 10 Texts and rename them in “Hierarchic tree” as Figure 1-42.

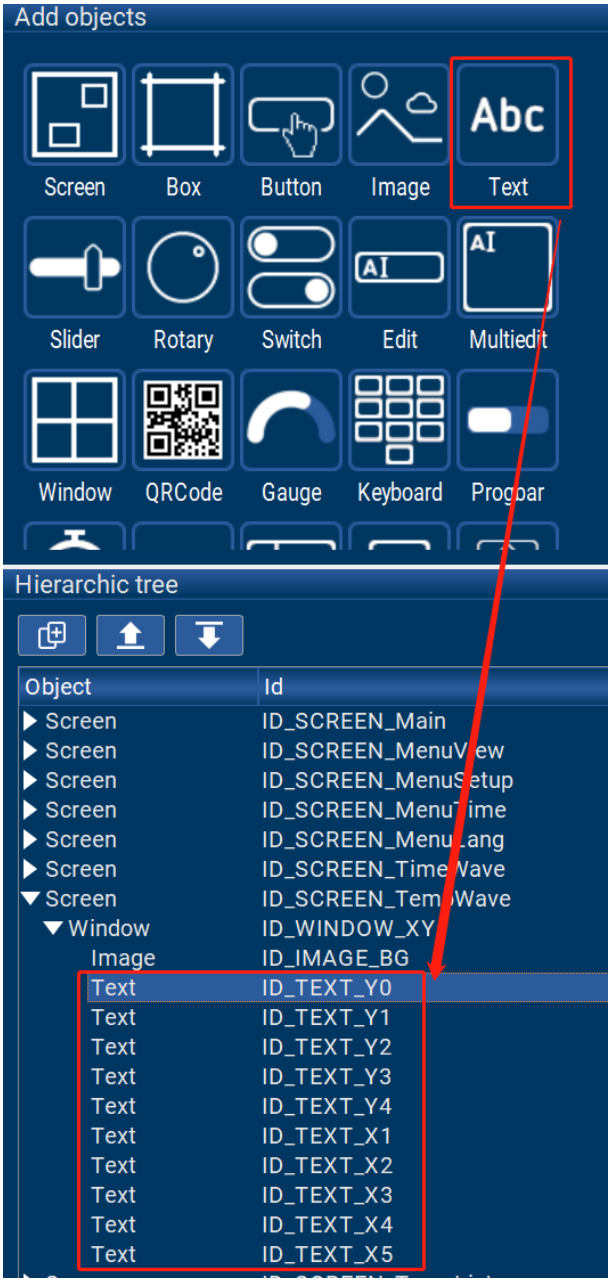


Figure 1-42 Add Text - Coordinate Value Display

Among them, 5 Texts are used to display text. In “Properties”, set “text color” to RGB (255, 255, 255), “text alignment” to “offset x” = “offset y” = 0 and “Alignment” to center, “font” to MicrosoftYaHeiUI_20_Normal_EXT_AA4. Set the coordinates and “text” according to Table 1-18. The actual setting items are shown in Figure 1-43.

ID	**Coordinates (left, top, width, height)**	**text**
ID_TEXT_X0	(100, 212, 40, 24)	ID_RTEXT_AREA1
ID_TEXT_X1	(180, 212, 40, 24)	ID_RTEXT_AREA2
ID_TEXT_X2	(260, 212, 40, 24)	ID_RTEXT_AREA3
ID_TEXT_X3	(340, 212, 40, 24)	ID_RTEXT_AREA4
ID_TEXT_X4	(420, 212, 40, 24)	ID_RTEXT_AREA5

Table 1-18 Description of Text Properties - x Coordinate Value Display

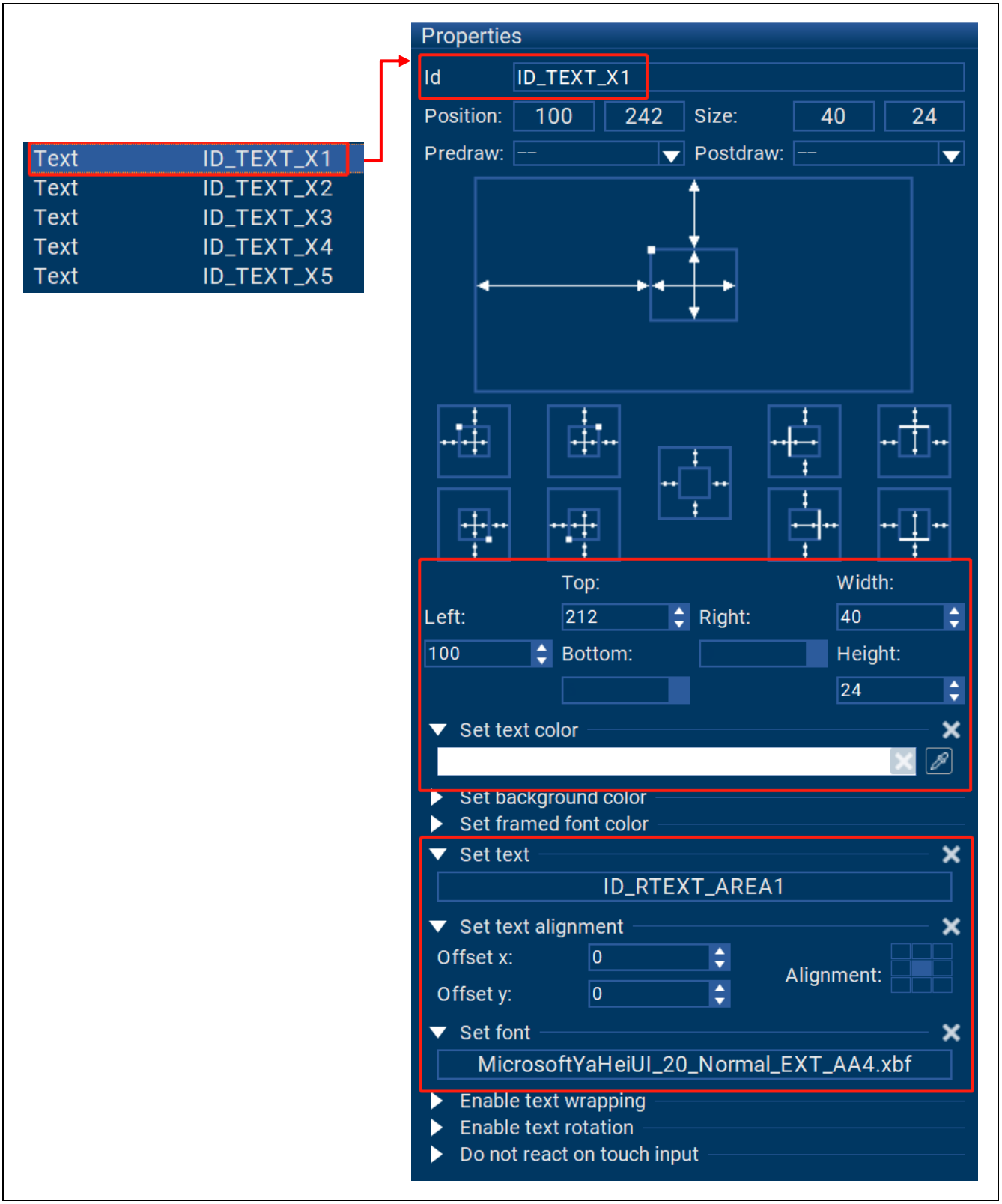


Figure 1-43 Set Text Properties - x Coordinate Display

Among them, 5 Texts are used to display numbers. In “Properties”, set “text color” to RGB (255, 255, 255), “font” to MicrosoftYaHeiUI_20_Normal_EXT_AA4. Set the coordinates, “decimal mode”, “rang”, and “text alignment” according to Table 1-19. The actual setting items are shown in Figure 1-44.

ID	**Coordinates (left, top, width, height)**	**Decimal mode**	**Rang (min, max)**	**text alignment (Offset x, offset y, Alignment)**
ID_TEXT_Y0	(2, 196, 24, 20)	Mask:	(0, 0)	(0, 0, Right)
ID_TEXT_Y1	(2, 146, 24, 20)	Mask:	(100, 100)	(0, 0, Right)
ID_TEXT_Y2	(2, 96, 24, 20)	Mask:	(200, 200)	(0, 0, Right)
ID_TEXT_Y3	(2, 46, 24, 20)	Mask:	(300, 300)	(0, 0, Right)
ID_TEXT_Y4	(2, 0, 24, 20)	Mask:	(400, 400)	(0, 0, Right)

Table 1-19 Description of Text Properties - y Coordinate Display

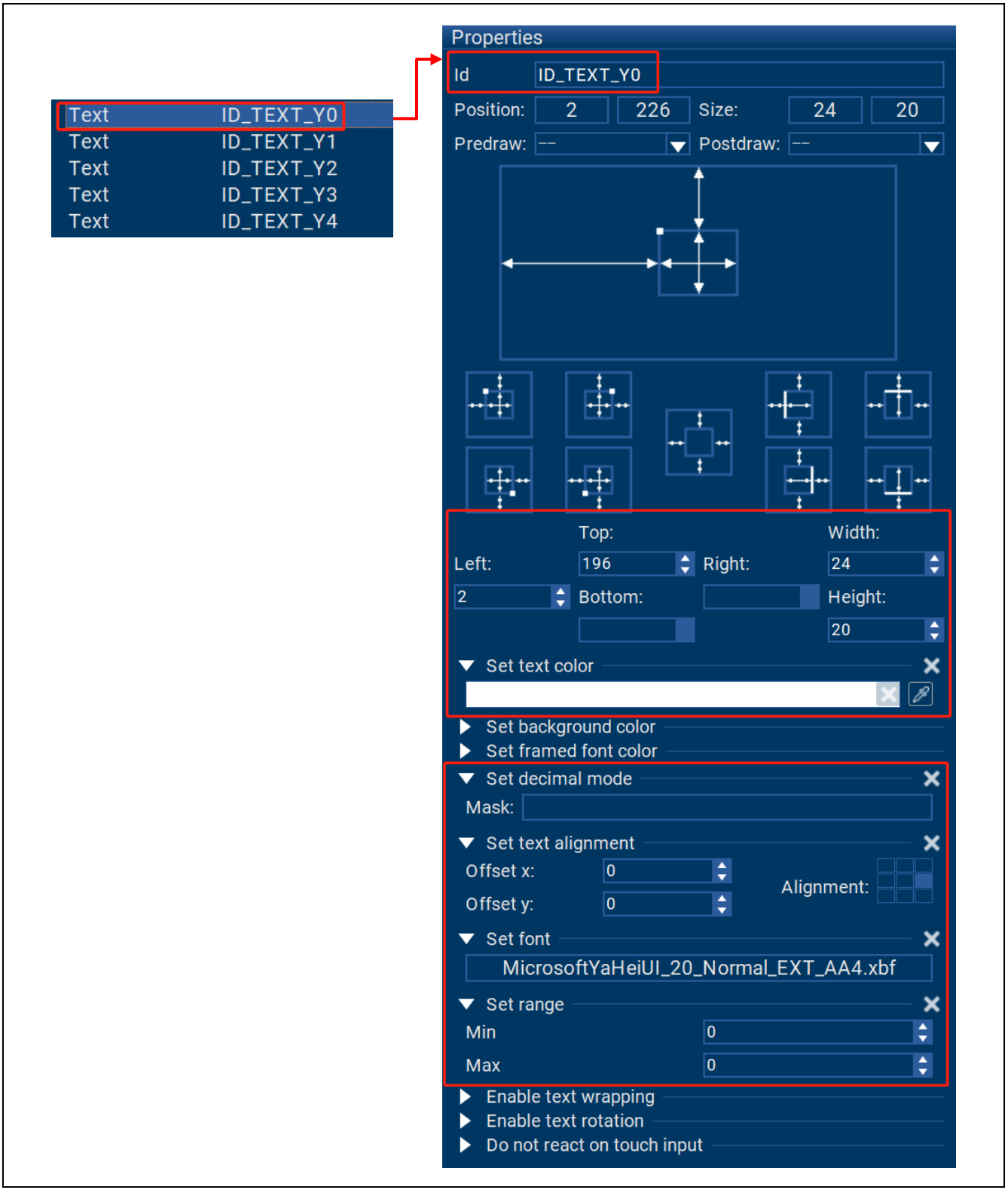


Figure 1-44 Set Text Properties - y Coordinate Display

3. Add Window

Add a window to draw Time-Temperature waveform. First select ID_SCREEN_TempWave, click “Window” in “Add Objects” to add a window, and rename it to ID_WINDOW_Wave in “Hierarchic tree”. In “Properties”, set the coordinates to (38, 5, 400, 200). Select Pre-draw action as ID_DRAW_BLK in the drop-down menu of “Predraw”. Select the drawing action as ID_Draw_TempWave in the drop-down menu of “Postdraw”.

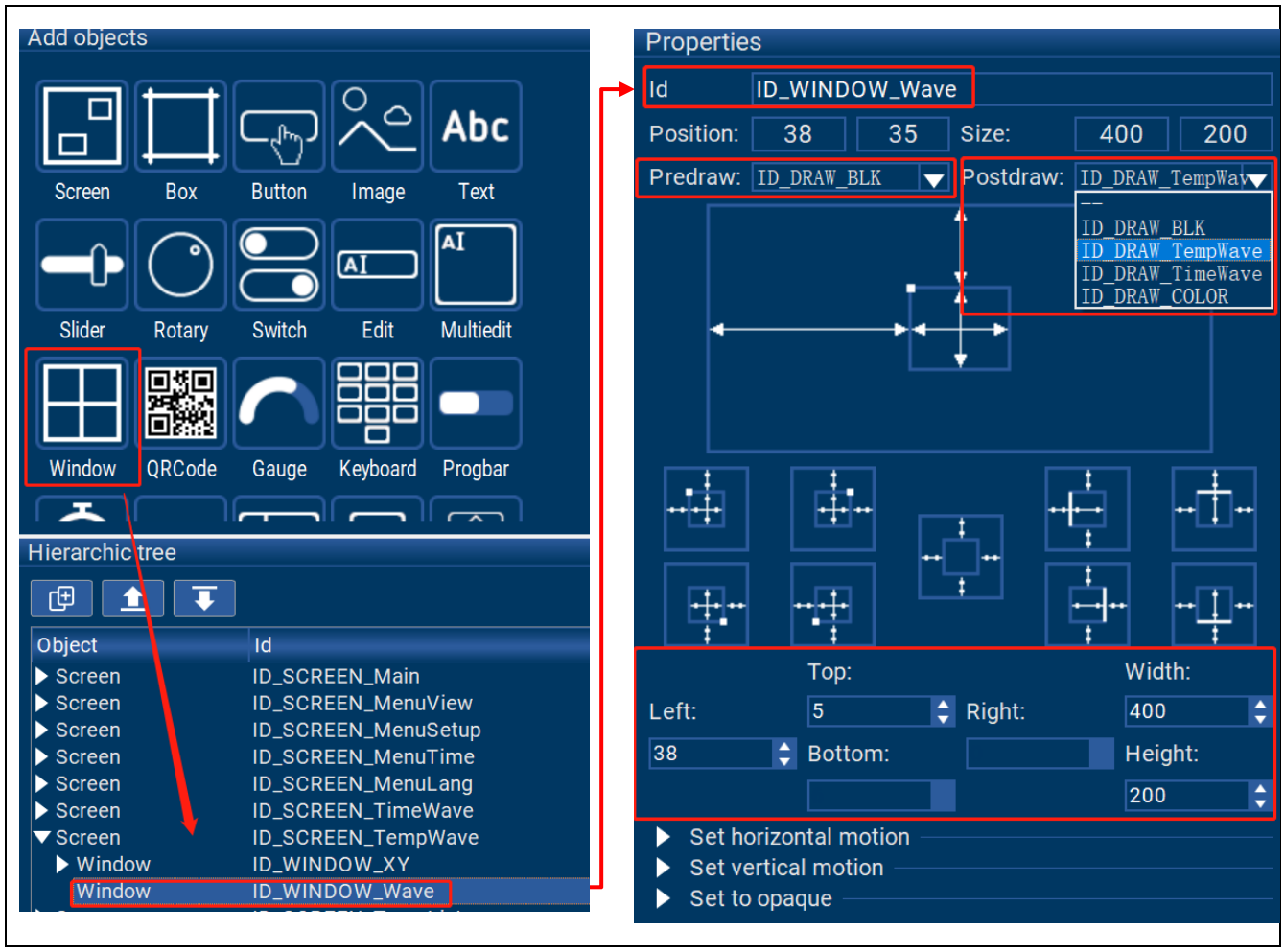


Figure 1-45 Add a Window - Area-Temperature Waveform Drawing

4. Area-Temperature Waveform After Completion of Design

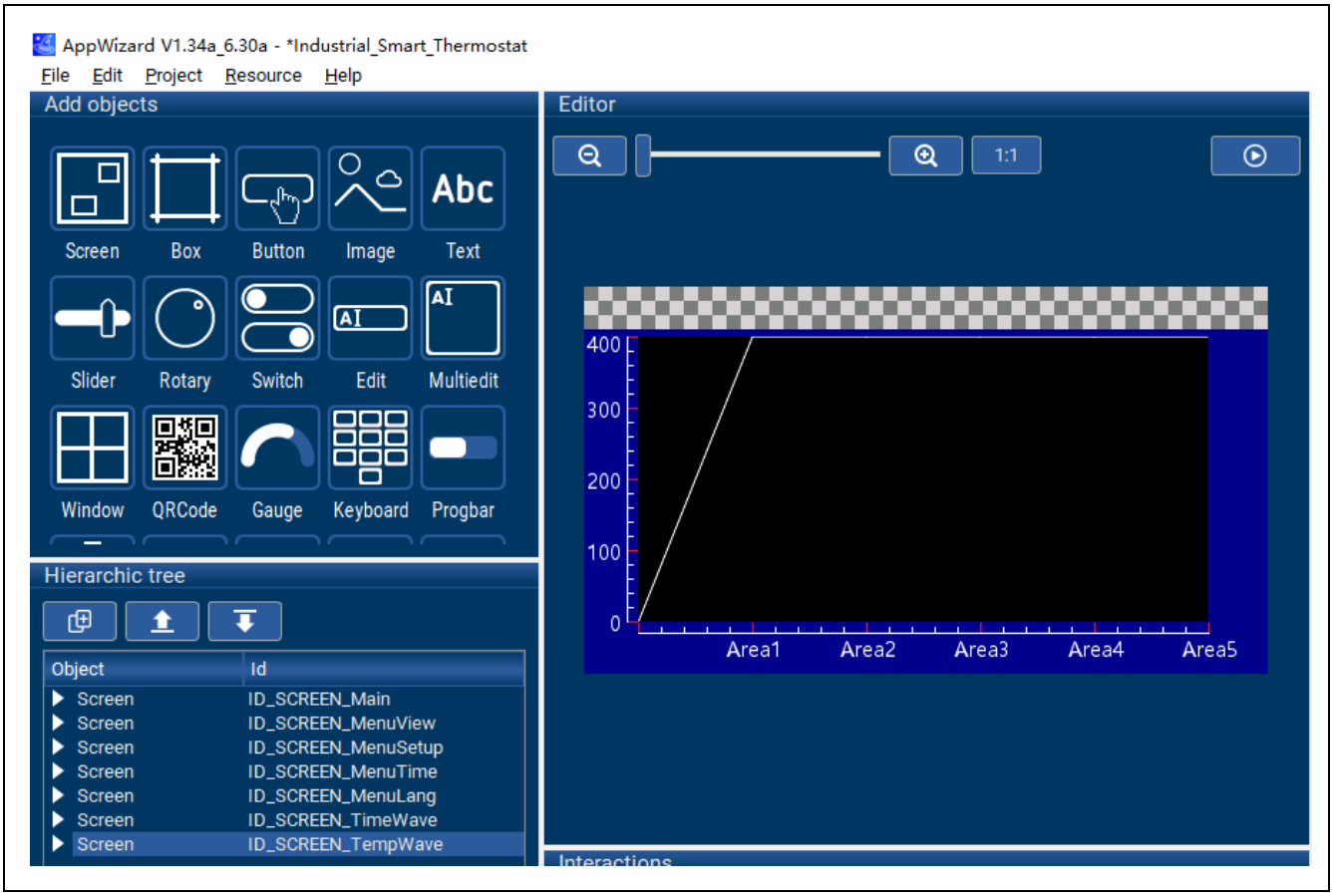


Figure 1-46 Area-Temperature Waveform Screen

1.2.2.12 Create Temperature List Screen

1. Add Screen

Click “Screen” in “Add Objects” to add a screen, rename it to ID_SCREEN_TempList in “Hierarchic tree”, and set the coordinates to (0, 30, 480, 242) in “Properties”.

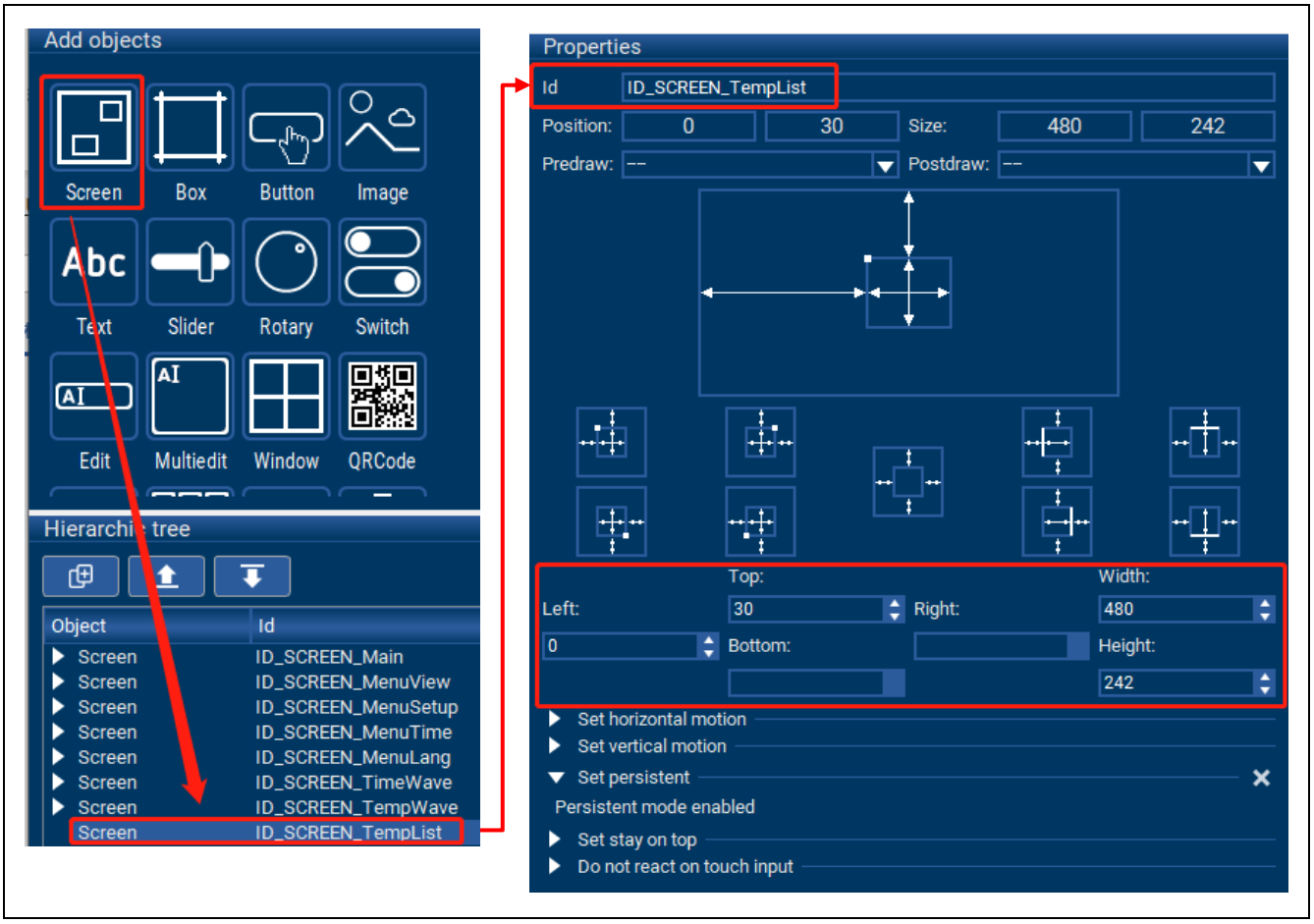


Figure 1-47 Create Temperature List Screen

2. Add Window

Select ID_SCREEN_TempList in “Hierarchic tree”. Click “Window” in “Add Objects” to add a window to the screen, rename it to ID_WINDOW_TempList in “Hierarchic tree”, and set the coordinates to (0, 0, 480, 242) in “Properties”.

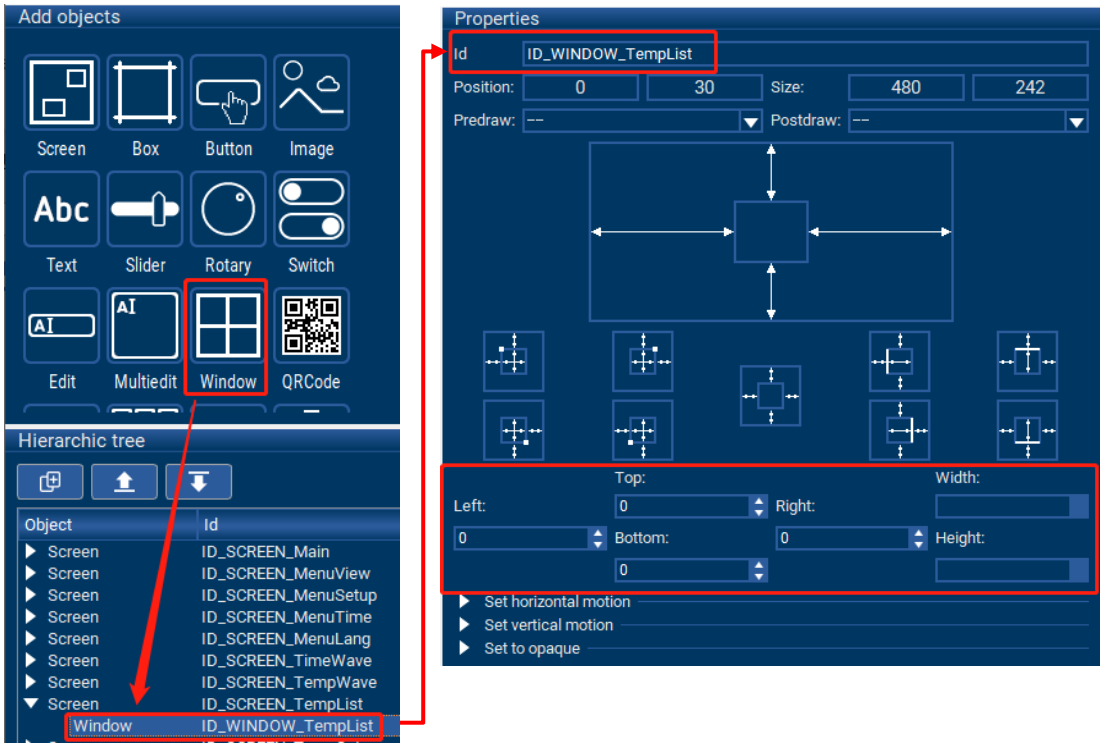


Figure 1-48 Add a Window - Area Temperature List Display

3. Add Image

As shown in Figure 1-49, click “Image” in “Add Objects”, add an Image to the window, and rename it to ID_IMAGE_BG in “Hierarchic tree”. In “Properties”, set the coordinates to (0, 0, 400, 242) and select BKG_TemperatureList_480x242.png in “bitmap” as background image. The actual setting items are shown in Figure 1-50.

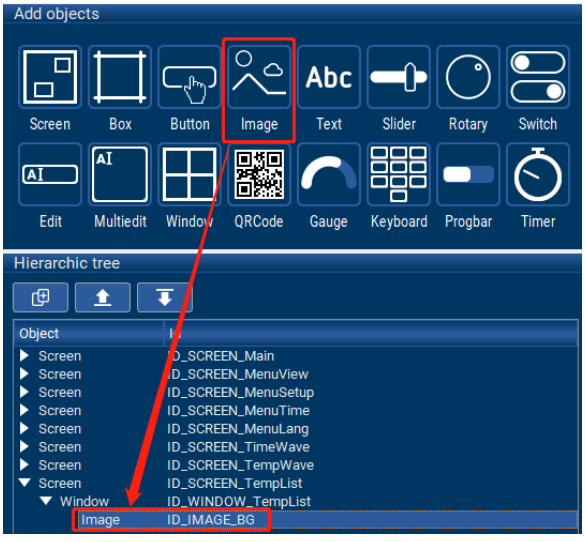


Figure 1-49 Add an Image - Background Display

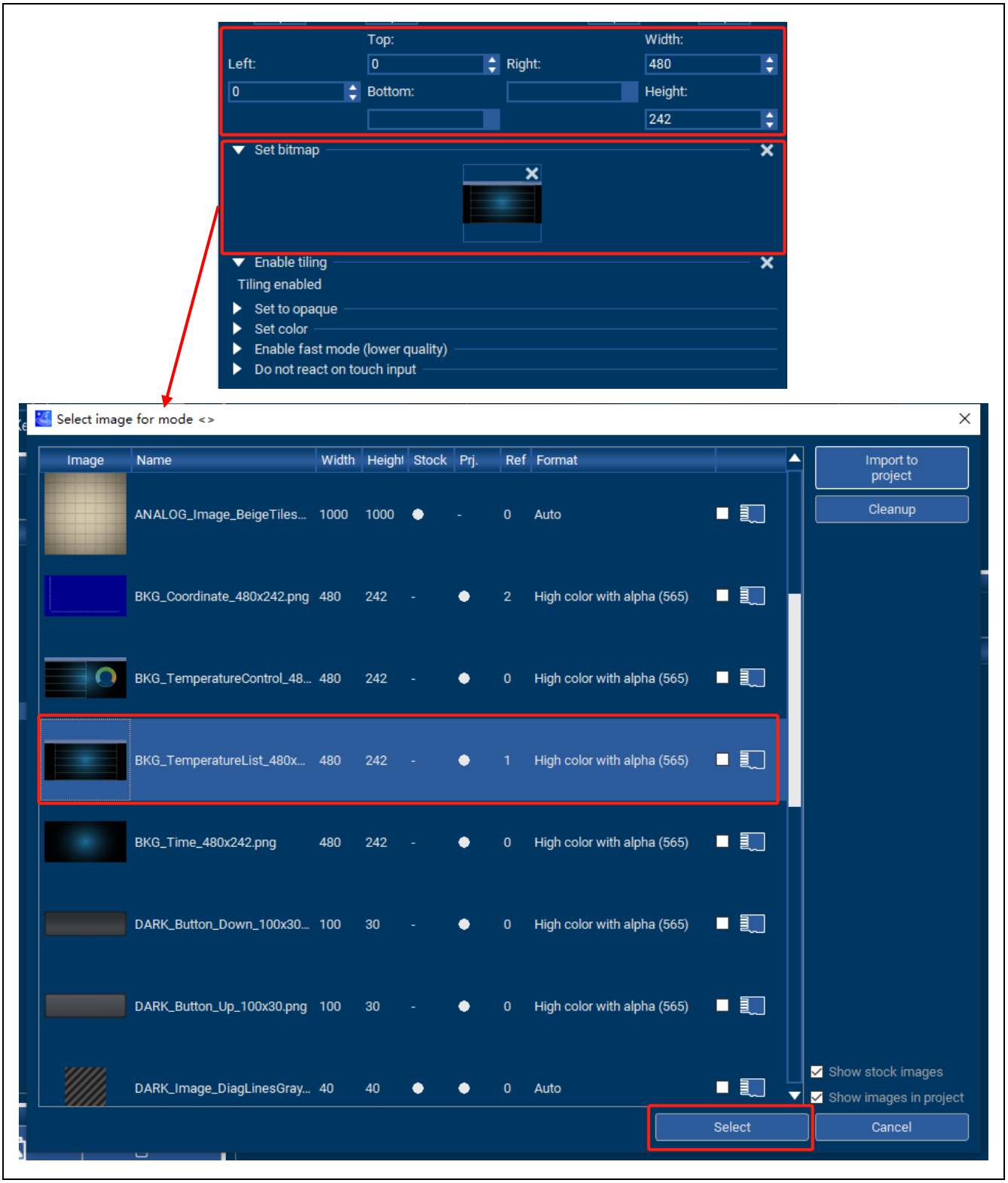


Figure 1-50 Set Image Properties

4. Add Text

Add multiple Texts to the window to display the names and temperature of all areas.

1) Add Text

Click “Text” in “Add Objects”, add 28 Texts, and rename them in “Hierarchic tree” as Figure 1-51.

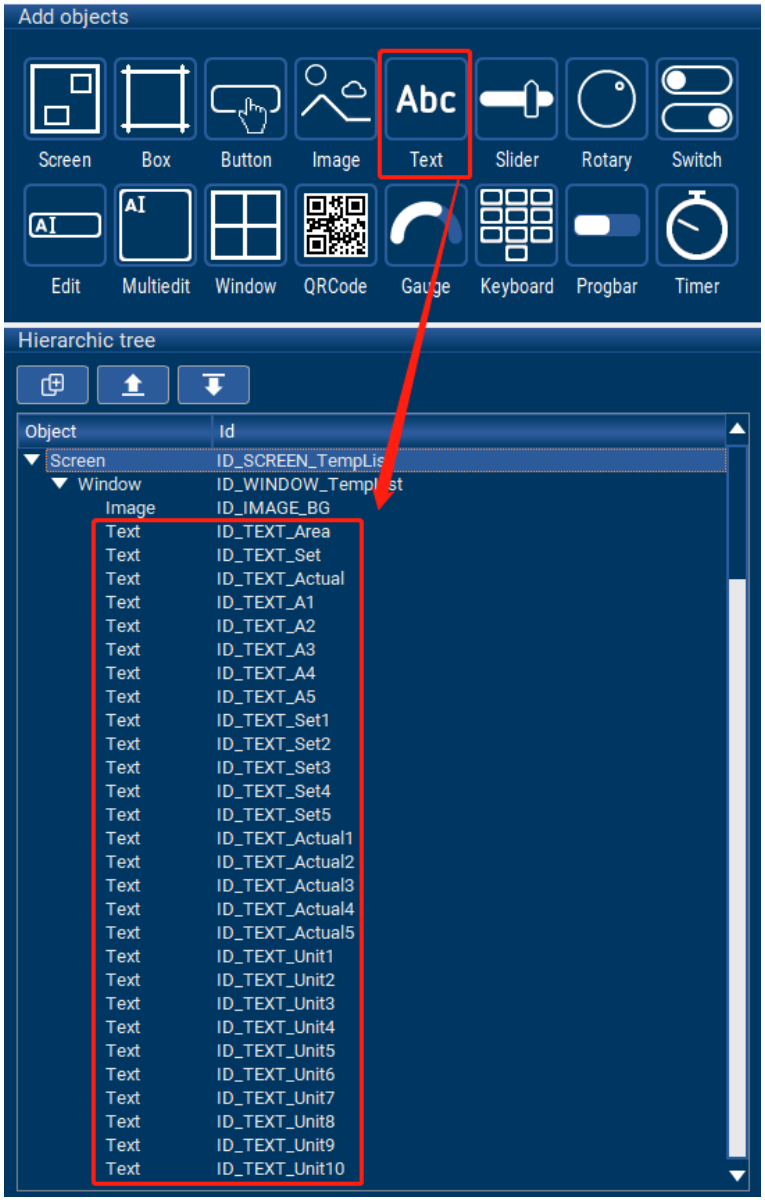


Figure 1-51 Add Texts - Area Set and Actual Temperature Display

2) Set Properties of Text Display

Among them, 18 Texts are used to display text. In “Properties”, set “text color” to RGB (192, 192, 192), “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4. Set the coordinates, “text”, and “text alignment” according to Table 1-20. The actual setting items are shown in Figure 1-52.

ID	**Coordinates (left, top, width, height)**	**text**	**text alignment (offset x, offset y, Alignment)**
ID_TEXT_Area	(88, 0, 80, 24)	ID_RTEXT_AREA	(0, 0, Left)
ID_TEXT_Set	(214, 0, 80, 24)	ID_RTEXT_SET_TEMP	(0, 0, Left)
ID_TEXT_Actual	(340, 0, 80, 24)	ID_RTEXT_ACTUAL_TEMP	(0, 0, Left)
ID_TEXT_A1	(88, 34, 80, 24)	ID_RTEXT_AREA1	(0, 0, Left)
ID_TEXT_A2	(88, 77, 80, 24)	ID_RTEXT_AREA2	(0, 0, Left)
ID_TEXT_A3	(88, 120, 80, 24)	ID_RTEXT_AREA3	(0, 0, Left)
ID_TEXT_A4	(88, 163, 80, 24)	ID_RTEXT_AREA4	(0, 0, Left)
ID_TEXT_A5	(88, 206, 80, 24)	ID_RTEXT_AREA5	(0, 0, Left)
ID_TEXT_Unit1	(246, 34, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit2	(246, 77, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit3	(246, 120, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit4	(246, 163, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit5	(246, 206, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit6	(372, 34, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit7	(372, 77, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit8	(372, 120, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit9	(372, 163, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit10	(372, 206, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)

Table 1-20 Description of Text Properties Setting - Text Display

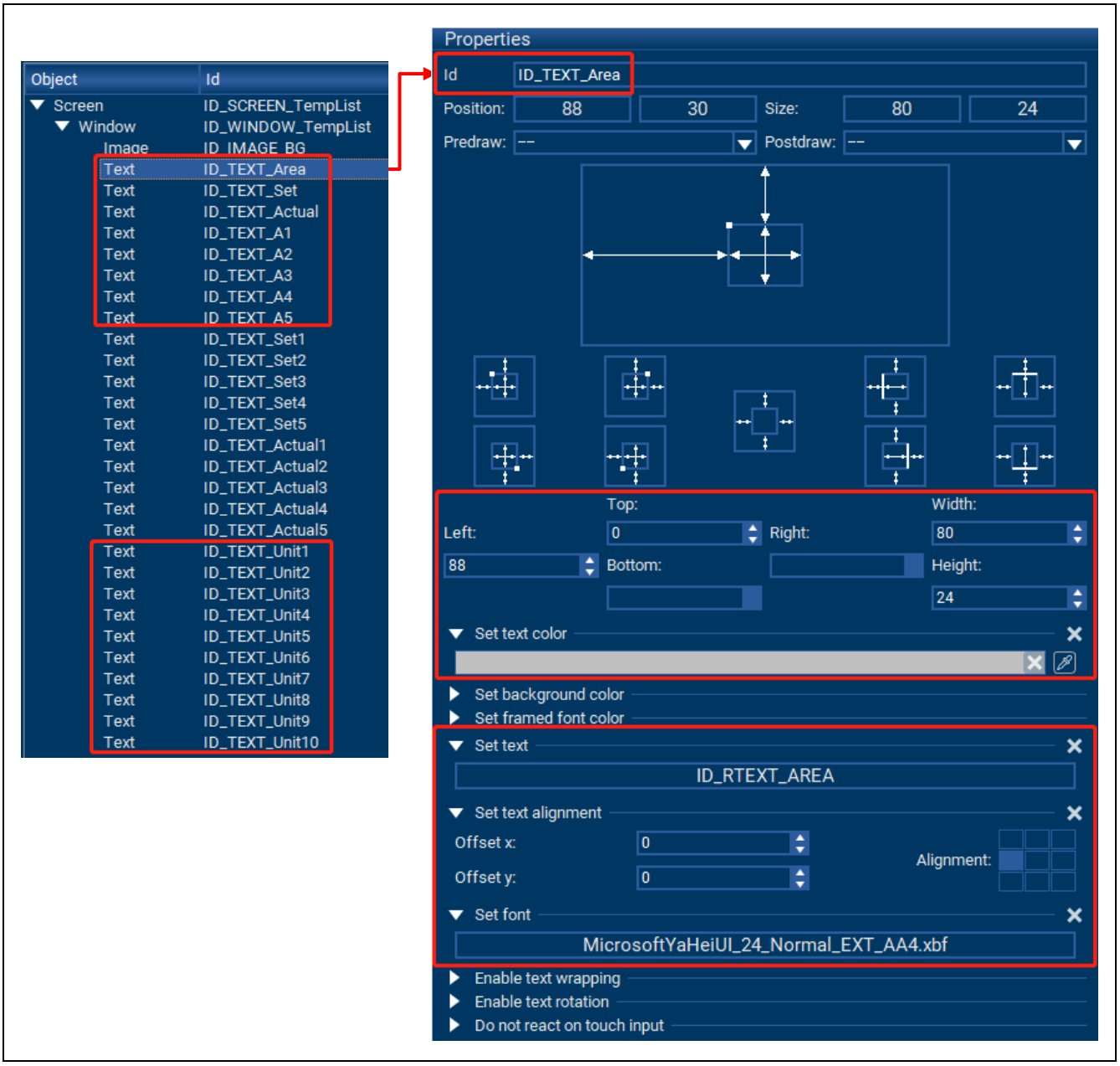


Figure 1-52 Set Text Properties - Text Display

3) Set Properties of Digital Display

Among them, 10 Texts are used to display numbers. In “Properties”, set “text color” to RGB (192, 192, 192), “decimal mode”, “text alignment” to “offset x” = “offset y” = 0 and “alignment” to right, “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4, “range” to Min = 0 and Max = 400. The coordinates are set according to Table 1-21, and the actual setting items are shown in Figure 1-53.

ID	Coordinates (left, top, width, height)
ID_TEXT_Set1	(214, 34, 32, 24)
ID_TEXT_Set2	(214, 77, 32, 24)
ID_TEXT_Set3	(214, 120, 32, 24)
ID_TEXT_Set4	(214, 163, 32, 24)
ID_TEXT_Set5	(214, 206, 32, 24)
ID_TEXT_Act1	(340, 34, 32, 24)
ID_TEXT_ Act2	(340, 77, 32, 24)
ID_TEXT_ Act3	(340, 120, 32, 24)
ID_TEXT_ Act4	(340, 163, 32, 24)
ID_TEXT_ Act5	(340, 206, 32, 24)

Table 1-21 Description of Text Properties Setting - Temperature Display

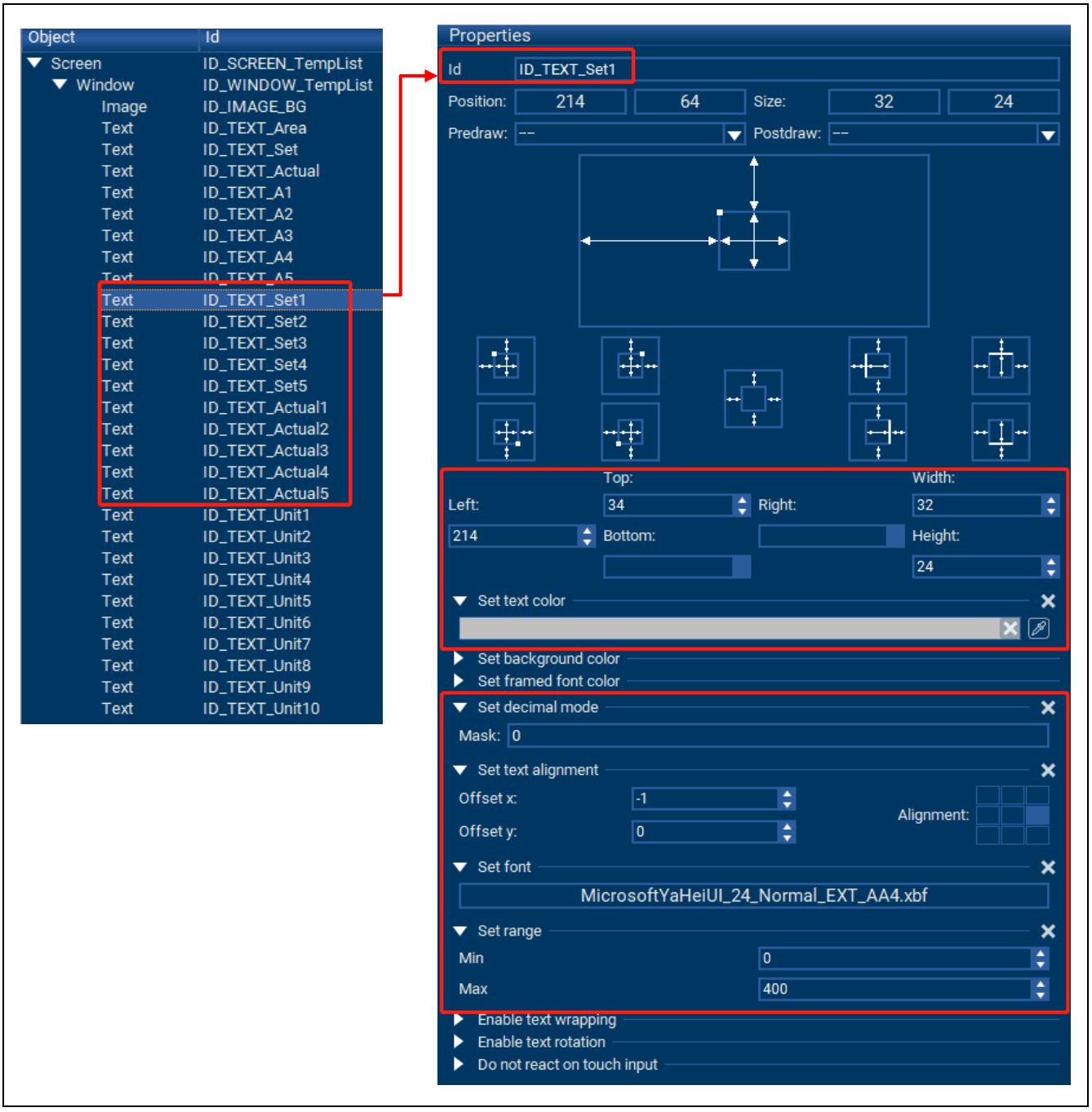


Figure 1-53 Add Text - Set and Actual Temperature Display

5. Temperature List Screen After Completion of Design

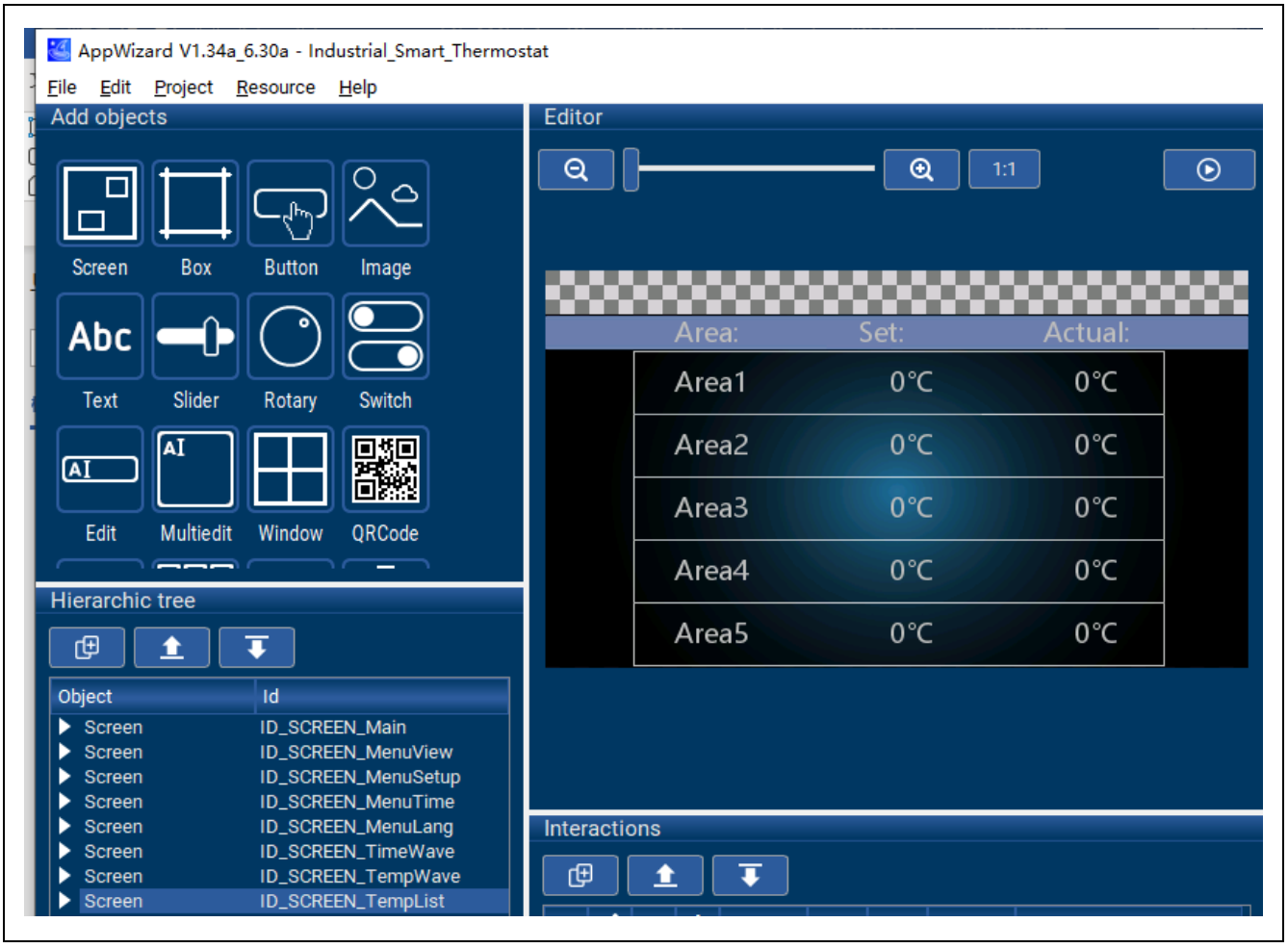


Figure 1-54 Temperature List Screen

1.2.2.13 Create Temperature Setting Screen

1. Add Screen

Click “Screen” in “Add Object” to add a screen, rename it to ID_SCREEN_TempSetup in “Hierarchic tree”, and set the coordinates to (0, 30, 480, 242) in “Properties”.

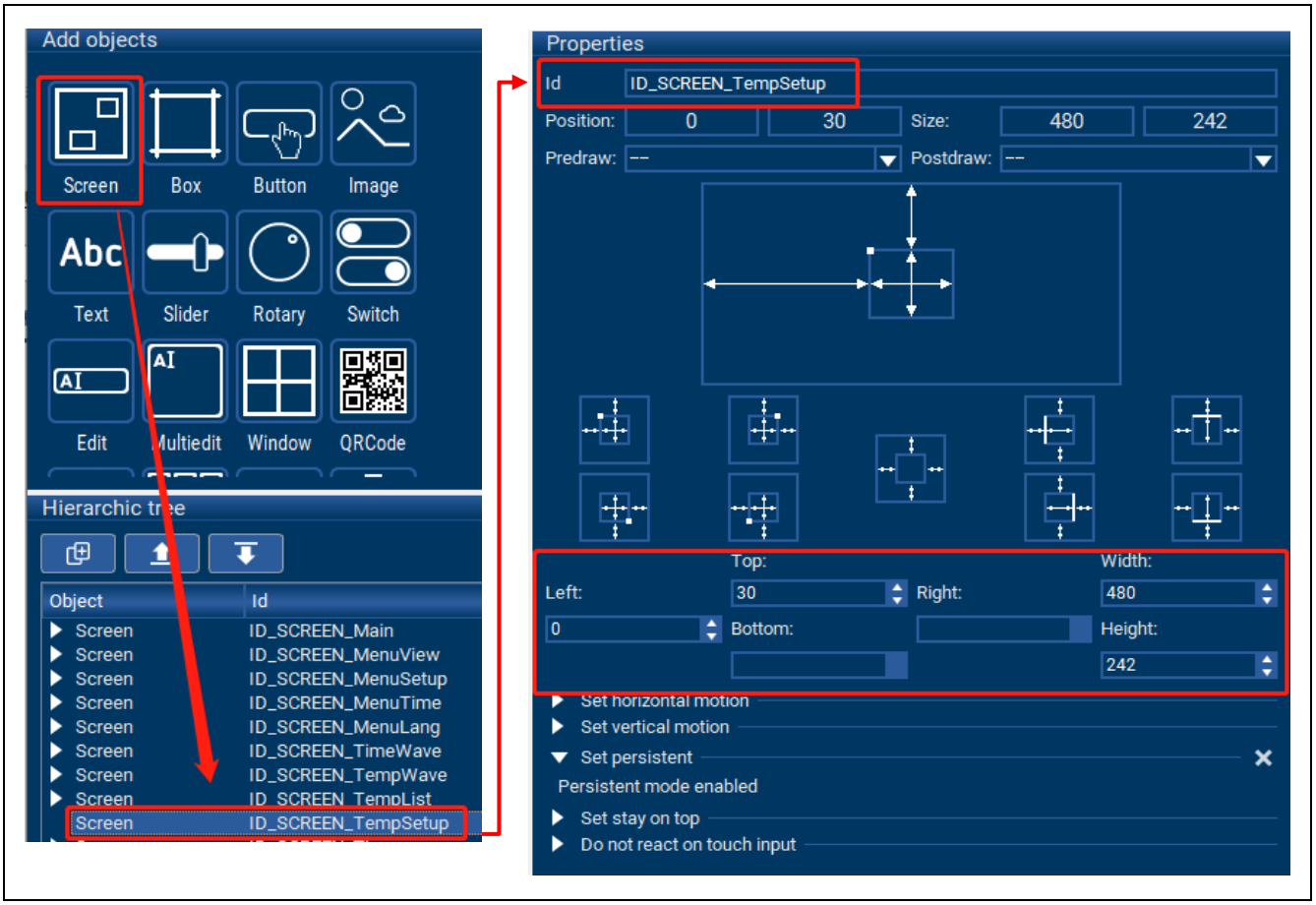


Figure 1-55 Create Temperature Setting Screen

2. Add Window

Click “Window” in “Add Object” to add a window to the screen, rename it to ID_WINDOW_TempSetup in “Hierarchic tree”, and set the coordinates to (0, 0, 480, 242) in “Properties”.

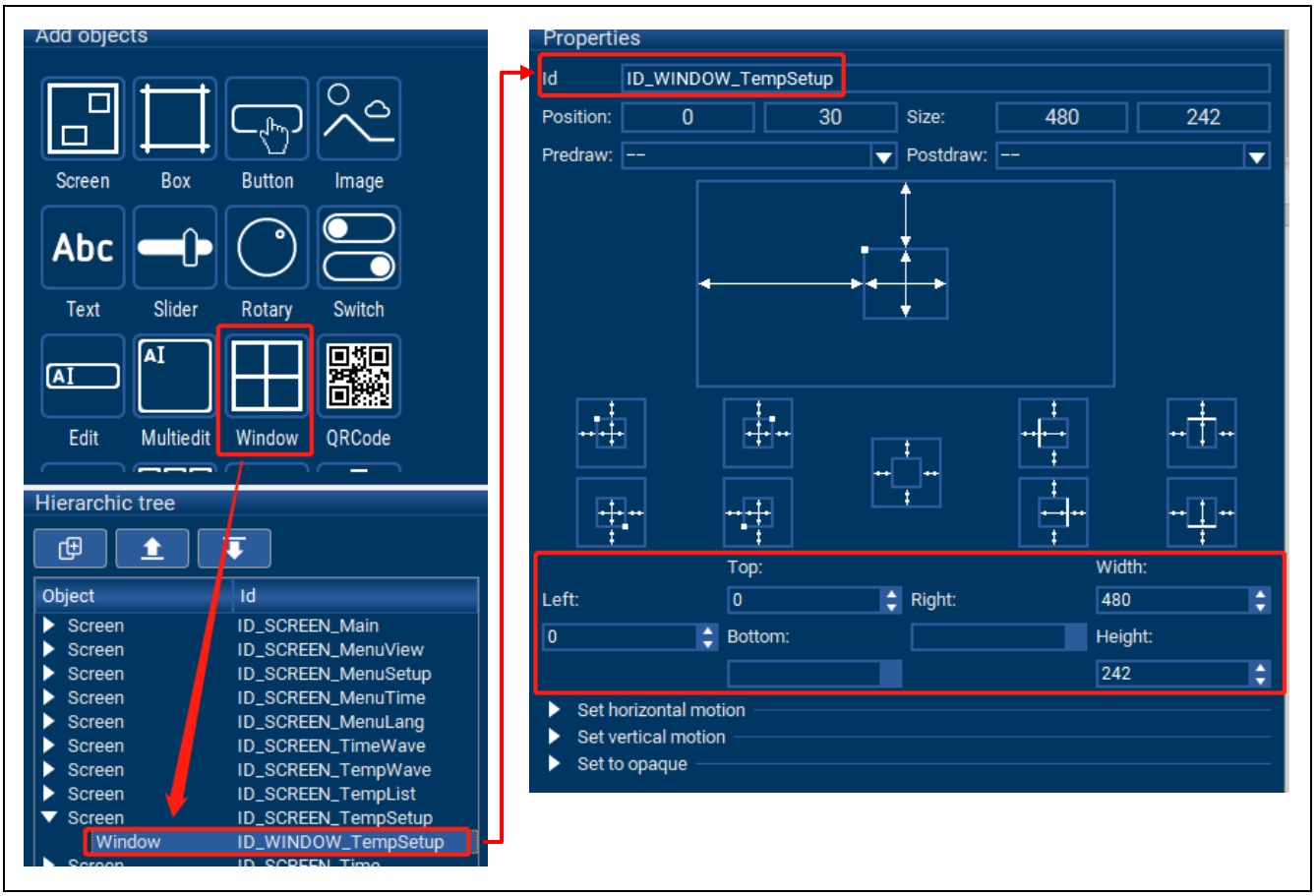


Figure 1-56 Add a Window - Area Temperature List Display

3. Add Image

As shown in Figure 1-57 and Figure 1-58, click “Image” in “Add Object” to add an Image to the window, and rename it to ID_IMAGE_BG in “Hierarchic tree”. In “Properties”, set the coordinates to (0, 0, 400, 242), and select BKG_TemperatureControl_480x242.png in “bitmap” as background image.

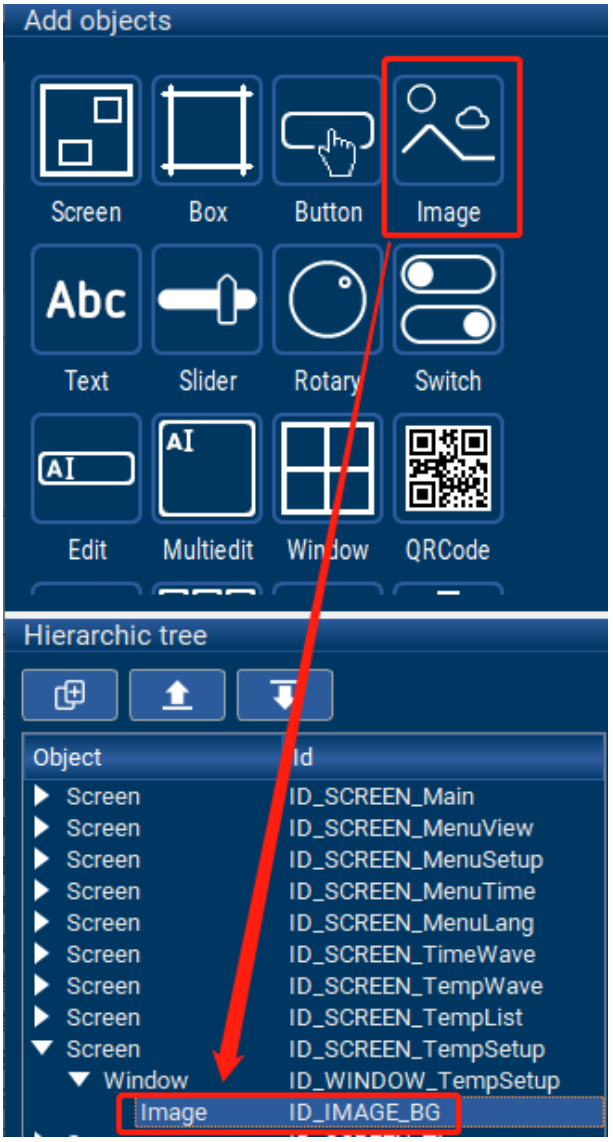


Figure 1-57 Add an Image - Background Display

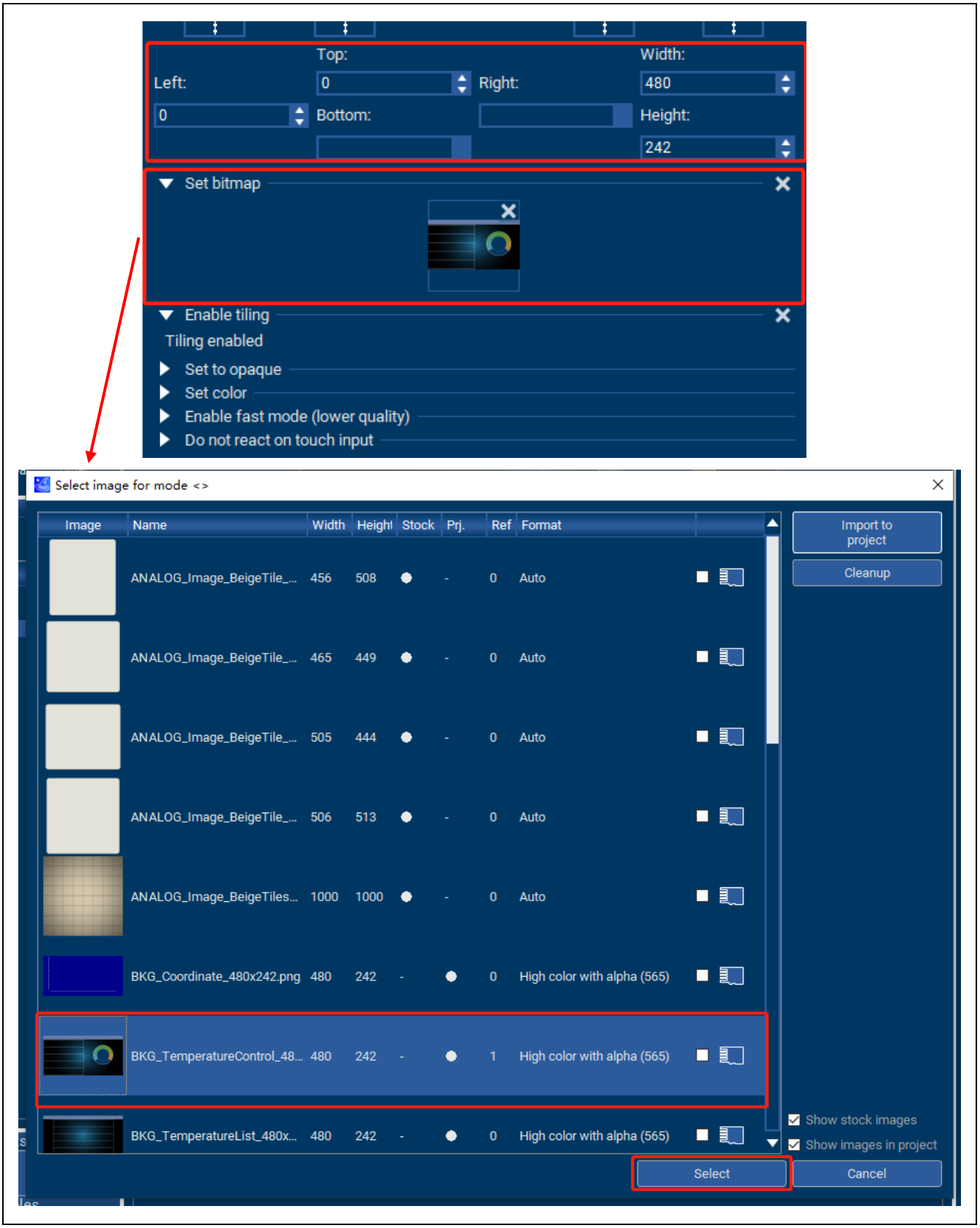


Figure 1-58 Set Image Properties

4. Add Highlighting of Selected Area

Add a Box to the window to highlight the selected area.

Click “Box” in “Add Object” and rename it to ID_BOX_IndexBar in “Hierarchic tree”. In “Properties”, set the coordinates to (3, 25, 241, 41) and “color” to RGB (51, 94, 128).

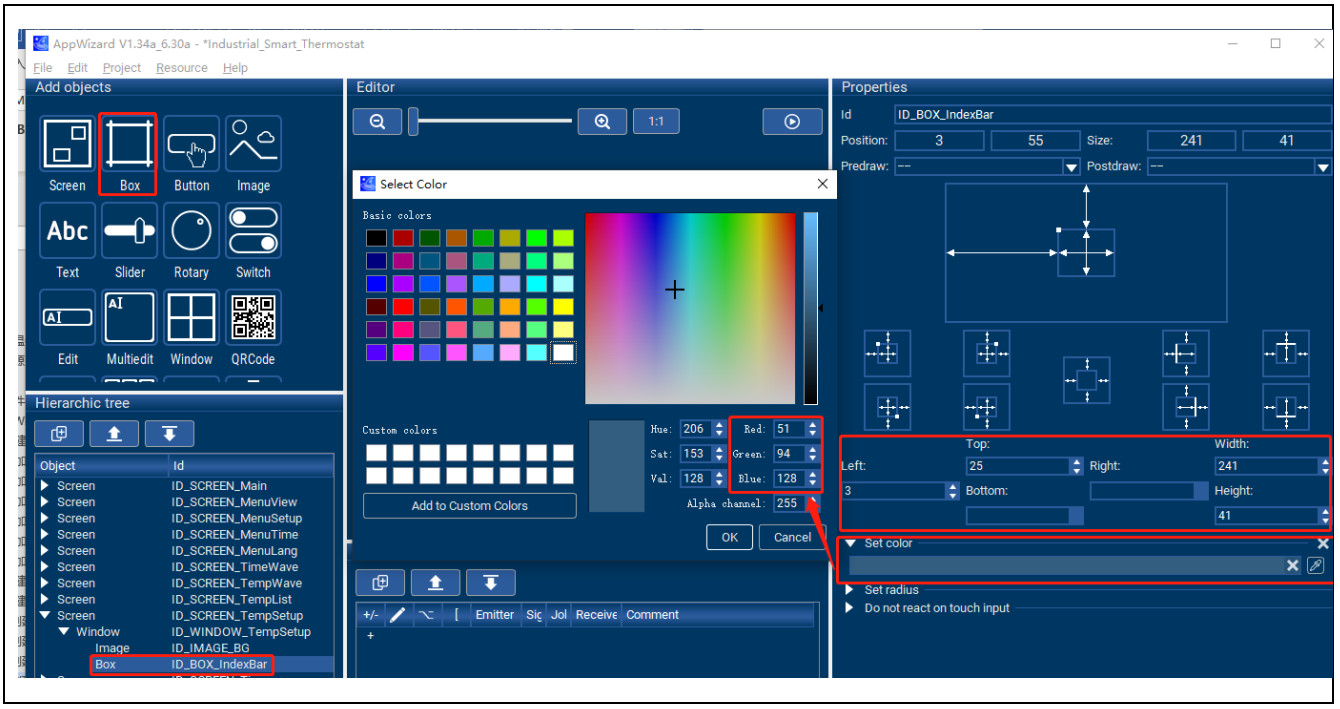


Figure 1-59 Add Box - Selected Area Highlighting

5. Add Display of Area Set Temperature and Area Actual Temperature

Add multiple Texts to the window to display the names and temperatures of all areas.

1) Add Text

Select ID_WINDOW_TempSetup in “Hierarchic tree”, click “Text” in “Add Object” to add 28 Texts, and rename them in “Hierarchic tree” according to Figure 1-60.

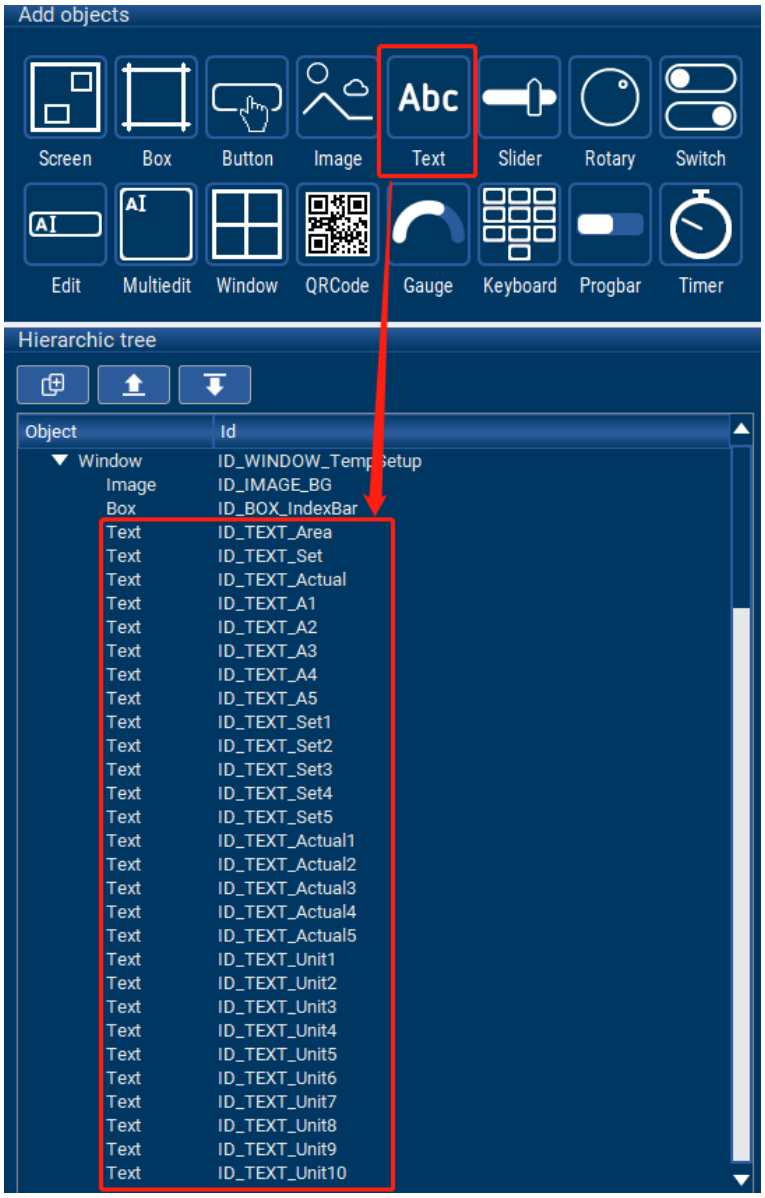


Figure 1-60 Add Texts - Area Set and Actual Temperature Display

2) Set Properties of Text Display

Among them, 18 Texts are used to display text. In “Properties”, set “text color” to RGB (192, 192, 192), “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4. Set the coordinates, “text”, and “text alignment” according to Table 1-22. The actual setting items are shown in Figure 1-61.

ID	Coordinates (left, top, width, height)	text	text alignment (offset x, offset y, Alignment)
ID_TEXT_Area	(18, 0, 52, 24)	ID_RTEXT_AREA	(0, 0, Left)
ID_TEXT_Set	(94, 0, 52, 24)	ID_RTEXT_SET_TEMP	(0, 0, Left)
ID_TEXT_Actual	(170, 0, 52, 24)	ID_RTEXT_ACTUAL_TEMP	(0, 0, Left)
ID_TEXT_A1	(18, 34, 70, 24)	ID_RTEXT_AREA1	(0, 0, Left)
ID_TEXT_A2	(18, 77, 70, 24)	ID_RTEXT_AREA2	(0, 0, Left)
ID_TEXT_A3	(18, 120, 70, 24)	ID_RTEXT_AREA3	(0, 0, Left)
ID_TEXT_A4	(18, 163, 70, 24)	ID_RTEXT_AREA4	(0, 0, Left)
ID_TEXT_A5	(18, 206, 70, 24)	ID_RTEXT_AREA5	(0, 0, Left)
ID_TEXT_Unit1	(126, 34, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit2	(126, 77, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit3	(126, 120, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit4	(126, 163, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit5	(126, 206, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit6	(126, 34, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit7	(126, 77, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit8	(126, 120, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit9	(126, 163, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)
ID_TEXT_Unit10	(126, 206, 20, 24)	ID_RTEXT_DEG	(0, 0, Left)

Table 1-22 Description of Text Properties Setting - Text Display

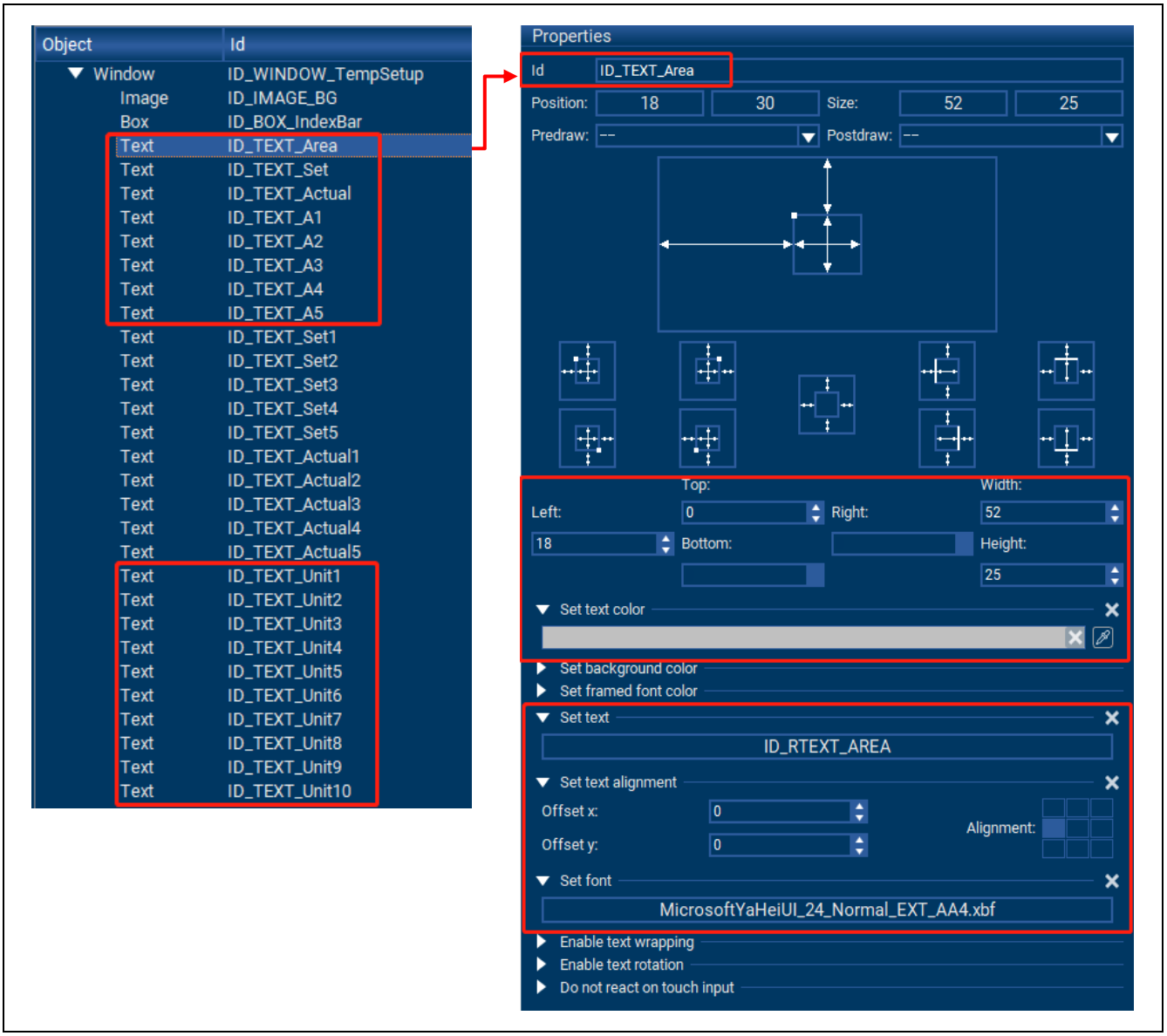


Figure 1-61 Set Text Properties -Text Display

3) Set Properties of Digital Display

Among them, 10 Texts are used to display numbers. In “Properties”, set “text color” to RGB (192, 192, 192), “decimal mode” to no Mask, “text alignment” to “offset x” = “offset y” = 0 and “Alignment” to right, “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4, and “range” to “Min” = 0, “Max” = 400. Set the coordinates according to Table 1-23. The actual setting items are shown in Figure 1-62.

ID	Coordinates (left, top, width, height)
ID_TEXT_Set1	(94, 34, 32, 24)
ID_TEXT_Set2	(94, 77, 32, 24)
ID_TEXT_Set3	(94, 120, 32, 24)
ID_TEXT_Set4	(94, 163, 32, 24)
ID_TEXT_Set5	(94, 206, 32, 24)
ID_TEXT_Act1	(170, 34, 32, 24)
ID_TEXT_ Act2	(170, 77, 32, 24)
ID_TEXT_ Act3	(170, 120, 32, 24)
ID_TEXT_ Act4	(170, 163, 32, 24)
ID_TEXT_ Act5	(170, 206, 32, 24)

Table 1-23 Description of Text Properties Setting - Temperature Display

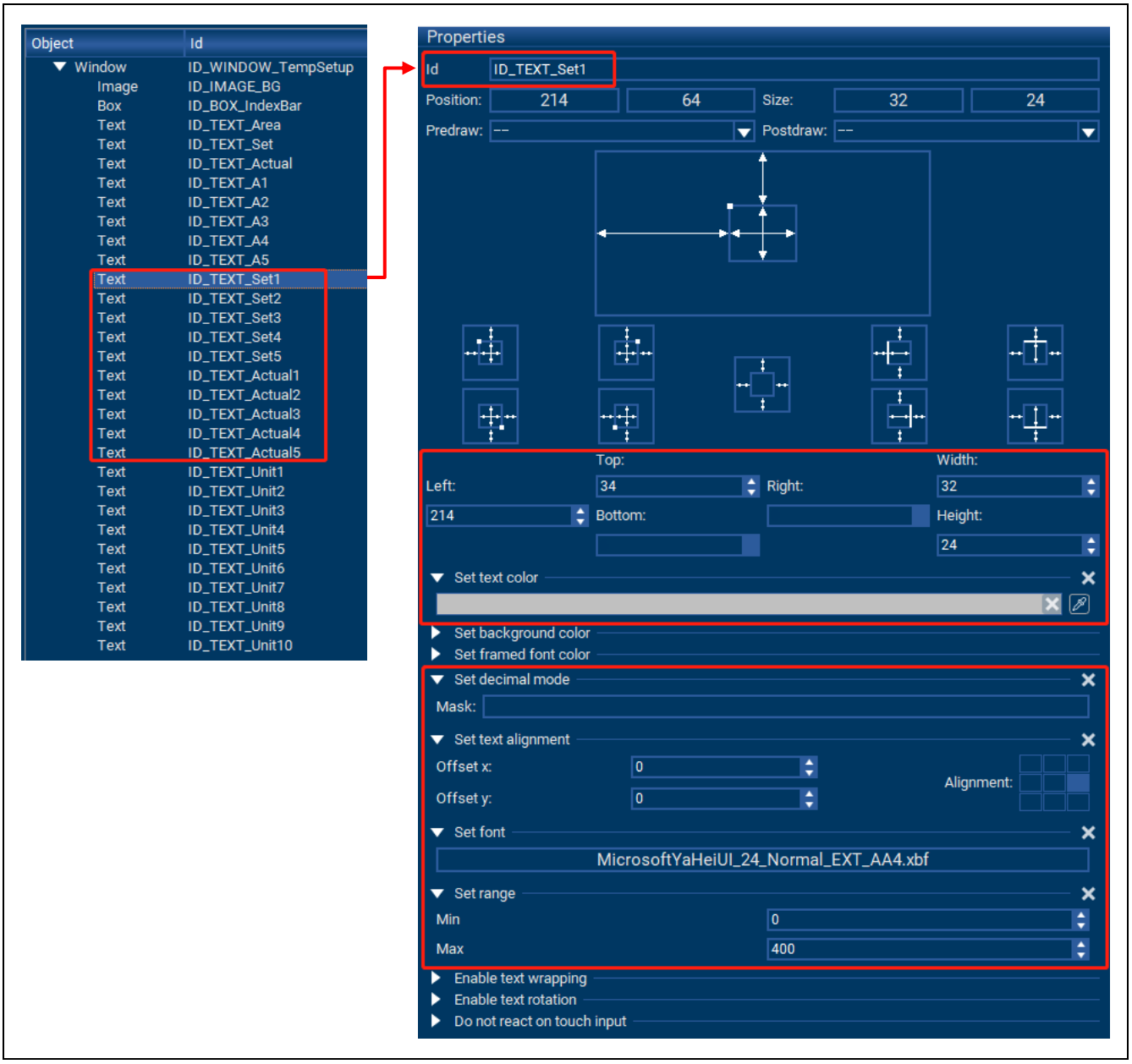


Figure 1-62 Add Texts - Temperature Display

6. Add Area Selection Button

Add five buttons to ID_WINDOW_TempSetup for area selection. The ID and coordinates of Button are shown in Table 1-24. In “bitmaps”, remove background image. The actual setting items are shown in Figure 1-63.

ID	Coordinates (left, top, width, height)
ID_BUTTON_A1	(3, 25, 240, 41)
ID_BUTTON_A2	(3, 68, 240, 41)
ID_BUTTON_A3	(3, 109, 240, 41)
ID_BUTTON_A4	(3, 152, 240, 41)
ID_BUTTON_A5	(3, 195, 240, 41)

Table 1-24 Description of Button Properties Setting - Area Selection

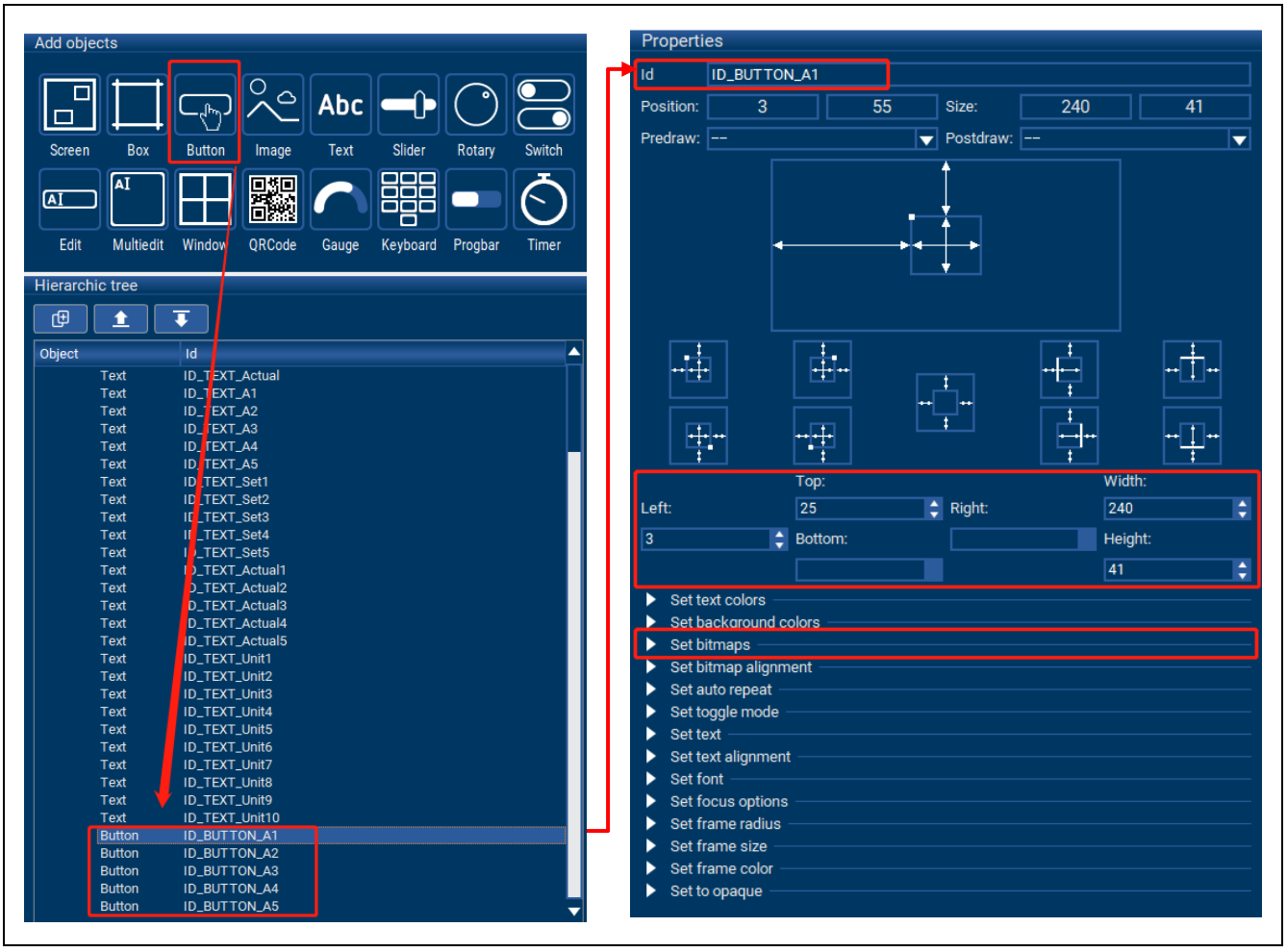


Figure 1-63 Add a Button - Area Selection

7. Add Rotary for Temperature Selection

1) Add Rotary

Select ID_WINDOW_TempSetup in “Hierarchic tree”, click “Rotary” in “Add objects” to add a Rotary to the window and rename it to ID_ROTARY_00 in “Hierarchic tree”. In “Properties”, set the coordinates to (307, 66, 105, 105), “Backgroud” to GUI_Knob_Icon_106x106.png and “Marker” to GUI_KnobMark_Icon_16x9.png, both of which are in “bitmaps.

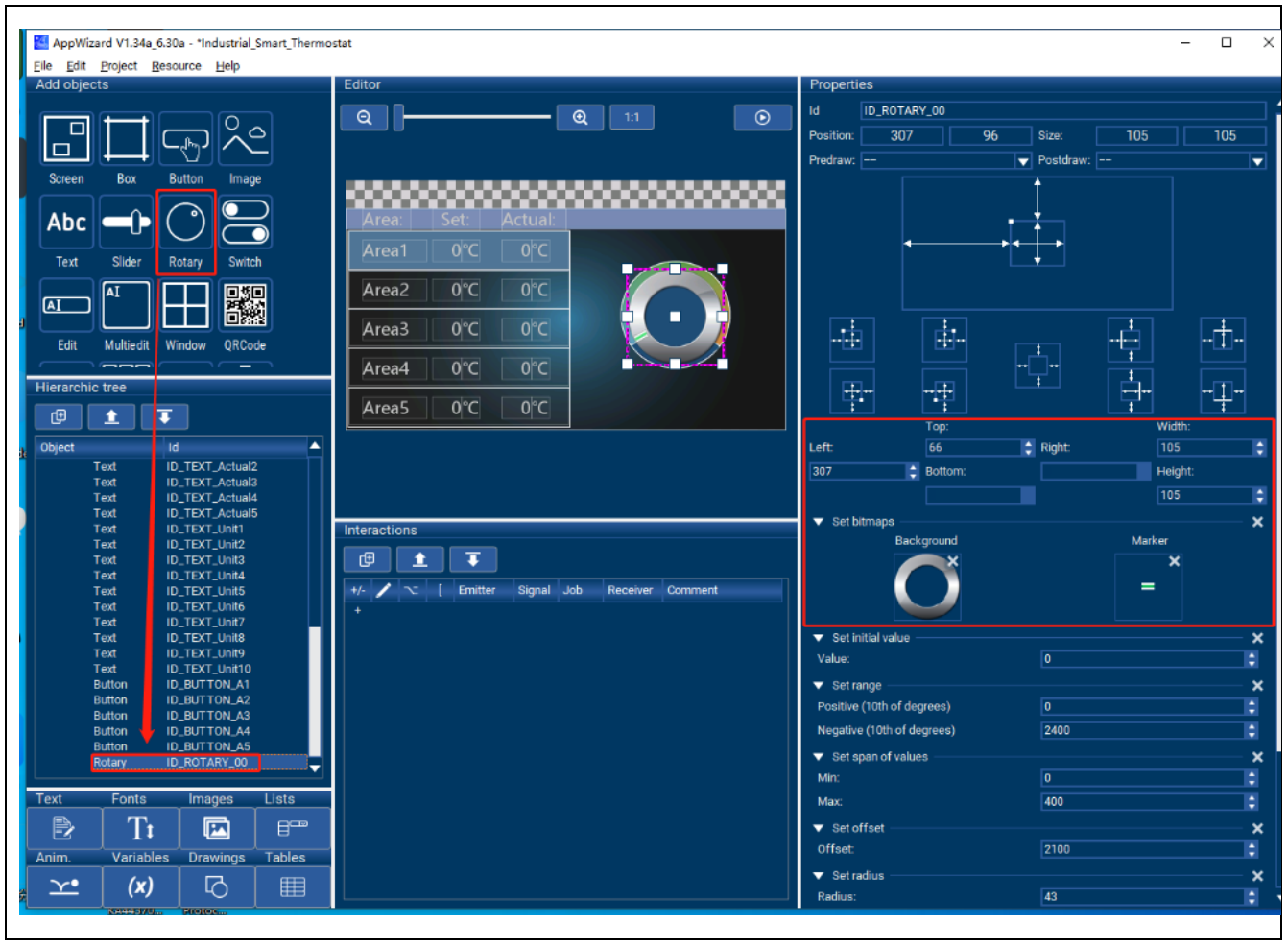


Figure 1-64 Add Rotary - Temperature Selection

2) Set Angle Parameters of Rotary

The system's default angle position of 0° is shown in Figure 1-65. To match the rotation angle and rotation value of the picture, as shown in Table 1-25 , set “initial value”, “range”, “span of value”, and “offset”. The actual setting items are shown in Figure 1-66.

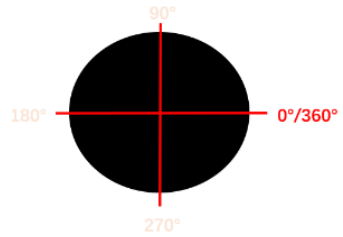


Figure 1-65 Rotary's initial angle

Properties	Value	Description
initial value	200	Default value
span of values	Min: 0	Set range of value
	Max: 400	
rang	Positive: 0	Set the adjustment range of the angle. unit 0.1°
	Positive: 2400	
offset	2100	Offset angle of starting point (relative to 0° position). Unit 0.1°

Table 1-25 Description of Rotary's Angle Setting

3) Set Parameters of Rotary's Marker

To align the rotation position of Marker, as shown in Table 1-26, set “marker alignment” and “radius”. The actual setting items are shown in Figure 1-66.

Properties	Value	Description
radius	43	Radius of Rotary
Rotate marker	Marker rotation enabled	The rotation angle of Marker is 90°.
Marker alignment	Offset x: 0. Offset y: 0. Alignment: cent	Marker position

Table 1-26 Description of Rotary's Marker Settings

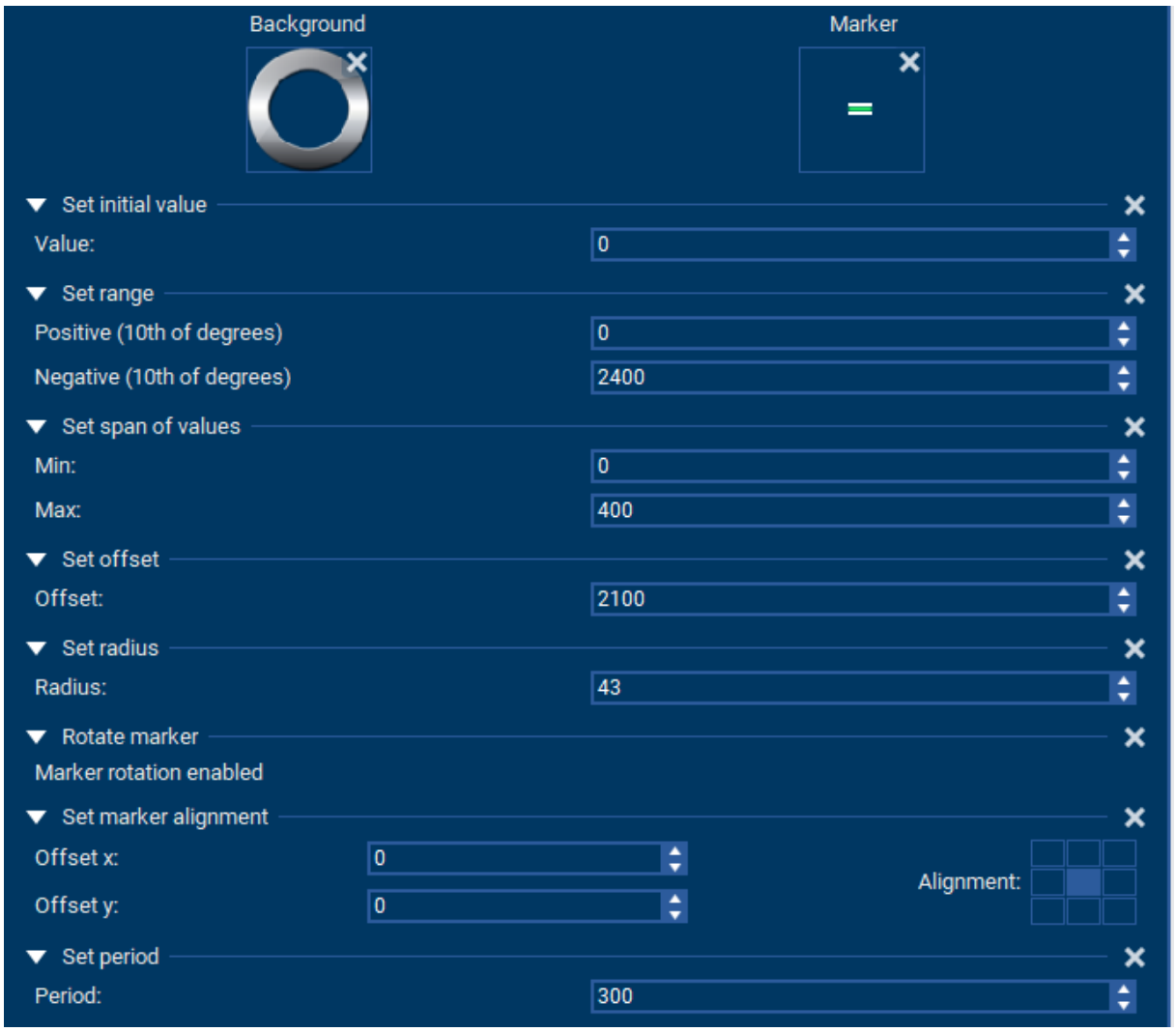


Figure 1-66 Set Rotary Properties

8. Add Text to Display Selected Temperature

Add 2 Texts to display value and unit of the selected temperature, respectively.

1) Add Text

As shown in Figure 1-67, select ID_WINDOW_TempSetup in “Hierarchic tree”, click “Text” in “Add Object” to add 2 Texts, and rename them to ID_TEXT_Unit and ID_TEXT_Temp respectively in “Hierarchic tree”.

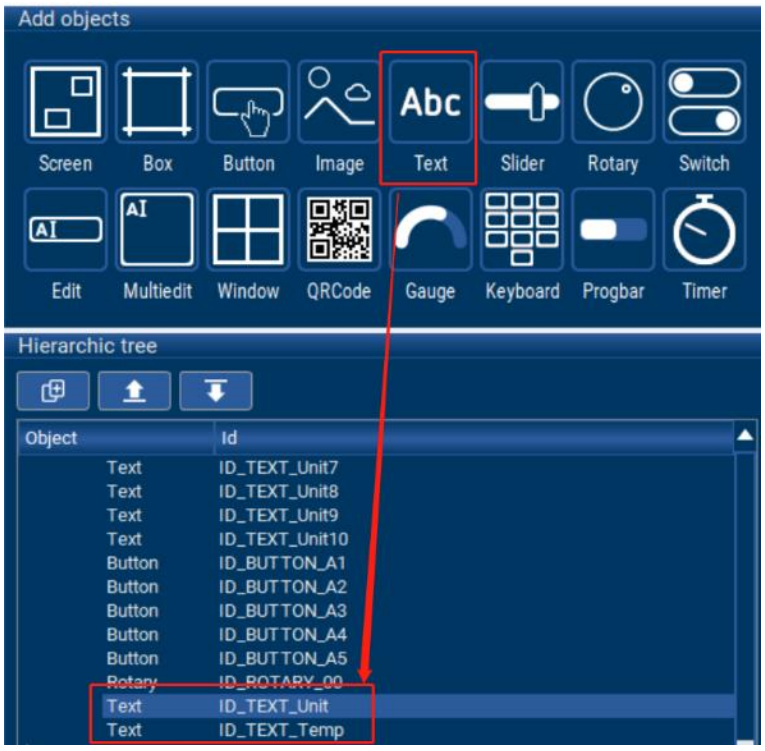


Figure 1-67 Add Text - Set Temperature Display

2) Set Text Properties

Set Properties of ID_TEXT_Unit and ID_TEXT_Temp in sequence according to Table 1-27. The actual setting items are shown in Figure 1-68.

Properties	**ID**		**Description**
	ID_TEXT_Unit	**ID_TEXT_Temp**	
coordinate	Left: 368 Top: 108 Width: 20 Height: 24	Left: 334 Top: 108 Width: 32 Height: 24	Set the coordinates of text
text color	RGB (192, 192, 192)	RGB (192, 192, 192)	Set text color
Decimal mode		Mask:	Enable Decimal mode
Text	ID_RTEXT_DEG		Set content when displaying text
text alignment	Offset x: 0. Offset y: 0. Alignment: left	Offset x: 0. Offset y: 0. Alignment: Right	Text position
font	MicrosoftYaHeiUI_24_Normal_EXT_AA4	MicrosoftYaHeiUI_24_Normal_EXT_AA4	Set font
rang		Min: 0 Max: 400	Set value range

Table 1-27 Description of Text Properties Setting - Temperature Display

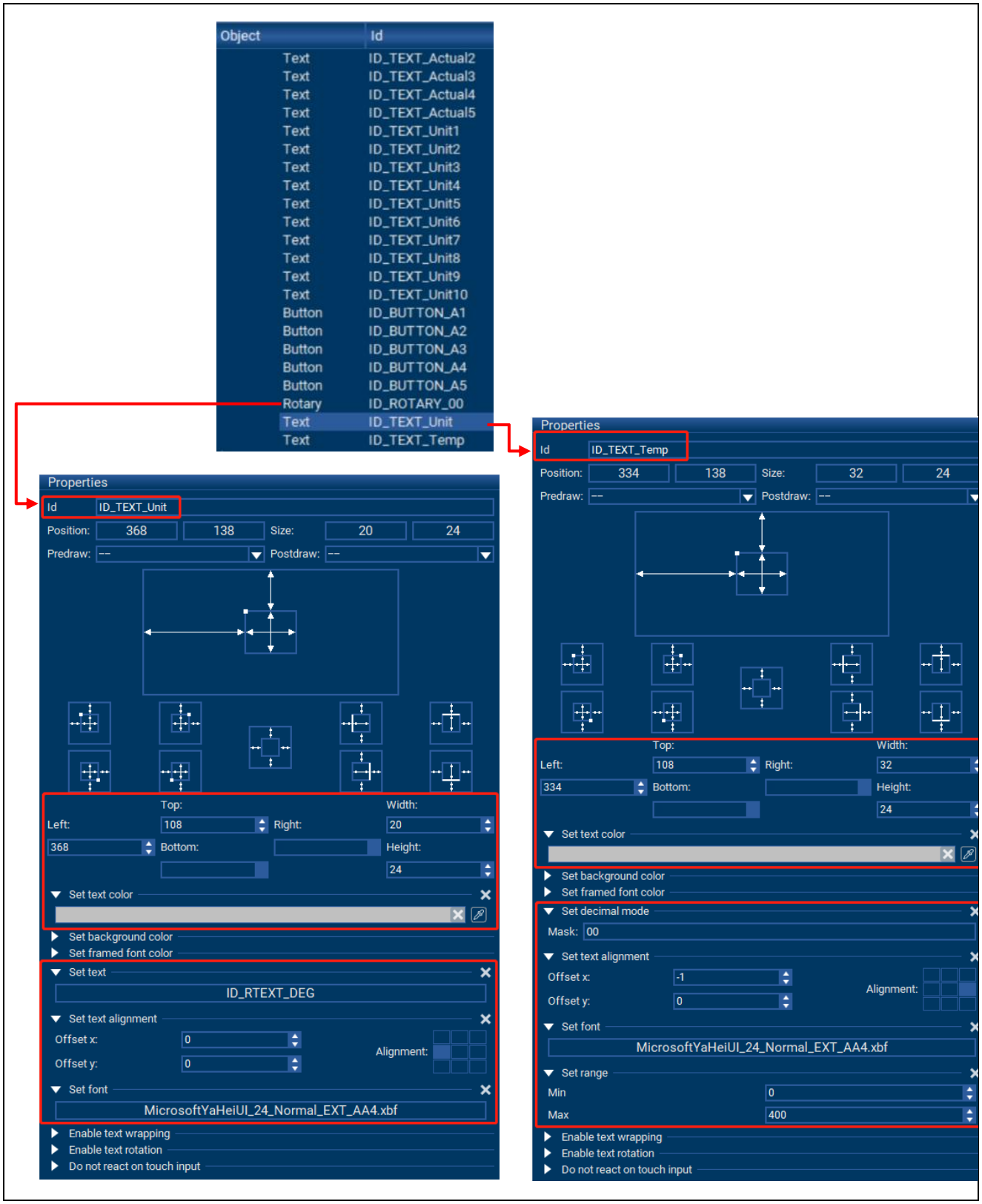


Figure 1-68 Set Text Properties - Temperature Display

9. Temperature Setting Screen After Completion of Design

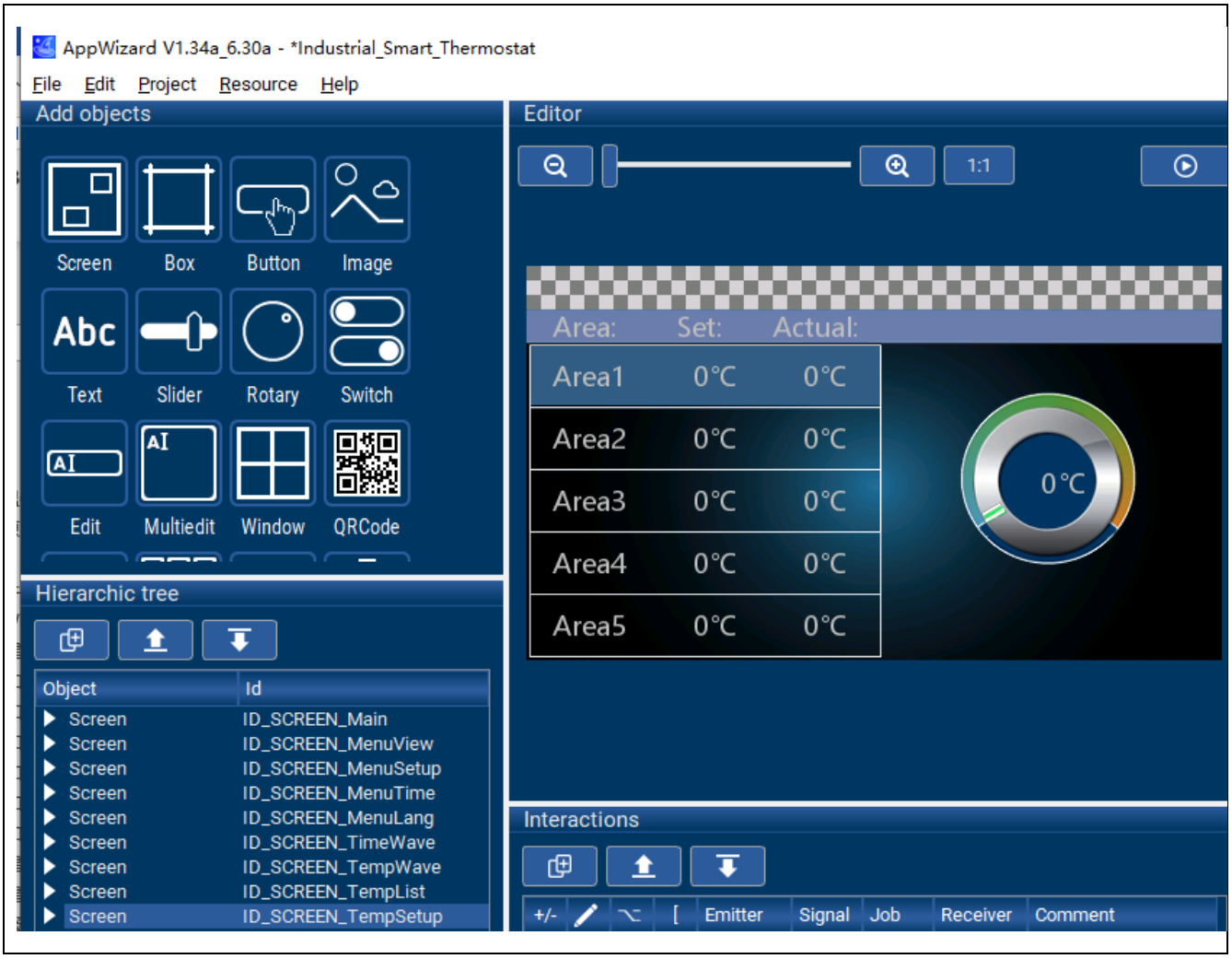


Figure 1-69 Temperature Settings Screen

1.2.2.14 Create Time Setting Screen

1. Add Screen

Click “Screen” in “Add Object” to add a screen, rename it to ID_SCREEN_Time in “Hierarchic tree”, and set the coordinates to (0, 30, 480, 242) in “Properties”.

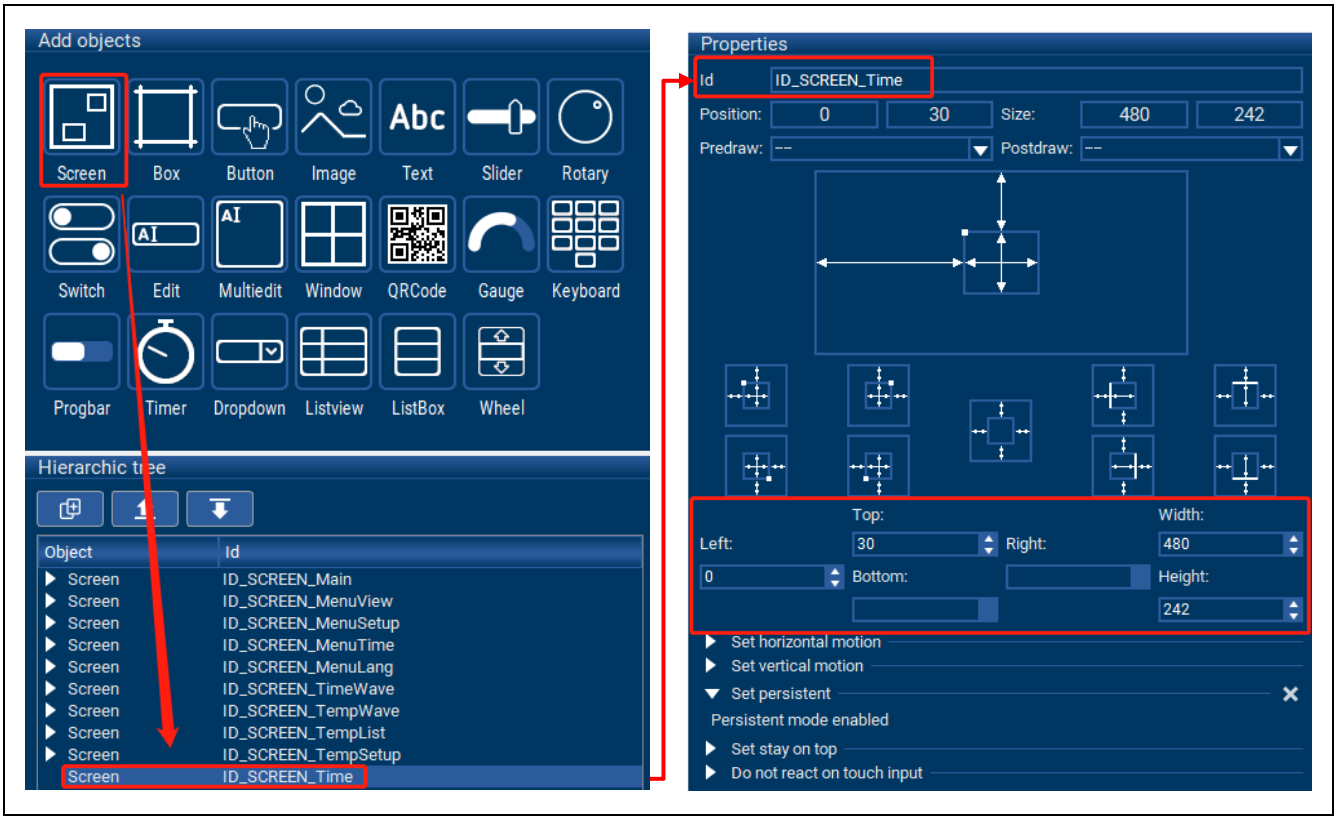


Figure 1-70 Create Time Setting Screen

2. Add Window

Click “Window” in “Add Object” to add a window to the screen, rename it to ID_WINDOW_Time in “Hierarchic tree”. Set the coordinates to (0, 0, 480, 242) in “Properties”.

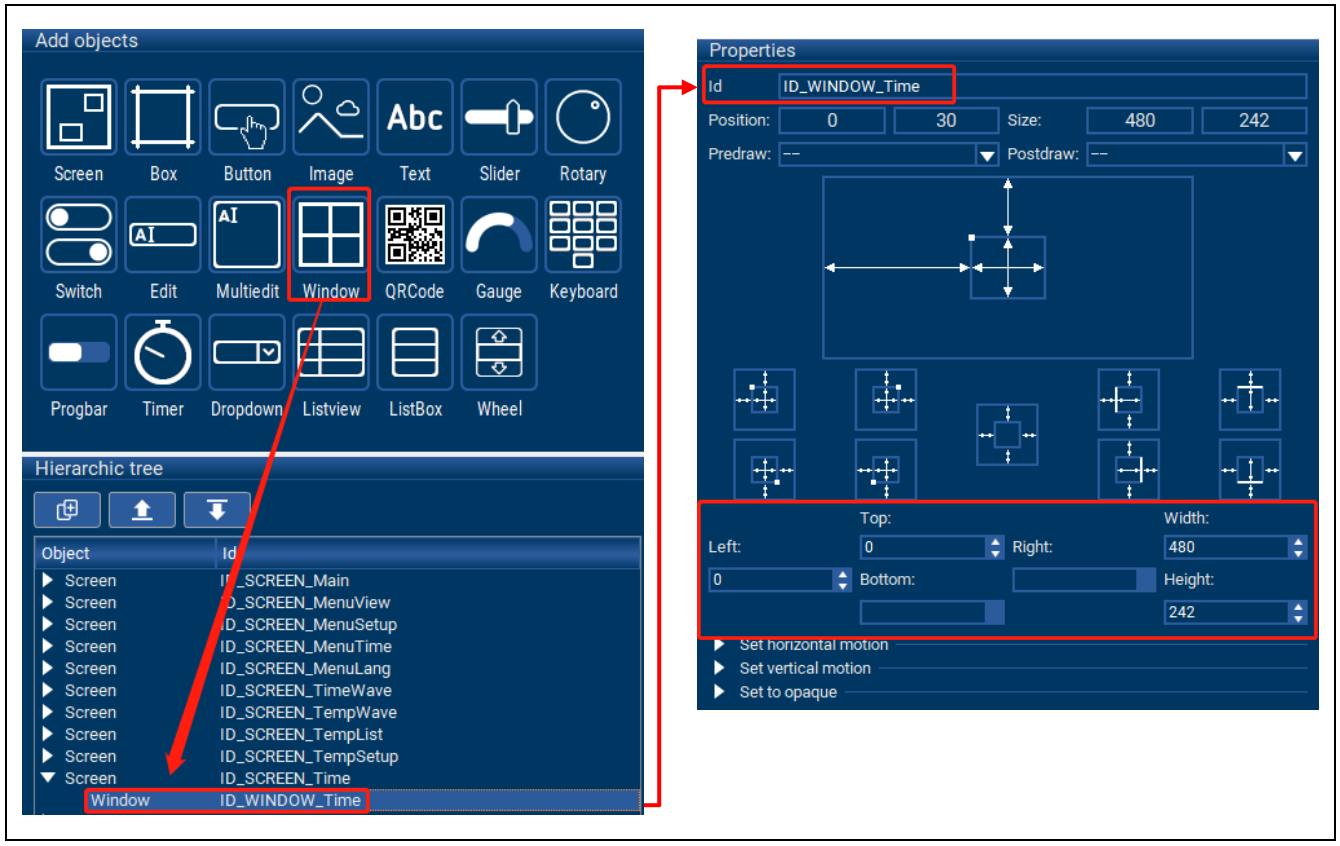


Figure 1-71 Add Window - Time Setting

3. Add Image

Add Images for background and screen function display. Click “Image” in “Add Object” to add 2 Images to the window. As shown in Table 1-28, rename in "Hierarchic tree" and set the coordinates and "bitmap" in "Properties". The actual operation is shown in Figure 1-72.

ID	coordinates (left, top, width, height)	bitmap	Description
ID_IMAGE_BG	(0, 0, 480, 242)	BKG_Time_480x242.png	Background Image
ID_IMAGE_Type	(19, 19, 37, 39)	GUI_Time_Icon_37x38.png	The function of the display screen is time setting.

Table 1-28 Description of Image Properties Setting

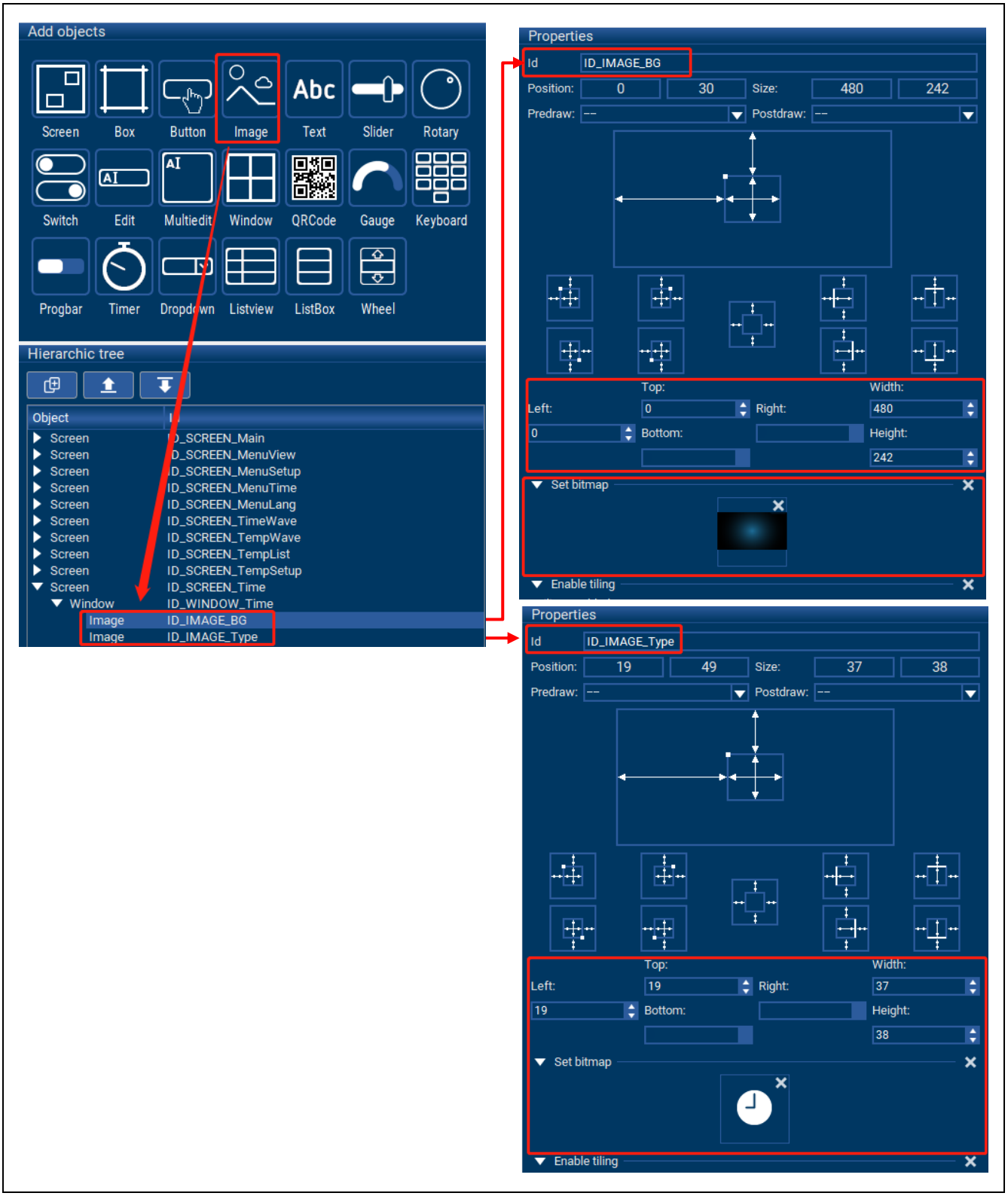


Figure 1-72 Add Image – Background and Type Display

4. Add Button

Buttons are used to save current settings and exit the current screen. Click “Button” in “Add Objects” to add 2 buttons. According to Table 1-29, rename them in “Hierarchic tree” and set the coordinates and “bitmap” in “Properties”. The actual setting items are shown in Figure 1-73.

Button	**Coordinates (Left, Top, Width, Height)**	**bitmap**
ID_BUTTON_Back	(402, 164, 78, 78)	Unpressed: GUI_Btn_Back_78x78.png Pressed: GUI_Btn_Back_78x78.png
ID_BUTTON_OK	(402, 0, 78, 78)	Unpressed: GUI_Btn_OK_78x78.png Press: GUI_Btn_OK_78x78.png

Table 1-29 Description of Button Properties Setting - Time Setting Screen

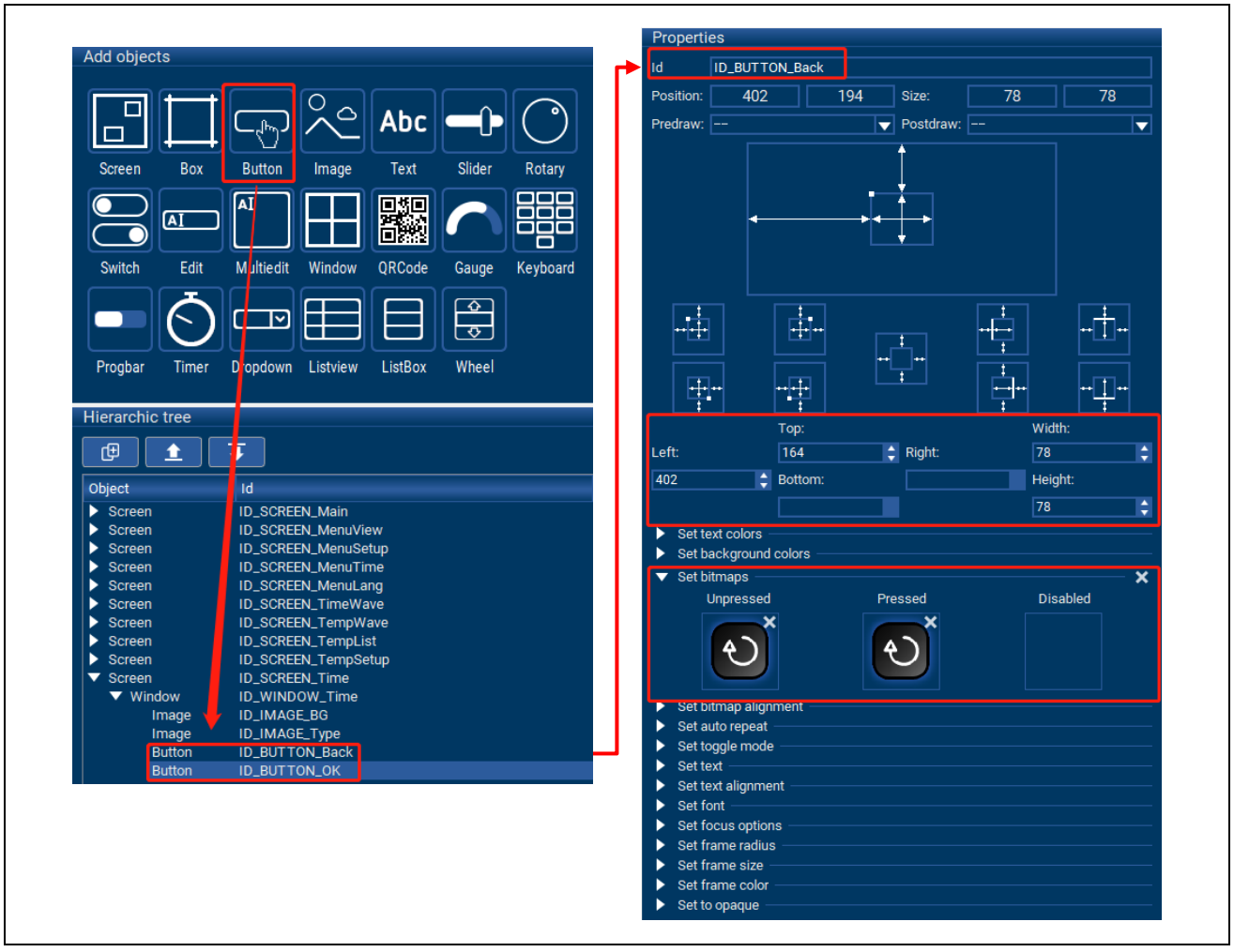


Figure 1-73 Add Buttons - Ok and Back

5. Add Box

Use Box as background for time setting options. Click “Box” in “Add Object” to add a Box and rename it to ID_BOX_00 in “Hierarchic tree”. In “Properties”, set the coordinates to (120, 21, 240, 200), “color” to (255, 255, 255), and “radius” to 25.

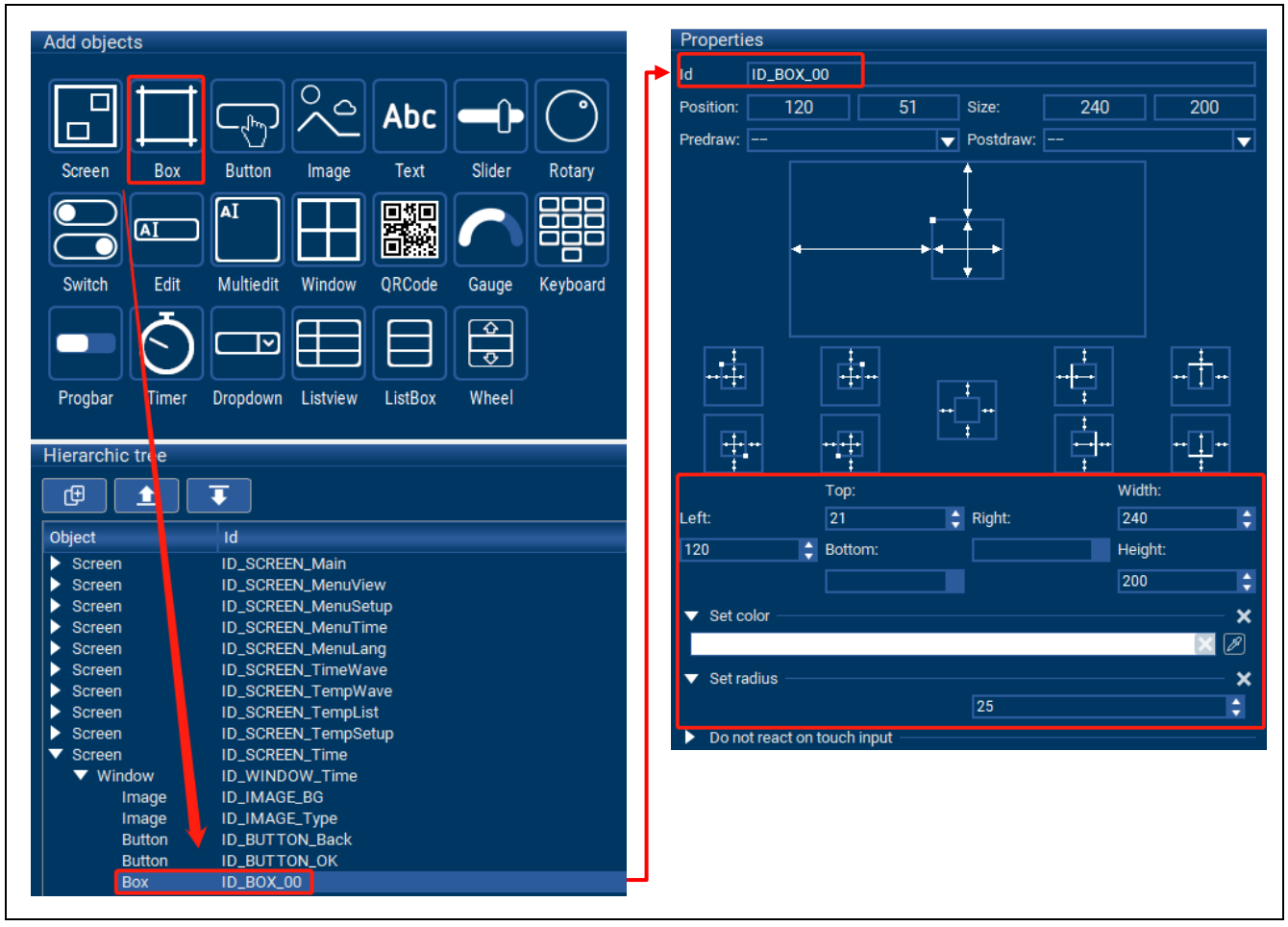


Figure 1-74 Add Box - Background of Time Selection Options

6. Add Text

Add Texts to the window to display text hour, minute, second and colon.

Click “Text” in “Add Object” to add 5 Texts, and rename them in "Hierarchic tree", as shown in Table 1-30. In “Properties”, set “text color” to RGB (0, 0, 0) and “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4. In “Properties”, set coordinates, “text”, and “text alignment” according to Table 1-30. The actual setting items are shown in Figure 1-75.

ID	Coordinates (left, top, width, height)	text	text alignment (offset x, offset y, Alignment)
ID_TEXT_Hour	(132, 31, 50, 24)	ID_RTEXT_HR	(0, 0, Right)
ID_TEXT_Colon1	(182, 122, 24, 24)	ID_RTEXT_COLON	(0, 0, Right)
ID_TEXT_Min	(194, 31, 70, 24)	ID_RTEXT_MIN	(0, 0, Right)
ID_TEXT_Colon2	(264, 122, 24, 24)	ID_RTEXT_ COLON	(0, 0, Right)
ID_TEXT_Sec	(276, 31, 70, 24)	ID_RTEXT_SEC	(0, 0, Right)

Table 1-30 Description of Text Properties Setting - Text Display

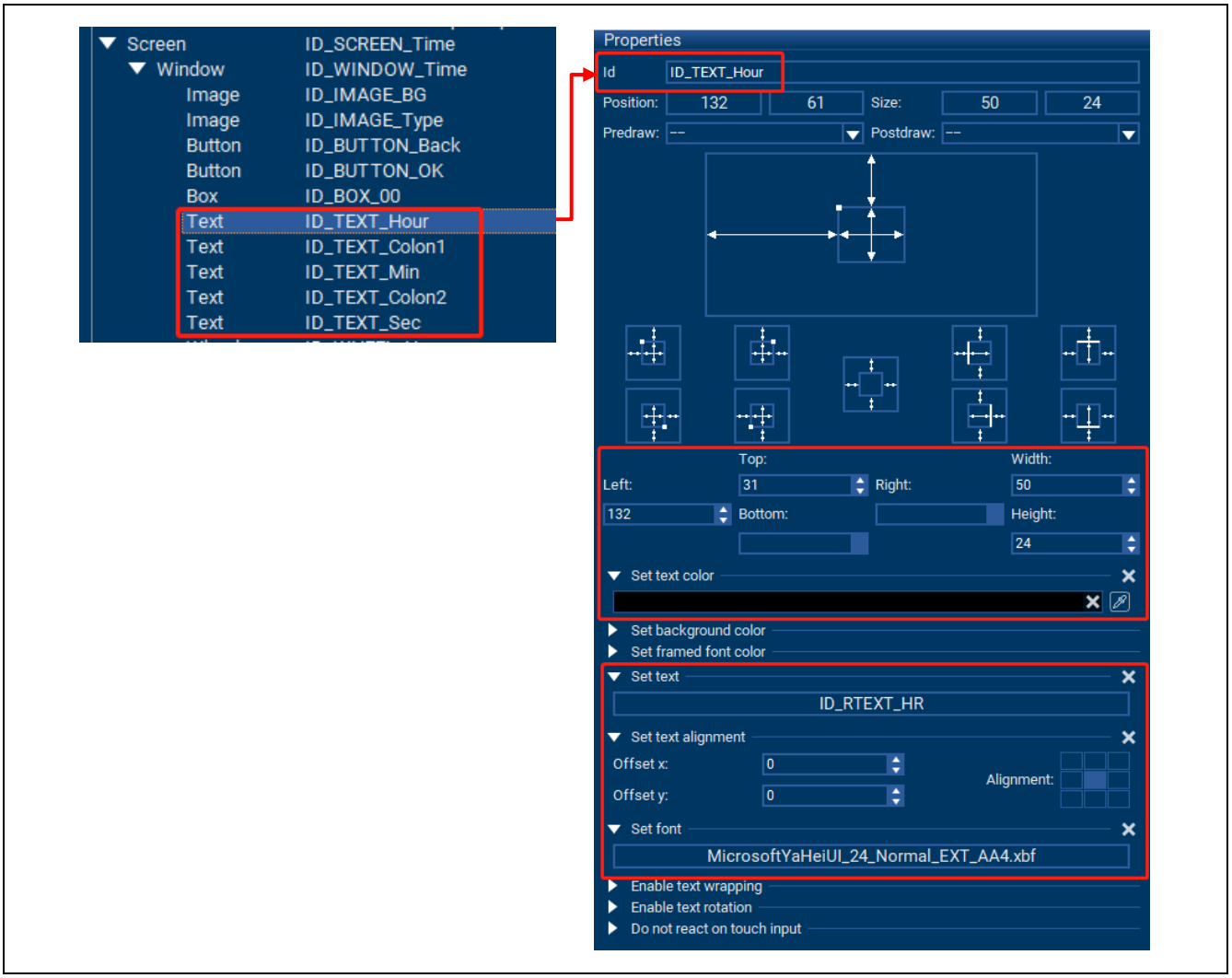


Figure 1-75 Set Text Properties - Text Display

7. Add Wheel

Add Wheels for hour, minute, and second selection.

1) Add Wheel

Click “Wheel” in “Add objects” to add 3 Wheels. As shown in Table 1-31, rename them in “Hierarchic tree”, and set the coordinates in “Properties”. The actual setting items are shown in Figure 1-76.

ID	Coordinates (left, top, width, height)
ID_WHEEL_Hour	(142, 61, 30, 150)
ID_WHEEL_Min	(214, 61, 30, 150)
ID_WHEEL_Sec	(296, 61, 30, 150)

Table 1-31 Coordinates of Wheel

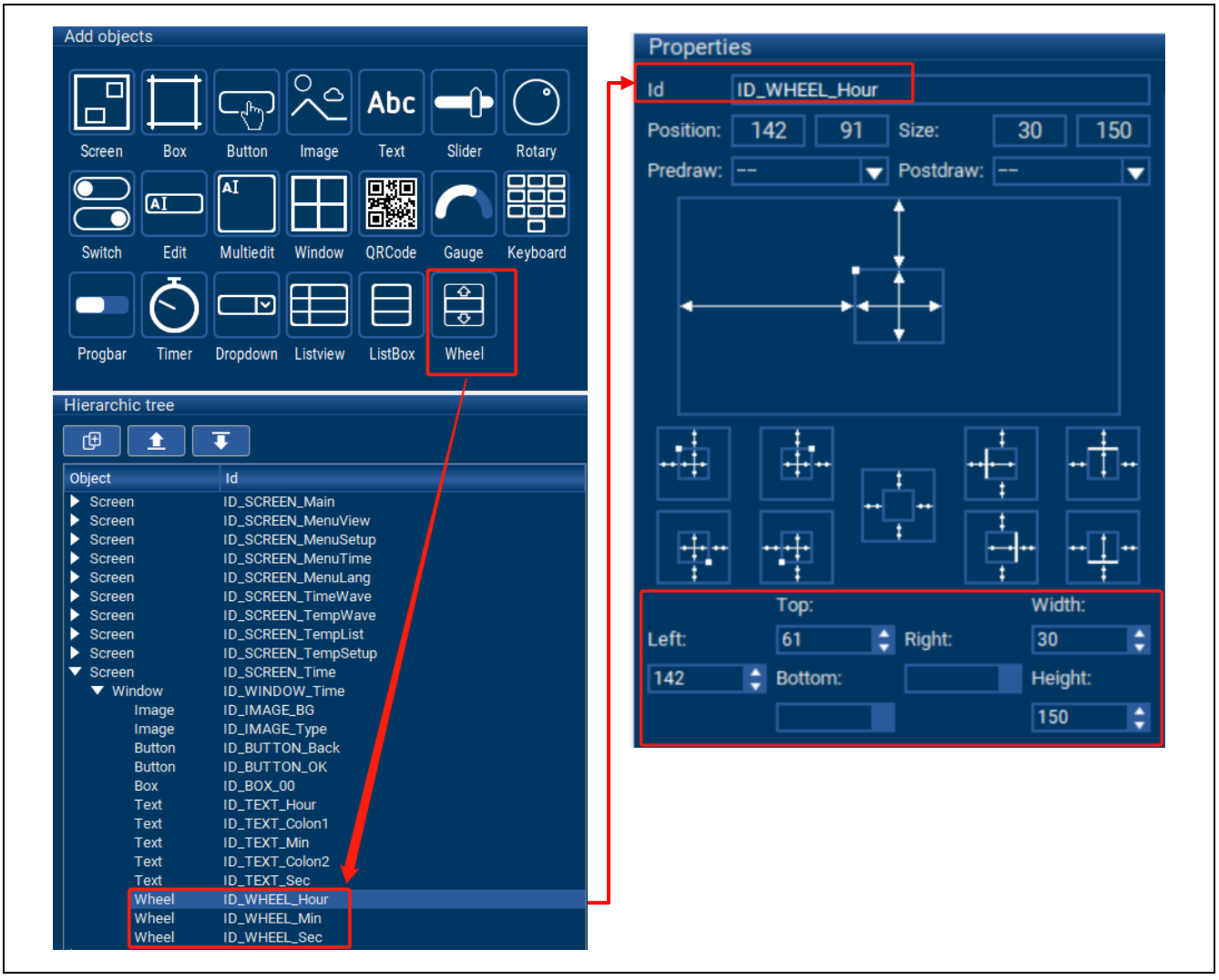


Figure 1-76 Add Wheels - Time Setting Options

2) Set Cell Parameters of Wheel

In “Properties”, set “endless mode”, “Cell and center size” and “color” according to Table 1-32. The actual setting items are shown in Figure 1-77.

Properties	Value	Description
endless mode		Enable endless mode
Cell and center size	cell: 32 center: 32	Set the size of unselected and selected cells
color	Background cell：RBG (255, 255, 255) Background Center：RBG (255, 255, 255) Background：RBG (255, 255, 255)	Set background color of unselected and selected cells

Table 1-32 Description of Cell Properties Setting

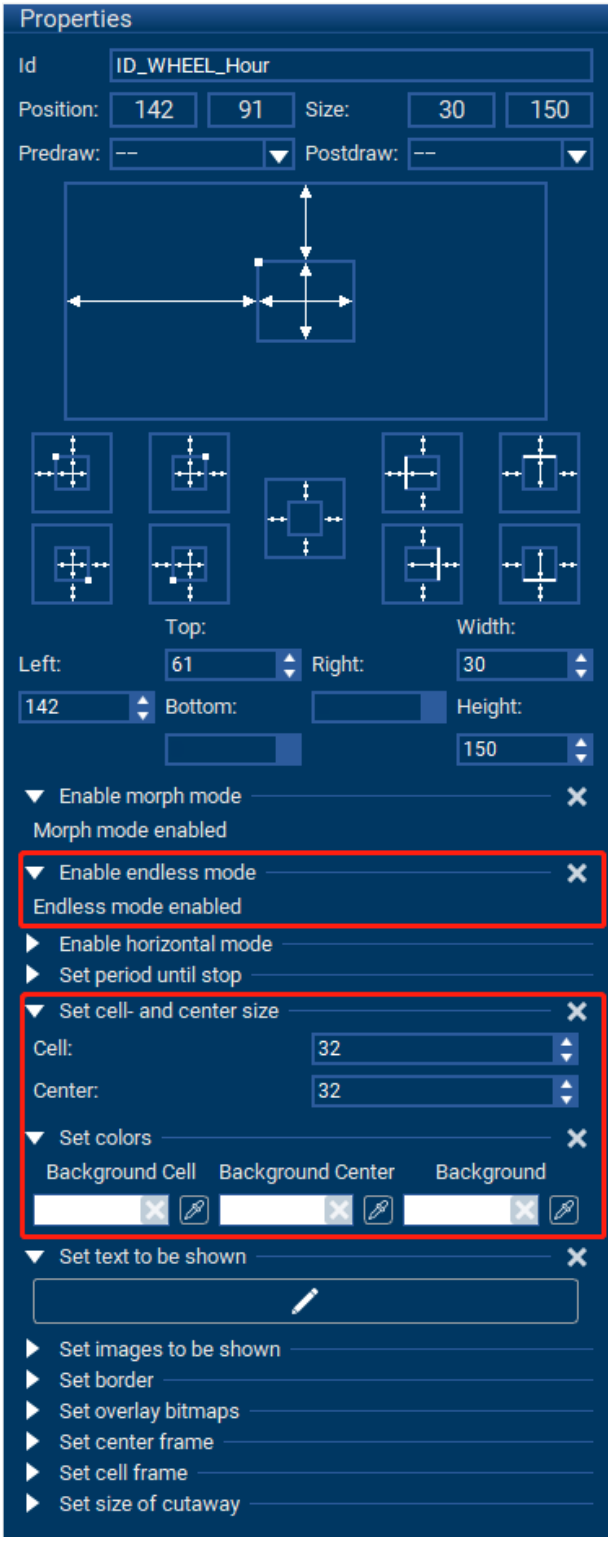


Figure 1-77 Set Cell Properties

3) Set Text for Display

In “Properties”, according to Table 1-33, set text content displayed by the Wheel in “text to be show”. The actual setting items are shown in Figure 1-78.

Properties	ID			Description
	ID_WHEEL_Hour	ID_WHEEL_Min	ID_WHEEL_Sec	
Content	ID_CONTENT_HOUR	ID_CONTENT_MIN	ID_CONTENT_SEC	Set text list for display
Cell alignment	Offset x：0 Offset y：0 Alignment：Center			Alignment of unselected cells
Cell color (R, G, B)	(170, 170, 170)			Font color of unselected cells
Cell Font	MicrosoftYaHeiUI_24_Normal_EXT_AA4			Font
Center alignment	Offset x：0 Offset y：0 Alignment：Center			Alignment of selected cells
Center Color (R, G, B)	(0, 0, 0)			Font color of selected cells

Table 1-33 Wheel Display Content Settings

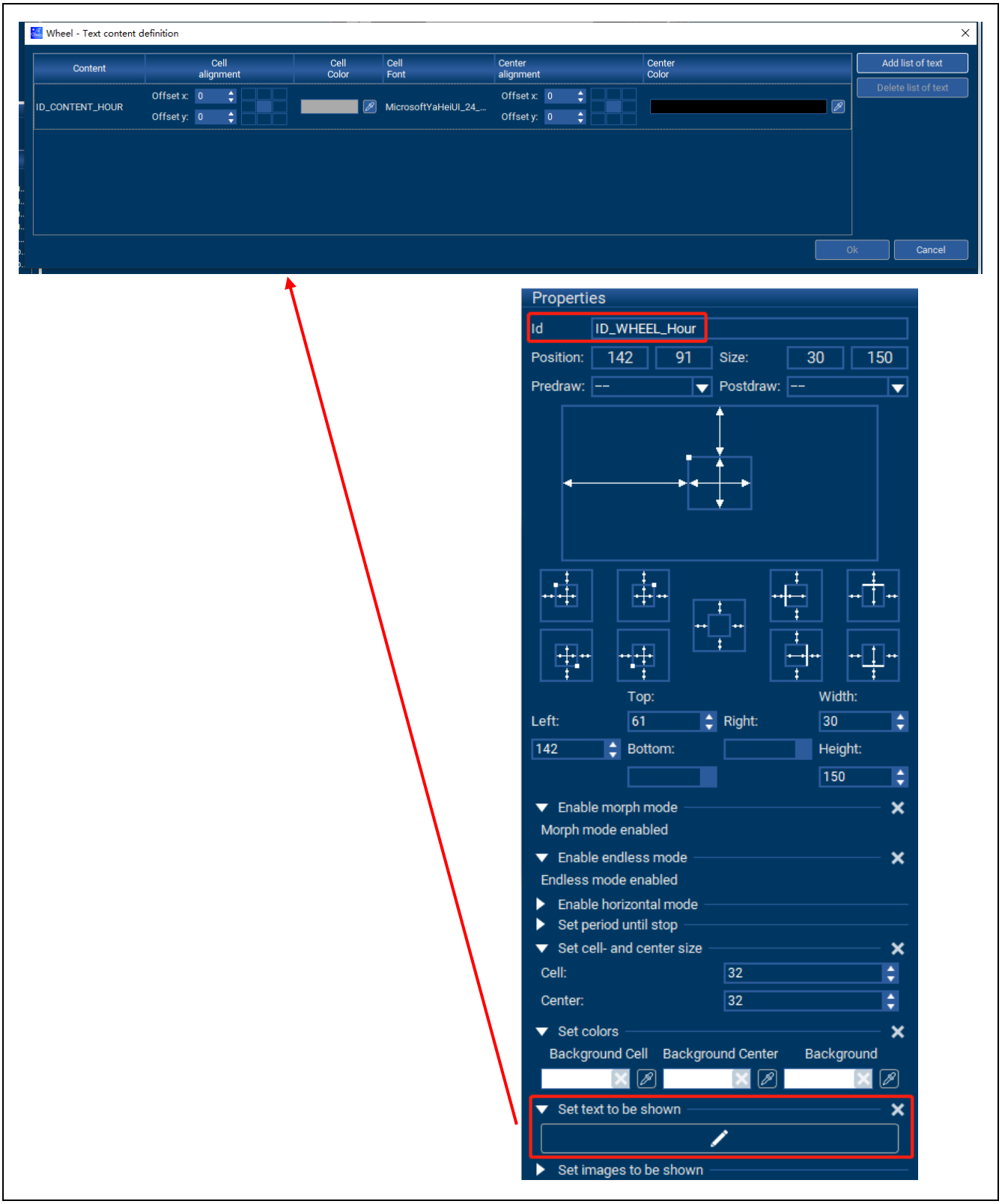


Figure 1-78 Text to be Shown

8. Time Setting Screen After Completion of Design

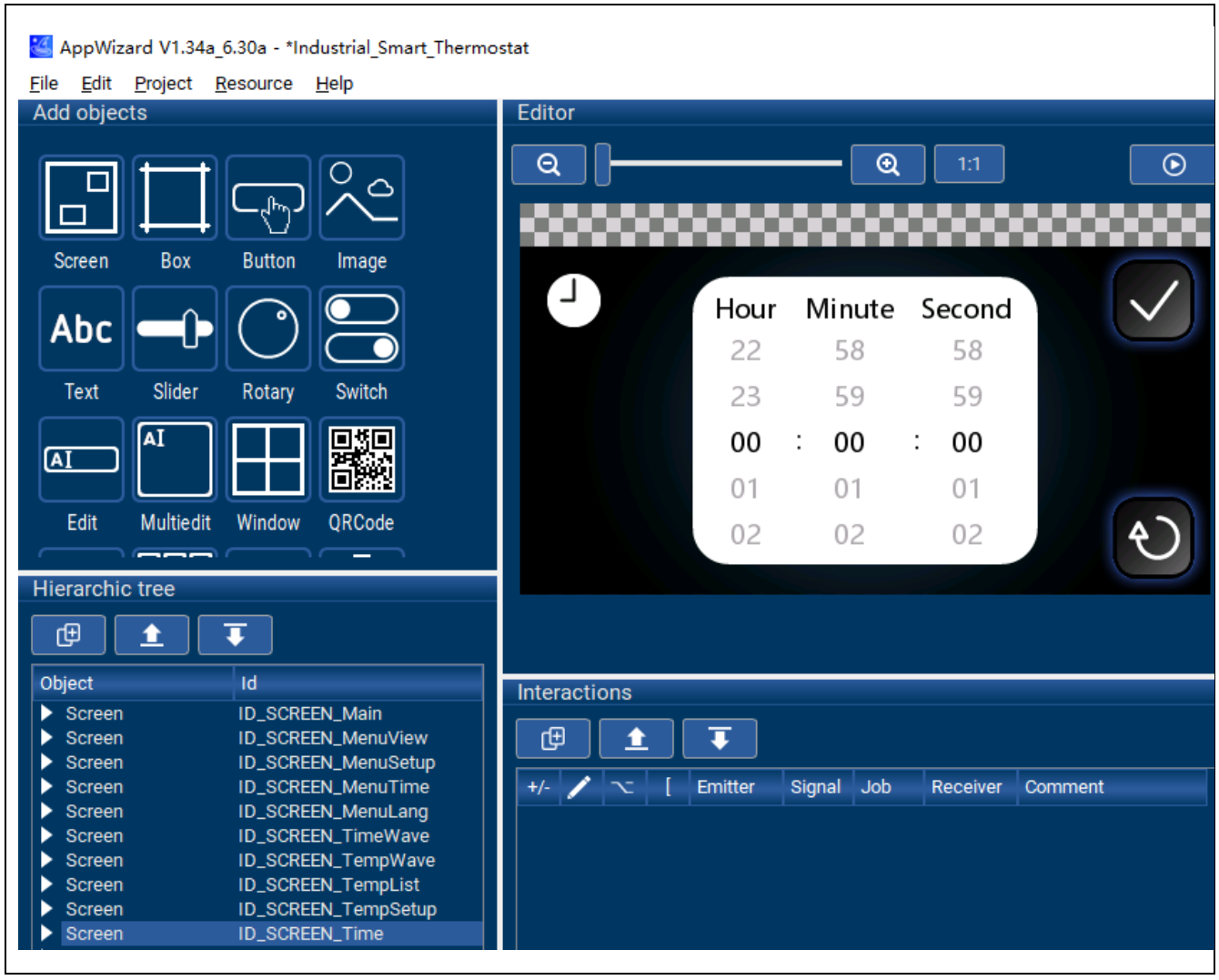


Figure 1-79 Time Setting Screen

1.2.2.15 Create Date Setting Screen

1. Add Screen

Click “Screen” in “Add Object” to add a screen, rename it to ID_SCREEN_Date in “Hierarchic tree”, and set the coordinates to (0, 30, 480, 242) in “Properties”.

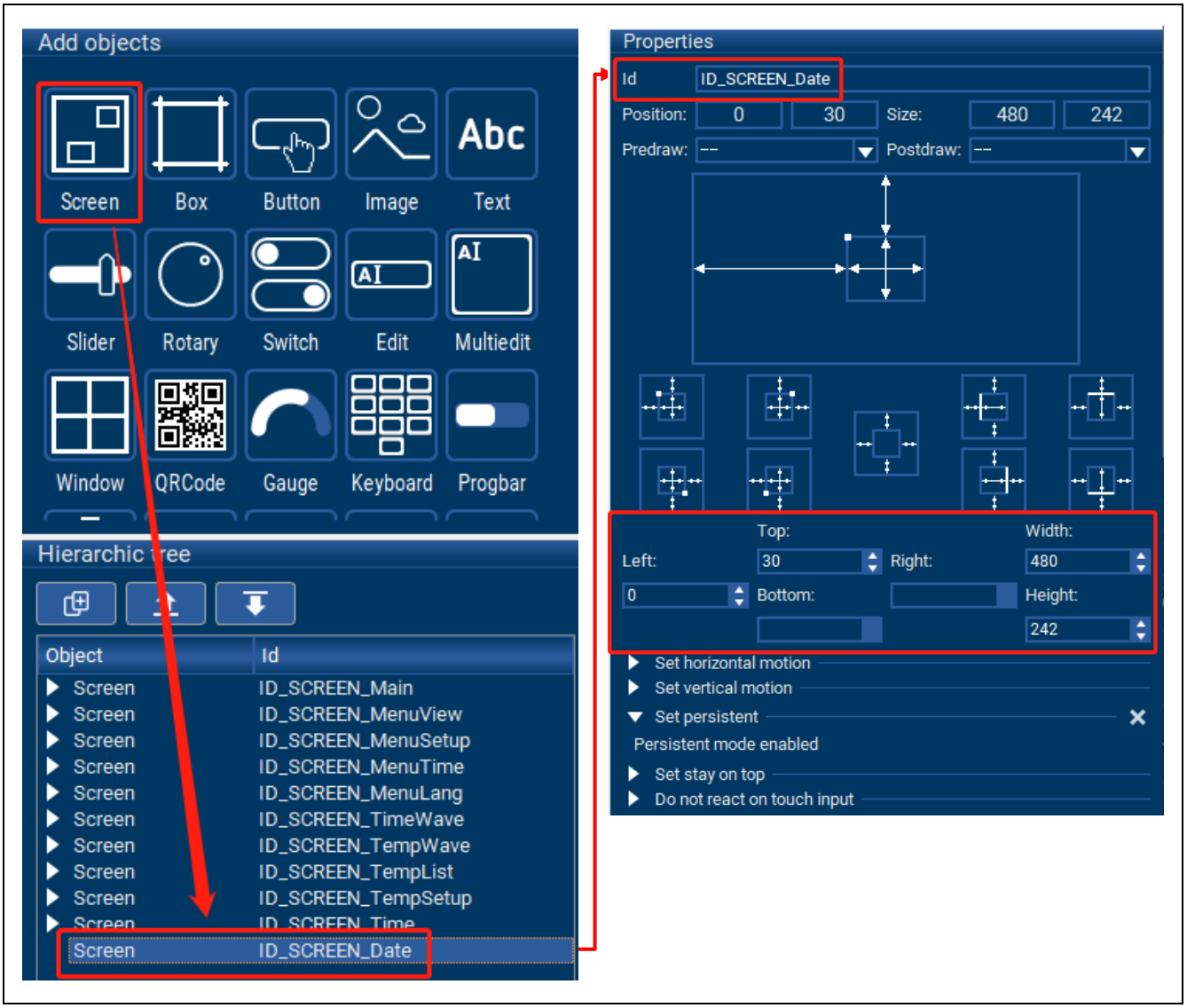


Figure 1-80 Create Date Setting Screen

2. Add Window

Click “Window” in “Add Object” to add a window to the screen, rename it to ID_WINDOW_Date in “Hierarchic tree”, and set the coordinates to (0, 0, 480, 242) in “Properties”.

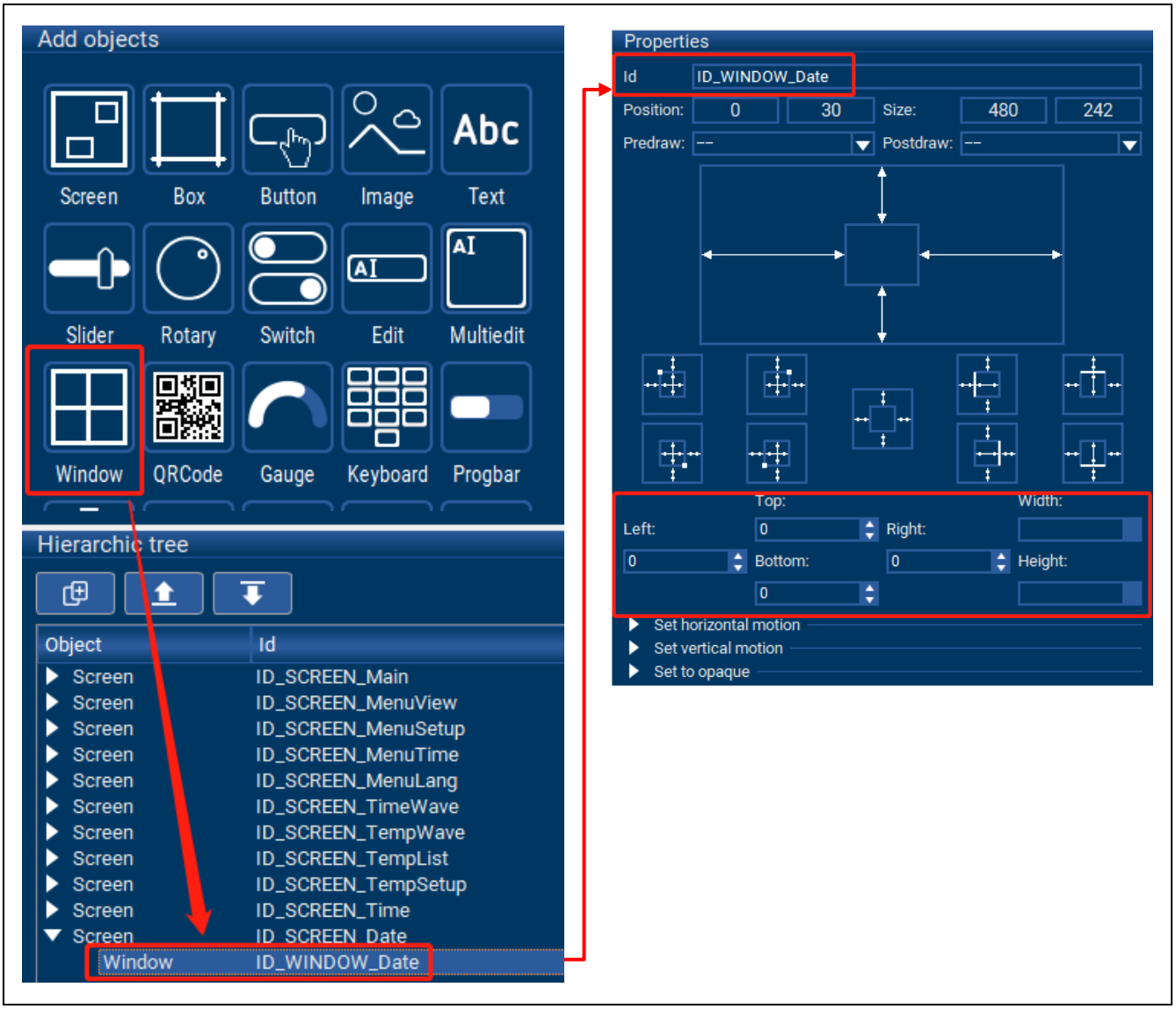


Figure 1-81 Add Window - Date Setting

3. Add Image

Add an Image for display of background and screen functions. Click “Image” in “Add Object” to add 2 Images to the window. According to Table 1-34, rename in “Hierarchic tree” and set the coordinates and “bitmap” in “Properties”. The actual settings are shown in Figure 1-82.

ID	Coordinates (left, top, width, height)	bitmap	Description
ID_IMAGE_BG	(0, 0, 480, 242)	BKG_Time_480x242.png	Background Image
ID_IMAGE_Type	(19, 19, 37, 39)	GUI_Date_Icon_37x38.png	The function of the display screen is date setting.

Table 1-34 Description of Image Properties Setting

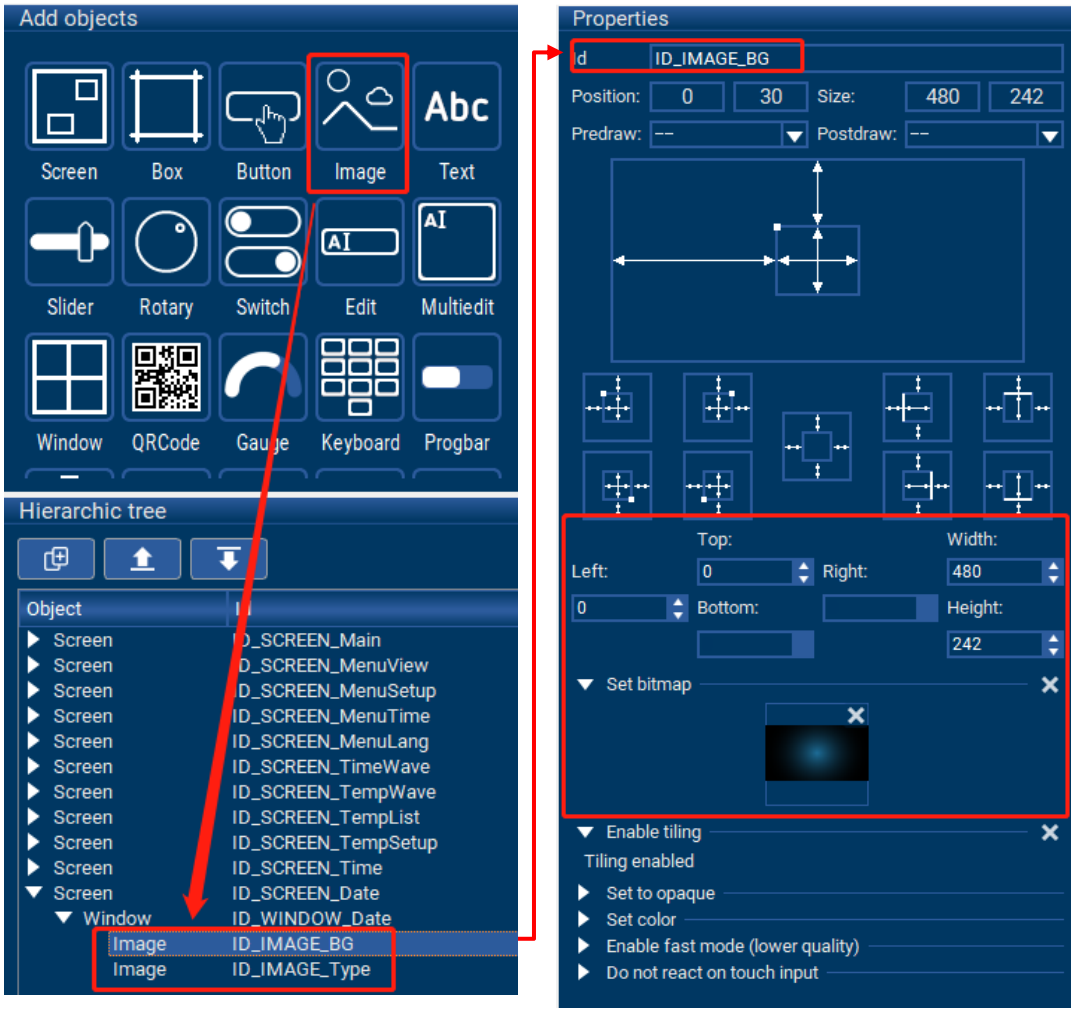


Figure 1-82 Add an Image - Background and Type Display

4. Add Button

Buttons are used to save current settings and exit the current screen. Click “Button” in “Add Objects” to add 2 buttons. As shown in Table 1-35, rename Buttons in “Hierarchic tree”, and set the coordinates and “bitmap” in “Properties”. The actual setting items are shown in Figure 1-83.

Button	**Coordinates (Left, Top, Width, Height)**	**bitmap**
ID_BUTTON_Back	(402, 164, 78, 78)	Unpressed: GUI_Btn_Back_78x78.png Pressed: GUI_Btn_Back_78x78.png
ID_BUTTON_OK	(402, 0, 78, 78)	Unpressed: GUI_Btn_OK_78x78.png Press: GUI_Btn_OK_78x78.png

Table 1-35 Description of Button Properties Setting - Date Setting Screen

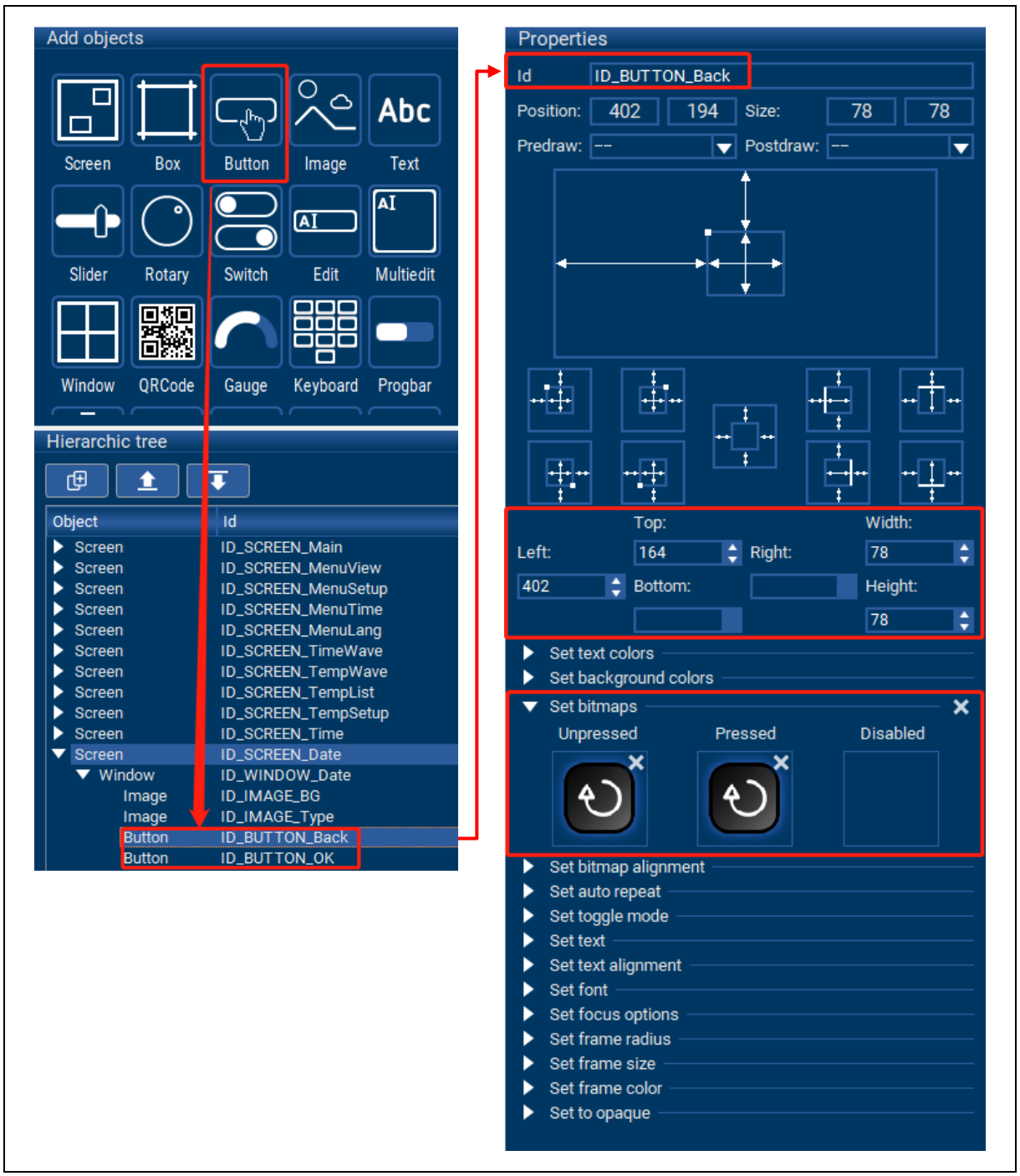


Figure 1-83 Add Button

5. Add Box

Use Box to display background of the time setting options. Click “Box” in “Add Objects” to add a Box and rename it to ID_BOX_00 in “Hierarchic tree”. Set the coordinates in “Properties” to (100, 21, 280, 200), “color” to (255, 255, 255), and “radius” to 25.

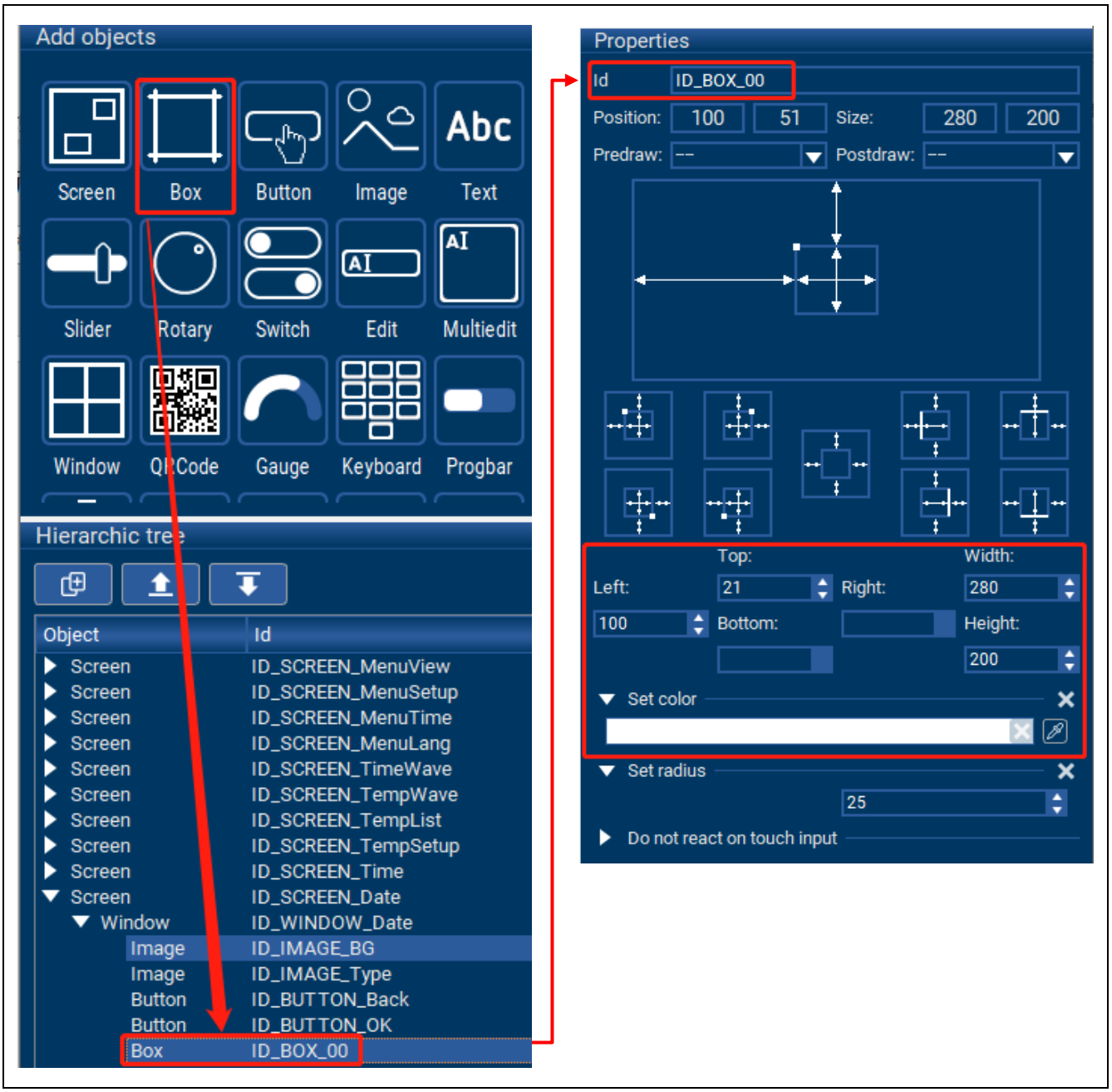


Figure 1-84 Add Box - Background of Date Selection Options

6. Add Text

Add Texts to display the year, month, and day.

Click “Text” in “Add Objects” to add 3 Texts and rename them in “Hierarchic tree” according to Table 1-36. In “Properties”, set “text color” to RGB (0, 0, 0) and “font” to MicrosoftYaHeiUI_24_Normal_EXT_AA4. Set coordinates, “text”, and “text alignment” according to Table 1-36. The actual setting items are shown in Figure 1-85.

ID	Coordinates (left, top, width, height)	text	text alignment (offset x, offset y, Alignment)
ID_TEXT_Day	(112, 31, 50, 24)	ID_RTEXT_DATE	(0, 0, Right)
ID_TEXT_Month	(194, 31, 70, 24)	ID_RTEXT_MONTH	(0, 0, Right)
ID_TEXT_Year	(301, 31, 70, 24)	ID_RTEXT_YEAR	(0, 0, Right)

Table 1-36 Description of Text Properties Setting -Text Display

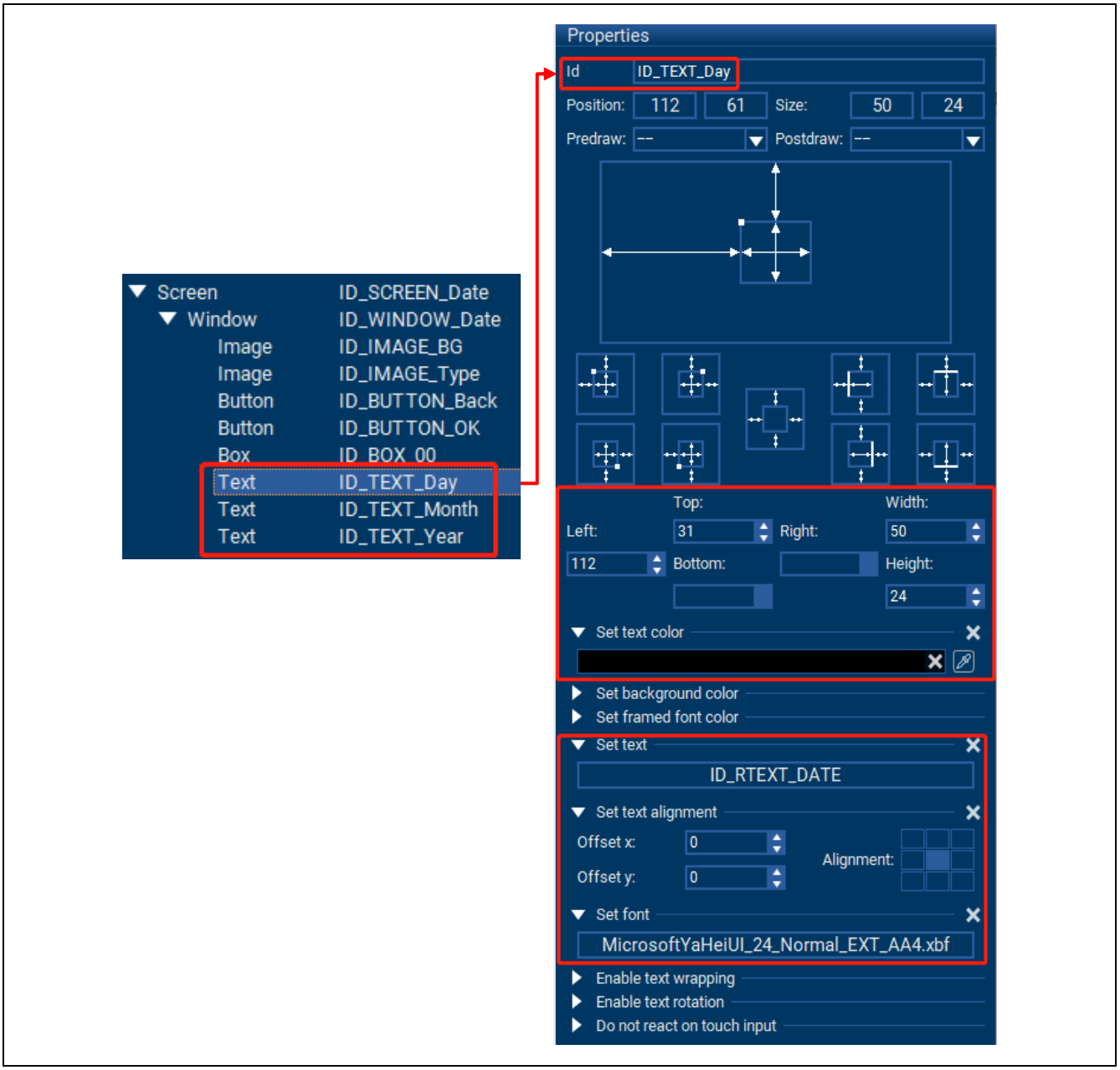


Figure 1-85 Set Text Properties -Text Display

7. Add Wheel

Add Wheels for year, month, and day selection.

1) Add Wheel

Click “Wheel” in “Add objects” to add 3 Wheels. As shown in Table 1-37, rename them in “Hierarchic tree” and set the coordinates in “Properties”.

ID	Coordinates (left, top, width, height)
ID_WHEEL_Day	(122, 61, 30, 150)
ID_WHEEL_Month	(179, 61, 100, 150)
ID_WHEEL_Year	(306, 61, 60, 150)

Table 1-37 Coordinates of Wheel

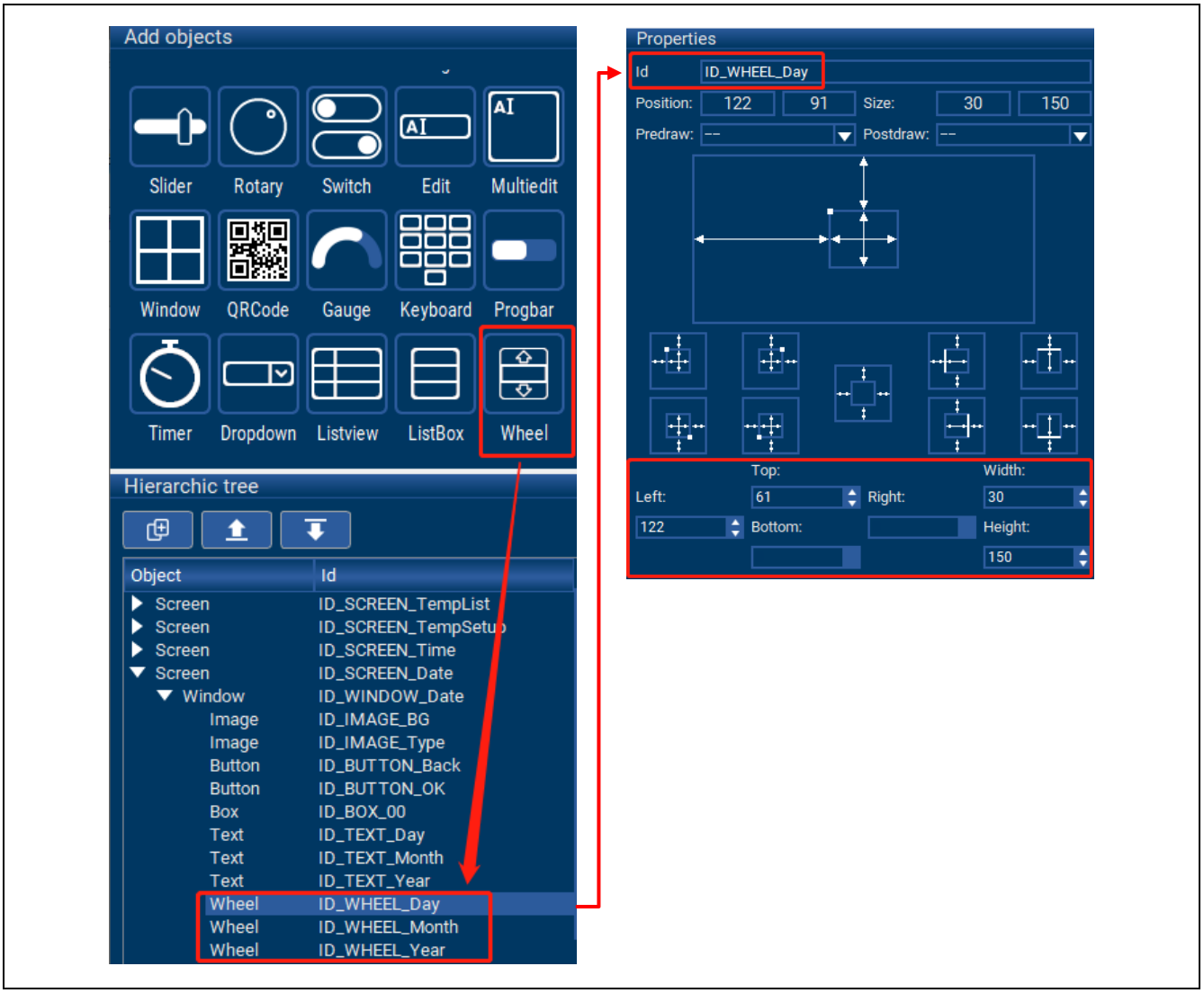


Figure 1-86 Add Wheel - Date Setting Options

2) Set Cell Parameters of Wheel

Set “endless mode”, “Cell and center size” and “color” in “Properties” according to Table 1-38. The actual setting items are shown in Figure 1-87.

Properties	Value	Description
endless mode		Enable endless mode
Cell and center size	cell: 32 center: 32	Set the size of unselected and selected cells
color	Background cell：RGB (0, 0, 0) Background Center：RGB (0, 0, 0) Background：RGB (0, 0, 0)	Set background color of unselected and selected cells

Table 1-38 Description of Cell Properties Setting

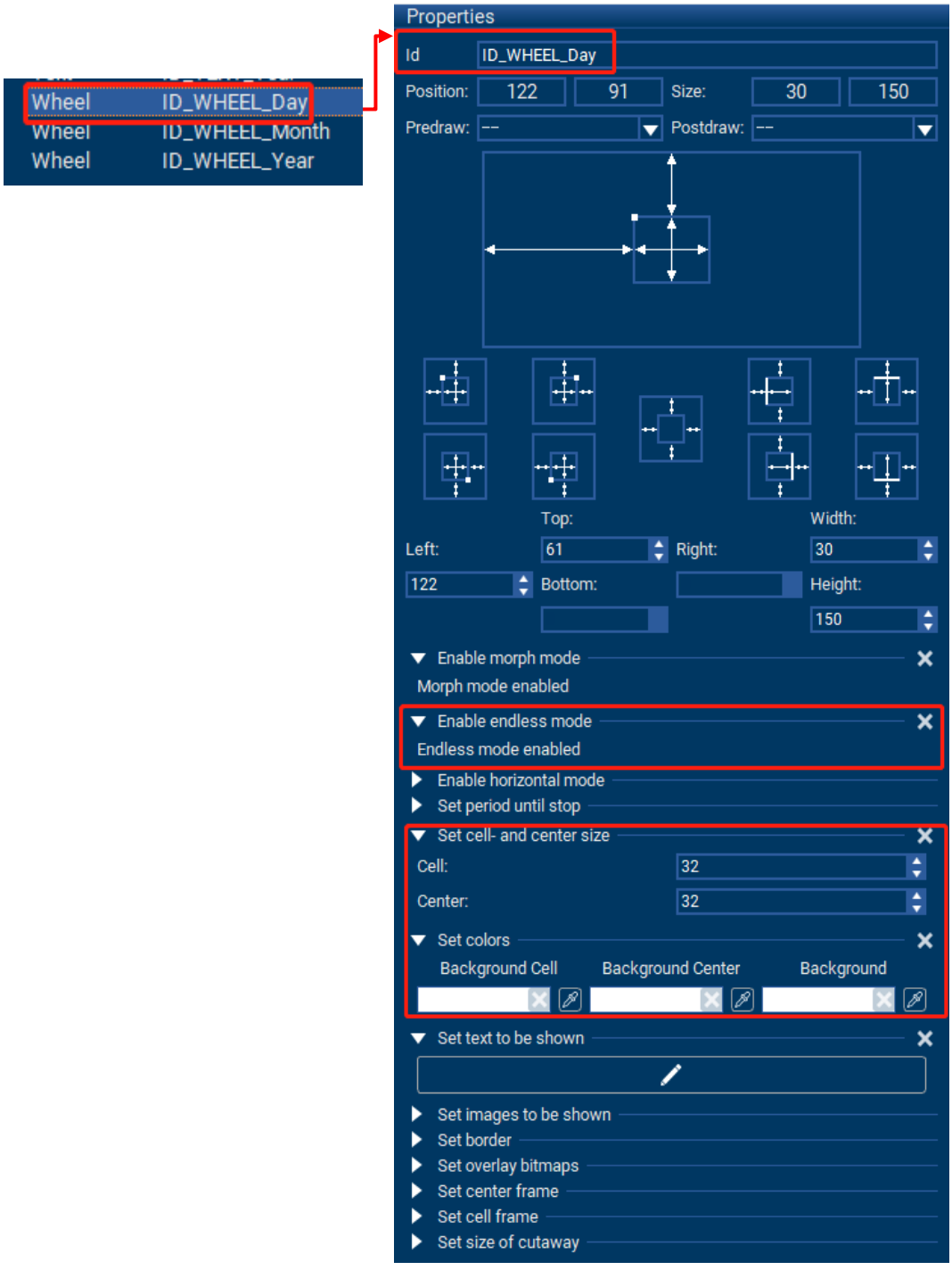


Figure 1-87 Set Cell Properties

3) Set Text for Display

According to Table 1-39, In “Properties”, set text content displayed by the Wheel in “text to be show”. The actual setting items are shown in Figure 1-88.

Properties	ID			Description
	ID_WHEEL_Day	ID_WHEEL_Month	ID_WHEEL_Year	
Content	ID_CONTENT_DAY	ID_CONTENT_MONTH	ID_CONTENT_YEAR	Set text list for display
Cell alignment	Offset x：0 Offset y：0 Alignment：Center			Alignment of unselected cells
Cell color (R, G, B)	(170, 170, 170)			Font color of unselected cells
Cell Font	MicrosoftYaHeiUI_24_Normal_EXT_AA4			Font
Center alignment	Offset x：0 Offset y：0 Alignment：Center			Alignment of selected cells
Center Color (R, G, B)	(0, 0, 0)			Font color of selected cells

Table 1-39 Wheel Display Content Settings

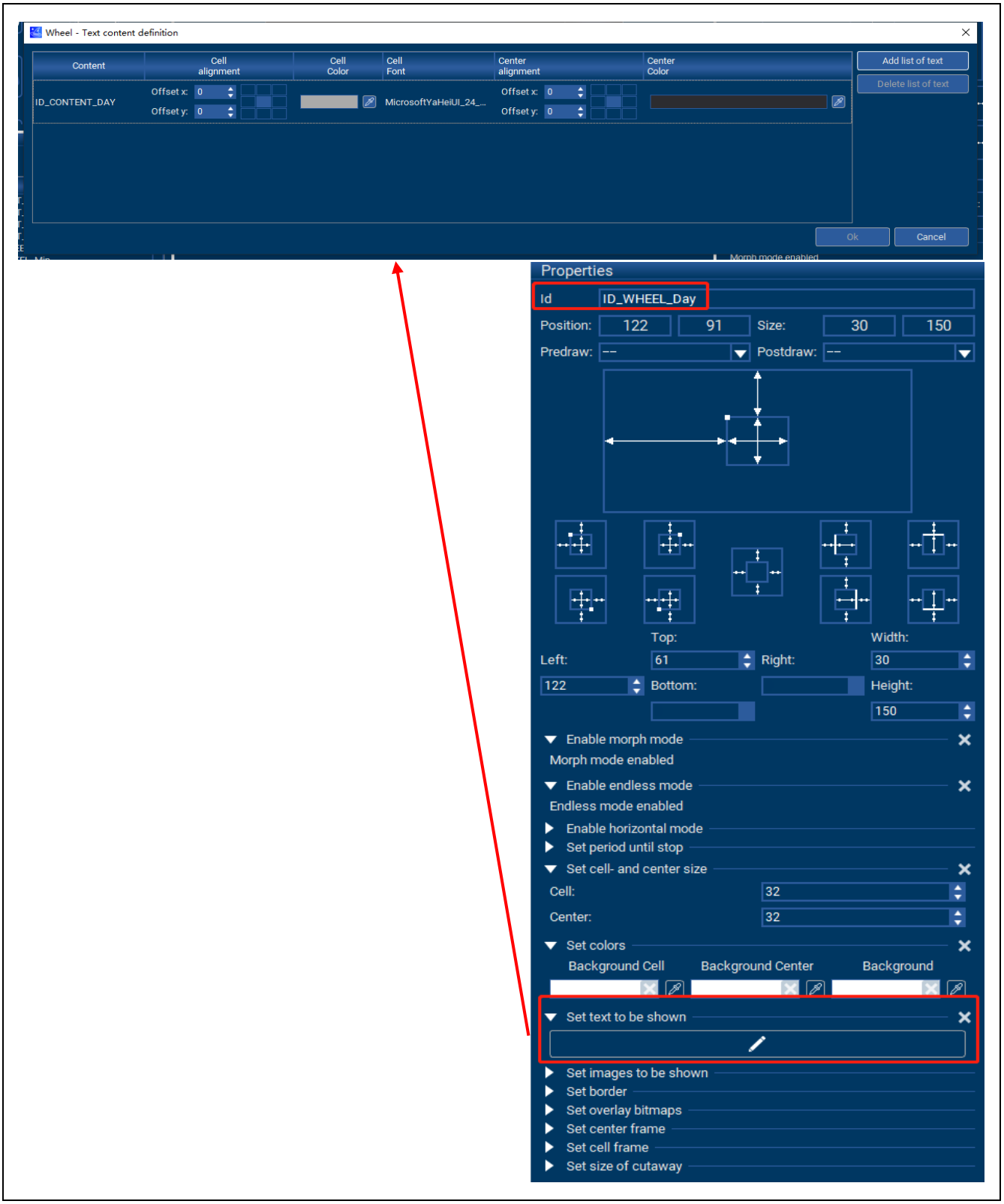


Figure 1-88 Text to Be Shown

8. Date Setting Screen After Completion of Design

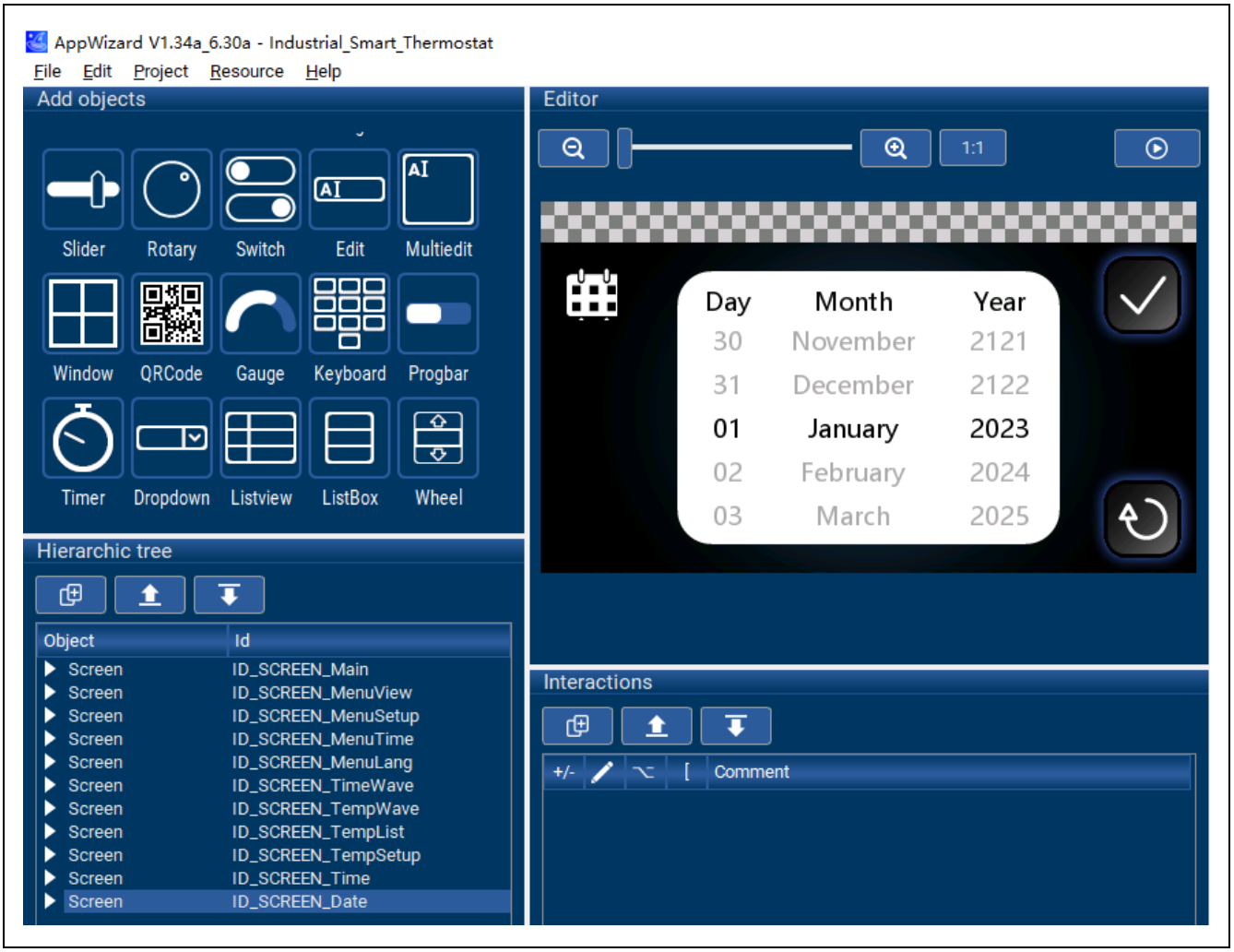


Figure 1-89 Date Setting Screen

1.2.2.16 Set Interaction

Interaction implementation method:

When Emitter A does Signal, Receiver B does Job.

Or when Emitter A does/occurs Signal, does Job to Receiver B.

The complete setting items are shown in Figure 1-90, and description of setting items is shown in Table 1-40.

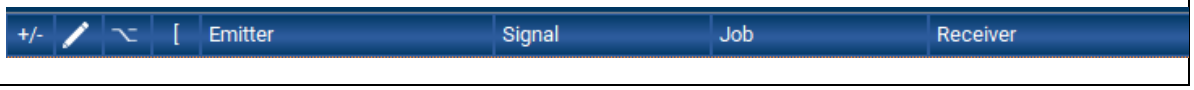


Figure 1-90 Set Interaction

Content	Description
	Add/remove Interaction items
	Set parameters of interaction
	Conditions for interaction to occur, not necessary
	Groups setting
Emitter	emitter
Signal	Event that occurred with Emitter
Job	The event to happen for Receiver
Receiver	receiver
Comment	comment

Table 1-40 Description of Interaction Setting

1. Set Interaction - Main Menu

   1) Display Main Menu and Temperature List When Powering On

   When powering on, it is necessary to display main menu and temperature list simultaneously. This is achieved by triggering display of ID_SCREEN_TempList during initialization ID_SCREEN_Main. The actual settings are shown in Figure 1-91.

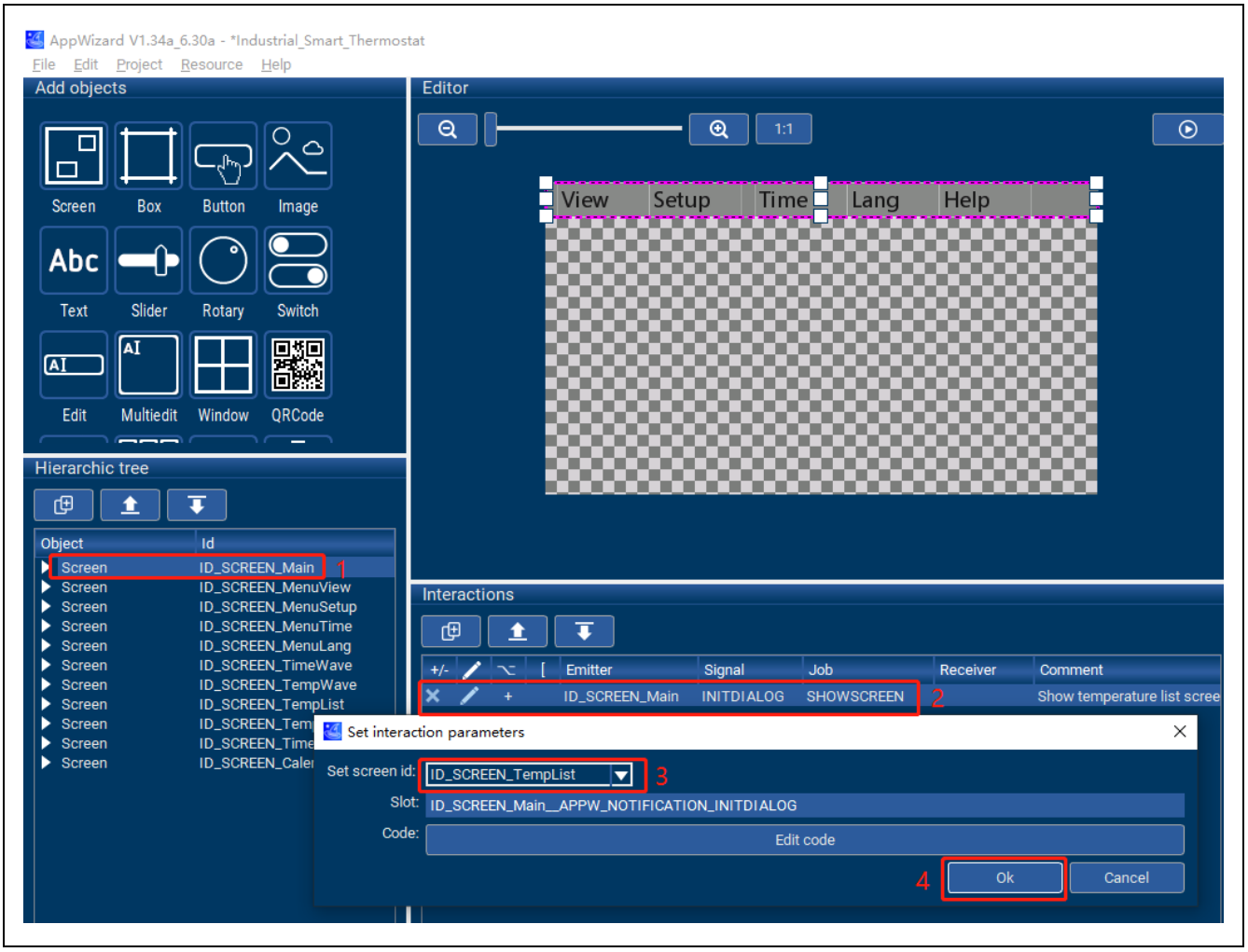
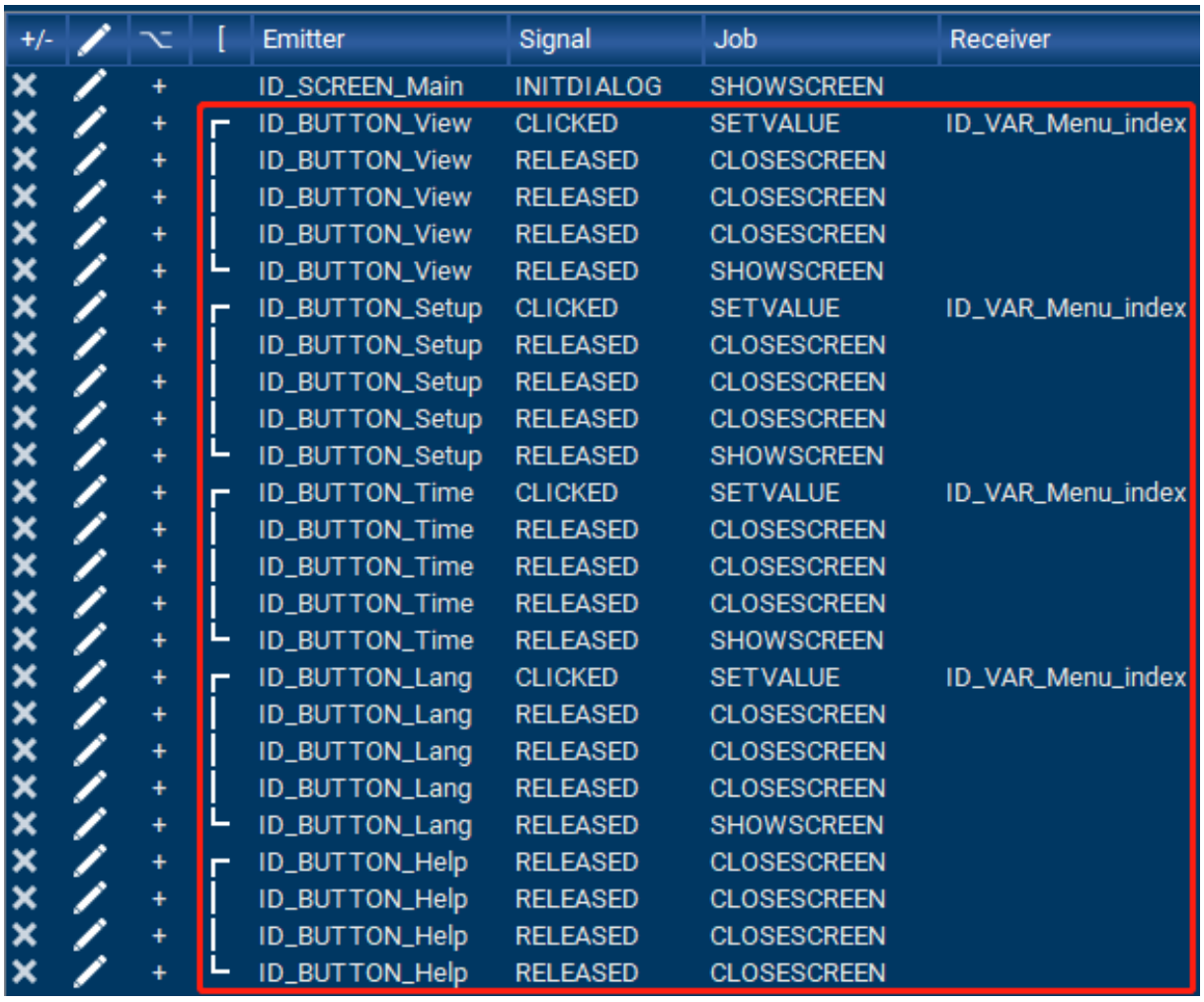


Figure 1-91 Set Interaction - Display Temperature List When Initializing Main Menu

2) Clcik Button in Menu

When a button in main menu is clicked, the corresponding drop-down menu needs to be displayed and other drop-down menus hidden. Note that ID_BUTTON_Help does not have a drop-down menu and will only hide other drop-down menus when pressed. The actual setting items are shown in Figure 1-92, and the detailed description of the actions is Table 1-41.



+/-		⌥	[	Emitter	Signal	Job	Receiver
✕		+		ID_SCREEN_Main	INITDIALOG	SHOWSCREEN	
✕		+	┌	ID_BUTTON_View	CLICKED	SETVALUE	ID_VAR_Menu_index
✕		+	│	ID_BUTTON_View	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_View	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_View	RELEASED	CLOSESCREEN	
✕		+	└	ID_BUTTON_View	RELEASED	SHOWSCREEN	
✕		+	┌	ID_BUTTON_Setup	CLICKED	SETVALUE	ID_VAR_Menu_index
✕		+	│	ID_BUTTON_Setup	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Setup	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Setup	RELEASED	CLOSESCREEN	
✕		+	└	ID_BUTTON_Setup	RELEASED	SHOWSCREEN	
✕		+	┌	ID_BUTTON_Time	CLICKED	SETVALUE	ID_VAR_Menu_index
✕		+	│	ID_BUTTON_Time	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Time	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Time	RELEASED	CLOSESCREEN	
✕		+	└	ID_BUTTON_Time	RELEASED	SHOWSCREEN	
✕		+	┌	ID_BUTTON_Lang	CLICKED	SETVALUE	ID_VAR_Menu_index
✕		+	│	ID_BUTTON_Lang	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Lang	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Lang	RELEASED	CLOSESCREEN	
✕		+	└	ID_BUTTON_Lang	RELEASED	SHOWSCREEN	
✕		+	┌	ID_BUTTON_Help	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Help	RELEASED	CLOSESCREEN	
✕		+	│	ID_BUTTON_Help	RELEASED	CLOSESCREEN	
✕		+	└	ID_BUTTON_Help	RELEASED	CLOSESCREEN	

Figure 1-92 Set Interaction – Buttons of Main Menu

Parameter	Emitter	Signal	Job
ID_SCREEN_MenuSetup	ID_BUTTON_View	RELEASED	CLOSESCREEN
ID_SCREEN_MenuTime			
ID_SCREEN_MenuLang			
ID_SCREEN_MenuView			SHOWSCREEN
ID_SCREEN_MenuView	ID_BUTTON_Setup	RELEASED	CLOSESCREEN

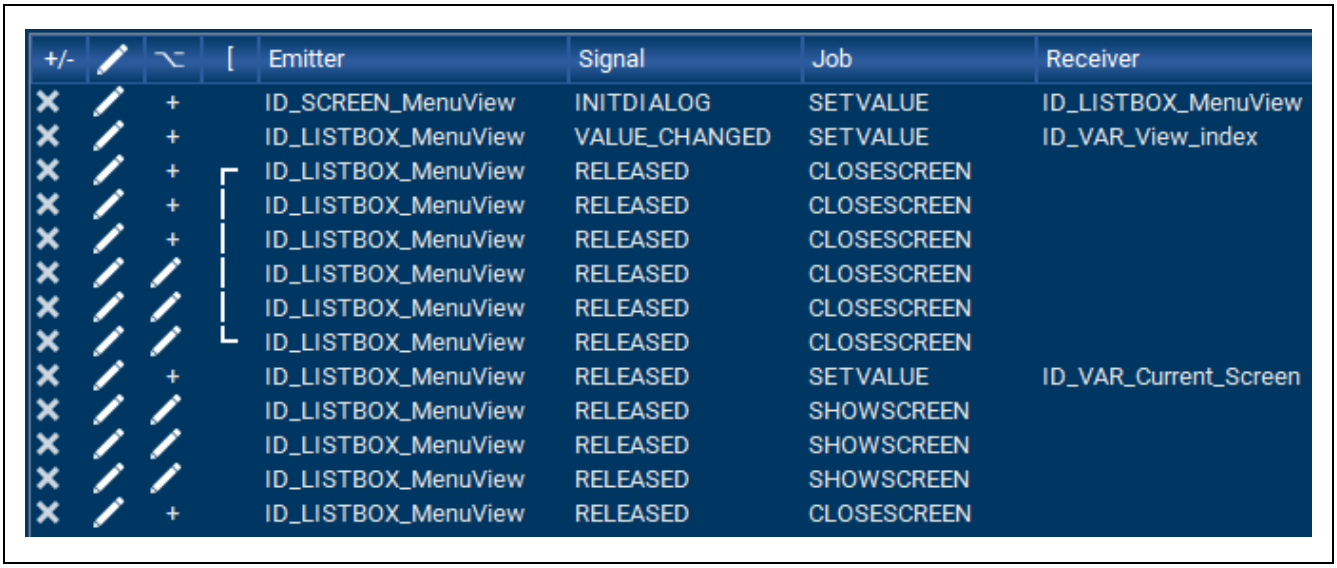
Parameter	Emitter	Signal	Job
ID_SCREEN_MenuTime			
ID_SCREEN_MenuLang			
ID_SCREEN_MenuSetup			SHOWSCREEN
ID_SCREEN_MenuView	ID_BUTTON_Time	RELEASED	CLOSESCREEN
ID_SCREEN_MenuSetup			
ID_SCREEN_MenuLang			
ID_SCREEN_MenuTime			SHOWSCREEN
ID_SCREEN_MenuView	ID_BUTTON_Lang	RELEASED	CLOSESCREEN
ID_SCREEN_MenuSetup			
ID_SCREEN_MenuTime			
ID_SCREEN_MenuLang			SHOWSCREEN
ID_SCREEN_MenuView	ID_BUTTON_Setup	RELEASED	CLOSESCREEN
ID_SCREEN_MenuSetup			
ID_SCREEN_MenuTime			
ID_SCREEN_MenuLang			

Table 1-41 Description of Interaction - Buttons of Main Menu

2. Set Interaction - Drop-down Menu

1) Interaction of View Drop-down Menu

When ID_SCREEN_MenuView screen is displayed, default focus is on the first option. When an option is selected, the corresponding screen needs to be displayed and the focus position needs to be saved. Define a variable ID_VAR_View_index to save index. When screen is initialized, set the variable value to 0. When ID_LISTBOX_Menuview value changes, save index of the option to the variable. When option is released, the corresponding screen is displayed according to the variable value, and the index of screen is saved. The actual setting items are shown in Figure 1-93, and description of actions is shown in Table 1-42.



+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_MenuView	INITDIALOG	SETVALUE	ID_LISTBOX_MenuView
+	ID_LISTBOX_MenuView	VALUE_CHANGED	SETVALUE	ID_VAR_View_index
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	
+	ID_LISTBOX_MenuView	RELEASED	SETVALUE	ID_VAR_Current_Screen
	ID_LISTBOX_MenuView	RELEASED	SHOWSCREEN	
	ID_LISTBOX_MenuView	RELEASED	SHOWSCREEN	
	ID_LISTBOX_MenuView	RELEASED	SHOWSCREEN	
+	ID_LISTBOX_MenuView	RELEASED	CLOSESCREEN	

Figure 1-93 Set Interaction - View Drop-down Menu

Parameter	Condition	Emitter	Signal	Job	Receiver
ID_VAR_View_index		ID_SCREEN_MenuView	INITDALOG	SETVAL	ID_LISTBOX_MenuVIew
		ID_LISTBOX_MenuVIew	VALUE_CHANGED	SETVAL	ID_VAR_View_index
ID_SCREEN_TempSetup		ID_LISTBOX_MenuVIew	RELEASED	CLOSESCREEN	
ID_SCREEN_Time					
ID_SCREEN_Date					
ID_SCREEN_TempList	ID_VAR_View_index ≠ 0				
ID_SCREEN_TempWave	ID_VAR_View_index ≠ 1				
ID_SCREEN_TimeWave	ID_VAR_View_index ≠ 2				
ID_VAR_View_index				SETVALUE	ID_VAR_Current_Screen

Parameter	Condition	Emitter	Signal	Job	Receiver
ID_SCREEN_TempList	ID_VAR_View_index = 0			SHOWSCREEN	
ID_SCREEN_TempWave	ID_VAR_View_index = 1				
ID_SCREEN_TimeWave	ID_VAR_View_index = 2				
ID_SCREEN_MenuView		ID_LISTBOX_ MenuVIew	RELEASED	CLOSESCREEN	

Table 1-42 Description of Interaction - View Drop-down Menu

2) Interactions of Setup drop-down menu and Time drop-down menu are similar to View drop-down menu, such as Figure 1-94 and Figure 1-95.

+/-		⌥	[	Emitter	Signal	Job	Receiver
✕	✎	+	┌	ID_LISTBOX_MenuSetup	CLICKED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuSetup	CLICKED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuSetup	CLICKED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuSetup	CLICKED	CLOSESCREEN	
✕	✎	+	└	ID_LISTBOX_MenuSetup	CLICKED	CLOSESCREEN	
✕	✎	+		ID_LISTBOX_MenuSetup	CLICKED	SETVALUE	ID_VAR_Current_Screen
✕	✎	+		ID_LISTBOX_MenuSetup	CLICKED	SHOWSCREEN	

Figure 1-94 Set Interaction - Setup Drop-down Menu

+/-		⌥	[	Emitter	Signal	Job	Receiver
✕	✎	+	┌	ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	
✕	✎	+	│	ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	
✕	✎	+	└	ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	
✕	✎	✎	┌	ID_LISTBOX_MenuTime	RELEASED	SHOWSCREEN	
✕	✎	✎	└	ID_LISTBOX_MenuTime	RELEASED	SHOWSCREEN	
✕	✎	✎	┌	ID_LISTBOX_MenuTime	RELEASED	SETVALUE	ID_VAR_Current_Screen
✕	✎	✎	└	ID_LISTBOX_MenuTime	RELEASED	SETVALUE	ID_VAR_Current_Screen
✕	✎	+		ID_LISTBOX_MenuTime	RELEASED	CLOSESCREEN	

Figure 1-95 Set Interaction - Time Drop-down

3) Set Interaction - Lang Drop-down Menu

When Lang drop-down menu is selected, need to set corresponding language to Chinese or English and hide the drop-down menu. In addition, if current display is Date Settings screen, the screen needs to be re-displayed for language setting to take effect. The actual settings are shown in Figure 1-96.

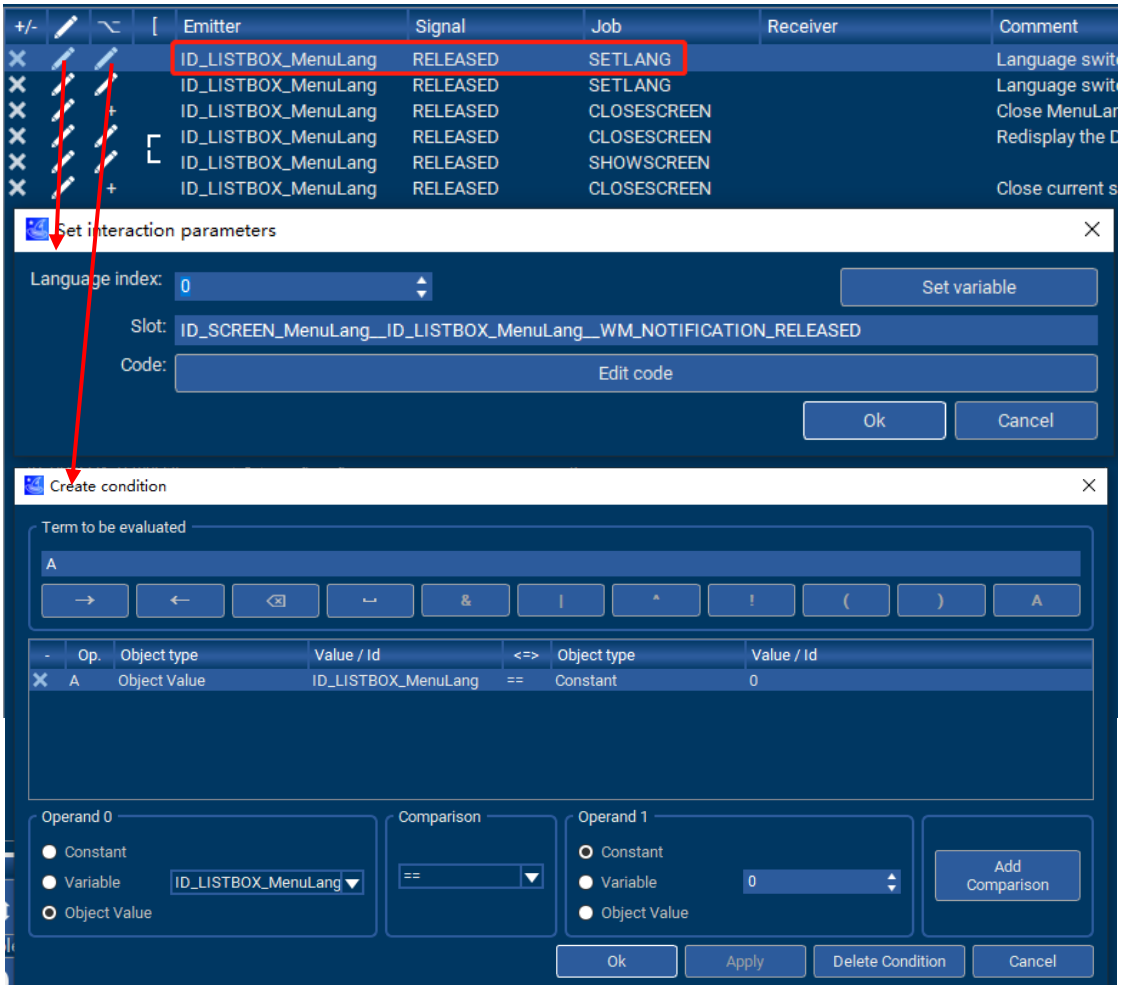


Figure 1-96 Set Interaction - Lang Drop-down Menu

3. Set Interaction - Time-Temperature Setting Screen

The drawing area size is defined as 400x200, divided into 100 points, and refresh time is set to 60 s, which means it takes 100 minutes to refresh 100 points. As shown in Figure 1-97, x-coordinate scale is divided into 5 grids, and time interval of each grid is 20 minutes.

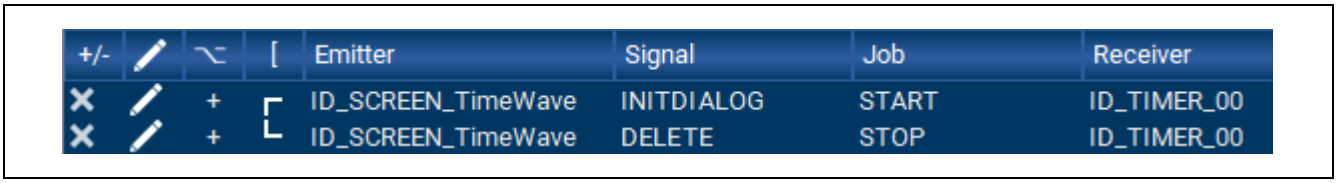
Figure 1-97 X-Coordinate (time)

Display starts from left and increases by one point with each refresh. When 100 points are reached, each refresh will move the original waveform one point to left, and the last point will always display the latest data.

When screen is first displayed, current time is time of the first point. After 100 points are displayed, current time corresponds to the last point.

1) Start Timer

Use a 1 s timer as a beat timer and control waveform refresh interval by counting timeout events of the timer. Start timer when screen is initialized and stop timer when screen is closed.



+/-		⌥	[	Emitter	Signal	Job	Receiver
✕		+		ID_SCREEN_TimeWave	INITDIALOG	START	ID_TIMER_00
✕		+		ID_SCREEN_TimeWave	DELETE	STOP	ID_TIMER_00

Figure 1-98 Set Interaction – Start and Stop Timer

2) Calculate Pixel and Time Increment

When Job = CALC is set, calculation formula will be called to calculate the value of Receiver when trigger event occurs. The calculation formula is defined when variable is defined as shown in Figure 1-99.

Pixel increment between two points = X-axis pixels ÷ number of points = 400 ÷ 100 = 4 pixels

Time increment = refresh interval × number of points ÷ number of scale divisions = 60 s × 100 ÷ 5 = 20 min

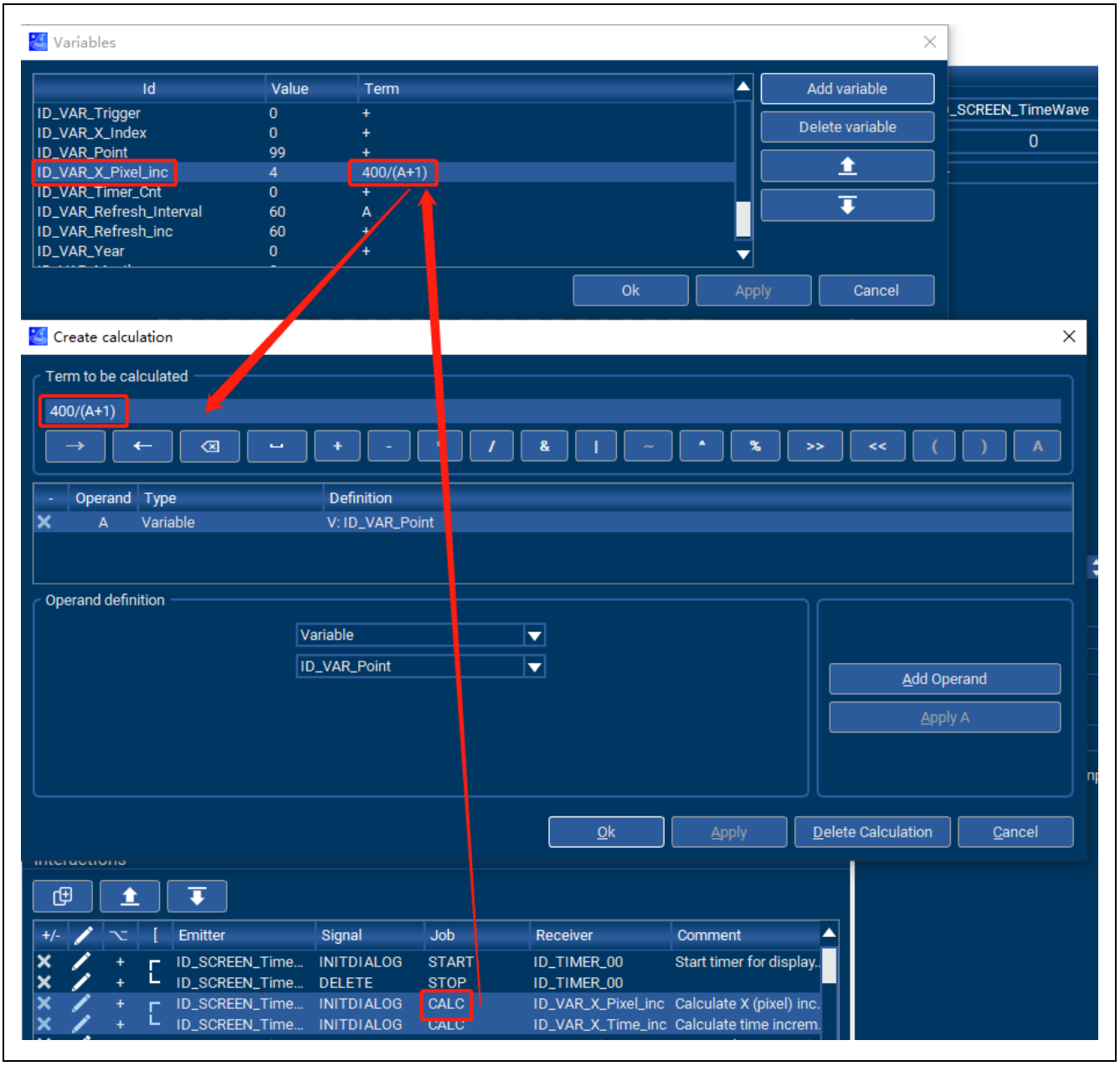


Figure 1-99 Set Interaction - Calculate Pixel and Time Increment

3) Initialize Display of X-Coordinate

Initially, only one point is displayed, with the first scale of x coordinate corresponding to current time. Convert the current time to seconds, calculate the time corresponding to each scale, then convert it into hours and minutes, and display them at position of the corresponding scale.

+/-		⌥	[	Emitter	Signal	Job	Receiver
✕		+		ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Time_In_Sec
✕		+	┌	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_VAR_Time_In_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_VAR_Time_In_Sec_Display
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Hour_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Min_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Hour0
✕		+	└	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Min0
✕		+	┌	ID_SCREEN_TimeWave	INITDIALOG	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Time_In_Sec_Display
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Hour_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Min_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Hour1
✕		+	└	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Min1
✕		+	┌	ID_SCREEN_TimeWave	INITDIALOG	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Time_In_Sec_Display
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Min_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Hour_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Hour2
✕		+	└	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Min2
✕		+	┌	ID_SCREEN_TimeWave	INITDIALOG	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Time_In_Sec_Display
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Min_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Hour_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Hour3
✕		+	└	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Min3
✕		+	┌	ID_SCREEN_TimeWave	INITDIALOG	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Time_In_Sec_Display
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Min_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Hour_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Hour4
✕		+	└	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Min4
✕		+	┌	ID_SCREEN_TimeWave	INITDIALOG	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Time_In_Sec_Display
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Sec_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Min_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	CALC	ID_VAR_Hour_Tmp
✕		+	│	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Hour5
✕		+	└	ID_SCREEN_TimeWave	INITDIALOG	SETVALUE	ID_TEXT_Min5

Figure 1-100 Set Interaction - Initialize the Display of X-Coordinate

4) Initialize X and Y Values

X and Y both refer to pixels. Initially, set X coordinate values of 100 points to 0, as shown in Figure 1-101. Set Y coordinate values of 100 points in A5 to 0, as shown in Figure 1-102, Y coordinate Values of A1~A4 are set in the same way.

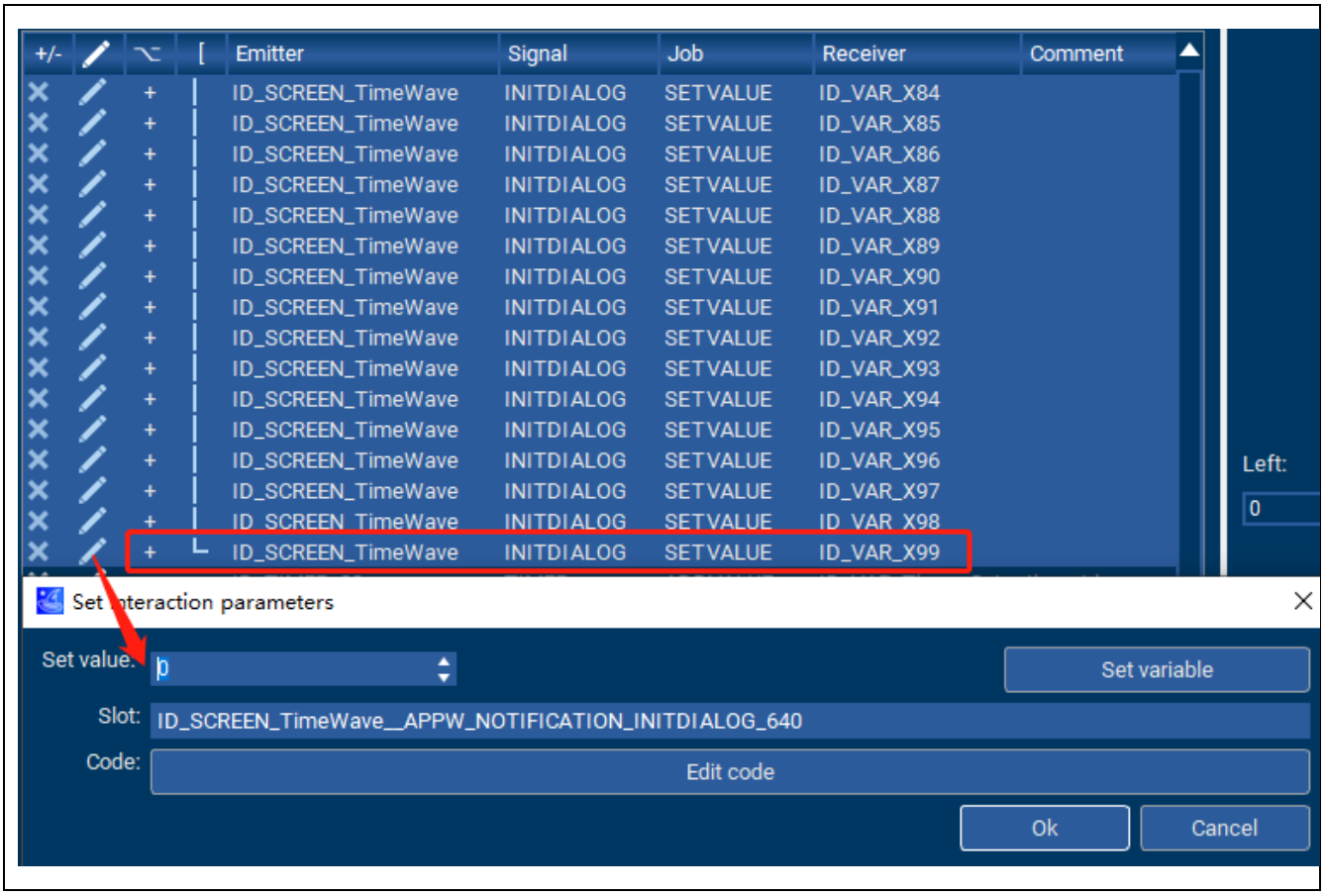


Figure 1-101 Set Interaction - Initialize X Value

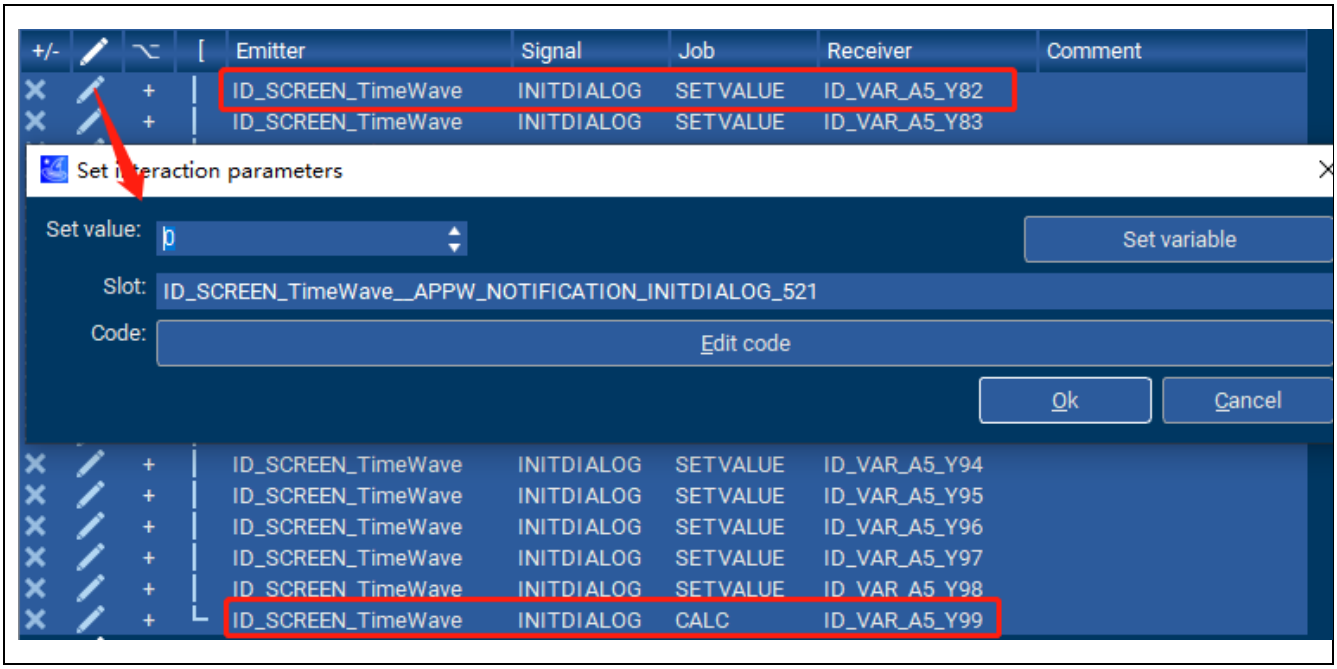


Figure 1-102 Set Interaction - Initialize Y Value

5) Generate Refresh Event

During initialization, a 1 s timer is set as beat timer. To achieve the purpose of refreshing once every 60 seconds, timer timeout event needs to be counted. When count reaches 60 times, a refresh event is generated. As shown in Figure 1-103, when timer event occurs, add 1 to ID_VAR_Timer_CNT. When ID_VAR_Timer_CNT is equal to 60, add 1 to ID_VAR_Trigger and clear ID_VAR_Timer_CNT count value. Then use the change of ID_VAR_Trigger value to trigger display refresh.

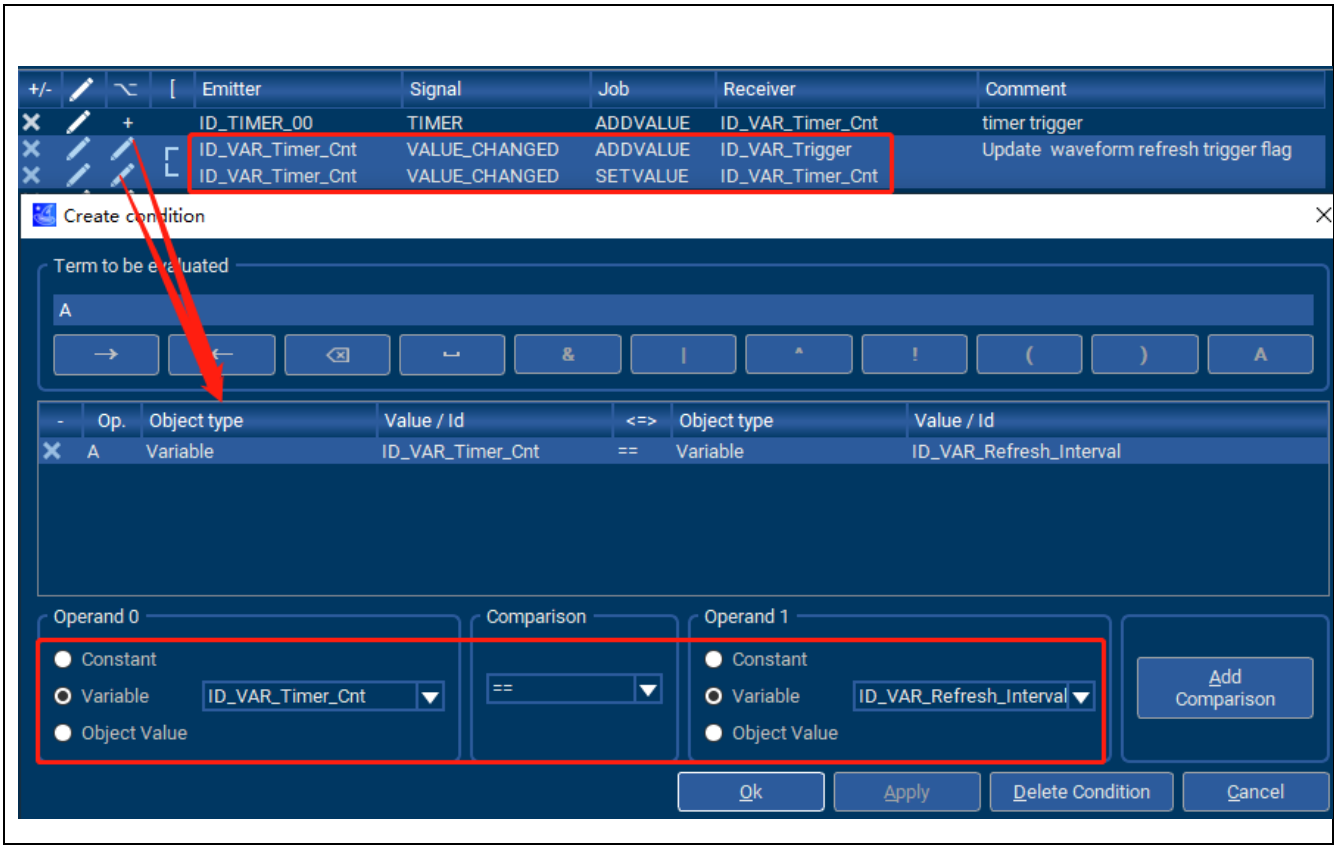


Figure 1-103 Set Interaction - Generate Waveform Refresh Event

6) Refresh Time Scale Display

After currently displayed points reach 100 points, time scale display needs to be refreshed every time a refresh event occurs. Read current time displayed at the last scale position. Subtract the time increment (20 minutes) shown at the second to last scale position, and so on.

+/-	Emitter	Signal	Job	Receiver
✕	ID_VAR_Timer_Cnt	VALUE_CHANGED	SETVALUE	ID_VAR_Timer_Cnt
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Display
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Hour_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Min_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour0
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Min0
✕	ID_VAR_Trigger	VALUE_CHANGED	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Display
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Hour_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Min_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour1
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Min1
✕	ID_VAR_Trigger	VALUE_CHANGED	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Display
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Hour_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Min_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour2
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Min2
✕	ID_VAR_Trigger	VALUE_CHANGED	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Display
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Hour_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Min_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour3
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Min3
✕	ID_VAR_Trigger	VALUE_CHANGED	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Display
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Hour_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Min_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour4
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Min4
✕	ID_VAR_Trigger	VALUE_CHANGED	ADDVALUE	ID_VAR_Time_In_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Time_In_Sec_Display
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Hour_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Min_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	CALC	ID_VAR_Sec_Tmp
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour5
✕	ID_VAR_Trigger	VALUE_CHANGED	SETVALUE	ID_TEXT_Min5

Figure 1-104 Set Interaction - Refresh Display of Time Scale

7) Refresh X and Y Values

Refresh X Values

When a refresh event occurs, if displayed points are less than 100, set Xn-1 = Xn, X99 = X99 + increment. If displayed points reach 100, X0~X99 remain unchanged, as shown in Figure 1-105.

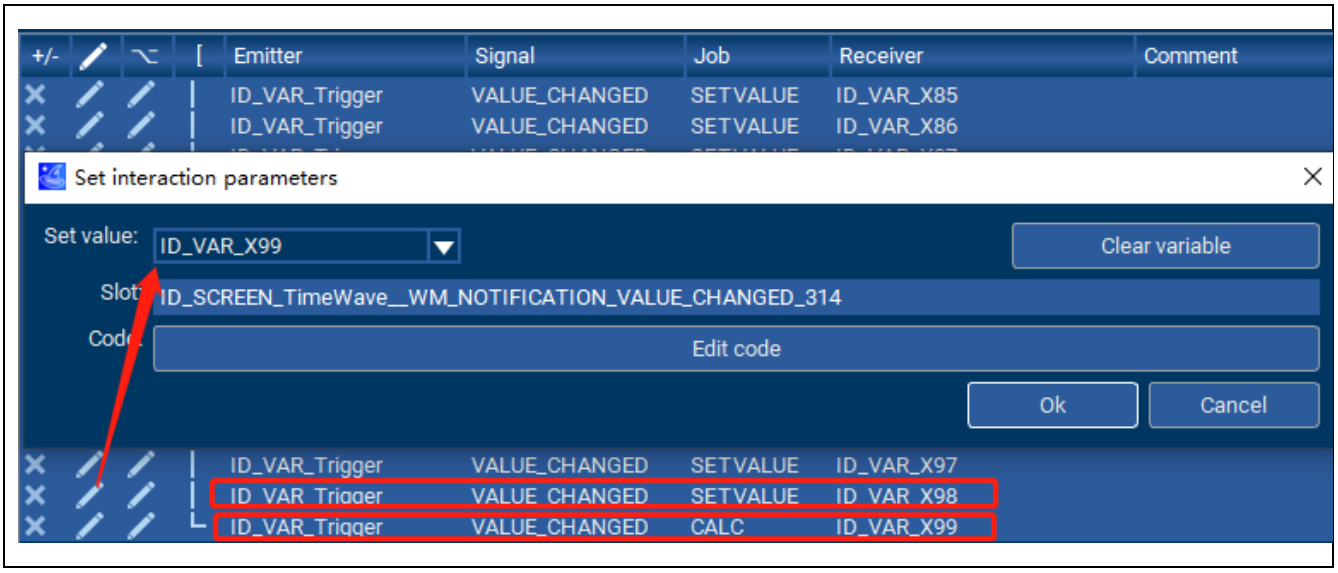


Figure 1-105 Set Interaction - Refresh X Values

Refresh Y Values

There are 5 groups of Y values, corresponding to A1~A5 areas. When refresh event occurs, set Yn-1 = Yn, then convert current area temperature into pixels and set it to Y99, as shown in Figure 1-106.

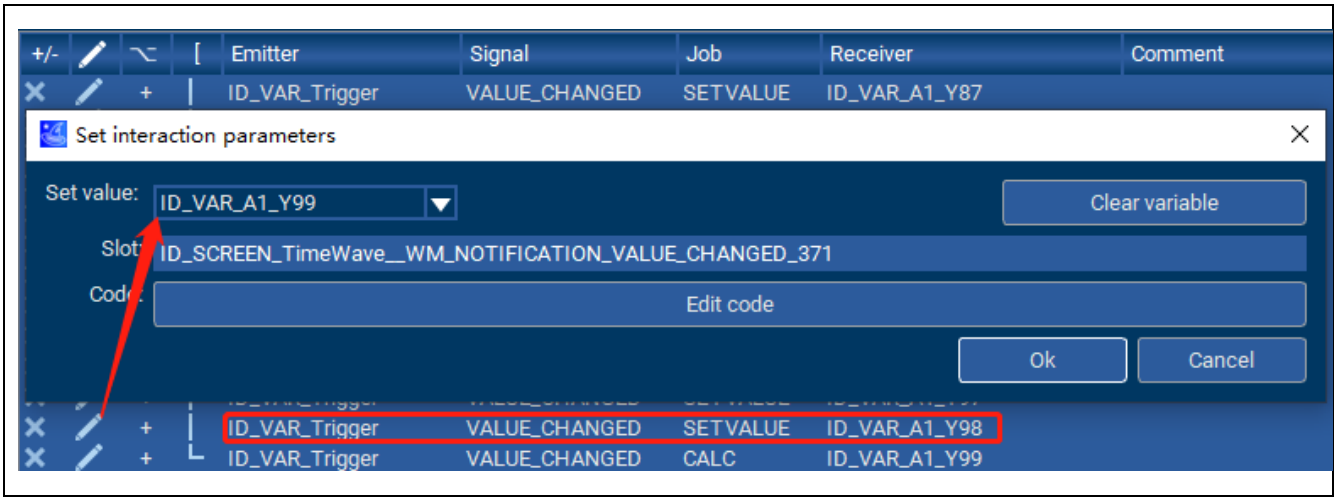


Figure 1-106 Set Interaction - Refresh Y Values

8) Update Points and Redraw Waveforms

When the number of points displayed is less than 100, the number needs to be recorded. When a refresh event occurs, Waveforms need to be redrawn, as shown in Figure 1-107.

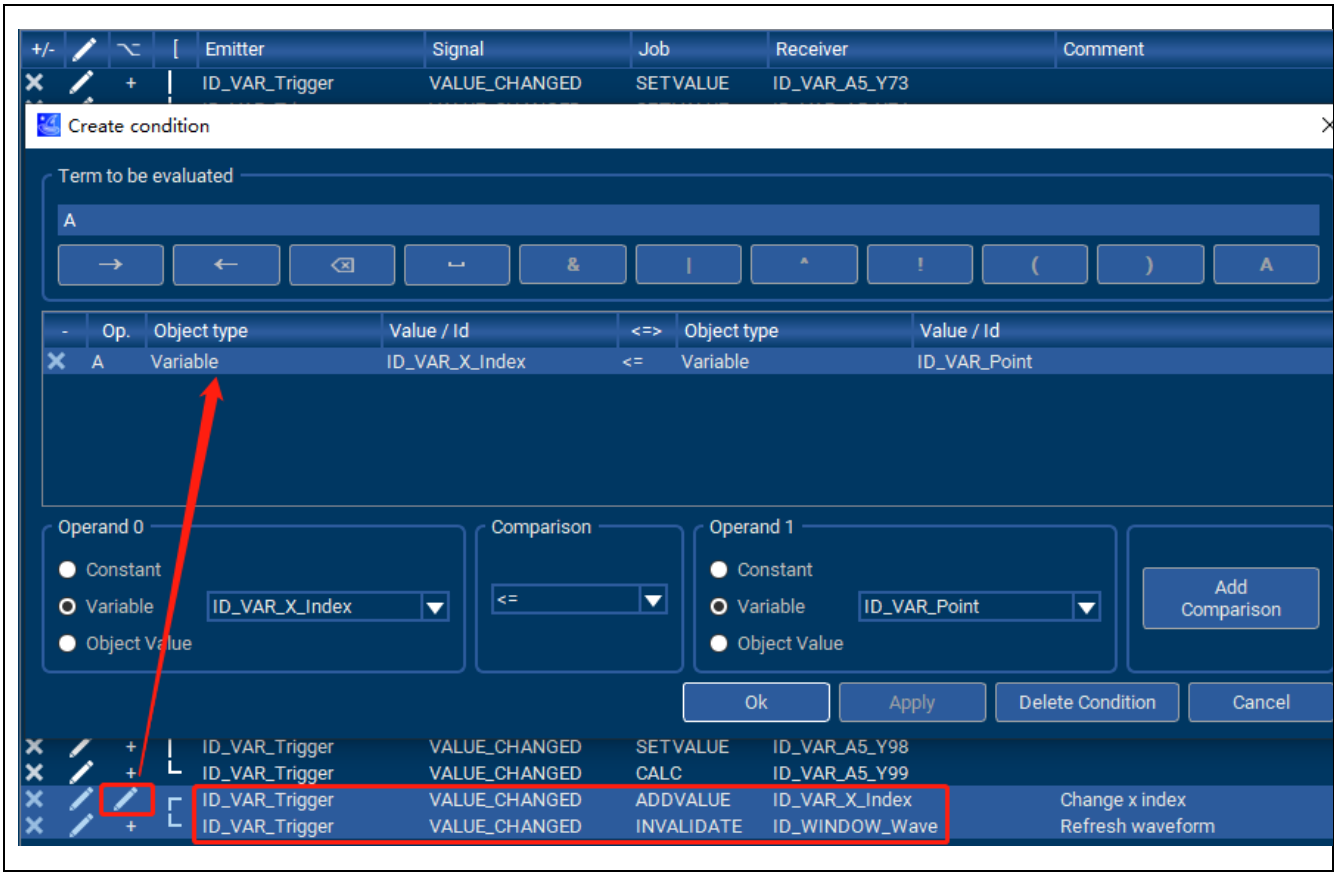
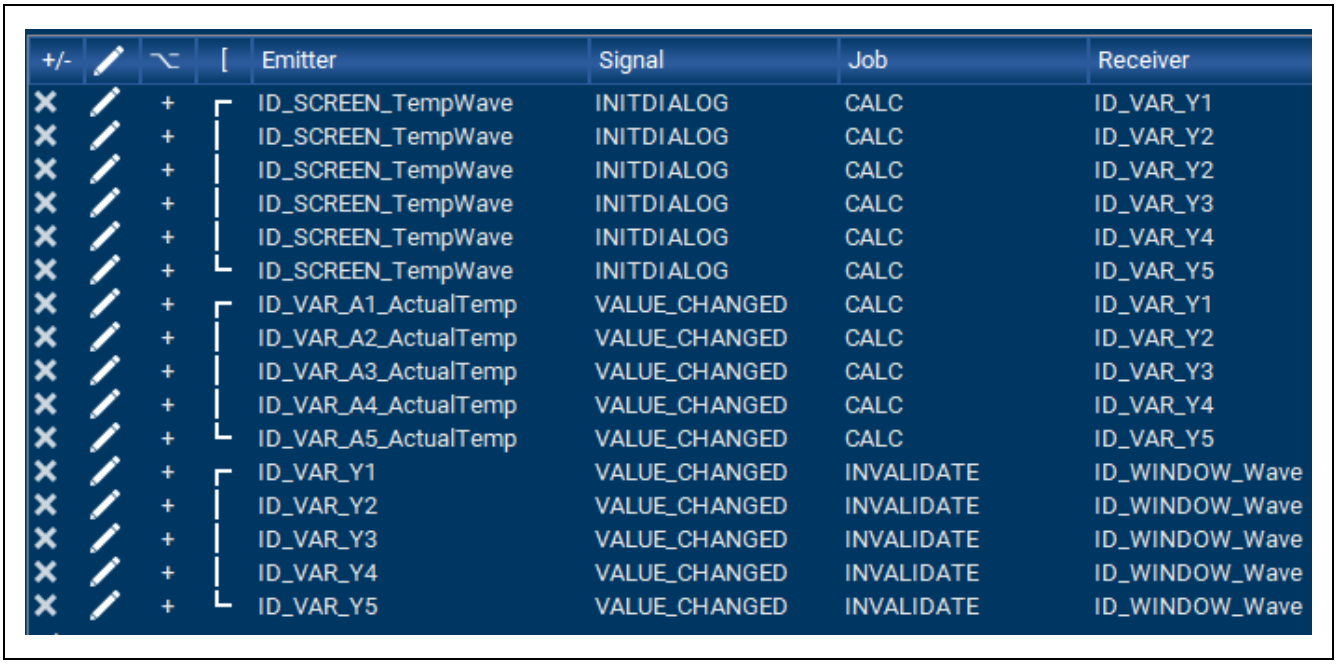


Figure 1-107 Set Interaction - Update Points and Redraw Waveforms

4. Set Interaction - Area-Temperature

Five X coordinate values in Area-Temperature waveform are fixed. When temperatures change, update Y values, redraw waveform.

During initial display, actual temperature of each area is converted into Y value. When actual temperature value changes, it triggers calculation of Y value. When Y value changes, it triggers redrawing of waveform. As shown in Figure 1-108.

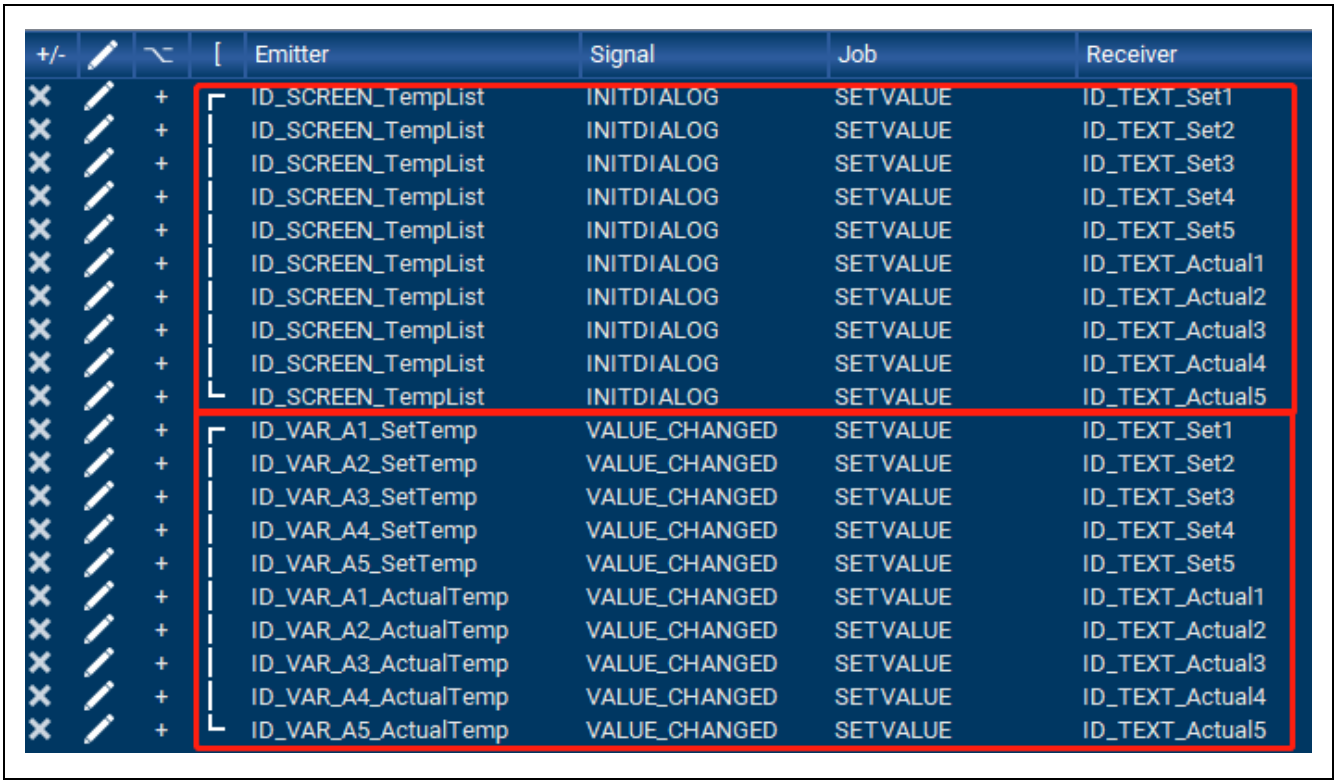


+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_TempWave	INITDIALOG	CALC	ID_VAR_Y1
+	ID_SCREEN_TempWave	INITDIALOG	CALC	ID_VAR_Y2
+	ID_SCREEN_TempWave	INITDIALOG	CALC	ID_VAR_Y2
+	ID_SCREEN_TempWave	INITDIALOG	CALC	ID_VAR_Y3
+	ID_SCREEN_TempWave	INITDIALOG	CALC	ID_VAR_Y4
+	ID_SCREEN_TempWave	INITDIALOG	CALC	ID_VAR_Y5
+	ID_VAR_A1_ActualTemp	VALUE_CHANGED	CALC	ID_VAR_Y1
+	ID_VAR_A2_ActualTemp	VALUE_CHANGED	CALC	ID_VAR_Y2
+	ID_VAR_A3_ActualTemp	VALUE_CHANGED	CALC	ID_VAR_Y3
+	ID_VAR_A4_ActualTemp	VALUE_CHANGED	CALC	ID_VAR_Y4
+	ID_VAR_A5_ActualTemp	VALUE_CHANGED	CALC	ID_VAR_Y5
+	ID_VAR_Y1	VALUE_CHANGED	INVALIDATE	ID_WINDOW_Wave
+	ID_VAR_Y2	VALUE_CHANGED	INVALIDATE	ID_WINDOW_Wave
+	ID_VAR_Y3	VALUE_CHANGED	INVALIDATE	ID_WINDOW_Wave
+	ID_VAR_Y4	VALUE_CHANGED	INVALIDATE	ID_WINDOW_Wave
+	ID_VAR_Y5	VALUE_CHANGED	INVALIDATE	ID_WINDOW_Wave

Figure 1-108 Set Interaction - Area-Temperature Waveform

5. Set Interaction – Temperature List

Temperature List screen is used to display set and actual temperature of 5 areas. Display values of the set and actual temperature variables in Text. Display is updated when the variables of actual temperature change. As shown in Figure 1-109.



+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Set1
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Set2
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Set3
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Set4
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Set5
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Actual1
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Actual2
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Actual3
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Actual4
+	ID_SCREEN_TempList	INITDIALOG	SETVALUE	ID_TEXT_Actual5
+	ID_VAR_A1_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set1
+	ID_VAR_A2_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set2
+	ID_VAR_A3_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set3
+	ID_VAR_A4_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set4
+	ID_VAR_A5_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set5
+	ID_VAR_A1_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual1
+	ID_VAR_A2_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual2
+	ID_VAR_A3_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual3
+	ID_VAR_A4_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual4
+	ID_VAR_A5_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual5

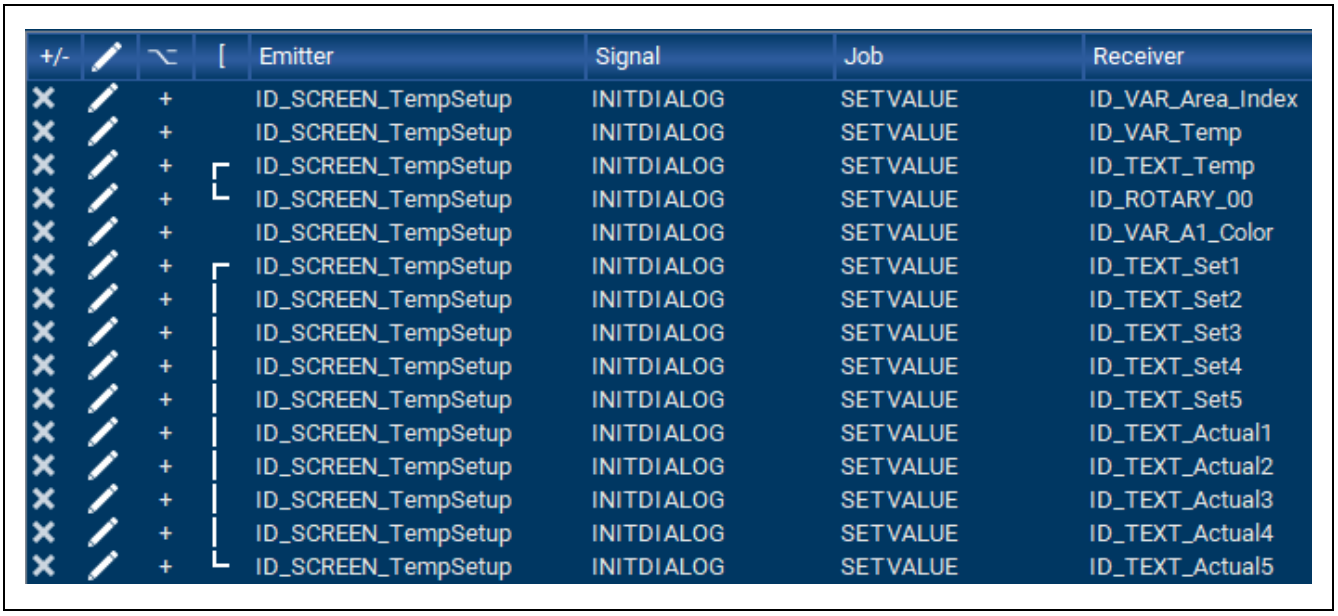
Figure 1-109 Set Interaction - Temperature List Screen

6. Set Interaction - Temperature Setting Screen

In addition to displaying temperature list, Temperature Setting screen also adds area selection. When an area is selected, the selected area is highlighted, and set temperature of the selected area is displayed in Rotary and Text box at the center of Rotary. When the Rotary is rotated, Text box and temperature list change with Rotary.

1) Initialize Display

During initial display, default selected area is set to area 1. BOX background and text color are used to indicate highlighting. Set actual and set temperatures of five areas to Texts in the list. The set temperature of area 1 is displayed in Rotary and the Text box in the Rotary. As shown in Figure 1-110.

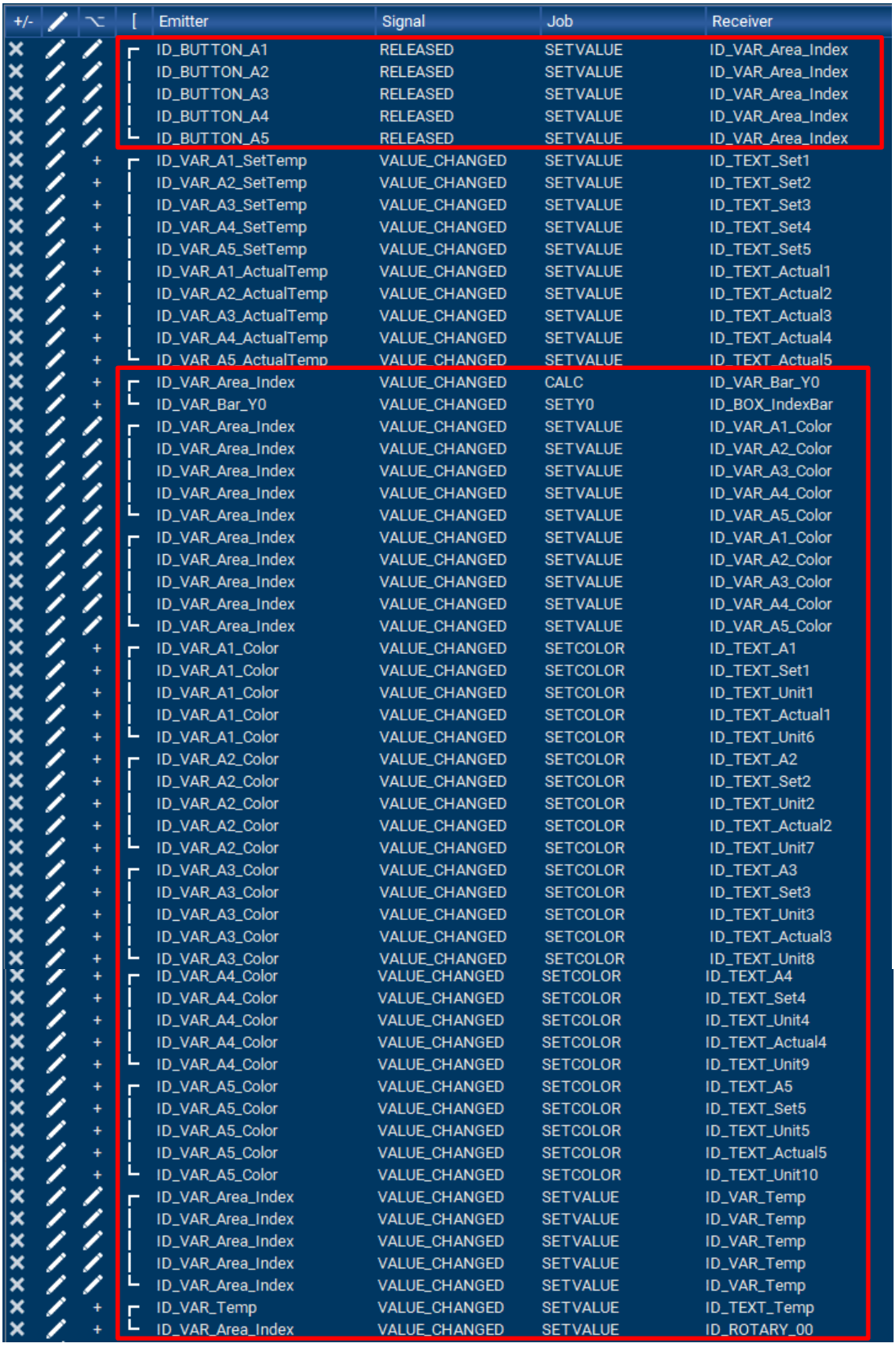


+/-		⌥	[	Emitter	Signal	Job	Receiver
✕		+		ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_VAR_Area_Index
✕		+		ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_VAR_Temp
✕		+	┌	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Temp
✕		+	└	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_ROTARY_00
✕		+		ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_VAR_A1_Color
✕		+	┌	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Set1
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Set2
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Set3
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Set4
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Set5
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Actual1
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Actual2
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Actual3
✕		+	│	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Actual4
✕		+	└	ID_SCREEN_TempSetup	INITDIALOG	SETVALUE	ID_TEXT_Actual5

Figure 1-110 Set Interaction - Initialize Temperature Setting Screen

2) Set Interaction - Area Selection Button

When an area button is pressed, it indicates that the area is selected. Need to set index variable of the selected area, coordinates of the highlighted box, text color of the selected area to yellow and the other areas to white. Then assign set temperature of the selected area to Rotary and Text in the center of Rotary. As shown in Figure 1-111.

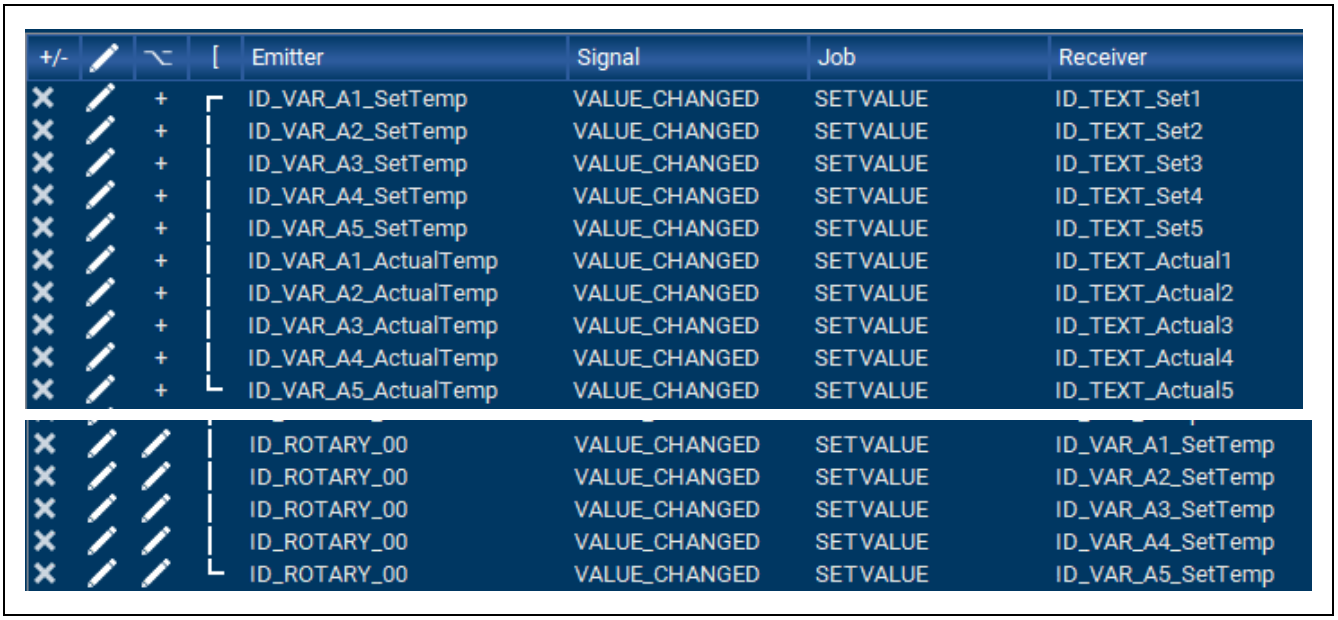


Emitter	Signal	Job	Receiver
ID_BUTTON_A1	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A2	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A3	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A4	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A5	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_VAR_A1_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set1
ID_VAR_A2_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set2
ID_VAR_A3_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set3
ID_VAR_A4_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set4
ID_VAR_A5_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set5
ID_VAR_A1_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual1
ID_VAR_A2_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual2
ID_VAR_A3_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual3
ID_VAR_A4_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual4
ID_VAR_A5_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual5
ID_VAR_Area_Index	VALUE_CHANGED	CALC	ID_VAR_Bar_Y0
ID_VAR_Bar_Y0	VALUE_CHANGED	SETY0	ID_BOX_IndexBar
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A1_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A2_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A3_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A4_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A5_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A1_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A2_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A3_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A4_Color
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_A5_Color
ID_VAR_A1_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_A1
ID_VAR_A1_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Set1
ID_VAR_A1_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit1
ID_VAR_A1_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Actual1
ID_VAR_A1_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit6
ID_VAR_A2_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_A2
ID_VAR_A2_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Set2
ID_VAR_A2_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit2
ID_VAR_A2_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Actual2
ID_VAR_A2_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit7
ID_VAR_A3_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_A3
ID_VAR_A3_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Set3
ID_VAR_A3_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit3
ID_VAR_A3_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Actual3
ID_VAR_A3_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit8
ID_VAR_A4_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_A4
ID_VAR_A4_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Set4
ID_VAR_A4_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit4
ID_VAR_A4_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Actual4
ID_VAR_A4_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit9
ID_VAR_A5_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_A5
ID_VAR_A5_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Set5
ID_VAR_A5_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit5
ID_VAR_A5_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Actual5
ID_VAR_A5_Color	VALUE_CHANGED	SETCOLOR	ID_TEXT_Unit10
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_Temp
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_Temp
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_Temp
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_Temp
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_VAR_Temp
ID_VAR_Temp	VALUE_CHANGED	SETVALUE	ID_TEXT_Temp
ID_VAR_Area_Index	VALUE_CHANGED	SETVALUE	ID_ROTARY_00

Figure 1-111 Set Interaction - Area Button

3) Set Interaction - Rotary

When rotating the Rotary, set temperature in the list and Text display content in the center of Rotary need to be changed with Rotary. As shown in Figure 1-112.



+/-	Emitter	Signal	Job	Receiver
+	ID_VAR_A1_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set1
+	ID_VAR_A2_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set2
+	ID_VAR_A3_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set3
+	ID_VAR_A4_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set4
+	ID_VAR_A5_SetTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Set5
+	ID_VAR_A1_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual1
+	ID_VAR_A2_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual2
+	ID_VAR_A3_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual3
+	ID_VAR_A4_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual4
+	ID_VAR_A5_ActualTemp	VALUE_CHANGED	SETVALUE	ID_TEXT_Actual5
	ID_ROTARY_00	VALUE_CHANGED	SETVALUE	ID_VAR_A1_SetTemp
	ID_ROTARY_00	VALUE_CHANGED	SETVALUE	ID_VAR_A2_SetTemp
	ID_ROTARY_00	VALUE_CHANGED	SETVALUE	ID_VAR_A3_SetTemp
	ID_ROTARY_00	VALUE_CHANGED	SETVALUE	ID_VAR_A4_SetTemp
	ID_ROTARY_00	VALUE_CHANGED	SETVALUE	ID_VAR_A5_SetTemp

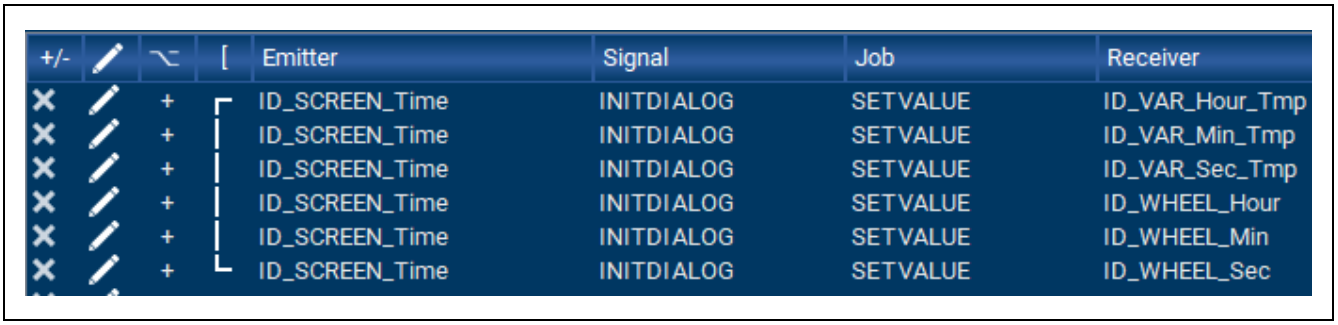
Figure 1-112 Set Interaction - Rotary

7. Set Interaction – Time Setting Screen

Time Setting screen needs to realize functions of displaying and modifying current time, saving set time, exiting current screen.

1) Initialize Display

When displaying screen, current time needs to be displayed in Wheel. When initializing screen, set actual time to Wheel. As shown in Figure 1-113.

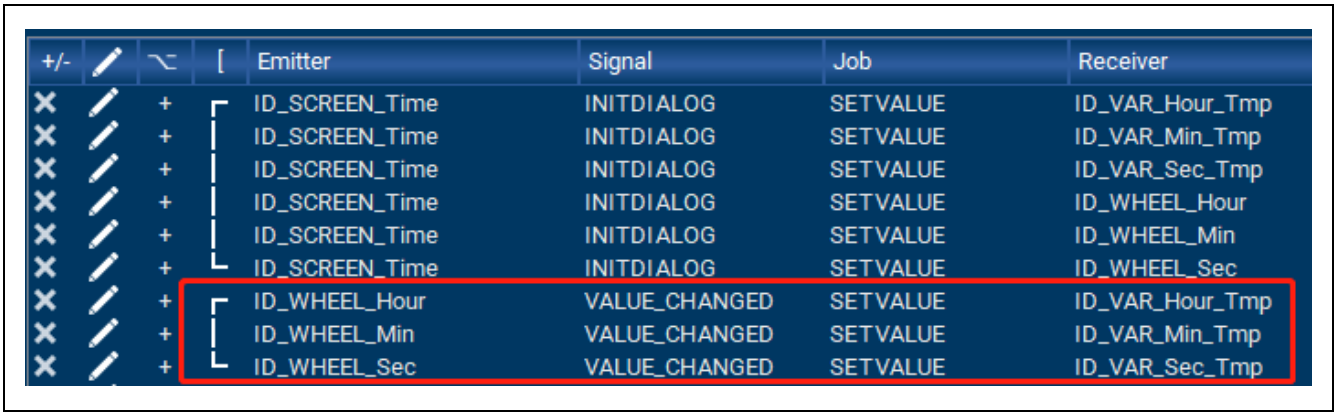


+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Hour_Tmp
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Min_Tmp
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Sec_Tmp
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Hour
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Min
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Sec

Figure 1-113 Set Interaction – Initial value of Wheel

2) Set Time in Wheel

When hours, minutes, and seconds are selected in Wheel, the selected time is saved in a temporary variable. As shown in Figure 1-114.

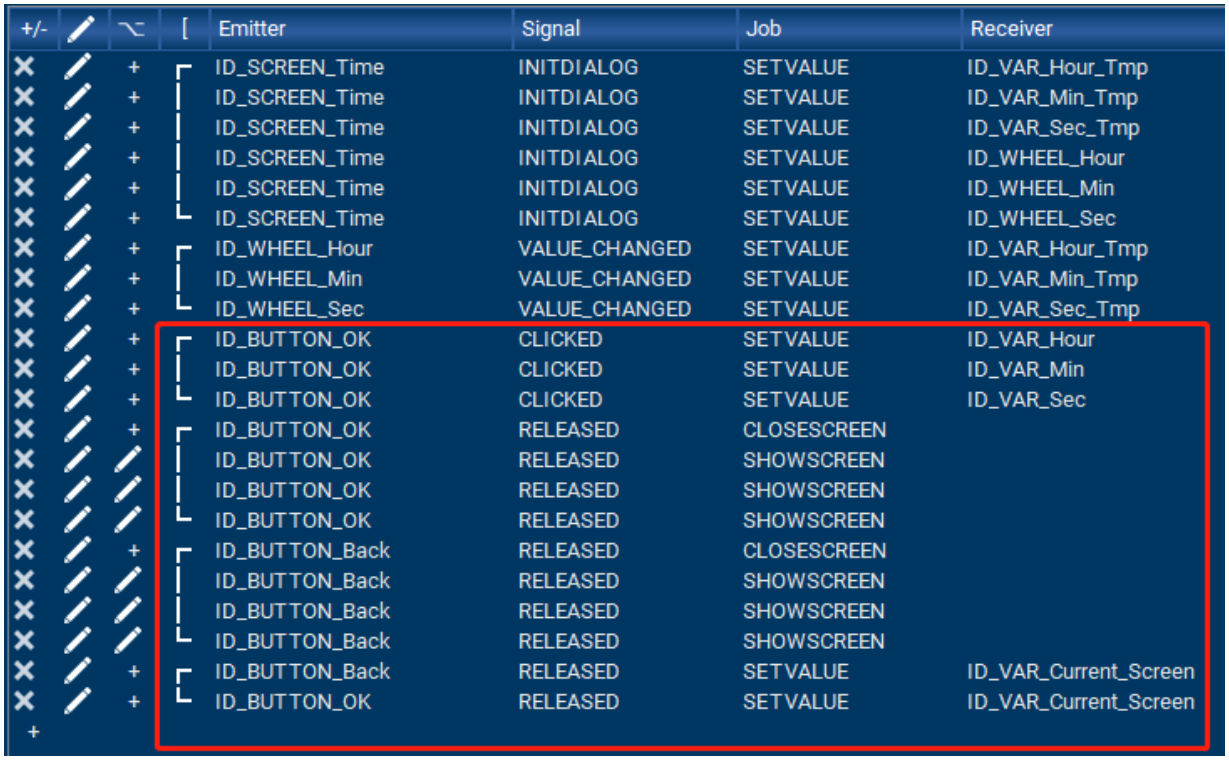


Emitter	Signal	Job	Receiver
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Hour_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Min_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Sec_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Hour
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Min
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Sec
ID_WHEEL_Hour	VALUE_CHANGED	SETVALUE	ID_VAR_Hour_Tmp
ID_WHEEL_Min	VALUE_CHANGED	SETVALUE	ID_VAR_Min_Tmp
ID_WHEEL_Sec	VALUE_CHANGED	SETVALUE	ID_VAR_Sec_Tmp

Figure 1-114 Set Interaction - Save The Set Time to a Temporary Variable

3) OK and Return Buttons

When OK button is pressed, set time will be saved, and the current screen will be exited. Temperature List, Time-Temperature waveform or Area-Temperature waveform screen will be displayed according to index. When return button is pressed, the set time is not saved, and other processing is the same as OK button. As shown in Figure 1-115.

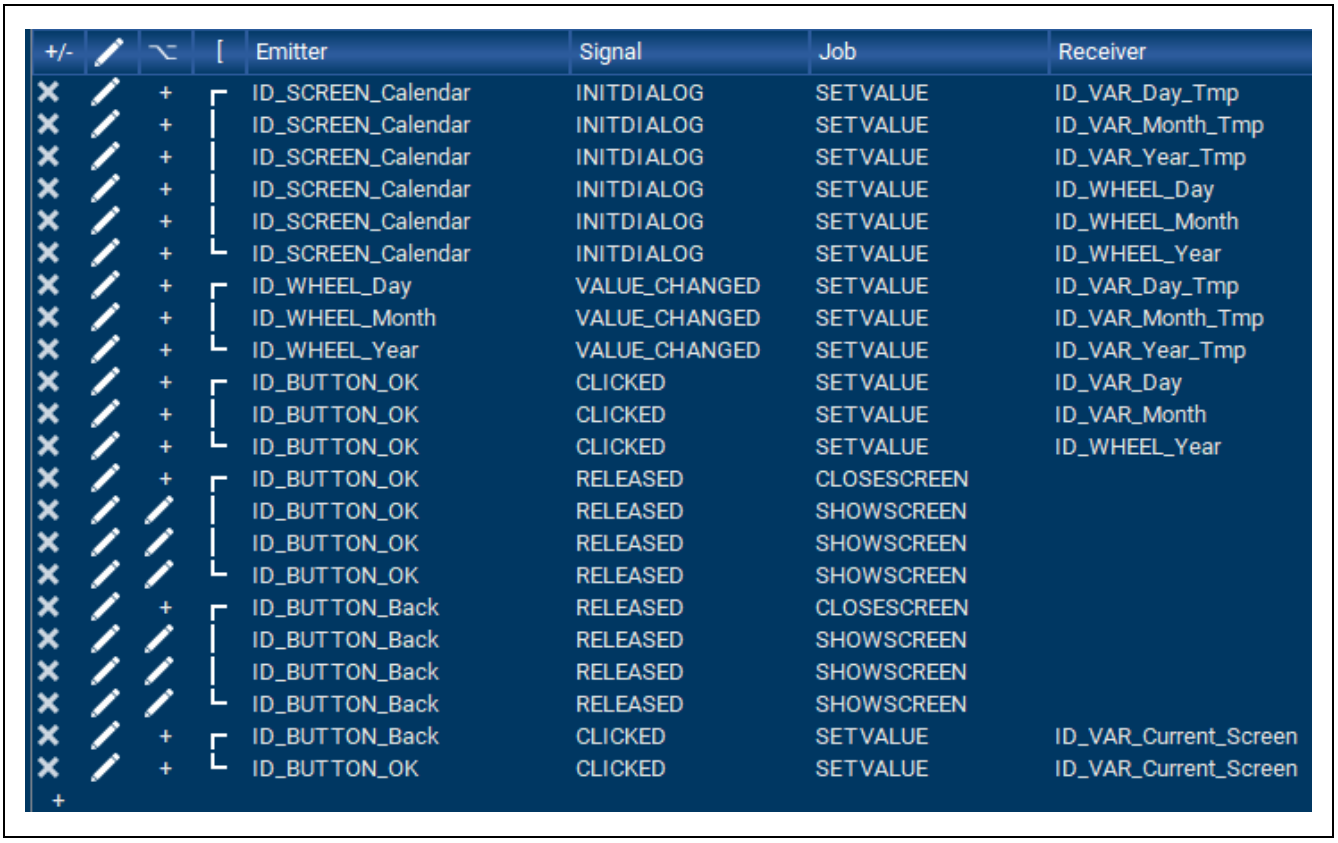


Emitter	Signal	Job	Receiver
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Hour_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Min_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Sec_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Hour
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Min
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Sec
ID_WHEEL_Hour	VALUE_CHANGED	SETVALUE	ID_VAR_Hour_Tmp
ID_WHEEL_Min	VALUE_CHANGED	SETVALUE	ID_VAR_Min_Tmp
ID_WHEEL_Sec	VALUE_CHANGED	SETVALUE	ID_VAR_Sec_Tmp
ID_BUTTON_OK	CLICKED	SETVALUE	ID_VAR_Hour
ID_BUTTON_OK	CLICKED	SETVALUE	ID_VAR_Min
ID_BUTTON_OK	CLICKED	SETVALUE	ID_VAR_Sec
ID_BUTTON_OK	RELEASED	CLOSESCREEN	
ID_BUTTON_OK	RELEASED	SHOWSCREEN	
ID_BUTTON_OK	RELEASED	SHOWSCREEN	
ID_BUTTON_OK	RELEASED	SHOWSCREEN	
ID_BUTTON_Back	RELEASED	CLOSESCREEN	
ID_BUTTON_Back	RELEASED	SHOWSCREEN	
ID_BUTTON_Back	RELEASED	SHOWSCREEN	
ID_BUTTON_Back	RELEASED	SHOWSCREEN	
ID_BUTTON_Back	RELEASED	SETVALUE	ID_VAR_Current_Screen
ID_BUTTON_OK	RELEASED	SETVALUE	ID_VAR_Current_Screen

Figure 1-115 Set Interaction - OK and Return Buttons

8. Set Interaction - Date Setting Screen

Date setting screen needs to realize functions of displaying and modifying current date, saving the set date, and exiting current display screen. The implementation method is the same as Time Setting Screen. As shown in Figure 1-116.



+/-		⌥	[	Emitter	Signal	Job	Receiver
		+	┌	ID_SCREEN_Calendar	INITDIALOG	SETVALUE	ID_VAR_Day_Tmp
		+	│	ID_SCREEN_Calendar	INITDIALOG	SETVALUE	ID_VAR_Month_Tmp
		+	│	ID_SCREEN_Calendar	INITDIALOG	SETVALUE	ID_VAR_Year_Tmp
		+	│	ID_SCREEN_Calendar	INITDIALOG	SETVALUE	ID_WHEEL_Day
		+	│	ID_SCREEN_Calendar	INITDIALOG	SETVALUE	ID_WHEEL_Month
		+	└	ID_SCREEN_Calendar	INITDIALOG	SETVALUE	ID_WHEEL_Year
		+	┌	ID_WHEEL_Day	VALUE_CHANGED	SETVALUE	ID_VAR_Day_Tmp
		+	│	ID_WHEEL_Month	VALUE_CHANGED	SETVALUE	ID_VAR_Month_Tmp
		+	└	ID_WHEEL_Year	VALUE_CHANGED	SETVALUE	ID_VAR_Year_Tmp
		+	┌	ID_BUTTON_OK	CLICKED	SETVALUE	ID_VAR_Day
		+	│	ID_BUTTON_OK	CLICKED	SETVALUE	ID_VAR_Month
		+	└	ID_BUTTON_OK	CLICKED	SETVALUE	ID_WHEEL_Year
		+	┌	ID_BUTTON_OK	RELEASED	CLOSESCREEN	
			│	ID_BUTTON_OK	RELEASED	SHOWSCREEN	
			│	ID_BUTTON_OK	RELEASED	SHOWSCREEN	
			└	ID_BUTTON_OK	RELEASED	SHOWSCREEN	
		+	┌	ID_BUTTON_Back	RELEASED	CLOSESCREEN	
			│	ID_BUTTON_Back	RELEASED	SHOWSCREEN	
			│	ID_BUTTON_Back	RELEASED	SHOWSCREEN	
			└	ID_BUTTON_Back	RELEASED	SHOWSCREEN	
		+	┌	ID_BUTTON_Back	CLICKED	SETVALUE	ID_VAR_Current_Screen
		+	└	ID_BUTTON_OK	CLICKED	SETVALUE	ID_VAR_Current_Screen
+							

Figure 1-116 Set Interaction – Date Setting Screen

1.2.2.17 Date Exchange Between MCU and AppWizard

1. To transfer temperatures sampled in MCU and the current time to AppWizard, Variable needs to be used. Set values to Variables in code, and then use Interaction function to set Variable values to Objects used for display, such as Text and Wheel. The process is shown in Figure 1-117.

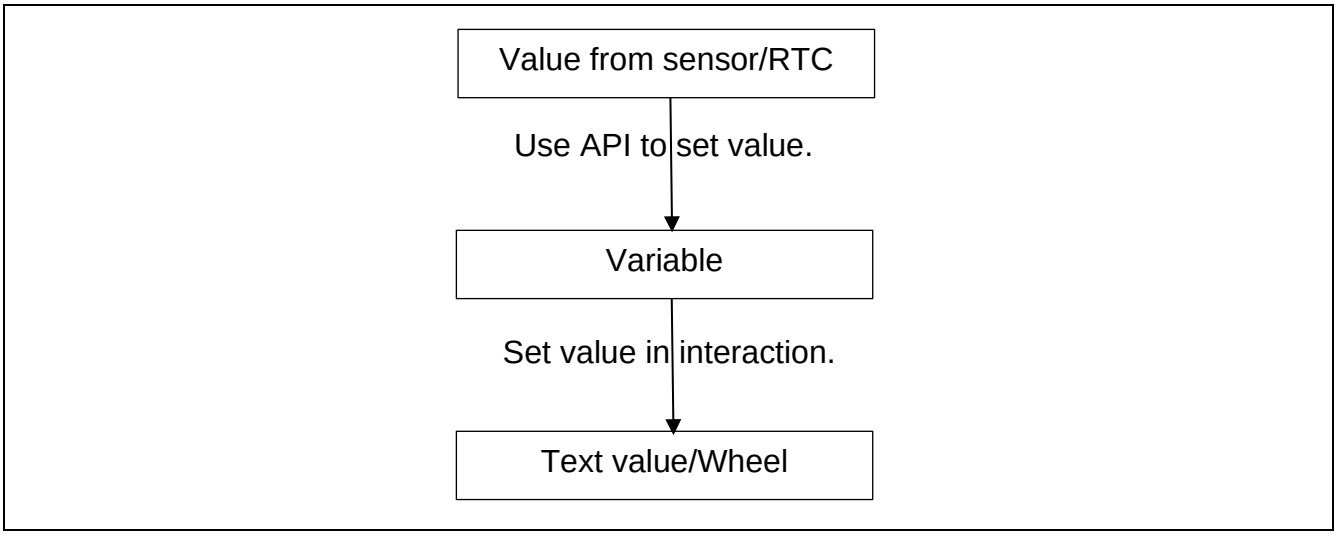


Figure 1-117 Pass Temperature and Time from MCU to AppWizard and Display

2. To pass date and time set by AppWizard to MCU, values of Wheel need to be set to Variables. Then read values of Variable in code to set RTC. The process is shown in Figure 1-118. For the definition of variables, please refer to 1.2.2.6. For the settings of Interaction, please refer to Figure 1-115 and Figure 1-116. For the code, please refer to Code Description.

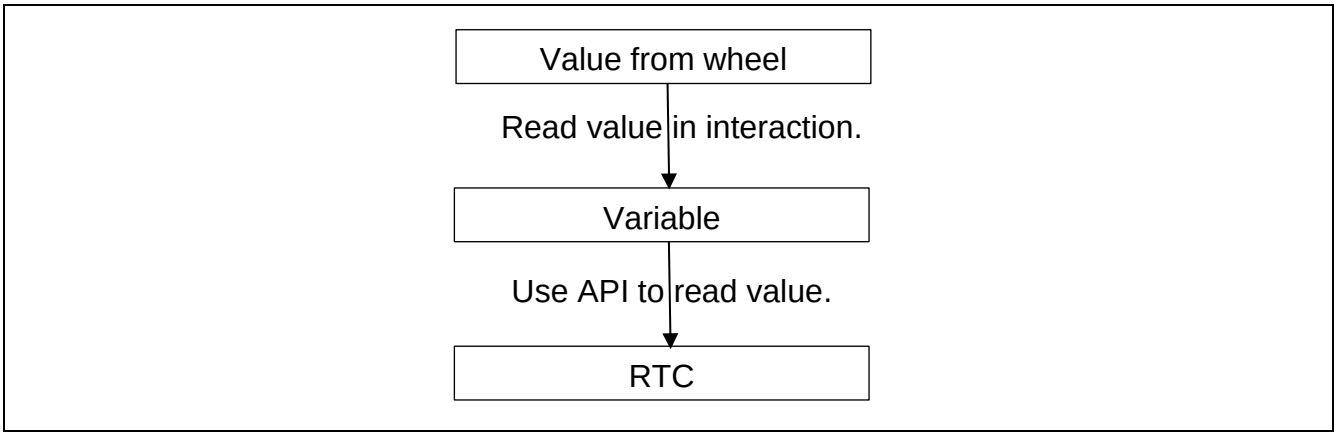


Figure 1-118 Set Time Set in AppWizard to RTC of MCU

3. The variables are defined in Chapter 1.2.2.6, and the variables are shown in Figure 1-119.

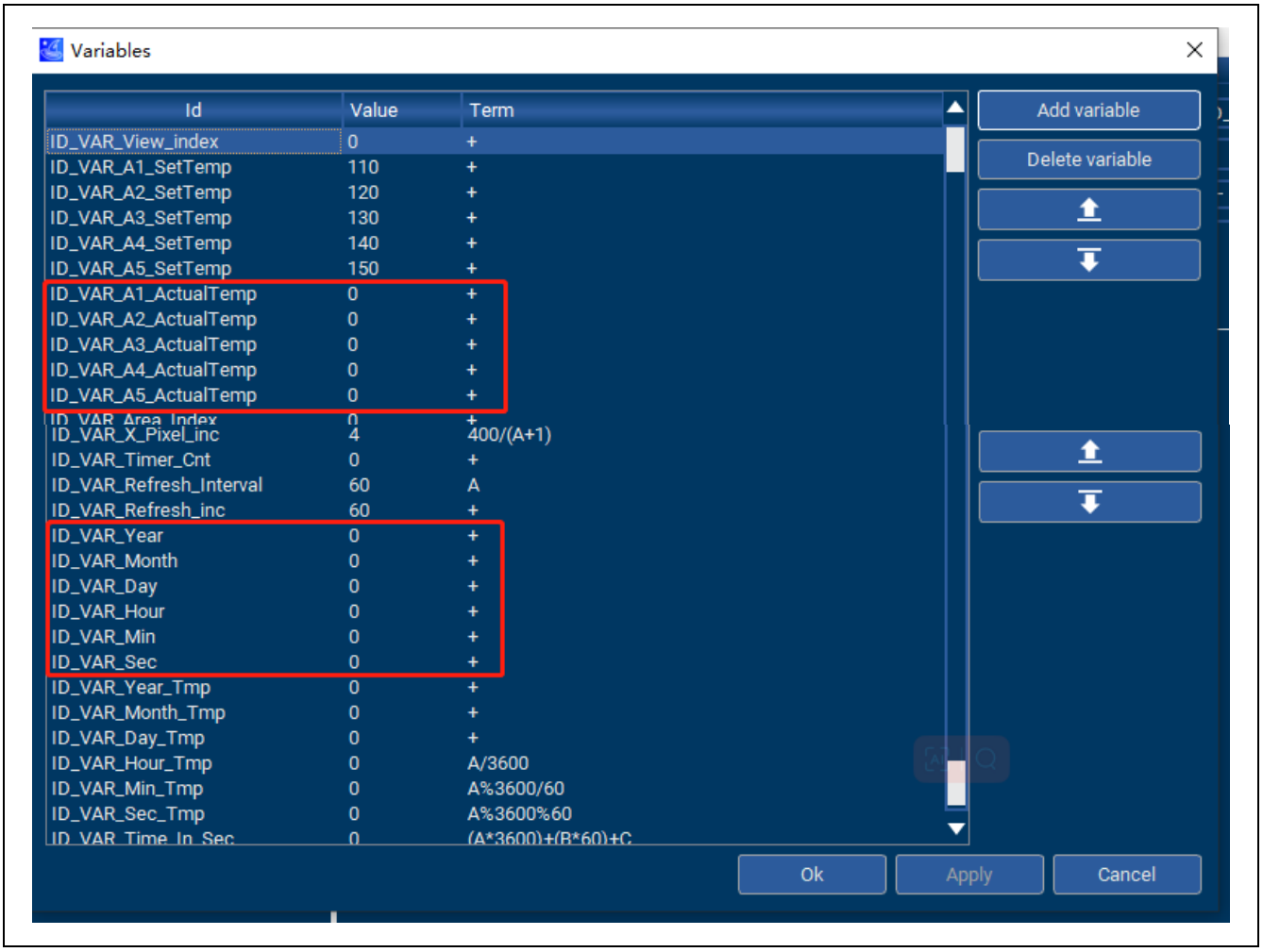


Figure 1-119 Variable Definition of Parameter Exchange Between MCU and AppWizard

4. The settings of Interaction are shown in Figure 1-120. They are used in many places and cannot be listed one by one. Please refer to section 1.2.2.16 for interactive settings of Temperature List screen, Temperature Setting screen, Time-Temperature waveform screen, Area-Temperature waveform screen, Time Setting screen, and Date Setting screen.

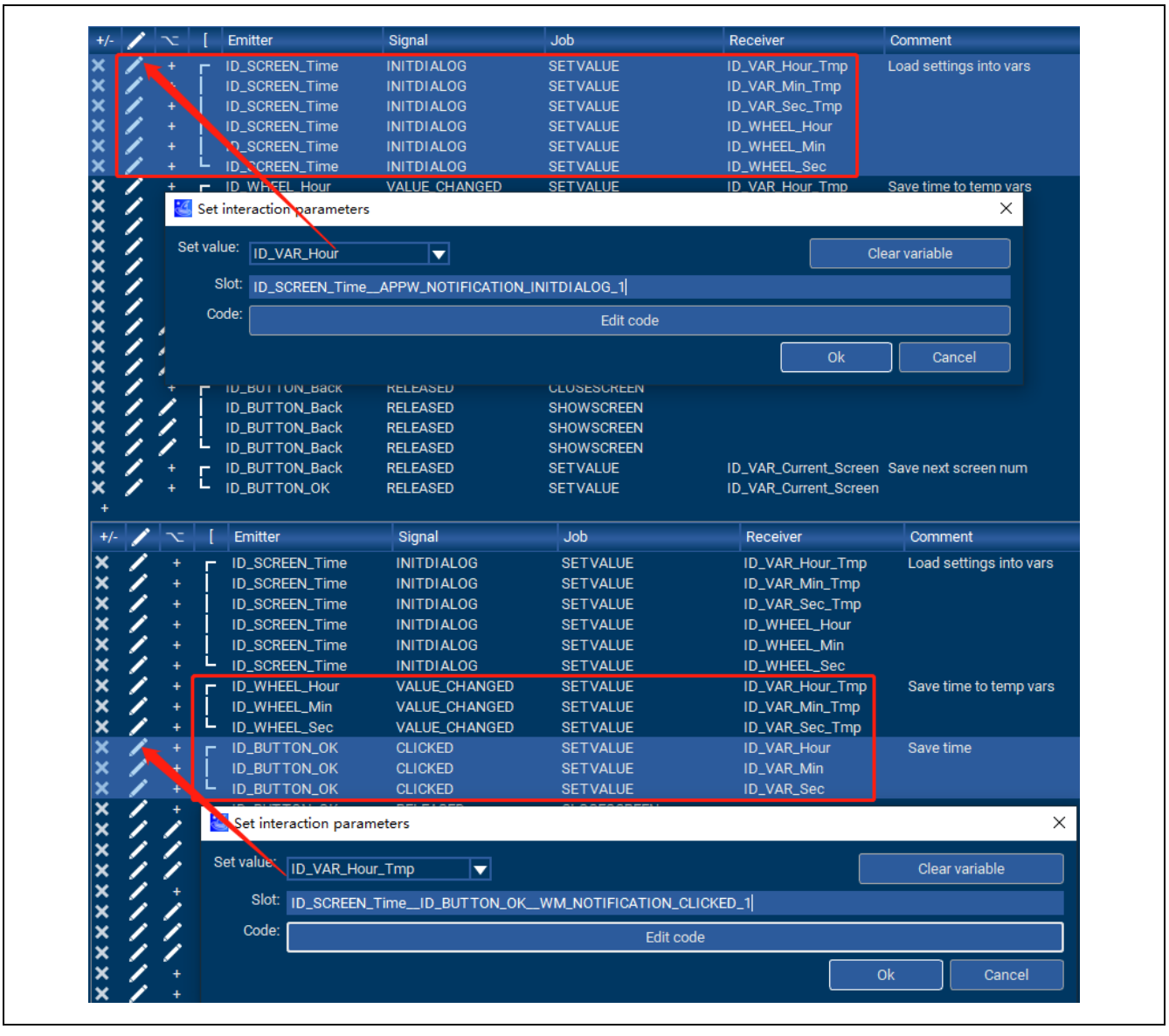


Figure 1-120 Setting Interaction - Variable

5. For code, refer to Code Description.

1.2.3 Example Code Editing

1. Export example code.

After the design is completed, click “File” -> “Export&Save” to output the example code, such as Figure1-121.

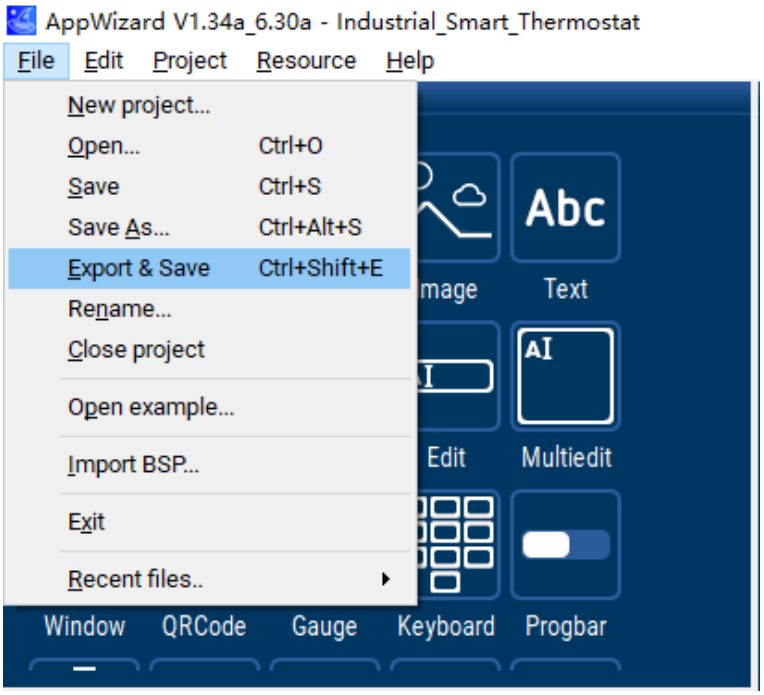


Figure1-121 Output Example Code

2. Copy exported code.

Click “Project” -> “Open containing folder” to open AppWizard project folder, copy Resource and Source folders, and overwrite original content in path EC_M467_AppWizard_ISMThermostat_V1.00\SampleCode\M467_AppWizard_ISMT\Application. The steps are shown in Figure1-122 and Figure1-123.

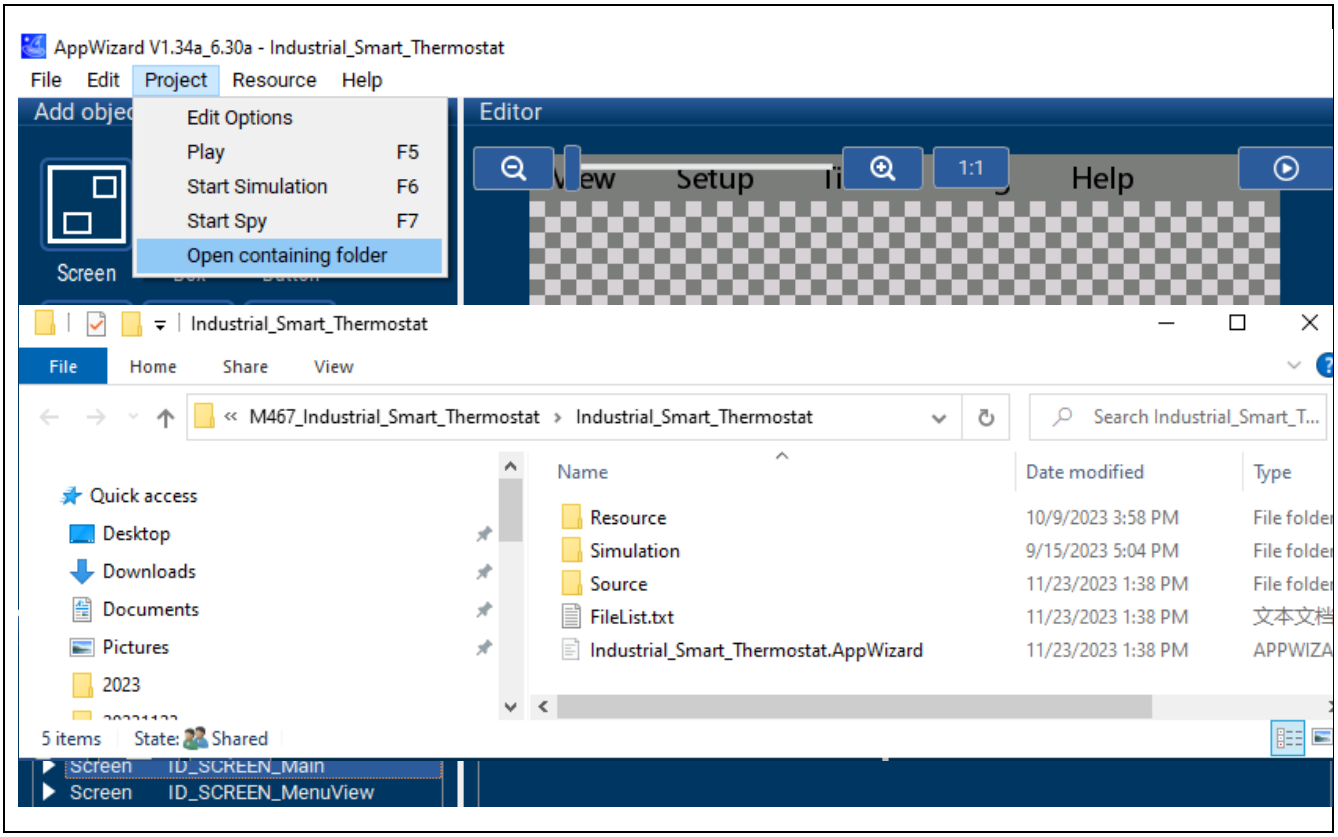


Figure1-122 Open AppWizard Project Folder

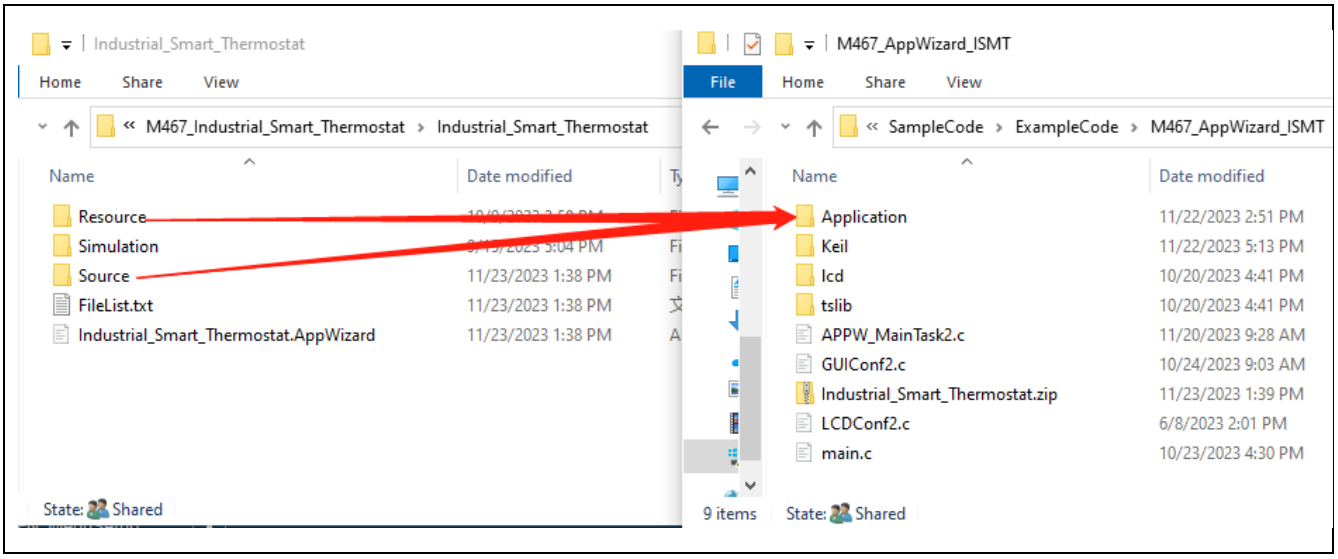


Figure1-123 Copy to Example Code Project

3. Import code generated by AppWizard into Keil project.

Open Keil project, right-click on Application, select Add Existing Files to Group “Application”, find Application folder, and load all *.c files in Resource and Source folders into the project.

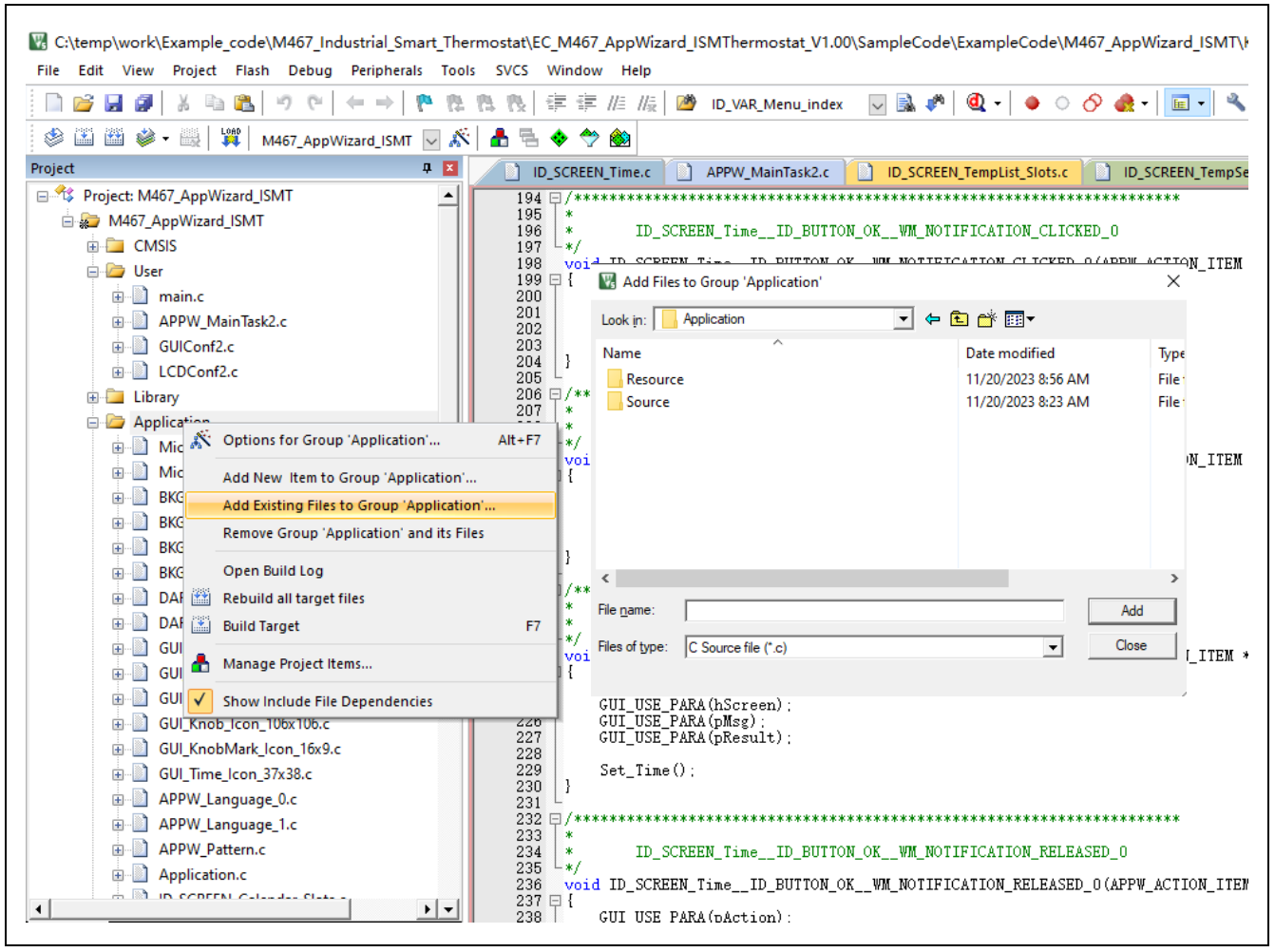


Figure1-124 Import Generated Code into Keil Project

4. Add code to Interact with AppWizard.

In application of industrial smart thermostat, time displayed on GUI comes from RTC of MCU. Temperatures are sampled by ADC of MCU and then passed to AppWizard for display. The time set on GUI also needs to be written to RTC to take effect. Therefore, the program needs to add ADC sampling and RTC reading and writing related codes, MCU transmits temperature and current time to AppWizard, and AppWizard transmits set time and date to MCU. The corresponding functions are defined in Application.c and declared in Application.h. The code for interaction between AppWizard and MCU involves MainTask2.c, ID_SCREEN_Date_Slots.c and ID_SCREEN_Time_Slots.c files. After AppWizard re-exports source code, when copying Resource and Source folders to Keil Project, be careful not to overwrite user code in Application.c, Application.h, ID_SCREEN_Date_Slots.c and ID_SCREEN_Time_Slots.c files.

```
void MainTask2(void)
{
    /* Setup configuration dependent pointers */
    APPW_X_Setup();

    /* Initialize AppWizard */
    APPW_Init(APPW_PROJECT_PATH);


    //    /* Create all persistent screens except initial screen */
    //    APPW_CreatePersistentScreens();

    /* Create initial screen... */
    //
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);

    /* ...and keep it alive */
//  GUI_MTOUCH_Enable(0);

    while (1)
    {
        /* Drawing a frame */
        GUI_Exec1();
        APPW_Exec();
        NVT_Draw1();

        /* Update time and temperature */
        if (RtcInt == 1)
        {
            RtcInt = 0;
            Read_Time();
            Read_Temperature();
        }
    }
}
```

Figure1-125 MCU Passes Temperature and Time to AppWizard

```
void ID_SCREEN_Time__ID_BUTTON_OK__WM_NOTIFICATION_RELEASED_2(APPW_ACTION_ITEM *pAction, WM_HWIN hScreen, WM_MESSAGE *pMsg, int *pResul
{
    GUI_USE_PARA(pAction);
    GUI_USE_PARA(hScreen);
    GUI_USE_PARA(pMsg);
    GUI_USE_PARA(pResult);

    Set_Time();
}
```

Figure1-126 When OK Button is Released, Time Set by AppWizard is Written to RTC of MCU

1.2.4 Programming and Running Program on Evaluation Board

Compile in Keil and download the program to NuMaker-HMI-M467 evaluation board to run.

1.3 Demo Result

The compilation and programming results of this example code are shown in Figure 1-127 and Figure 1-128.

Figure 1-127 Programming Results - English

Figure 1-128 Programming Results - Chinese

2. Code Description

2.1 Main.c

The main function is entered from *main.c* and contains three parts as follows:

- System clock and pin initialization: sys_init ()
- initialization: UART, SPIM, Timer, Touch, ADC, RTC, etc.
- Main application function: MainTask2()

```
#define SPI_CLK_FREQ          16000000   /* SPI clock: HZ */
#define DATA_WIDTH            8          /* data width: bit */
#define FIFO_LEVEL            8          /* FIFO level */
int main(void)
{
    /* Initialize system clock */
    sys_init();
    /* Initialize debug port */
    debug_port_init();
    SPIM_init();
    g_pu8FrameBufPtr  = (U8 *)((U32)g_u8FrameBuf);
    g_enable_Touch = 0;
    /* Initialize timer */
    tmr_init();
    RTC_init();
    ADC_Init();
    /* Startup CTP */
    st1633i_low_level_init();
    st1633i_startup_sequence();

    printf("\r\n");
    printf("CPU @ %d Hz\r\n", SystemCoreClock);
    printf("+-----------------------------------------------------------+\r\n");
    printf("|      Industrial Thermostat emWin Simple Demo              |\r\n");
    printf("+-----------------------------------------------------------+\r\n");
    printf("\r\n");

    g_enable_Touch = 1;
    /* Enable memory device before GUI_Init() to avoid flickering */
    printf("Main Task2 -> \n");
    /* Main Loop
       MainTask2 for anti-flickering
    */
    MainTask2();
    return 0;
}
```

Among them, the SPIM interface is used to store pictures and text files that occupy larger resources to the external SPI Flash. Using the SPI Synchronous Serial Interface Controller (SPI Master Mode), in the Direct Memory Mapping Mode (DMM) mode, SPI Flash can be regarded as a ROM, and SPI Flash can be read directly to the specified address; with the help of built-in cache, using quad read mode can improve overall reading efficiency.

The example code to initialize SPIM is as follows, which must include SPIM speed, SPI Flash detection, turning on SPIM cache mode, turning on SPIM DMM mode, and turning on fast quad read mode. For relevant parameter settings, please refer to the W25Q32JVTBIQ description.

```
void SPIM_init(void)
{
  uint8_t     idBuf[3];
  SYS_UnlockReg();              /* Unlock protected registers */
  SPIM_SET_CLOCK_DIVIDER(2);    /* Set SPIM clock as HCLK divided by 2, 192MHz/2 = 96 */
  SPIM_SET_RXCLKDLY_RDDLYSEL(0); /* Adjust the sampling clock to latch the correct data.*/
  SPIM_SET_RXCLKDLY_RDEDGE();/* Use SPI input clock rising edge to sample received data.*/

   SPIM_SET_DCNUM(4);         /* 4 is dummy cycle for command 0xEB(fast quad read) . */
   if (SPIM_InitFlash(1) != 0)    /* Initialized SPI flash */
   {
       printf("SPIM flash initialize failed!\n");
   }

   SPIM_ReadJedecId(idBuf, sizeof(idBuf), 1);
   printf("SPIM get JEDEC ID=0x%02X, 0x%02X, 0x%02X\n", idBuf[0], idBuf[1], idBuf[2]);

   SPIM_DISABLE_CCM();
   SPIM_ENABLE_CACHE();

   if (SPIM_CIPHER_ON)
       SPIM_ENABLE_CIPHER();
   else
       SPIM_DISABLE_CIPHER();

   if (SPIM_Enable_4Bytes_Mode(USE_4_BYTES_MODE, 1) != 0)
   {
       printf("SPIM_Enable_4Bytes_Mode failed!\n");
   }

   SPIM->CTL1 |= SPIM_CTL1_CDINVAL_Msk;        /* invalid cache */
   SPIM_SetQuadEnable(1, 1);    /* Set quad mode bit */
   SPIM_EnterDirectMapMode(USE_4_BYTES_MODE, CMD_DMA_FAST_QUAD_READ, 0);
}
```

In addition to SPIM initialization, the Scatter file must be edited to let compiler know the location and content of the SPI Flash. In KEIL environment, follow the path Option->Linker->Scatter File->Edit, open .sct file and edit the content as follows:

```
;APROM
LOAD_ROM_1  0x0
{
    APROM.bin  0x0
    {
     startup_M460.o (RESET, +FIRST)
     *(+RO)
    }

    SRAM  0x20000000 0x80000
    {
     * (+RW, +ZI)
```

```
    }
}

;SPI flash
LOAD_ROM_2  0x100000
{
    SPIM.bin  0x100000
    {
     MicrosoftYaHeiUI_20_Normal_EXT_AA4.o (+RO)
     MicrosoftYaHeiUI_24_Normal_EXT_AA4.o (+RO)
     BKG_Coordinate_480x242.o (+RO)
     BKG_TemperatureControl_480x242.o (+RO)
     BKG_TemperatureList_480x242.o (+RO)
     BKG_Time_480x242.o (+RO)
     GUI_Btn_Back_78x78.o (+RO)
     GUI_Btn_OK_78x78.o (+RO)
     GUI_Cal_Icon_40x36.o (+RO)
     GUI_Knob_Icon_106x106.o (+RO)
     GUI_KnobMark_Icon_16x9.o (+RO)
     GUI_Time_Icon_37x38.o (+RO)
    }
}
```

After entering the main application function MainTask2(), the program will execute AppWizard related functions and perform drawing actions in the while loop. This example code also adds Read_Time () and Read_Temperature (). These two functions are defined in Application. c file. In the function, the value is transferred from MCU to Variable in AppWizard by calling APPW_SetVarData() function, and then through the method described in section 1.2.2.17, the value will be displayed on screen.

```
void MainTask2(void)
{
    /* Setup configuration dependent pointers */
    APPW_X_Setup();
    /* Initialize AppWizard */
    APPW_Init(APPW_PROJECT_PATH);
    /* Create initial screen... */
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);

    /* ...and keep it alive */
    while (1)
    {
        /* Drawing a frame */
        GUI_Exec1();
        APPW_Exec();
        NVT_Draw1();
        /* Update time and temperature */
        if (RtcInt == 1)
        {
            RtcInt = 0;
            Read_Time();
            Read_Temperature();
        }
    }
}
```

2.2 Add User Processing

The content of screen executed in AppWizard will be read in ID_Screen_xxx_Slots.c. Each action set in Interaction has a corresponding processing function in ID_Screen_xxx_Slots.c file. The function naming rules are shown in Figure 2-1. Each screen has a cbID_SCREEN_xxx callback function. Users can add relevant action code to either of these two functions.

```
void <ScrID>__<EmitID>__<SignID>__<RecvID>__<JobID>(APPW_ACTION_ITEM * pAction,
                                                    WM_HWIN             hScreen,
                                                    WM_MESSAGE        * pMsg,
                                                    int               * pResult);
```

Figure 2-1 Slot Naming Rules

As follows, two methods are used to add user processing:

1. Add user code to Action Processing Function of Interaction.

Releasing OK button in Time Setting screen will trigger the closing of the time setting screen. Add the code to write the set time to the RTC in the function of this action.

In Time Setting screen, releasing OK button will trigger closing of the current screen. Add code to write set time to RTC in the action function.

```
/*********************************************************************
*
*       ID_SCREEN_Time__ID_BUTTON_OK__WM_NOTIFICATION_RELEASED
*/
void ID_SCREEN_Time__ID_BUTTON_OK__WM_NOTIFICATION_RELEASED(APPW_ACTION_ITEM *pAction,
WM_HWIN hScreen, WM_MESSAGE *pMsg, int *pResult)
{
    GUI_USE_PARA(pAction);
    GUI_USE_PARA(hScreen);
    GUI_USE_PARA(pMsg);
    GUI_USE_PARA(pResult);

    Set_Time();
}
```

2. Add User Processing in Callback Function of Screen.

In callback function of Date Setting screen, when OK button is released, the set date is written into the RTC of MCU. Code needs to be added to determine whether OK button is released and to write the set date to the RTC.

```
/*********************************************************************
*
*       cbID_SCREEN_Date
*/
void cbID_SCREEN_Date(WM_MESSAGE *pMsg)
{

    int Id, NCode;
```

```
    switch (pMsg->MsgId)
    {
        case WM_NOTIFY_PARENT:
            Id = WM_GetId(pMsg->hWinSrc);
            NCode = pMsg->Data.v;
            if (Id == ID_BUTTON_OK)
            {
                switch (NCode)
                {
                    case WM_NOTIFICATION_RELEASED:
                        Set_Date();
                        break;
                }
            }
            break;
    }
}
```

2.3 Application.c File

Application.c generated by AppWizard is an empty file, used to add user code. Note that if you modify GUI in AppWizard and regenerate code, do not overwrite user code. It includes RTC reading and writing and temperature sampling, as well as the interaction between MCU and AppWizard.

```
void Read_Time(void)
{
    S_RTC_TIME_DATA_T sReadRTC;
    /* Read rtc */
    RTC_GetDateAndTime(&sReadRTC);
    /* Update time variables in emWin */
    APPW_SetVarData(ID_VAR_Year, sReadRTC.u32Year - 2023);
    APPW_SetVarData(ID_VAR_Month, sReadRTC.u32Month - 1);
    APPW_SetVarData(ID_VAR_Day, sReadRTC.u32Day - 1);
    APPW_SetVarData(ID_VAR_Hour, sReadRTC.u32Hour);
    APPW_SetVarData(ID_VAR_Min, sReadRTC.u32Minute);
    APPW_SetVarData(ID_VAR_Sec, sReadRTC.u32Second);
}

void Set_Time(void)
{
    int error;
    S_RTC_TIME_DATA_T sTime;

    /* Read the time setting value in emWin */
    sTime.u32Hour = APPW_GetVarData(ID_VAR_Hour, &error);
    sTime.u32Minute = APPW_GetVarData(ID_VAR_Min, &error);
    sTime.u32Second = APPW_GetVarData(ID_VAR_Sec, &error);
    sTime.u32TimeScale = RTC_CLOCK_24;

    /* Modify RTC time */
    RTC_SetTime(sTime.u32Hour, sTime.u32Minute, sTime.u32Second, sTime.u32TimeScale, 0);
}
void Set_Date(void)
```

```
{
    int error;
    S_RTC_TIME_DATA_T sTime;

    /* Read the date setting value in emWin  */
    sTime.u32Year = APPW_GetVarData(ID_VAR_Year, &error) + 2023;
    sTime.u32Month = APPW_GetVarData(ID_VAR_Month, &error) + 1;
    sTime.u32Day = APPW_GetVarData(ID_VAR_Day, &error) + 1;

    if ((sTime.u32Month == 1) || \
            (sTime.u32Month == 3) || \
            (sTime.u32Month == 5) || \
            (sTime.u32Month == 7) || \
            (sTime.u32Month == 8) || \
            (sTime.u32Month == 10) || \
            (sTime.u32Month == 12))
    {
        if (sTime.u32Day > 31)
            sTime.u32Day = 31;
    }
    else if ((sTime.u32Month == 4) || \
             (sTime.u32Month == 6) || \
             (sTime.u32Month == 9) || \
             (sTime.u32Month == 11))
    {
        if (sTime.u32Day > 30)
            sTime.u32Day = 30;
    }
    else  /* February */
    {
        if (sTime.u32Year % 4 == 0) /* leap-year */
        {
            if (sTime.u32Day > 29)
                sTime.u32Day = 29;
        }
        else
        {
            if (sTime.u32Day > 28)
                sTime.u32Day = 28;
        }
    }
    /* Modify RTC date */
    RTC_SetDate(sTime.u32Year, sTime.u32Month, sTime.u32Day, sTime.u32Day);
}
float Calc_Temperature(int AdcValue)
{
    int Vi;             /* mV */
    float Rt;           /* Measured actual resistance value */
    float Temp;         /* Temperature value */
    float R = 10000;    /* Thermistor NTC103 */
    float B = 3950;     /* NTC fixed parameters */
    float T2 = (273.15 + 25.0);

    /* Value ==> Voltage */
    Vi = 3300 * AdcValue / 0xFFF; /* mV */
    /* Voltage ==> Resistance */
```

```
    Rt = (float)Vi * 10000 / (3300 - Vi);
    /* Resistance ==> Temperature */
    Temp = 1 / (log(Rt / R) / B + 1 / T2);
    /* Degrees Celsius */
    Temp = (Temp - (float)273.15);
    return Temp;
}
void Read_Temperature(void)
{
    int  i32ConversionData[5], Temp[5], i;
    int BIT;

    /* ADC conversion */
    for (i = 0 ; i < 5; i++)
    {
        BIT = 1 << i;
        EADC_START_CONV(EADC0, BIT);
        /* Wait conversion done */
        while (EADC_GET_DATA_VALID_FLAG(EADC0, BIT) != BIT);
        /* Read the conversion result */
        i32ConversionData[i] = EADC_GET_CONV_DATA(EADC0, i);
        Temp[i] = (int) Calc_Temperature(i32ConversionData[i]);
    }

    /* Update temperature variables in emWin */
    APPW_SetVarData(ID_VAR_A1_ActualTemp, Temp[0]);
    APPW_SetVarData(ID_VAR_A2_ActualTemp, Temp[1]);
    APPW_SetVarData(ID_VAR_A3_ActualTemp, Temp[2]);
    APPW_SetVarData(ID_VAR_A4_ActualTemp, Temp[3]);
    APPW_SetVarData(ID_VAR_A5_ActualTemp, Temp[4]);
}
```

3. Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M460_Series_BSP_CMSIS_V3.00.002
- IDE version
  - Keil uVersion 5.38

3.2 Hardware Requirements

- Circuit components
  - NuMaker-HMI-M467HJ V1.0

4. Directory Information

The directory structure is shown below.

EC_ M467_AppWizard_ISMThermostat _V1.00	
Library	**Sample code header and source files**
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	
Project	Source file of example code
ThirdParty	
emWin	

Figure 4-1 Directory Structure

5. Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *M467_AppWizard_ISMT.uvprojx.*
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session.
3. Enter debug mode.
   - Run

6. Revision History

Date	Revision	Description
2023.11.15	1.00	Initial version.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.