以 AppWizard 设计工业智能温控器 GUI 项目

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 emWin AppWizard 工具，结合 M467 系列开发平台与开发资源，完成工业智能温控器图形化使用接口。
BSP 版本	M460_Series_BSP_CMSIS_V3.00.002 emWin Software Package for M460 V3.0
开发平台	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. 概述

本范例代码将示范如何使用 SEGGER emWin AppWizard 工具，搭配 NuMaker-HMI-M467 开发平台，逐步完成一个工业智能温控器项目的画面设计、画面互动。

1.1 原理

1.1.1 框图

此范例代码直接使用新唐 GUI 开发平台 NuMaker-HMI-M467 开发板，开发时会使用到的组件包括分辨率 480x272 且屏幕大小 4.3 寸的 TFT-LCD、电容式触控面板、主控板上 NuMicro M467HJHAN 微控制器、以及容量大小 4 Mbyte 的 SPI Flash。其系统框图如图 1-2 所示。

图 1-1 NuMaker-HMI-M467 开发板

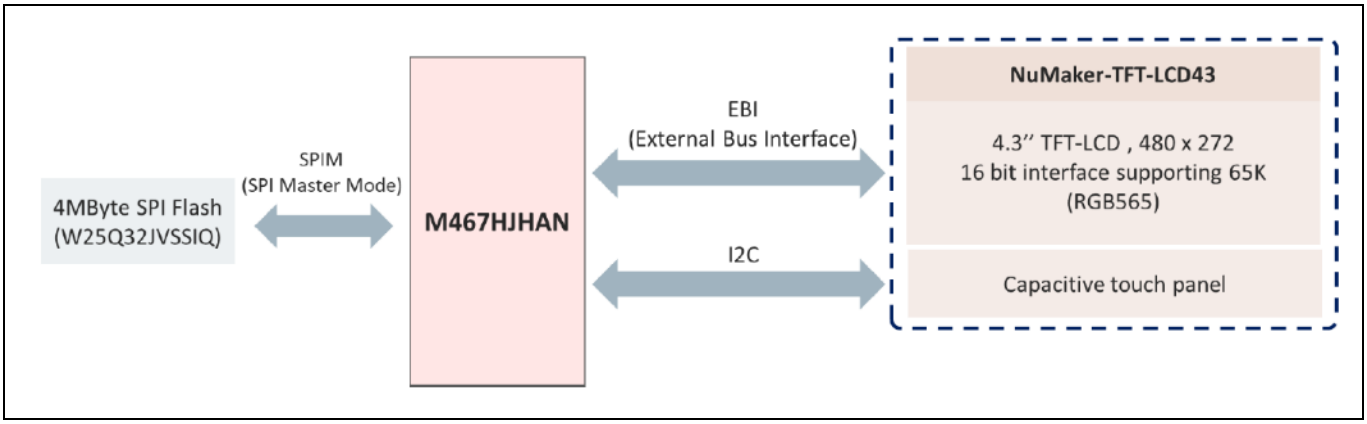
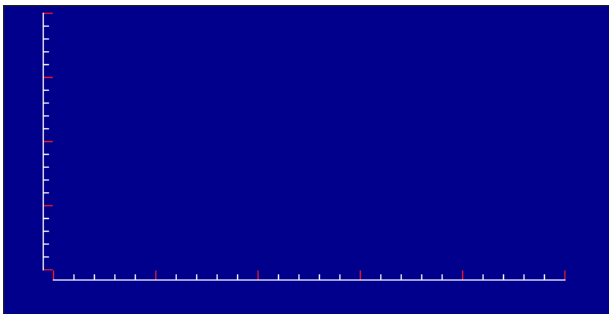
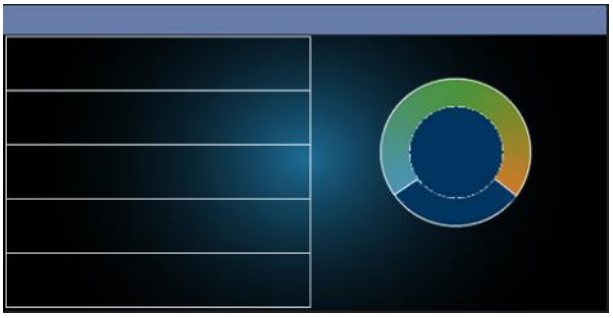
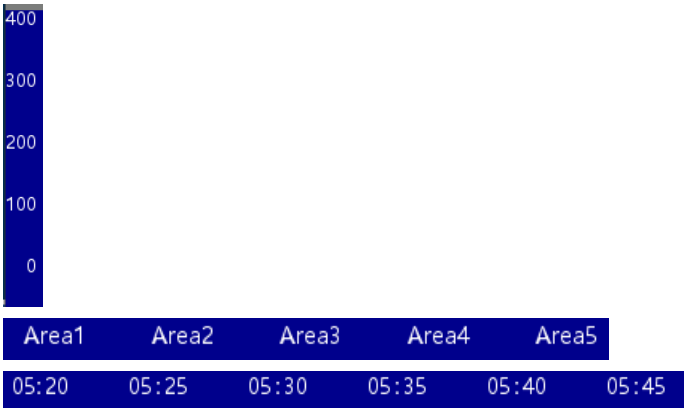
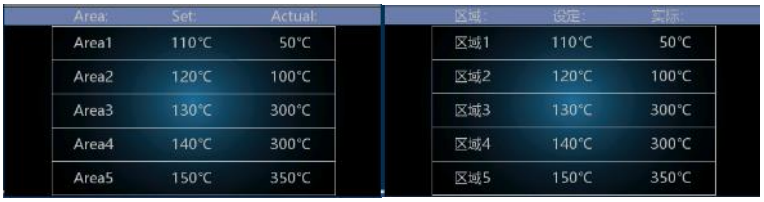
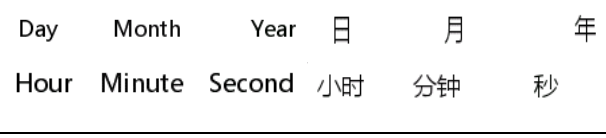
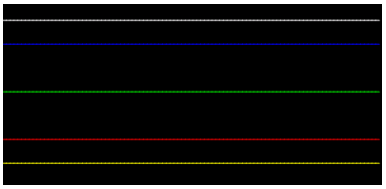
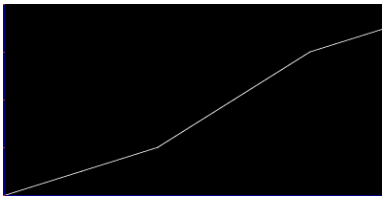


图 1-2 系统框图

1.1.2 工业智能温控器画面使用元素总览

应用说明	应用 GUI	对象
主菜单	View Setup Time Lang Help 显示 设置 时间 语言 帮助	Button
确定按键		
返回按键		
温度区域选择	透明无显示	
显示菜单	Area Temperature Area Temperature Time Temperature 区域温度列表 区域温度曲线 时间温度曲线	Listbox
温度设置菜单	Temperature Setup 温度设置	
时间设置菜单	Time Setup Calendar Setup 时间设置 日期设置	
语言设置菜单	EN CN 英文 中文	
480x242 坐标背景图片		Image
480x242 温度列表背景图片		

应用说明	应用 GUI	对象
480x242 温度设置背景图片		
480x242 时间、日期设置屏幕背景图片		
日期图标		
时间图标		
坐标的文字 温度、区域、时间		TEXT
区域名称、设定温度、实际温度、温度单位		
年、月、日、小时、分钟、秒		

应用说明	应用 GUI	对象
区域颜色的表示		Drawings
时间-温度曲线		
区域-温度曲线		
温度设定旋钮		Rotary
日期选择		Wheel
时间选择		
日期、时间选择的背景		Box

应用说明	应用 GUI	对象
选中区域的背景		

表 1-1 元素总览

1.1.3 配套资源

- SEGGER emWin 官方论坛：Segger emWin Forum
- 新唐官方论坛：Nuvoton emWin Forum
- SEGGER emWin AppWizard 工具使用手册下载：AppWizard User Manual

1.2 执行步骤

使用 AppWizard 工具完成一项 GUI 项目的执行步骤可分为四大阶段：

1. 准备硬件与下载软件。
2. 在 AppWizard 中完成对应的画面设计、object 设定、interaction 设定、Drawings 设定。
3. 使用 AppWizard 生成代码并加入用户代码，来完成更多应用功能。
4. 烧录到开发板上运行。

以下就各项阶段详细说明执行步骤。

1.2.1 准备硬件与软件资源

1. 硬件使用 NuMaker-HMI-M467，相关资料可参考 参考线路、开发板使用手册。
2. 软件资源需逐一下载与完成以下项目：
    1) M460 BSP，下载链接 M460 BSP，下载后解压缩。
    2) M460 emWin HMI Library，下载链接 M467_emWin_package，下载后对 M460_emWin_NonOS 文件夹解压缩。
    3) 请按表 1-2 路径，点击 AppWizard_V134a_630a_Install.exe 安装 AppWizard 工具，需注意若有旧的 AppWizard 版本，需将旧版本完全移除后，才能安装新版本。

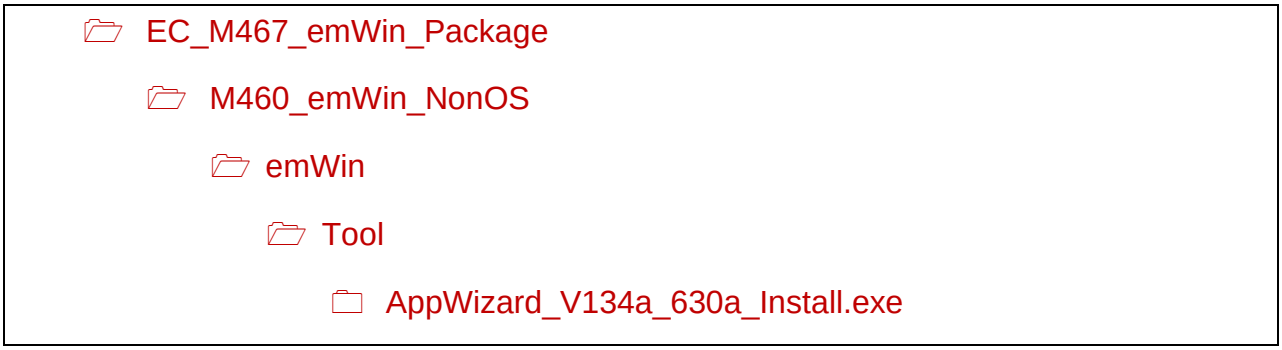


表 1-2 AppWizard 工具安装文件路径

3. 安装 M467_emWin_package 至 M460 BSP 中：

   1) 将 M467_emWin_Package\M460_emWin_NonOS\emWin 复制并贴上至 M460_Series_BSP_CMSIS_V3.00.002\ThirdParty 中。

   2) 将 M467_emWin_Package\M460_emWin_NonOS\emWin_SimpleDemoAppWizard 复制并贴上至 M460_Series_BSP_CMSIS_V3.00.002\SampleCode 中。

1.2.2 在 AppWizard 工具中编辑与设定 GUI

1.2.2.1 创建 emWin 工程

1. 桌面快捷方式开启 AppWizard 工具。

图 1-3 AppWizard 桌面快捷方式图标

2. 点选 Create new project 创建新工程。

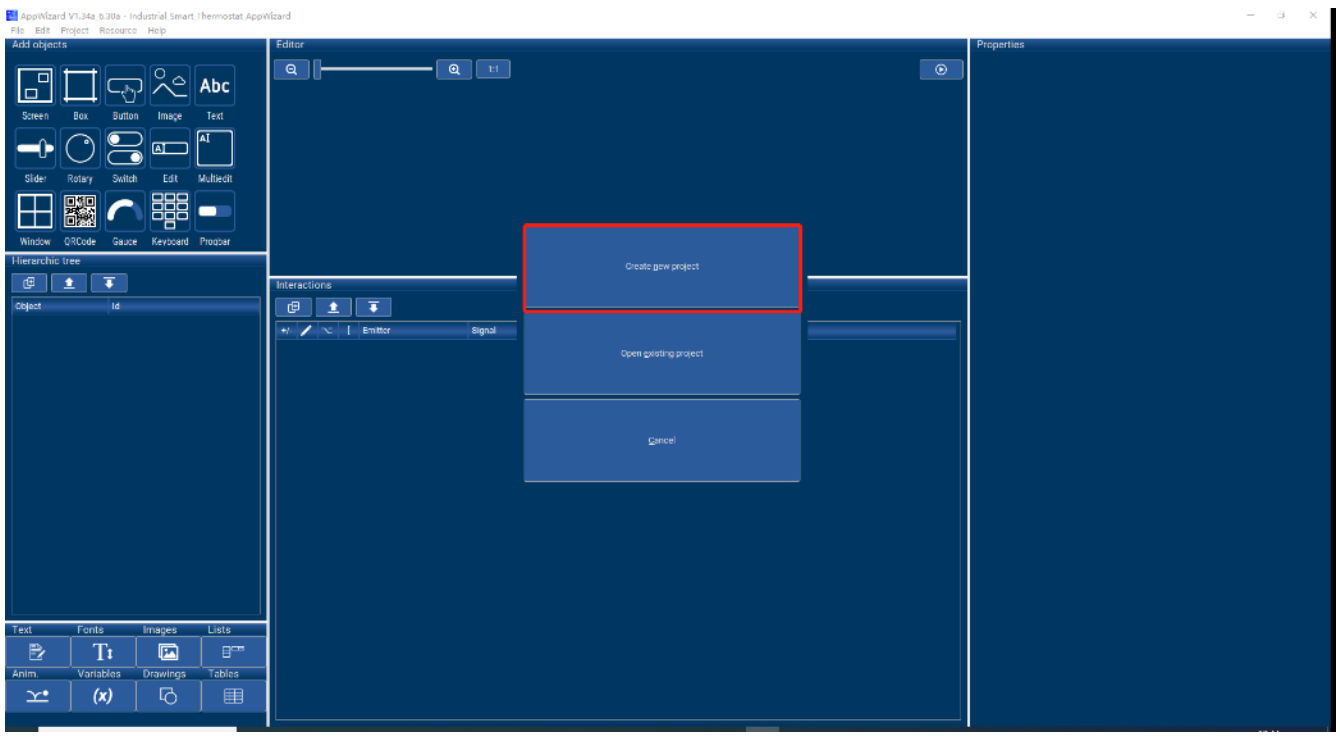


图 1-4 创建新工程

3. 按图 1-5 设置 Project path、Project name 名称，符合使用屏幕的颜色格式：16 Bit, GUICC_M565，选择 Enable Multibuffering，点选 Ok。

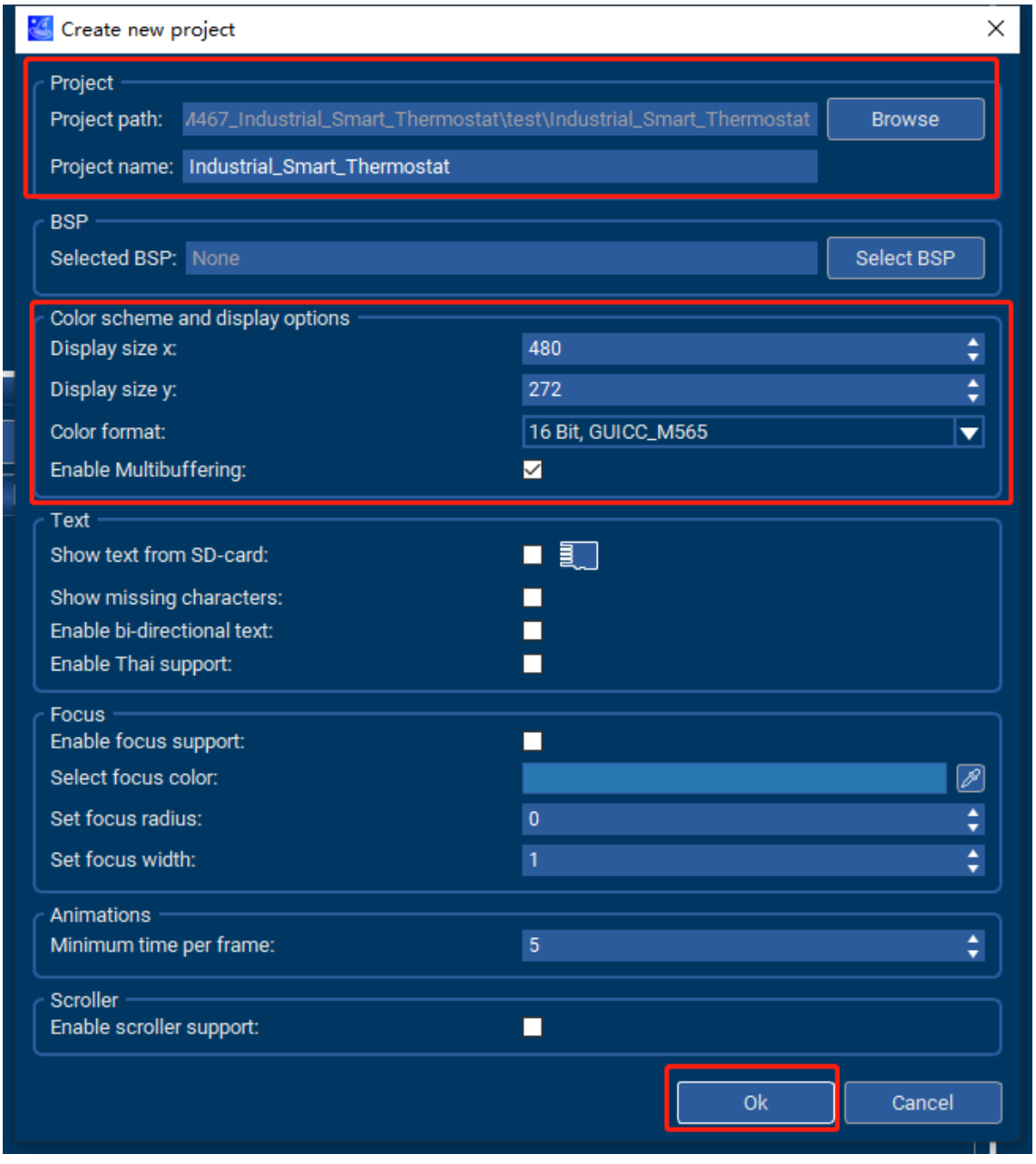


图 1-5 新工程设定

1.2.2.2 添加 Text

定义工程中用到的所有文本内容，开发过程中可以随时添加，文本如表 1-3 所示。

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	日期设置

ID	en	cn
ID_RTEXT_COLON	:	:

表 1-3 工程中的文本

添加文本的步骤如图 1-6 所示：

1. 第 1 步点击 Text，弹出 Choose/Edit Text 对话框。
2. 重复 2~4 步点击 New language 添加两种语言英文和中文。
3. 第 5~8 步点击 Add text 添加文本、修改 ID、定义英文和中文文本内容。
4. 重复 5~8 步完成表 1-3 中所有文本的定义。
5. 第 9 步点击 Ok 完成 Text 的定义。

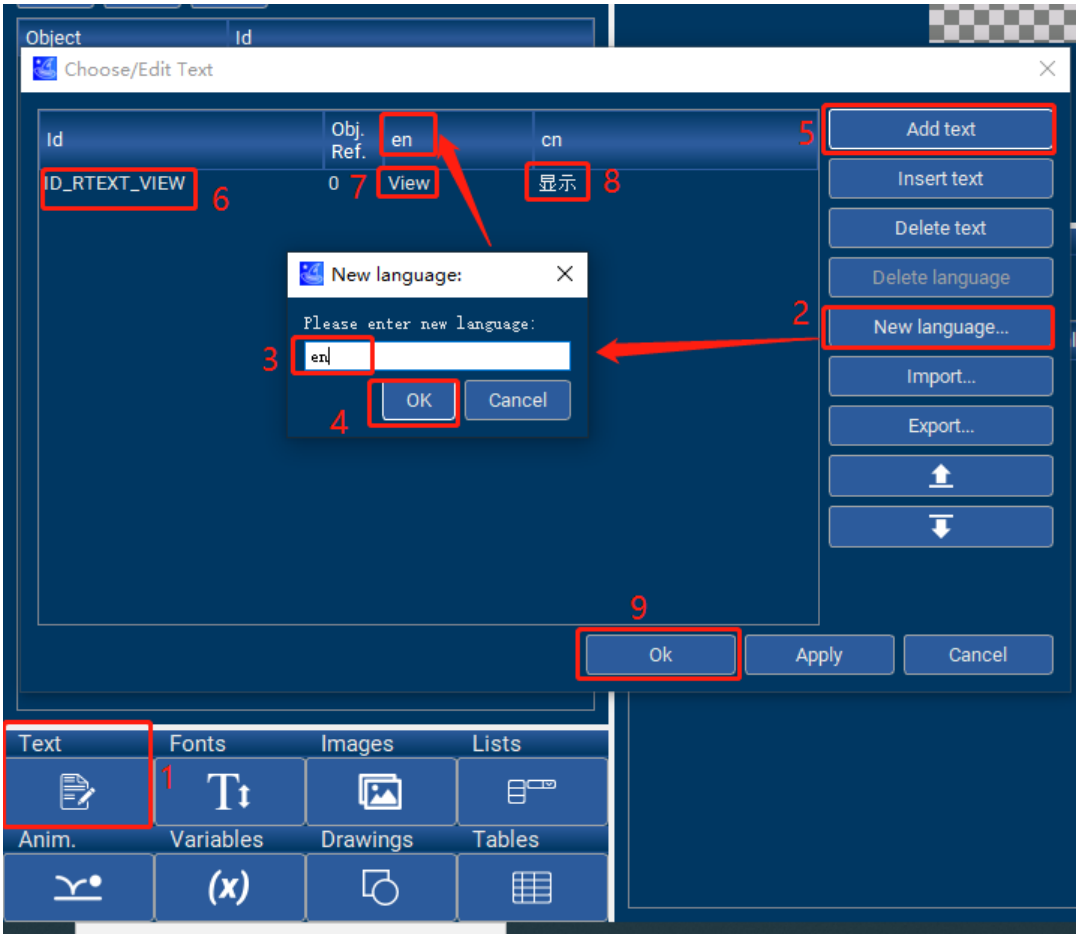


图 1-6 添加文本

1.2.2.3 添加 List

定义工程中用到的所有 List，开发过程中可以随时添加，List 如表 1-4 所示。

ID	内容	说明
ID_CONTENT_VIEW	ID_RTEXT_AREA_TEMP_L ID_RTEXT_AREA_TEMP_W ID_RTEXT_TEMP_TIME_W	View 下拉菜单显示列表
ID_CONTENT_SETUP	ID_RTEXT_TEMP_SETUP	Setup 下拉菜单显示列表
ID_CONTENT_TIME	ID_RTEXT_TIME_SETUP ID_RTEXT_DATE_SETUP	Time 下拉菜单显示列表
ID_CONTENT_LANG	ID_RTEXT_ENG ID_RTEXT_CH	语言下拉菜单显示列表
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	月份设置选项的内容
ID_CONTENT_HOUR	00~23	小时设置选项的内容
ID_CONTENT_MIN_SEC	00~59	分和秒的设置选项的内容
ID_CONTENT_DAY	01~31	日期设置选项的内容
ID_CONTENT_YEAR	2023~2122	年份设置选项的内容

表 1-4 工程中的 List

添加 List 的步骤如图 1-7 所示：

1. 第 1 步点击 List，弹出 Content edit 对话框。
2. 第 2~3 步添加 List 以及重命名。
3. 第 4 步点击 添加 List 的内容。
4. 重复 5~6 步为 List 添加需要的内容，内容可以是 Text 里定义的文本，也可以直接写入字符串。
5. 重复 2~7 步完成表 1-4 中所有 List 的定义。
6. 第 8 步点击 Ok 完成 List 的定义。

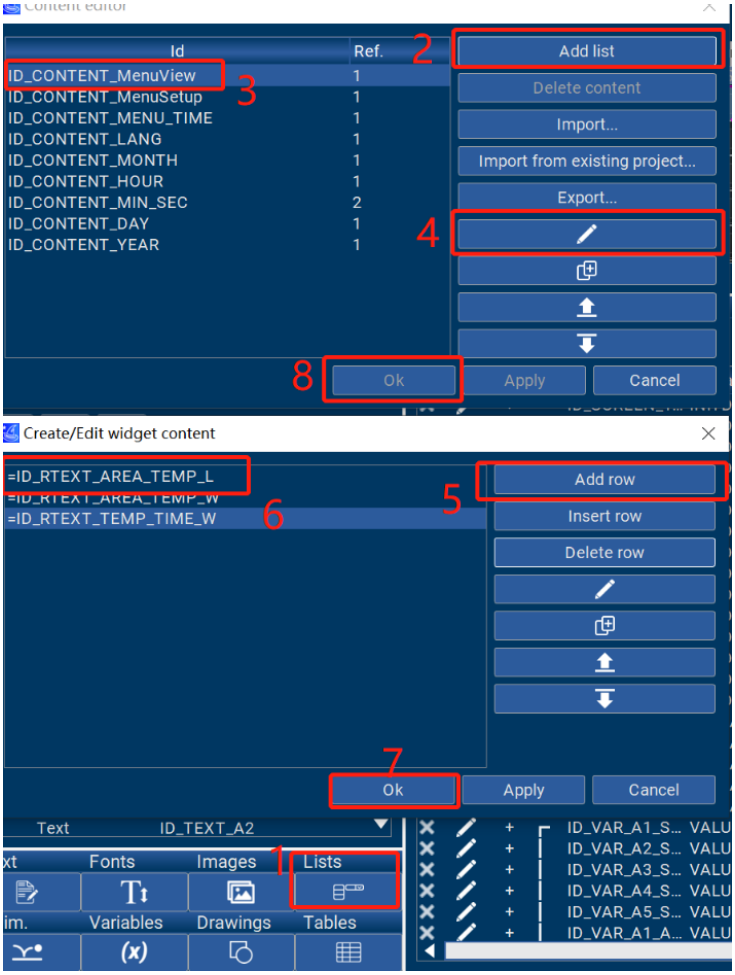


图 1-7 添加 List

1.2.2.4 添加 Fonts

按照图 1-8 创建两个字体 MicrosoftYaHeiUI_24_Normal_EXT_AA4 和 MicrosoftYaHeiUI_20_Normal_EXT_AA4，注意这里选择的字体要能同时显示中文和英文。

1. 第 1 步点击 Fonts 弹出 Choose Font 对话框。
2. 第 2 步点击 Create new 弹出 Create Font 对话框。
3. 第 3~7 步选择字体 MicrosoftYaHeiUI。
4. 第 8~10 步设置 codepoint 的范围，选择 Project text only 即仅选择项目会使用到的字符范围。
5. 重复 2~10 步完成 MicrosoftYaHeiUI_20_Normal_EXT_AA4 字体的定义。

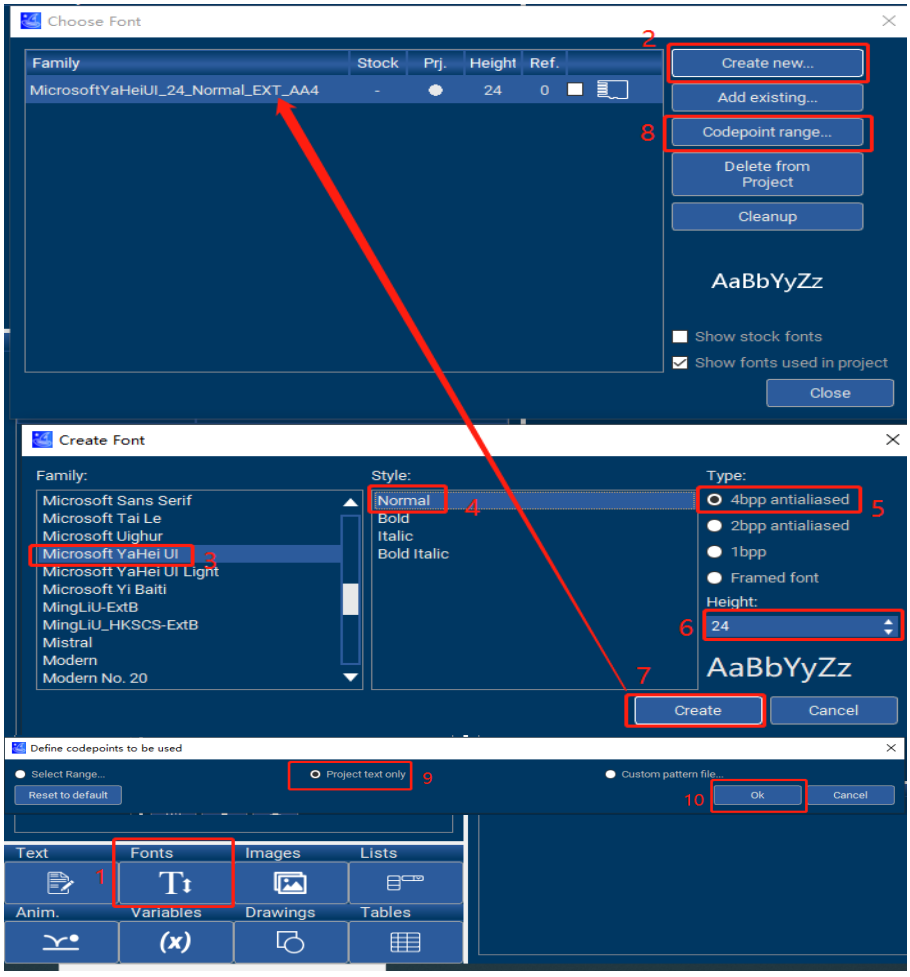


图 1-8 创建字体

1.2.2.5 添加 Images

加载工程中用到的图片。

1. 第 1 步点击 Images 弹出 Images 对话框。
2. 第 2 步点击 Import to project，弹出 Add Image 对话框。
3. 第 3~4 步选择工程需要的 image 导入。

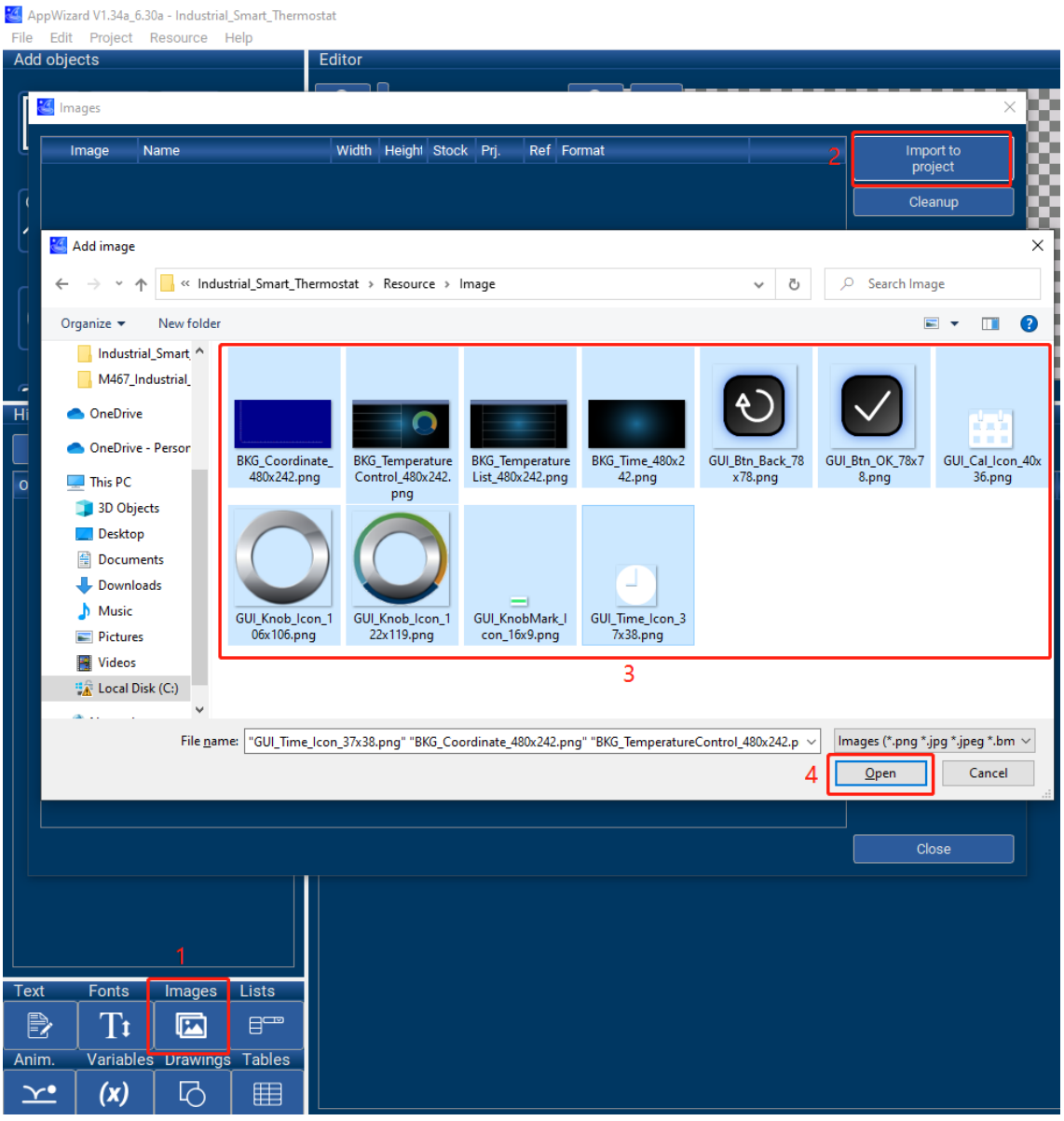


图 1-9 加载 Image

1.2.2.6 添加 Variables

定义工程中用到的所有变量，开发过程中可以随时添加。所有变量如表 1-5 所示。

ID	初始值	计算公式	说明
ID_VAR_View_index	0		View 下拉菜单的索引号
ID_VAR_A1_SetTemp	110		区域的设定温度 (5 个区域)
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		区域实际温度 (5 个区域)
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	选中的区域
ID_VAR_Bar_Y0	0		选中区域的坐标 Y0 值
ID_VAR_Temp	0		温度设置值的临时值
ID_VAR_Color_Yellow	-2698496		黄色
ID_VAR_Color_White	-1		白色
ID_VAR_A1_Color~ ID_VAR_A5_Color	-1		区域的字体显示颜色
ID_VAR_Y1~ ID_VAR_Y5	-1	200-(ID_VAR_An_ActualTemp*200/400) 其中 n=1~5	区域-温度曲线纵坐标

ID	初始值	计算公式	说明
ID_VAR_X0~ ID_VAR_X99	0	ID_VAR_X99 = ID_VAR_X99 +ID_VAR_X_Pixel_inc	时间-温度曲线横坐标
ID_VAR_A1_Y0~ ID_VAR_A1_Y99	0	ID_VAR_An_Y99 = 200-(ID_VAR_An_ActualTemp*200/400) 其中 n=1~5	区域曲线的纵坐标 (5 个)
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		时间-温度曲线刷新标志
ID_VAR_X_Index	0		x 轴的当前显示位置索引号
ID_VAR_Point	99		曲线上总共显示的点数=ID_VAR_Point+1
ID_VAR_X_Pixel_inc	4	400/(ID_VAR_Point+1)	时间-温度曲线中两点间的像素点数
ID_VAR_Timer_Cnt	0		1 s 定时器计数
ID_VAR_Refresh_Interval	60	ID_VAR_Refresh_Interval= ID_VAR_Refresh_inc	波形刷新间隔变量
ID_VAR_Refresh_inc	60		波形刷新间隔常量
ID_VAR_Year	0		时间、日期实际值变量
ID_VAR_Month	0		
ID_VAR_Day	0		

ID	初始值	计算公式	说明
ID_VAR_Hour	0		
ID_VAR_Min	0		
ID_VAR_Sec	0		
ID_VAR_Year_Tmp	0		时间、日期中间转换用的变量
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	当前时间 (换算成秒)
ID_VAR_Time_In_Sec_Tmp	0	86400+ID_VAR_Time_In_Sec-(5*ID_VAR_X_Time_inc)	根据当前时间换算出坐标刻度的中间量
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	x 轴上两个大刻度间的时间间隔
ID_VAR_Current_Screen	0		当前显示的画面

表 1-5 工程中的变量

添加变量步骤如图 1-10 所示：

1. 第 1 步点击 Variables 弹出 Variables 对话框。
2. 第 2~4 步点击 Add Variable 添加变量，并设置变量 ID 和初始值。
3. 第 5 步点击 Term 里的[redacted]弹出 Create calculated 对话框，为变量添加计算公式。注意只有需要计算公式的变量才需要添加公式。
4. 第 6~9 步在 Operand definition 里选择计算公式里需要的参数，可以是对象的值、变量、固定值等。点击 Add Operand 添加选中的参数，已经添加的参数会显示会在第 9 步的位置。重复 6~9 步可以添加多个参数。
5. 第 10 步在 Term to be calculated 里编辑公式。
6. 重复 2~11 步添加所有变量。如果不需要计算公式忽略第 5~11 步。
7. 第 12 步点击 Ok 保存添加的变量。

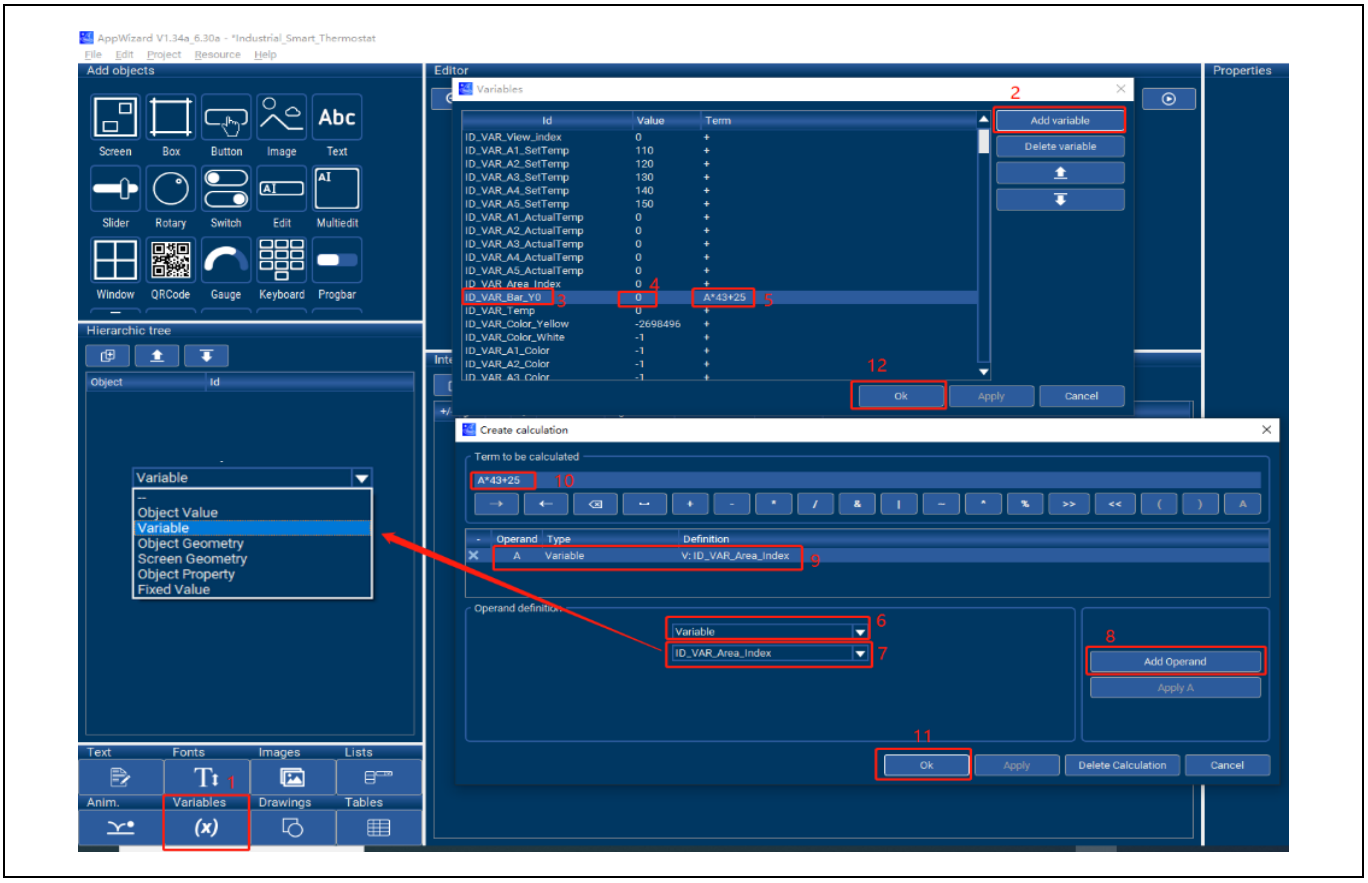


图 1-10 添加变量

1.2.2.7 添加 Drawings

定义工程中用到的所有画图动作，也可以在开发过程中定义，所有动作如表 1-6 所示。

动作	作用
ID_DRAW_BLK	清除曲线绘制区域
ID_DRAW_TempWave	绘制区域-温度曲线
ID_DRAW_TimeWave	绘制时间-温度曲线
ID_DRAW_COLOR	绘制曲线颜色指示

表 1-6 画图说明

1. 绘制纯黑背景色，步骤如图 1-11 所示：

   1) 第 1 步点击 Drawings 弹出 Drawing objects 对话框。

   2) 第 2~3 步点击 Add drawing 添加画图动作，并为此次动作命名为 ID_DRAW_BK。

   3) 第 4 步点击 弹出 Edit drawing 对话框。

   4) 第 5~7 步选择需要的功能。在 Available functions 里选择功能，点击 Add function 将选中的 function 添加到 Used functions 里，这里选择 SET_BKCOLOR 先设置背景色。

   5) 第 8~9 步在 Parameters of selected function 里选择需要设置的参数并设置。SET_BKCOLOR 只有一个参数 BkColor (设置背景色)，设置为 0xff000000 (黑色)。

   6) 重复 5~6 步添加 CLEAR，目的是用刚才设定背景色清除显示区域，注意 CLEAR 没有参数需要设置，所以无需第 8~9 步。

   7) 第 10~11 步点击 Ok 完成动作的设置。

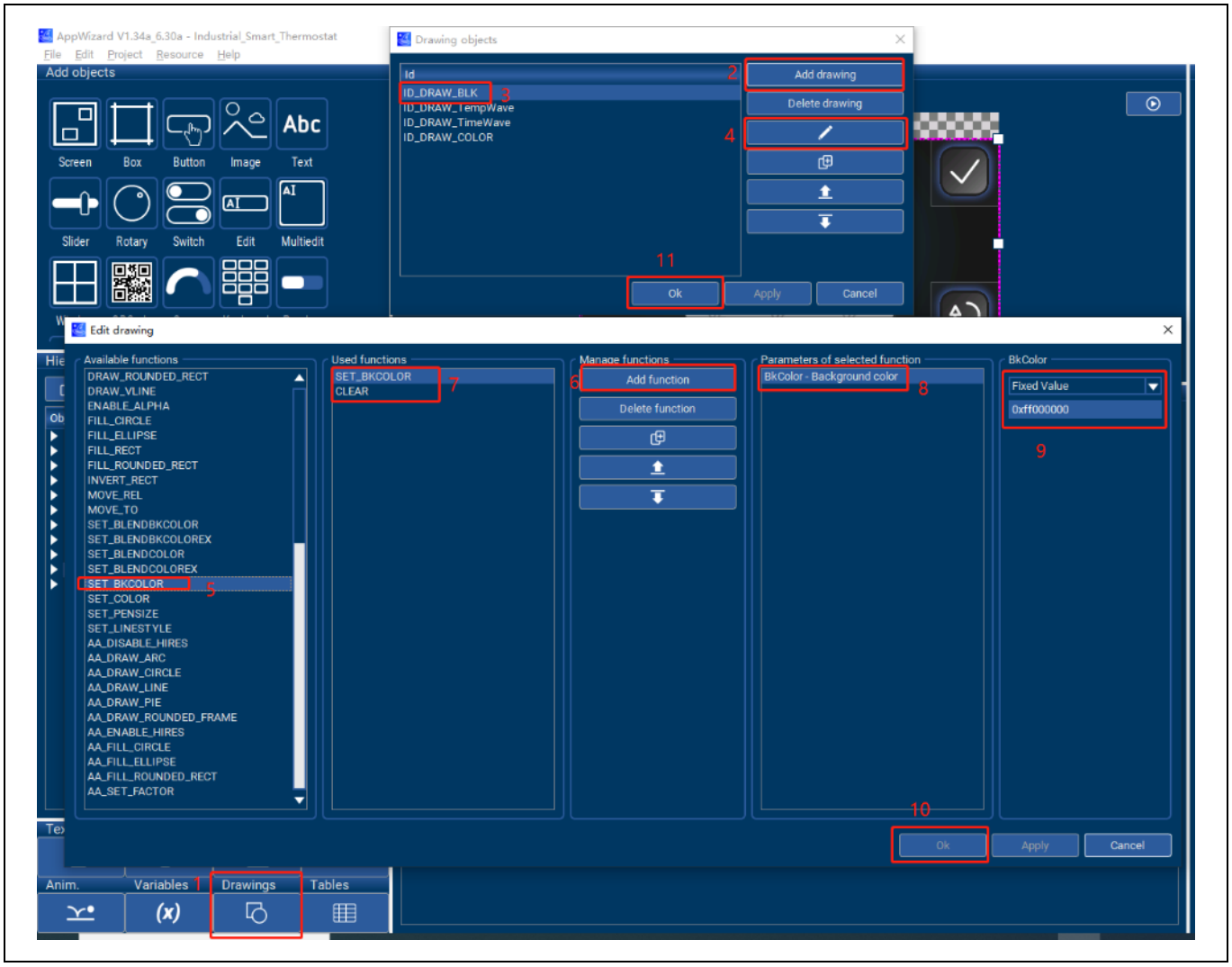


图 1-11 绘制纯黑背景

2. 绘制区域-温度曲线

区域-温度曲线画面中设置的曲线绘制区域大小是 400x200。X 轴 400 个像素点表示 5 个区域。Y 轴 200 个像素点表示 0 °C ~ 400 °C，所以 1 个像素点对应 2 °C。需要注意 AppWizard 里坐标是以 object 的左上角为起点向右和向下伸展。而被绘制的曲线需要以左下角为起点上右向上伸展，因此作图过程中需要进行坐标转换。如图 1-12 所示，红色是 Object 的坐标，绿色是曲线的坐标。

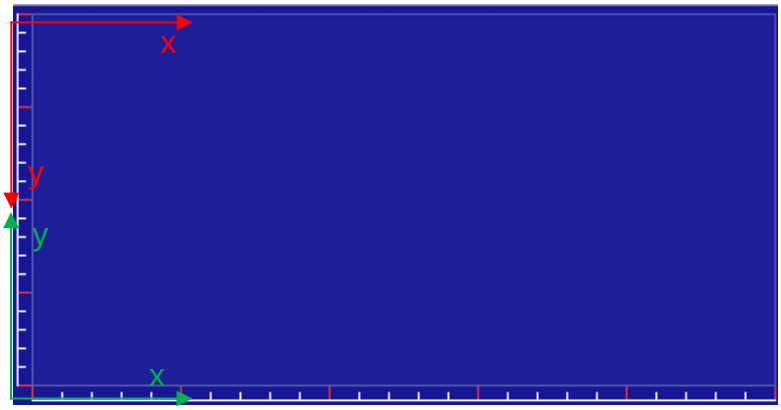


图 1-12 Object 坐标和曲线坐标

区域-温度曲线主要由坐标原点和 5 个区域的温度， 6 个点绘制成 5 条线段组成。绘图动作如表 1-7 所示。

功能	参数	说明
SET_BKCOLOR	BkColor = (Fixed value, 0xff000000)	设置背景颜色为黑色
CLEAR	无	用背景色清除显示内容
SET_COLOR	Color = (Fixed value，0xffffffff)	设置画图颜色为白色
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 200, 80, ID_VAR_Y1)	画从原点到区域 1 温度的线段。
AA_DRAW_LINE	(x0, y0, x1, y1) = (80, ID_VAR_Y1, 160, ID_VAR_Y2)	画从区域 1 温度到区域 2 温度的线段。
AA_DRAW_LINE	(x0, y0, x1, y1) = (160, ID_VAR_Y2, 240, ID_VAR_Y3)	画从区域 2 温度到区域 3 温度的线段。
AA_DRAW_LINE	(x0, y0, x1, y1) = (240, ID_VAR_Y3, 320, ID_VAR_Y4)	画从区域 3 温度到区域 4 温度的线段。
AA_DRAW_LINE	(x0, y0, x1, y1) = (320, ID_VAR_Y4, 400, ID_VAR_Y5)	画从区域 4 温度到区域 5 温度的线段。

表 1-7 区域-温度曲线绘制的参数

先用背景色清除显示，然后绘制五条线段，步骤如图 1-13 所示。

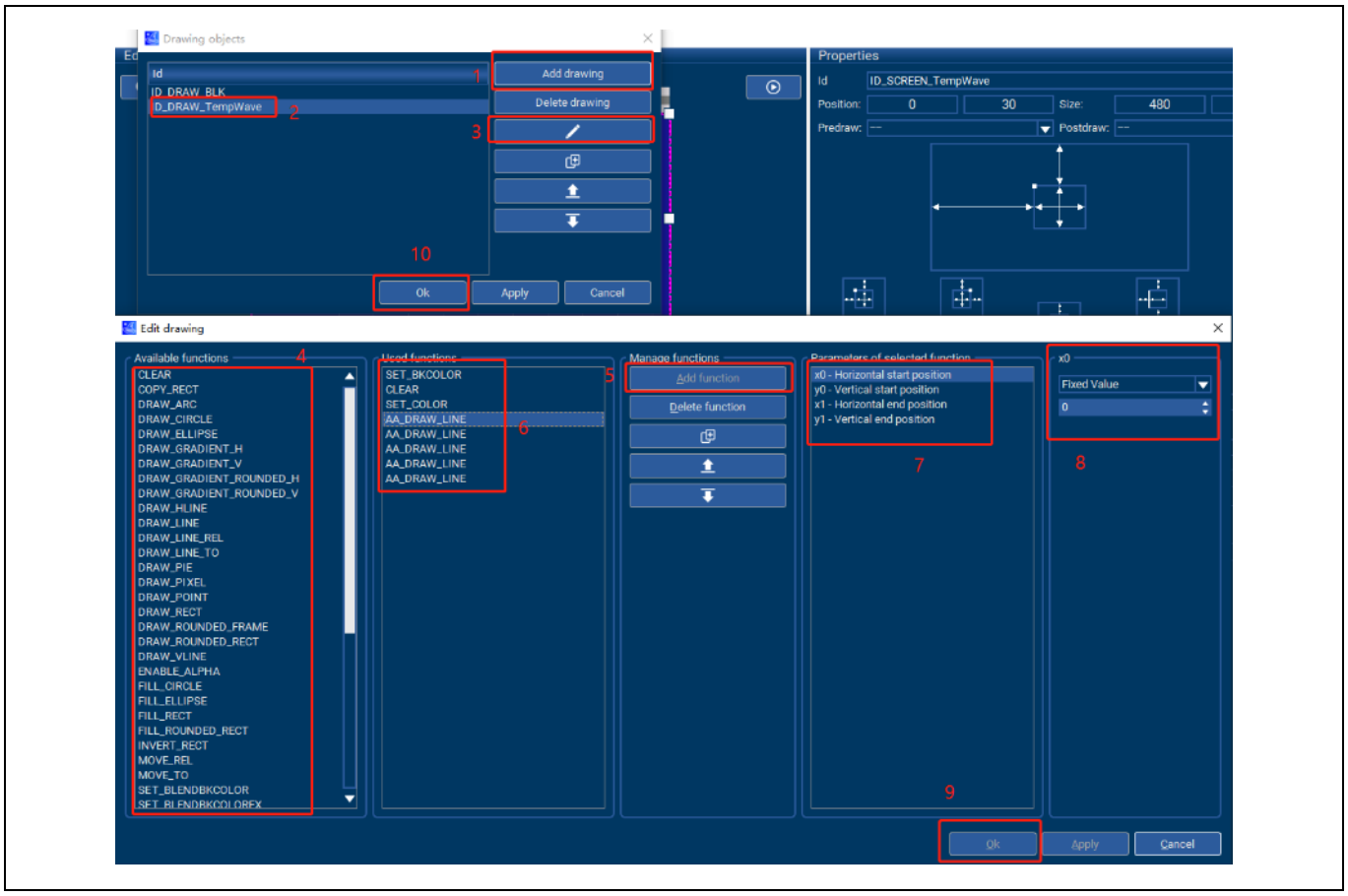


图 1-13 绘制区域-温度曲线

3. 绘制时间-温度曲线的颜色指示

当一个画面中有多条曲线时，通常用不同颜色来区分。因此需要说明颜色和曲线的对应关系。这里绘制不同颜色的线段来指示对应的曲线，绘图动作如表 1-8，操作步骤如图 1-14 所示。

功能	参数	说明
SET_COLOR	Color = (Fixed value, 0xffffff00)	设置画图颜色为黄色
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 24, 40, 24)	画一条黄色线段，表示黄色曲线是区域 1 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xffff0000)	设置画图颜色为红色
AA_DRAW_LINE	(x0, y0, , x1, y1) = (0, 54, 40, 54)	画一条红色线段，表示红色曲线是区域 2 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xff00ff00)	设置画图颜色为绿色
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 84, 40, 84)	画一条绿色线段，表示绿色曲线是区域 3 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xff0000ff)	设置画图颜色为蓝色
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 114, 40, 114)	画一条蓝色线段，表示蓝色曲线是区域 4 时间-温度曲线
SET_COLOR	Color = (Fixed value, 0xff000000)	设置画图颜色为白色
AA_DRAW_LINE	(x0, y0, x1, y1) = (0, 144, 40, 144)	画一条白色线段，表示白色曲线是区域 5 时间-温度曲线

表 1-8 曲线颜色指示绘制的参数

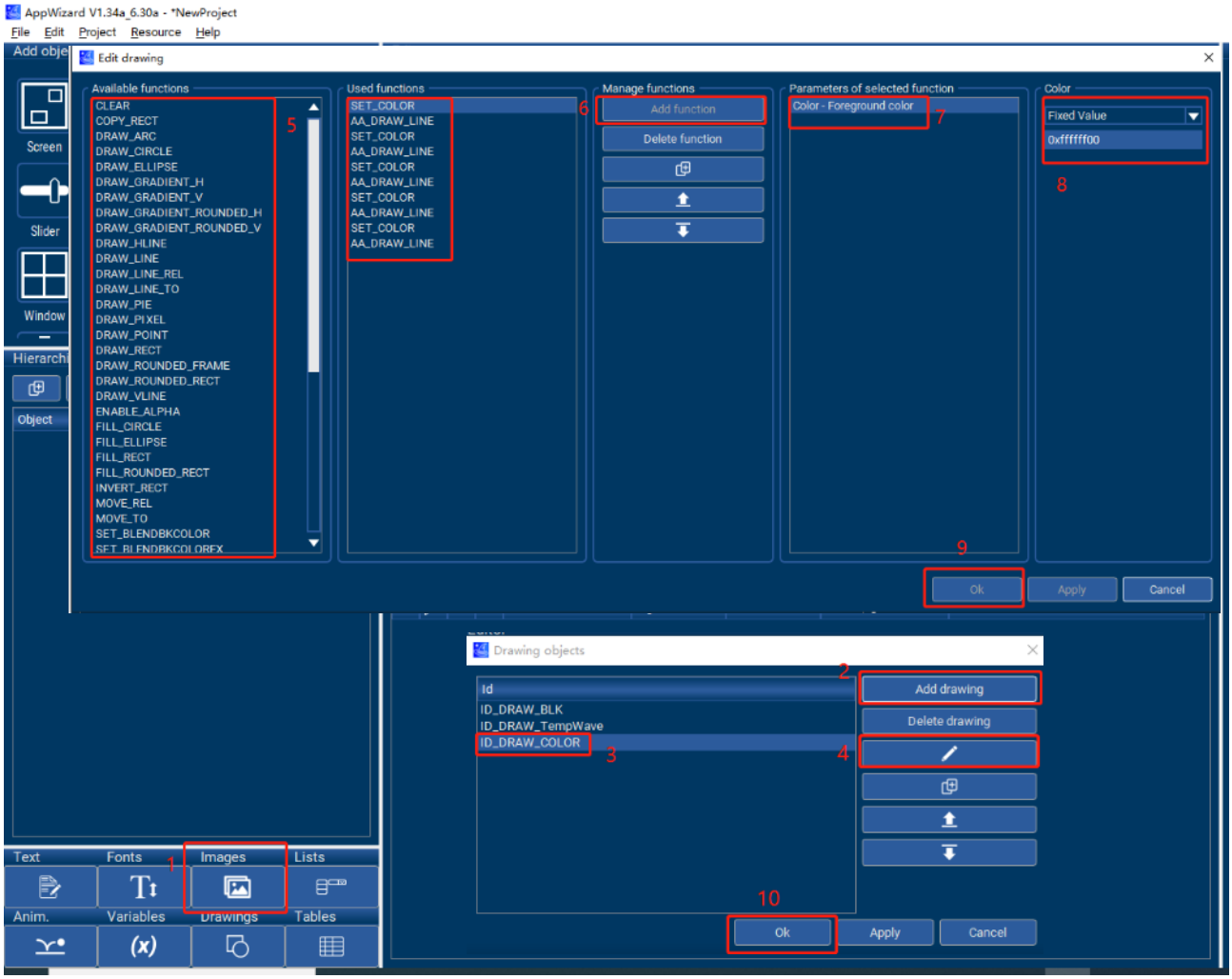


图 1-14 绘制曲线颜色指示

4. 绘制时间-温度曲线

时间-温度曲线画面中设置的曲线绘制区域的大小是 400x200。X 轴 400 个像素点分成 100 个点。Y 轴 200 个像素点表示 0 ℃ ~ 400 ℃，所以 1 个像素点对应 2 ℃。也就是 X 轴有 100 个变量，y 轴有 500 个变量如表 1-9 所示，变量已经在 1.2.2.6 章节定义，相邻两点间绘制线段，实现曲线的绘制。

变量	作用
ID_VAR_X0~ID_X99	时间-温度曲线的横坐标
ID_VAR_A1_Y0~ ID_VAR_A1_Y99	区域 1 时间-温度曲线的纵坐标
ID_VAR_A2_Y0~ ID_VAR_A2_Y99	区域 2 时间-温度曲线的纵坐标
ID_VAR_A3_Y0~ ID_VAR_A3_Y99	区域 3 时间-温度曲线的纵坐标
ID_VAR_A4_Y0~ ID_VAR_A4_Y99	区域 4 时间-温度曲线的纵坐标
ID_VAR_A5_Y0~ ID_VAR_A5_Y99	区域 5 时间-温度曲线的纵坐标

表 1-9 时间-温度曲线的坐标变量

这里需要绘制 5 条不同颜色曲线。绘图动作如表 1-10 所示。绘制动作重名命名为 ID_TimeWave，操作步骤参考区域-温度曲线的绘制图 1-13。

功能	参数	说明
SET_BKCOLOR	BkColor = (Fixed value, 0xff000000)	设置背景颜色为黑色
CLEAR	无	用背景色清除显示内容
SET_COLOR	Color = (Fixed value, 0xffffff00)	设置画图颜色为黄色
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A1_Yn, ID_VAR_Xn+1, ID_VAR_A1_Yn+1) 其中 n = 0~99	绘制区域 1 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xffff0000)	设置画图颜色为红色
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A2_Yn, ID_VAR_Xn+1, ID_VAR_A2_Yn+1) 其中 n = 0~99	绘制区域 2 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xff00ff00)	设置画图颜色为绿色

功能	参数	说明
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A3_Yn, ID_VAR_Xn+1, ID_VAR_A3_Yn+1) 其中 n = 0~99	绘制区域 3 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xff0000ff)	设置画图颜色为蓝色
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A4_Yn, ID_VAR_Xn+1, ID_VAR_A4_Yn+1) 其中 n = 0~99	绘制区域 4 时间-温度曲线
SET_COLOR	Color = (Fixed value，0xff000000)	设置画图颜色为白色
AA_DRAW_LINE	(x0, y0, x1, y1) = (ID_VAR_Xn, ID_VAR_A5_Yn, ID_VAR_Xn+1, ID_VAR_A5_Yn+1) 其中 n = 0~99	绘制区域 5 时间-温度曲线

表 1-10 时间-温度曲线绘制的参数

1.2.2.8 创建主菜单画面

1. 添加画面

在 Add Objects 里点击 Screen 创建一个画面，在 Hierarchic tree 中重命名为 ID_SCREEN_Main。在 Properties 里设置画面的坐标为 (0, 0, 480, 30)。

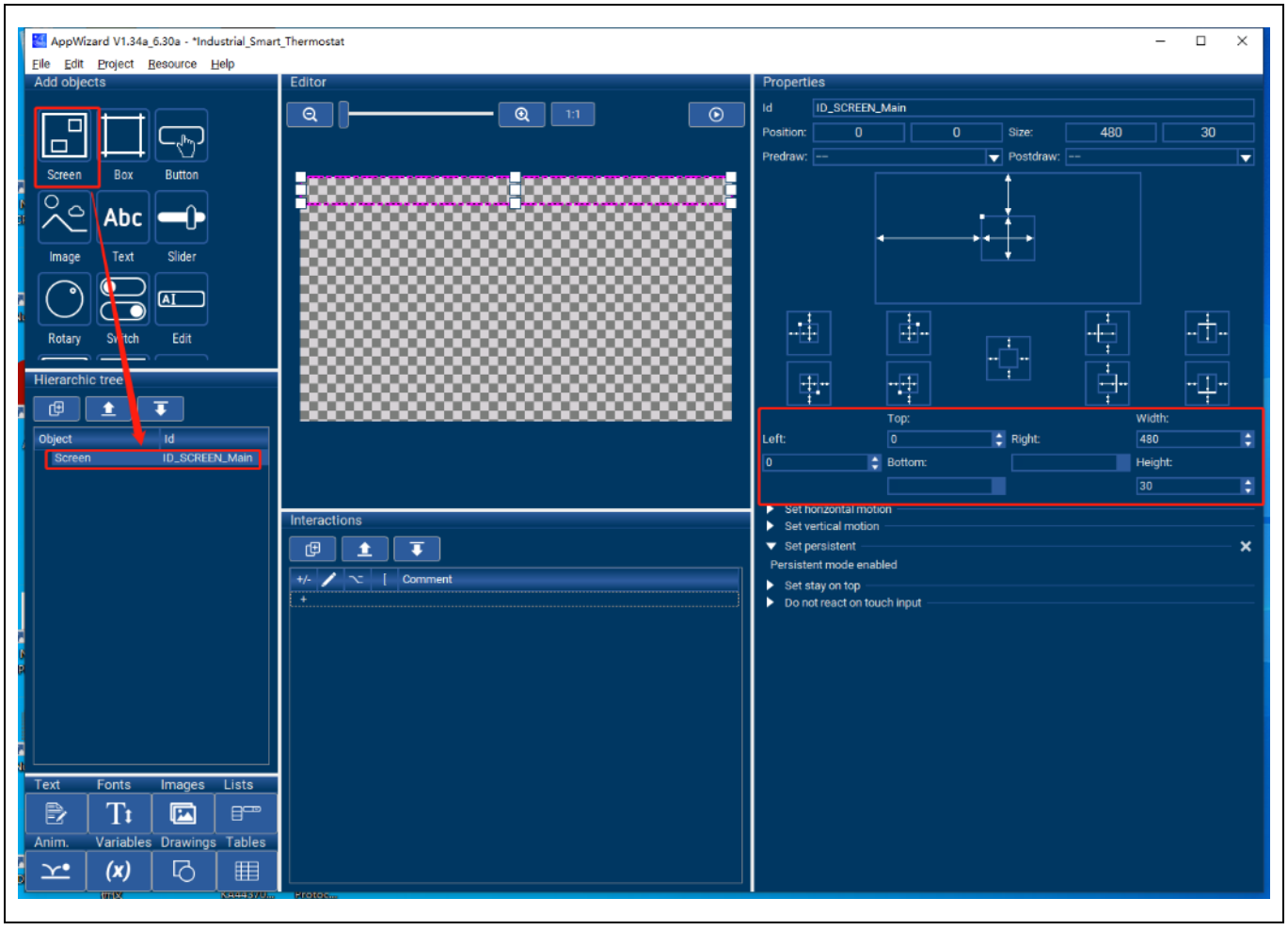


图 1-15 创建 Main 菜单画面

2. 添加素色背景

选中 ID_SCREEN_Main，开始编辑第一个画面。

1) 添加 Box

使用 Box 显示素色的背景。在 Add Objects 里点击 Box，添加一个 Box，并 Hierarchic tree 里重命名为 ID_BOX_BG。

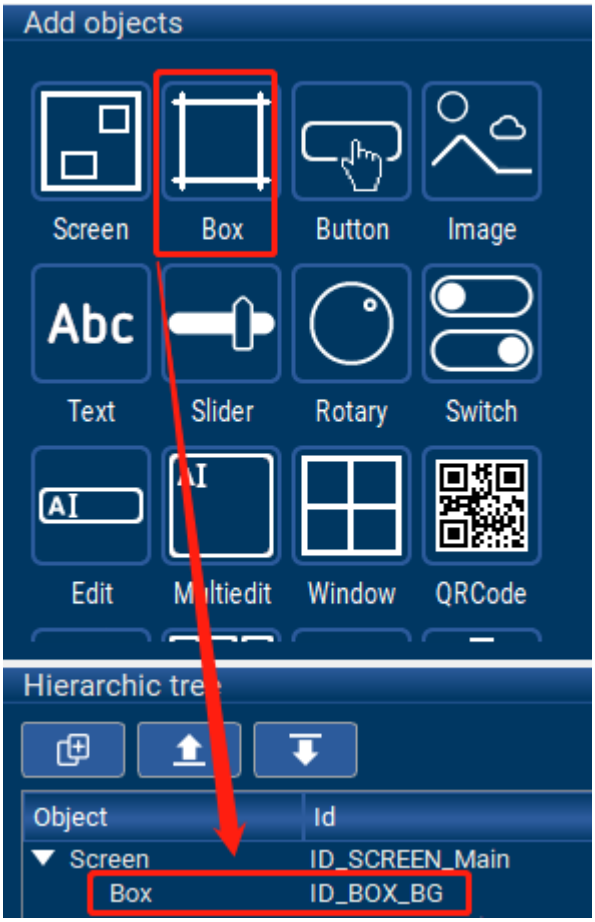


图 1-16 添加 Box - 素色背景

2) 设置 Box 的属性参数

在 Properties 里设置坐标为 (0, 0, 480, 30)，color 为 RGB (123, 125, 123)，这里 color 作为主菜单的背景色。

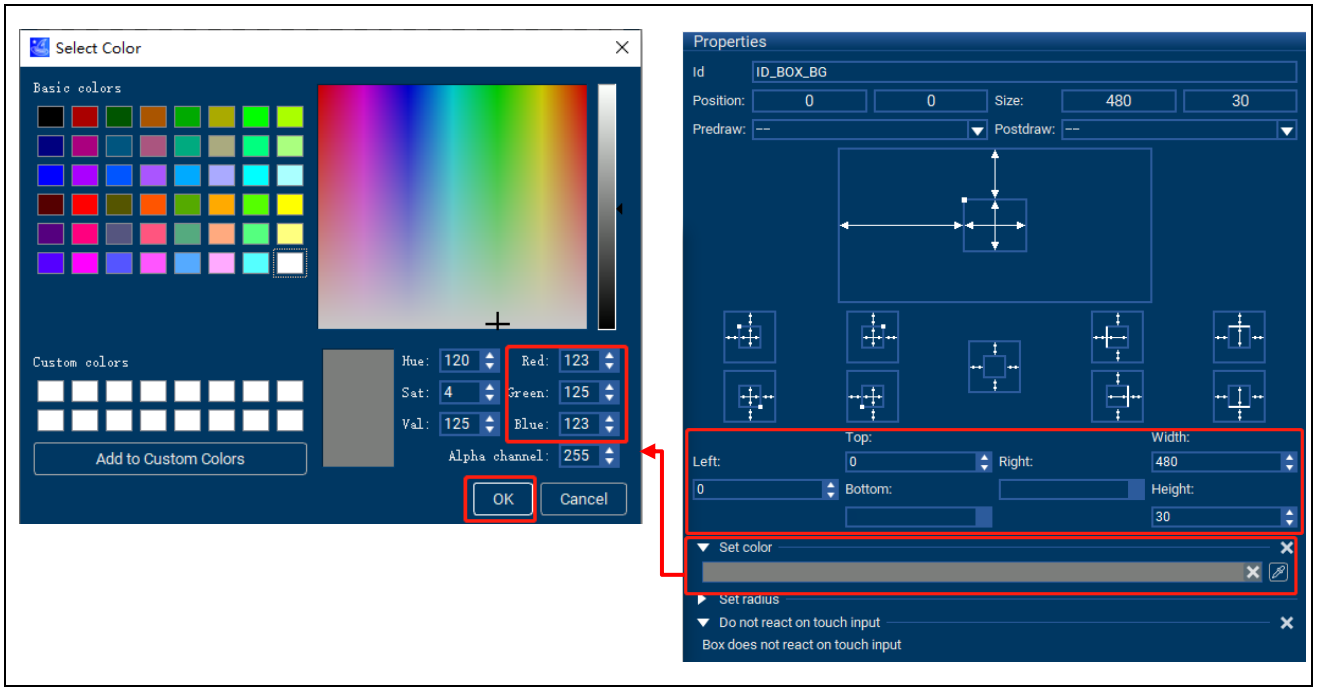


图 1-17 设置主菜单背景色

3. 添加按钮

1) 添加按钮

在 Add Objects 里点击 Button，为画面添加 5 个按键。在 Hierarchic tree 中按照图 1-18 为 Button 重命名。

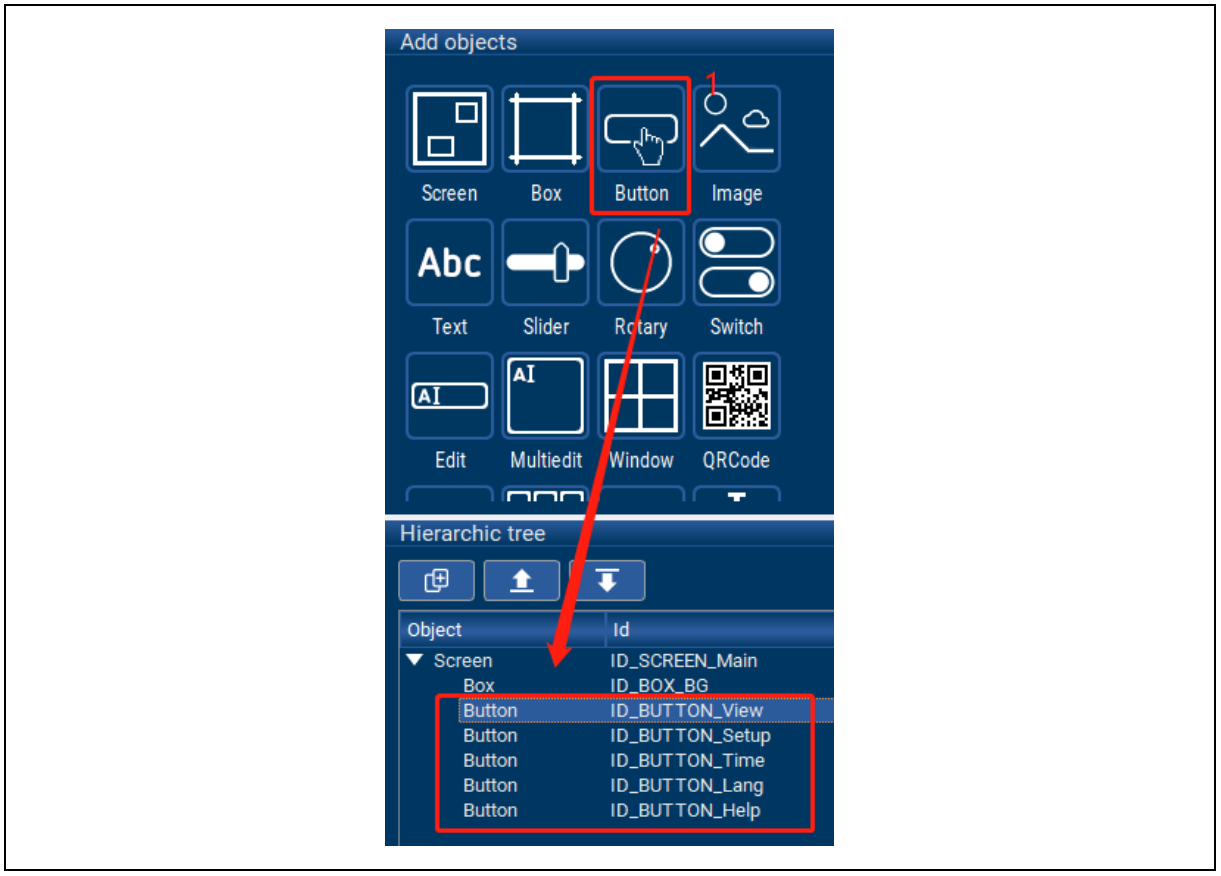


图 1-18 添加按钮 - 主菜单

2) 设置 Button 的属性参数

在 Properties 里删除 bitmaps，设置 text color 为 RGB (0, 0, 0)，text alignment 的 offset x = offset y = 0、Alignment 居左 ，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4，按照表 1-11 设置按钮的坐标和 text。实际设置项目如图 1-19 所示。

Button	**坐标 (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

表 1-11 Button 的属性参数设置说明 - 主菜单

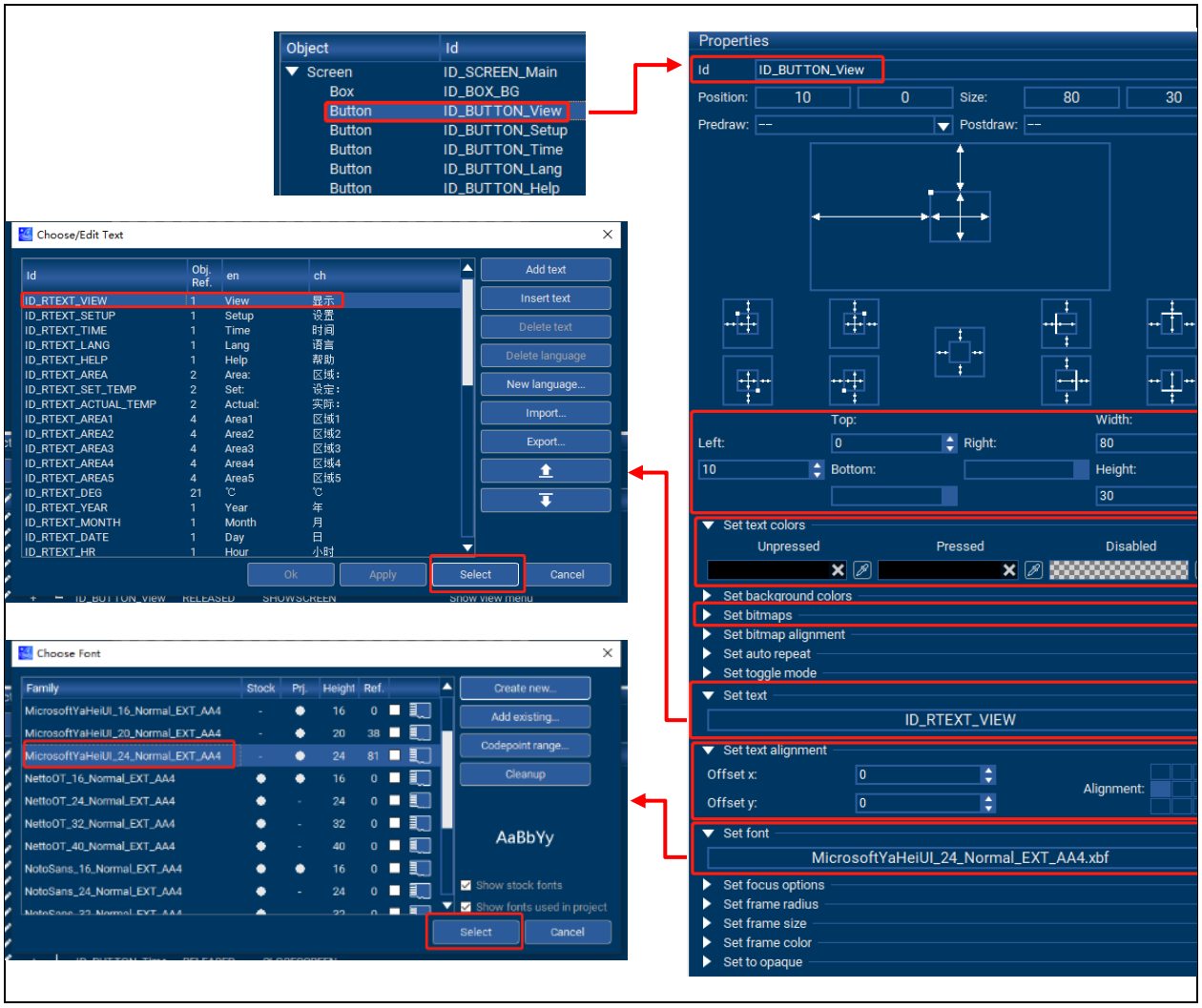


图 1-19 Button 的属性参数设置 - 主菜单

4. 设计完成后的主菜单

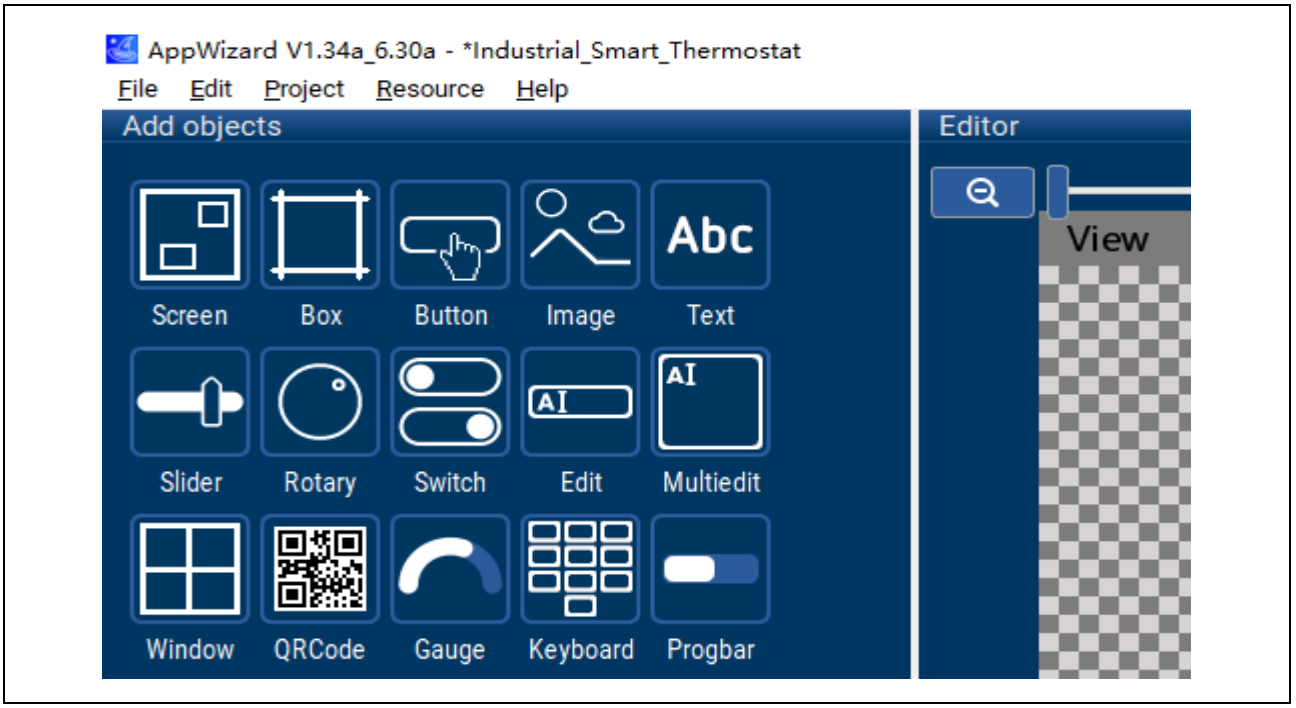


图 1-20 主菜单画面

1.2.2.9 创建下拉菜单画面

1. 添加画面

分别为菜单 View、Setup、Time、Lang 添加一个下拉菜单 (Help 无下拉菜单)。在 Add Objects 里点击 Screen，创建 4 个画面。如表 1-12 所示，在 Hierarchic tree 中重命名，并在 Properties 里设置画面坐标。实际步骤如图 1-21 所示。

ID	坐标 (Left, Top, Width, Height)	说明
ID_SCREEN_MenuView	(10, 30, 250, 76)	显示模式下拉菜单
ID_SCREEN_MenuSetup	(90, 30, 200, 30)	温度设置下拉菜单
ID_SCREEN_MenuTime	(182, 30, 150, 52)	时间设置下拉菜单
ID_SCREEN_MenuLang	(262, 30, 150, 30)	语言设置下拉菜单

表 1-12 画面的属性参数设置说明 - 下拉菜单画面

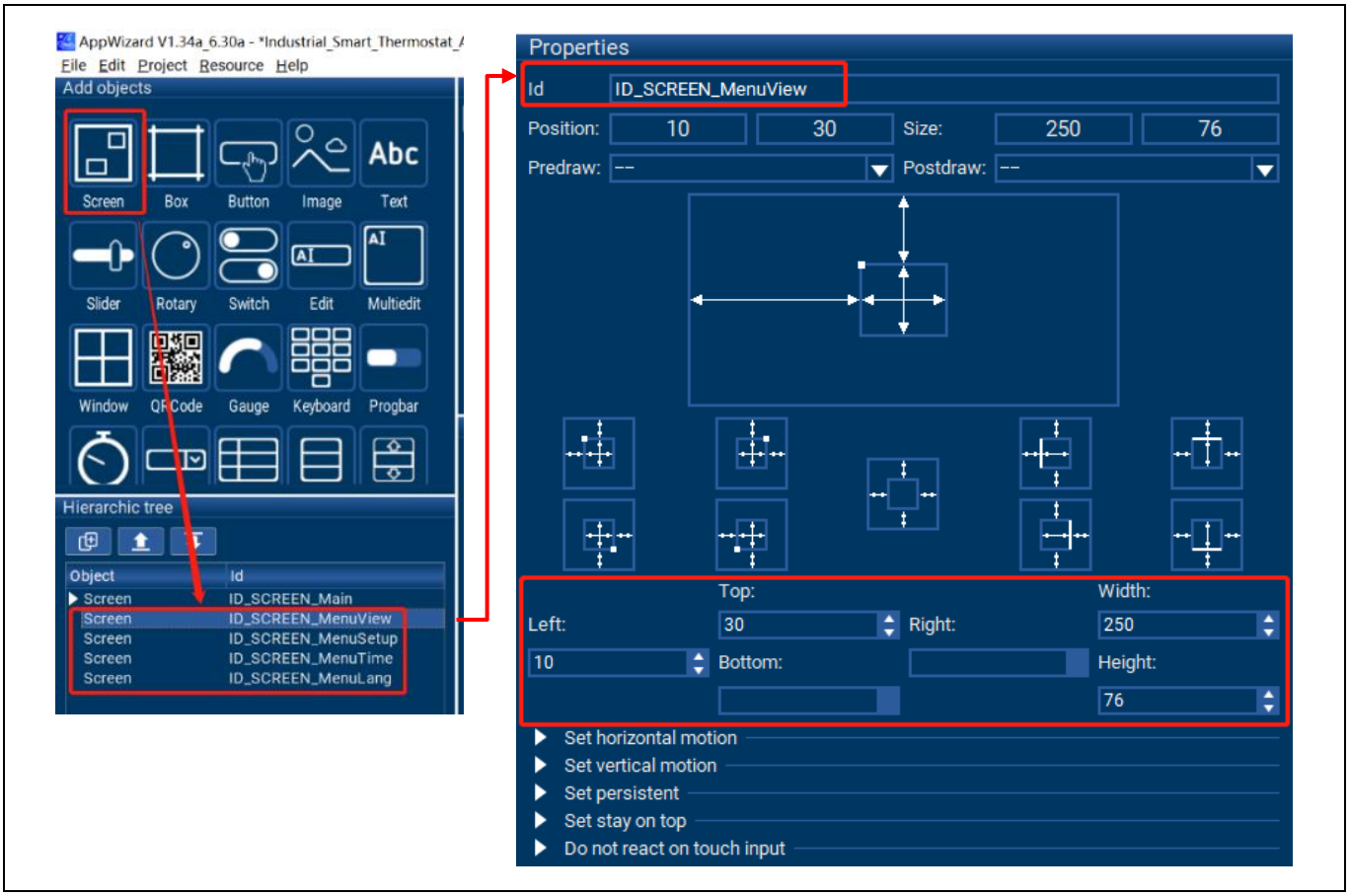


图 1-21 创建下拉菜单画面

2. 添加窗口

为下拉菜单画面分别添加一个窗口，窗口名称和坐标如表 1-13 所示。

ID	**坐标 (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)

表 1-13 窗口属性参数设置说明 - 下拉菜单窗口

在 Hierarchic tree 里选中 ID_SCREEN_MenuView 画面。在 Add Objects 里点击 Window 添加一个窗口。如表 1-13 所示，在 Hierarchic tree 中为窗口重命名，并在 Properties 里设置窗口的坐标。实际操作如图 1-22 所示。按照相同的方法添加其他 3 个窗口。

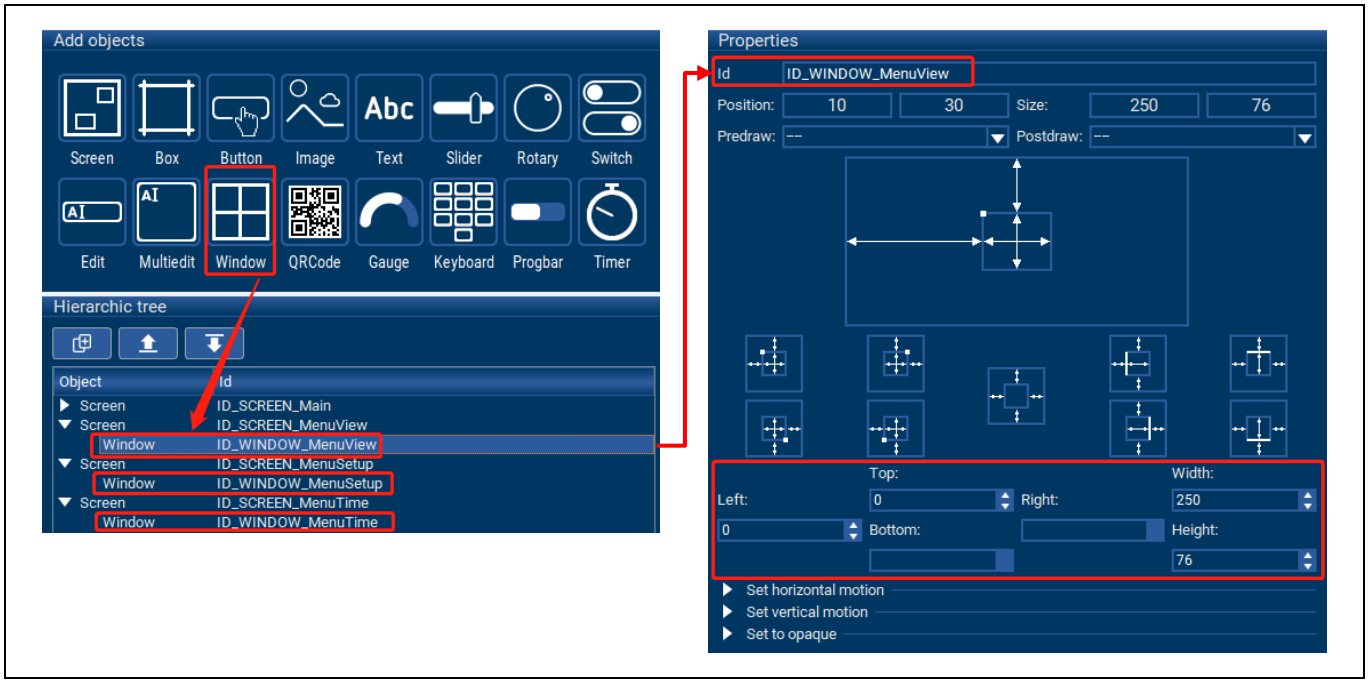


图 1-22 添加下拉菜单窗口

3. 添加 ListBox

ListBox 用于显示下拉菜单的内容。

1) 添加 ListBox

在 Hierarchic tree 中选中 ID_WINDOW_MenuView 。在 Add Objects 里点击 ListBox，添加一个 ListBox，并在 Hierarchic tree 中重命名为 ID_LISTBOX_MenuView。按照同样的方法添加另外 3 个 ListBox。

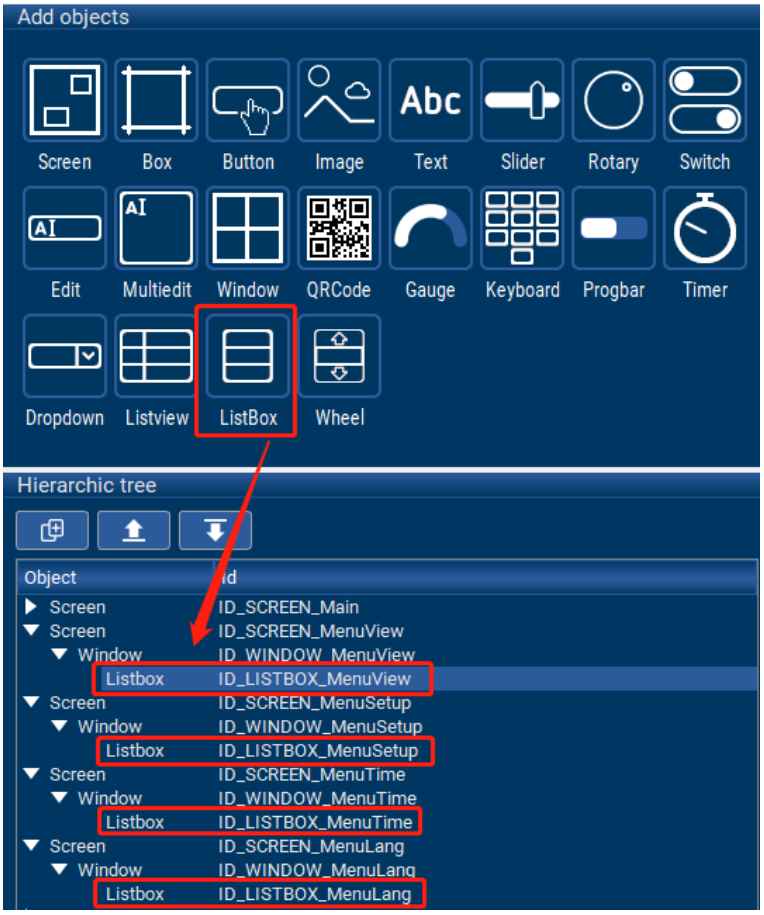


图 1-23 添加 ListBox - 下拉菜单显示

2) ListBox 属性参数设置

在 Properties 里按照表 1-14 设置坐标和 text，并设置 font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4，实际设定项目如图 1-24。按照同样的方法设置另外 3 个 ListBox 的属性参数。

ID	**坐标 (Left, Top, Width, Height)**	**显示内容**
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

表 1-14 ListBox 属性参数设置说明

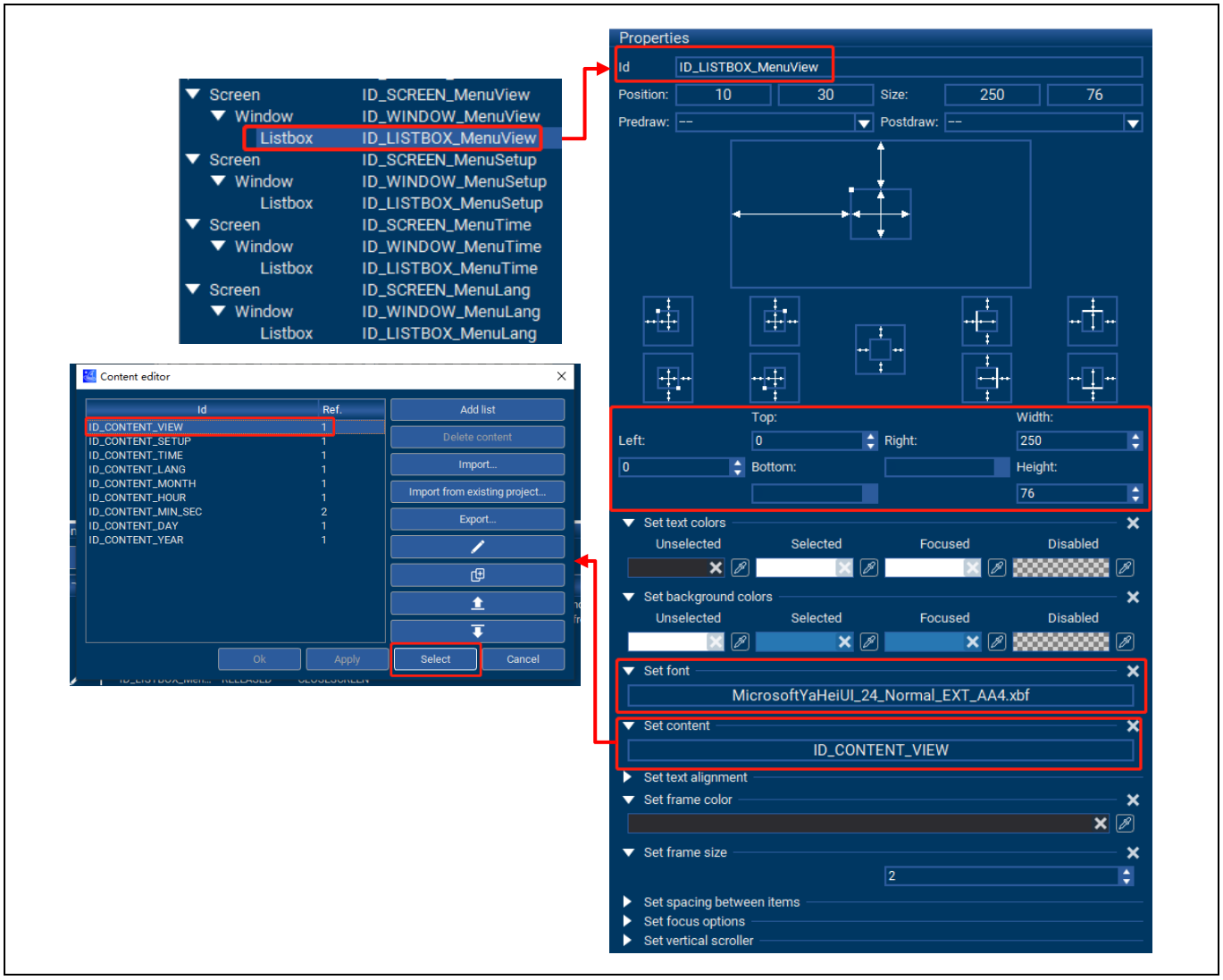


图 1-24 ListBox 属性参数设置

4. 完成后的下拉菜单

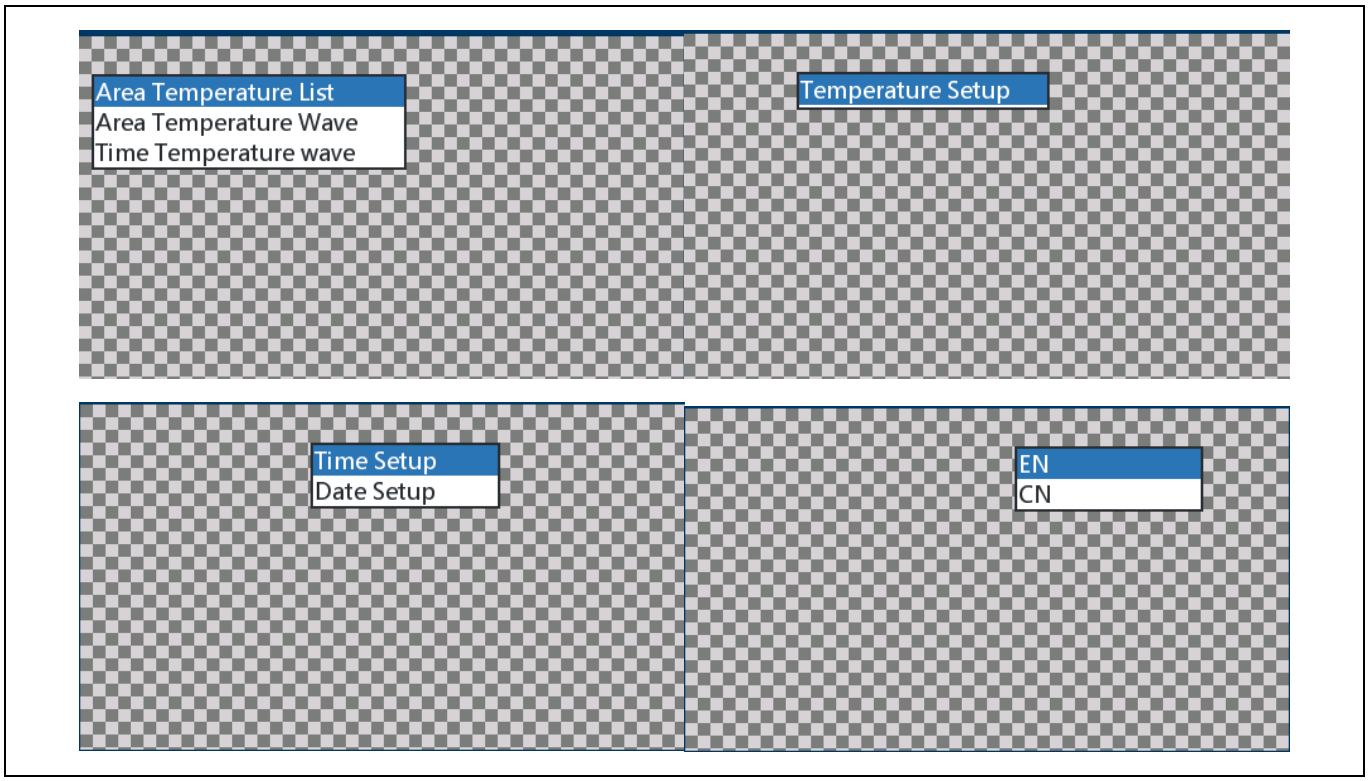


图 1-25 下拉菜单画面

1.2.2.10 创建时间-温度曲线画面

1. 添加画面

在 Add Objects 里点击 Screen 添加一个画面，在 Hierarchic tree 里重命名为 ID_SCREEN_TimeWave。在 Properties 里设置坐标为 (0, 30, 480, 242)。

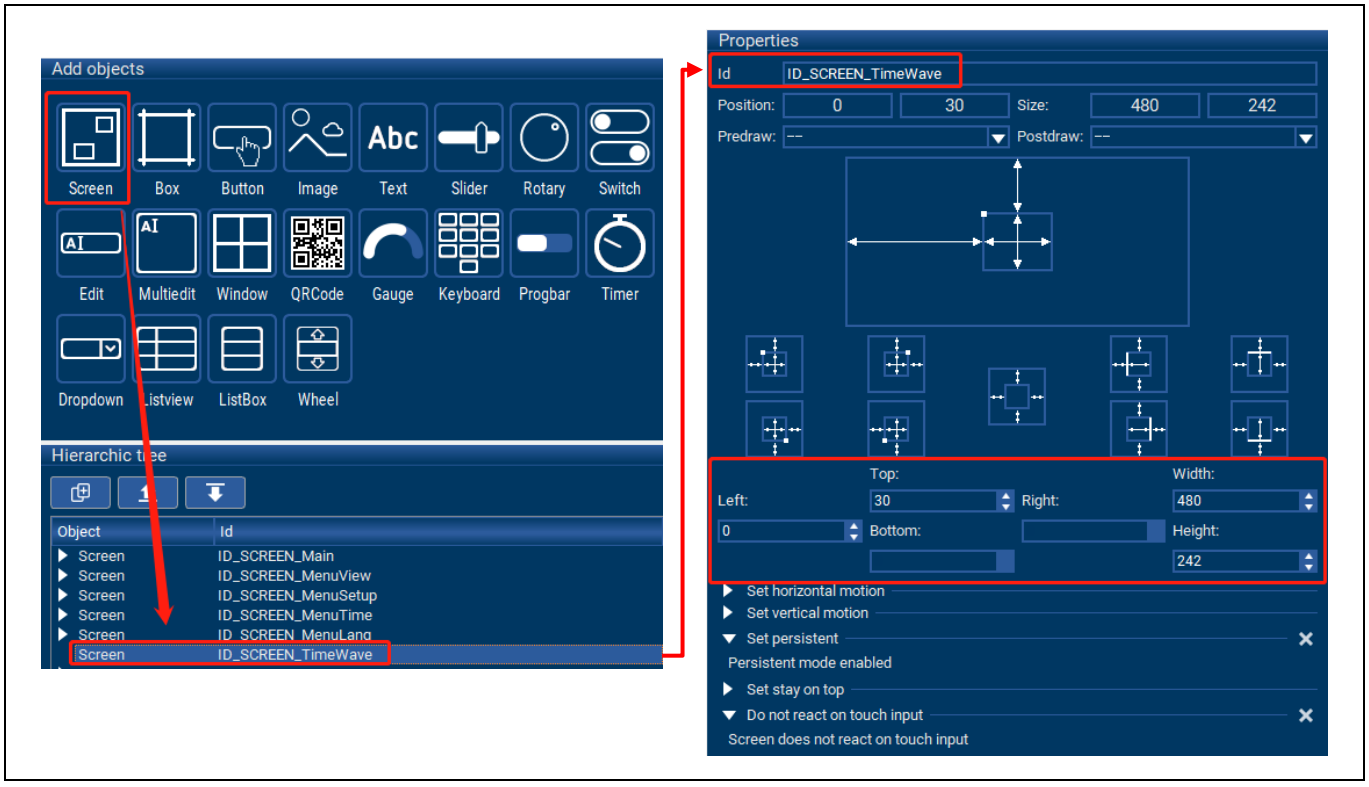


图 1-26 添加时间 - 温度曲线画面

2. 添加坐标显示

用于显示时间-温度曲线的坐标以及指示区域曲线使用的颜色。

1) 添加窗口

在 Add Objects 里点击 Window 添加一个窗口，在 Hierarchic tree 里重命名为 ID_WINDOW_XY。在 Properties 里设置坐标为 (0, 0, 480, 242)。

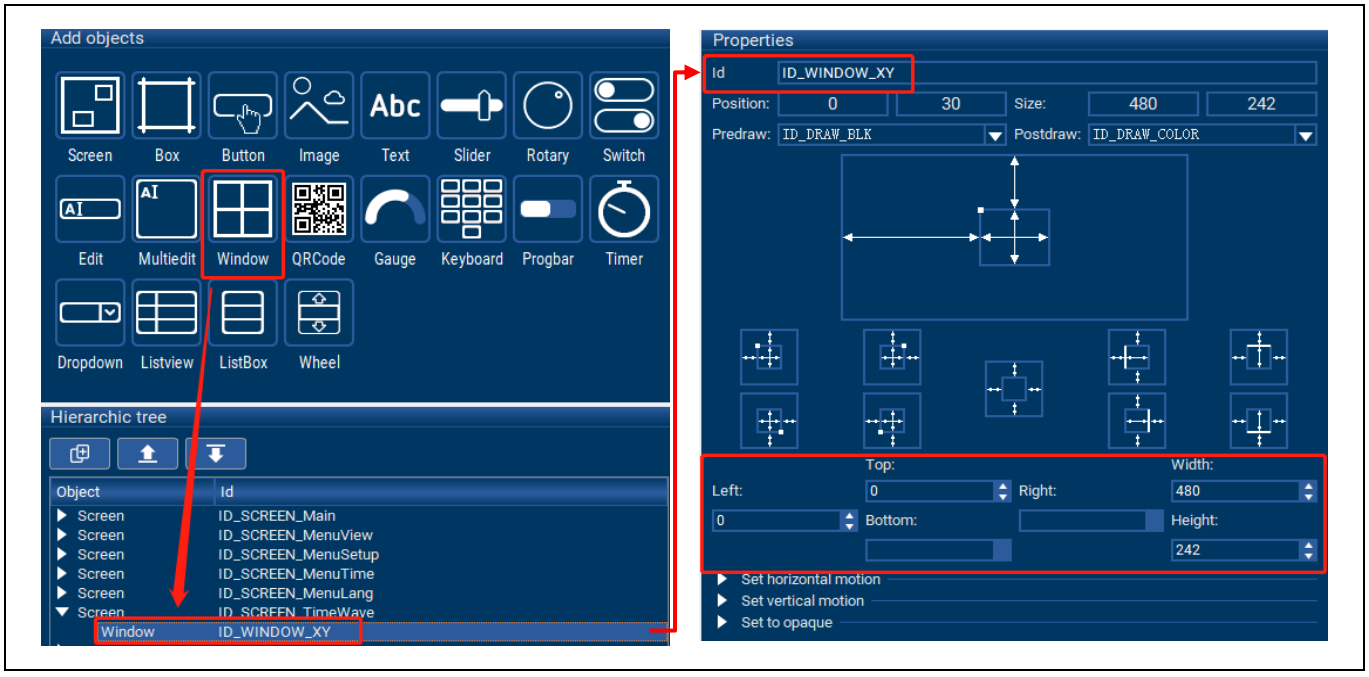


图 1-27 创建时间 - 温度曲线的坐标显示窗口

2) 添加 Inage

在 Add Objects 里点击 Image 添加一个 Image。在 Hierarchic tree 里重命名为 ID_IMAGE_BG，如图 1-28 所示。在 Properties 里设置坐标为 (0, 0, 400, 242)，bitmap 里选择图片 BKG_Coordinate_480x242.png 作为时间-温度曲线背景图片，主要用于显示坐标刻度，实际设置项目如图 1-29。

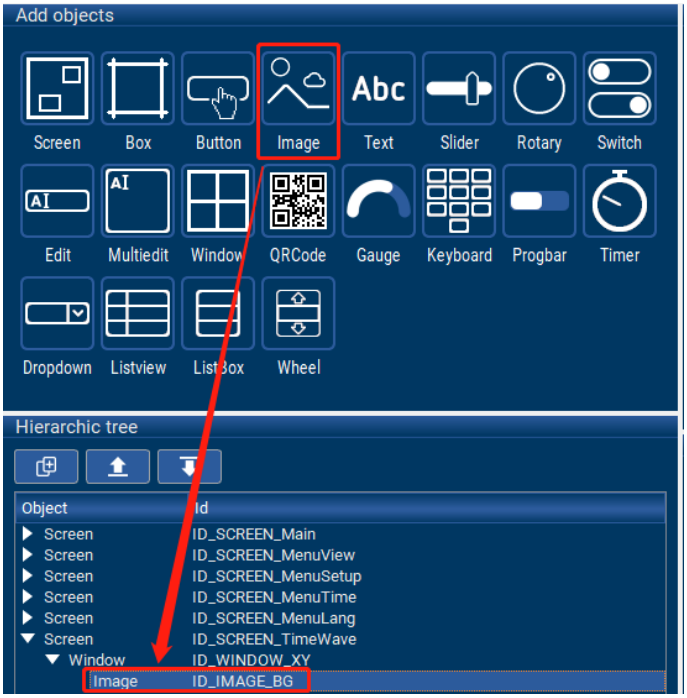


图 1-28 添加 Image - 坐标背景

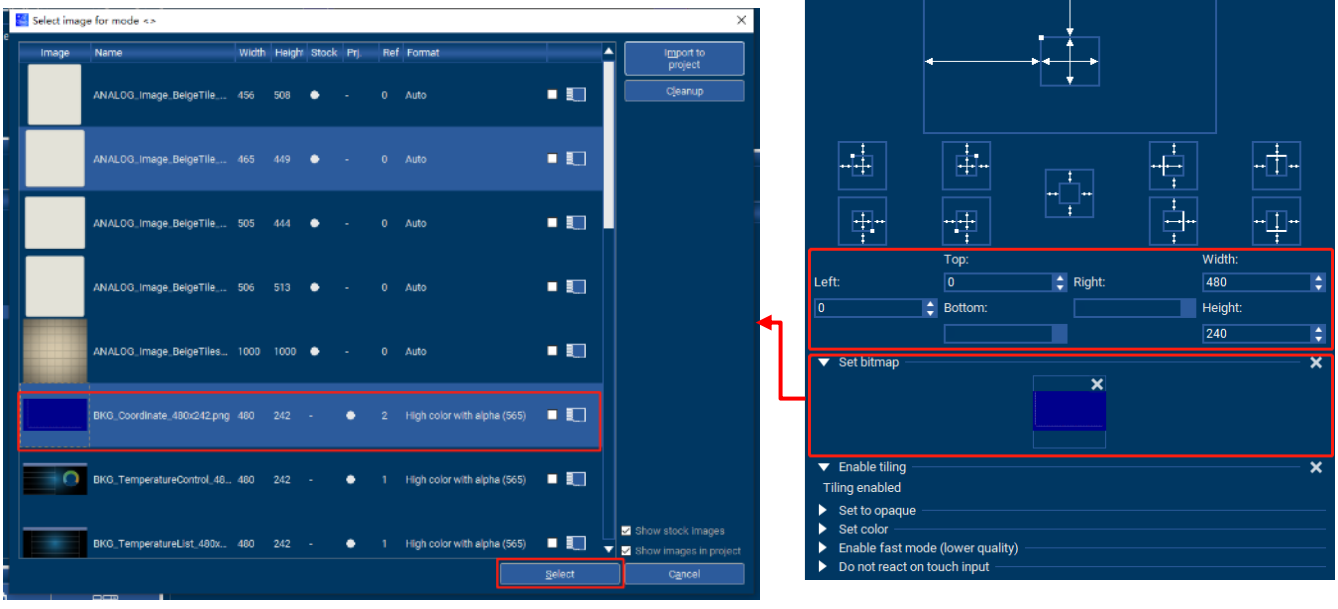


图 1-29 Image 属性参数设置

3) 添加 Text

为 ID_WINDOW_XY 窗口添加 Text，用于显示曲线的横坐标 (时间)、纵坐标 (温度)。

在 Add Objects 中点击 Text，添加 23 个 Text，并在 Hierarchic tree 里按图 1-30 重新命名。

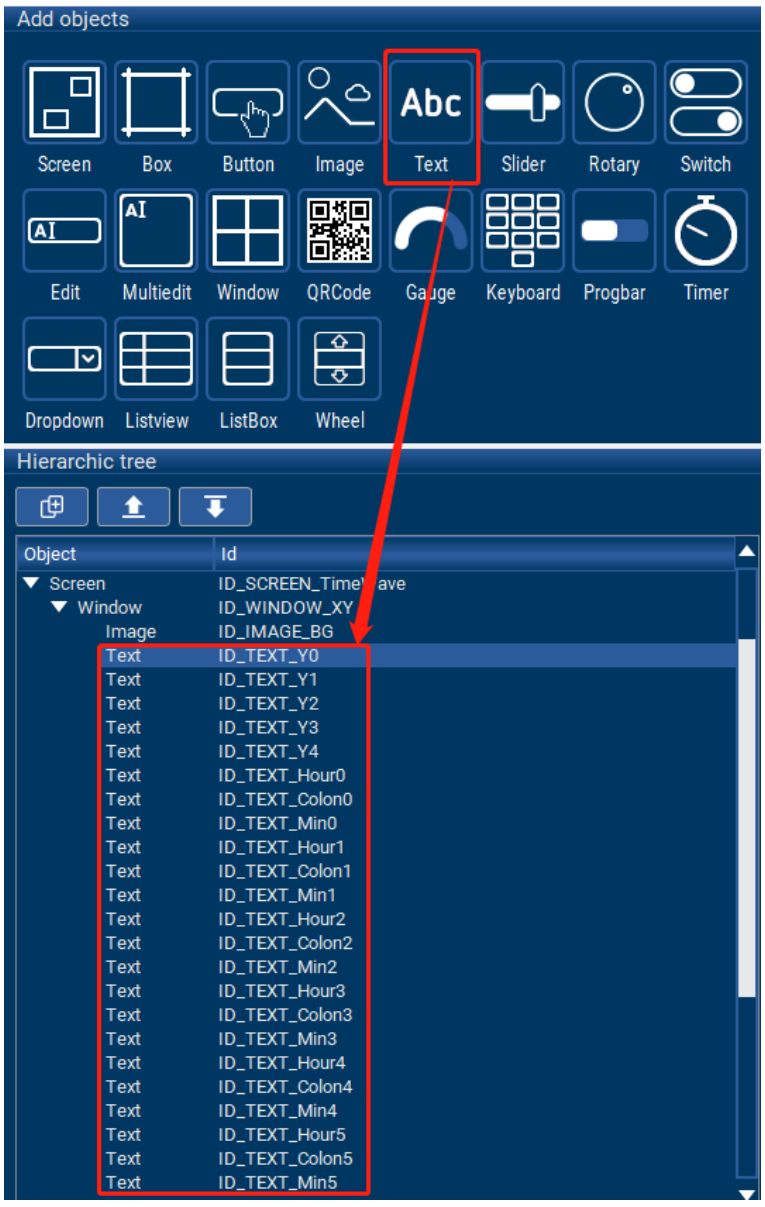


图 1-30 添加 Text - 坐标值显示

其中 6 个 Text 用于显示文本。在 Properties 里设置 text color 为 RGB (255, 255, 255)，text alignment 的 offset x = offset y = 0、Alignment 为居中，font 为

MicrosoftYaHeiUI_20_Normal_EXT_AA4。再按照表 1-15 设置坐标、text。实际设置项目如图 1-31。

ID	坐标 (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

表 1-15 Text 属性参数设置说明 - 冒号显示

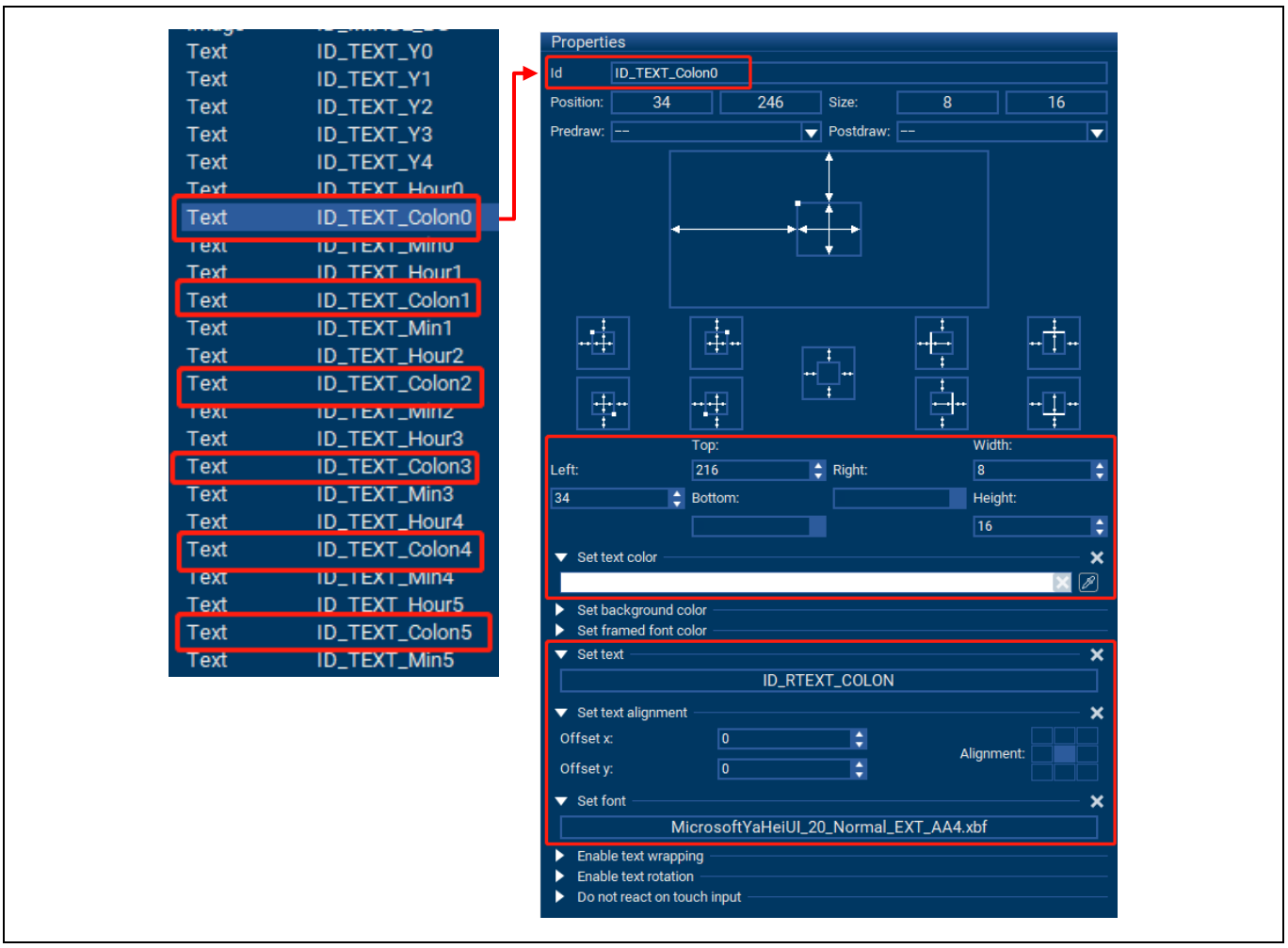


图 1-31 Text 属性参数设置 - 冒号显示

其中 17 个 Text 用于显示数字，包含温度和时间。在 Properties 里设置 text color 为 RGB (255, 255, 255)，font 为 MicrosoftYaHeiUI_20_Normal_EXT_AA4。再按照表 1-16 设置坐标、decimal mode、rang、text alignment，实际设置项目如图 1-32。

ID	坐标 (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, 居右)
ID_TEXT_Y1	(2, 146, 24, 20)	Mask:	(100, 100)	(0, 0, 居右)
ID_TEXT_Y2	(2, 96, 24, 20)	Mask:	(200, 200)	(0, 0, 居右)
ID_TEXT_Y3	(2, 46, 24, 20)	Mask:	(300, 300)	(0, 0, 居右)
ID_TEXT_Y4	(2, 0, 24, 20)	Mask:	(400, 400)	(0, 0, 居右)
ID_TEXT_Hour0	(18, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, 居中)
ID_TEXT_Min0	(42, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, 居中)
ID_TEXT_Hour1	(98, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, 居中)
ID_TEXT_Min1	(122, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, 居中)
ID_TEXT_Hour2	(178, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, 居中)
ID_TEXT_Min2	(202, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, 居中)
ID_TEXT_Hour3	(258, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, 居中)
ID_TEXT_Min3	(282, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, 居中)
ID_TEXT_Hour4	(338, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, 居中)
ID_TEXT_Min4	(362, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, 居中)
ID_TEXT_Hour5	(418, 216, 16, 16)	Mask:00	(0, 23)	(0, 0, 居中)
ID_TEXT_Min5	(442, 216, 16, 16)	Mask:00	(0, 59)	(0, 0, 居中)

表 1-16 Text 属性参数设置说明 - 数字显示

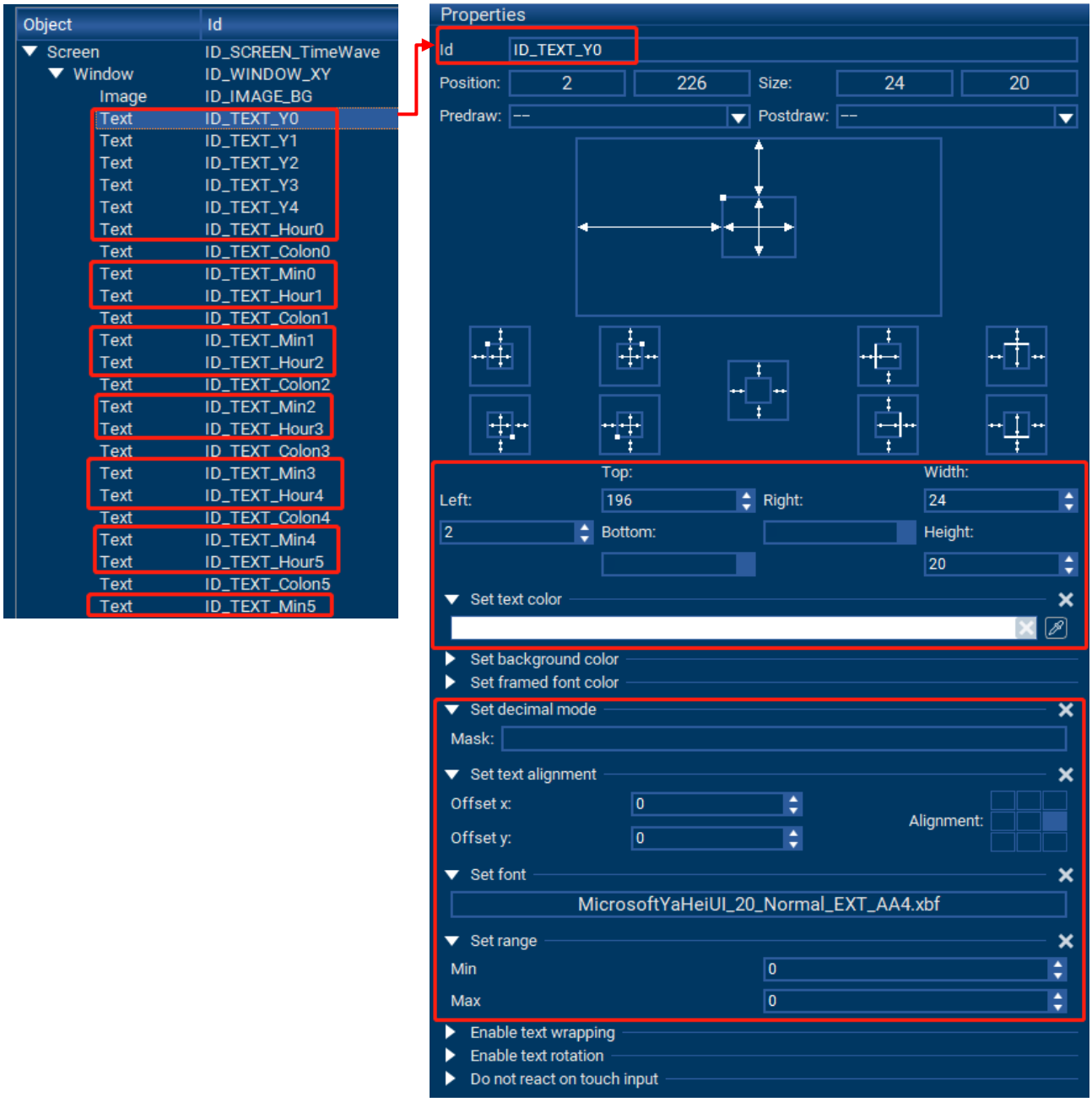


图 1-32 Text 属性参数设置 - 数字显示

4) 添加 Box

绘制 5 条不同颜色的线段来指示每条曲线使用的颜色。在 Add Objects 里点击 Box，为窗口添加一个 Box，在 Hierarchic tree 里重命名为 ID_BOX_0。在 Properties 里设置坐标为 (440, 0, 40, 200)，color 为 RGB (0, 0, 140)，在 Postdraw 的下拉菜单里选择 ID_DRAW_COLOR 用于绘制五条不同颜色的线段，实际设置项目如图 1-33。

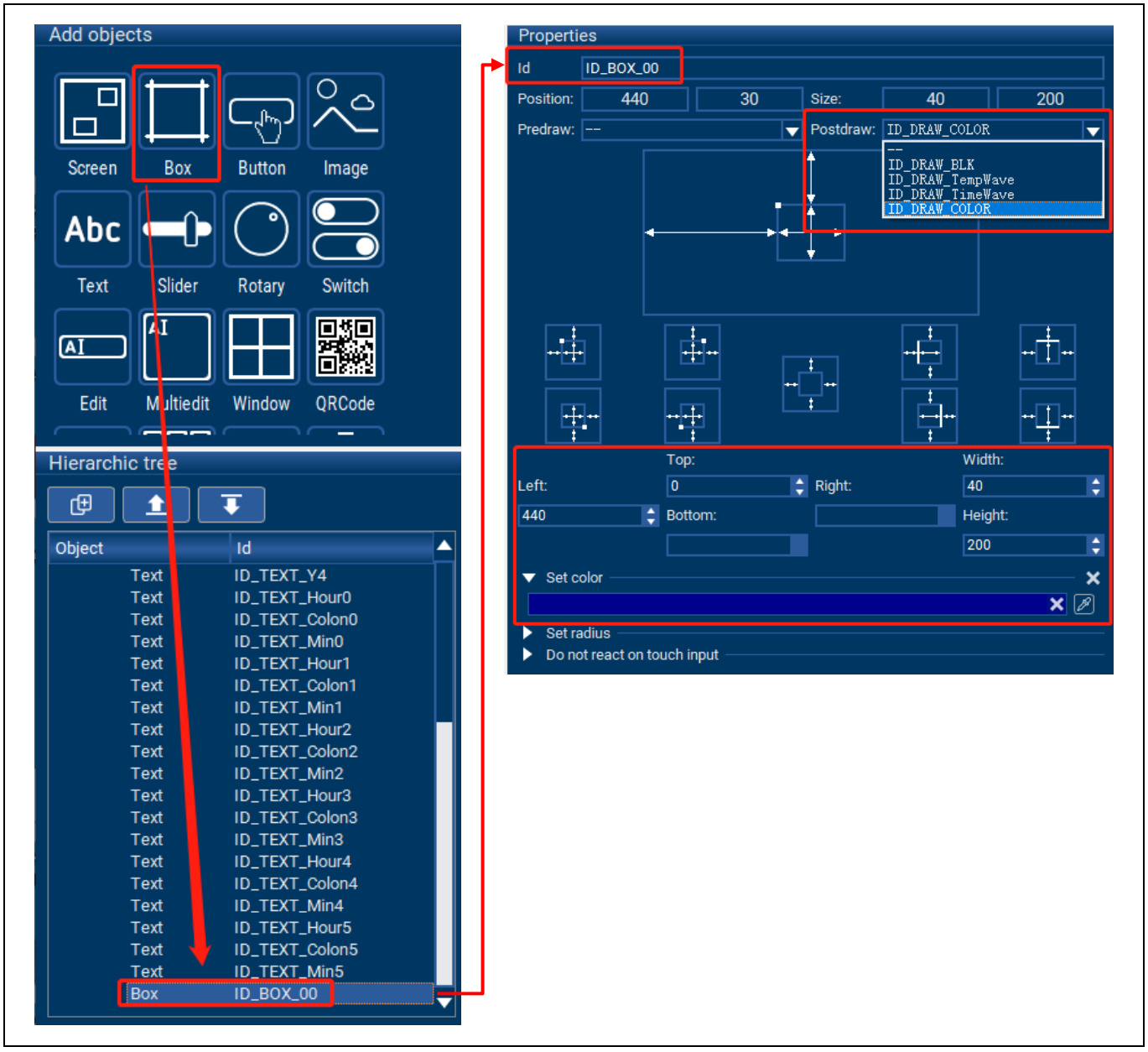


图 1-33 绘制五条不同颜色线段

5) 添加 Text

添加 5 个 Text 用于显示区域名称，和上一步绘制的 5 条线段组合在一起，指示每条曲线的颜色。

在 Add Objects 中点击 Text，添加 5 个 Text，并在 Hierarchic tree 里重新命名，如表 1-17 所示。在 Properties 里设置 text color 为 RGB (255, 255, 255)，text alignment 的 offset x=offset y= 0 、Alignment 为居中，font 为

MicrosoftYaHeiUI_20_Normal_EXT_AA4。按照表 1-17 设置坐标和 text。实际设置项目如图 1-34。

ID	坐标 (left, top, width, height)	文本内容
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

表 1-17 Text 属性参数设置说明 - 区域名称显示

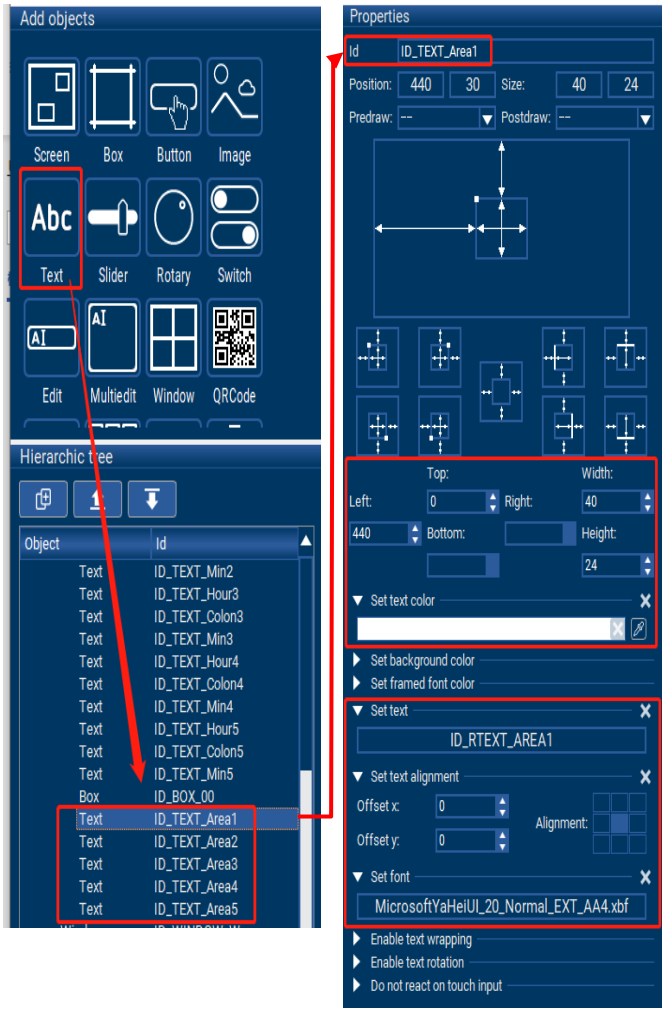


图 1-34 添加 Text - 区域名称显示

3. 添加窗口

添加一个窗口用于绘制时间-温度曲线。首先选中 ID_SCREEN_TimeWave，在 Add Objects 里点击 Window 添加一个窗口，并在 Hierarchic tree 里重命名为 ID_WINDOW_Wave。在 Properties 里设置坐标为 (38, 5, 400, 200)，Predraw 的下拉菜单里选择预画图动作为 ID_DRAW_BLK，Postdraw 的下拉菜单选择画图动作为 ID_Draw_TimeWave。

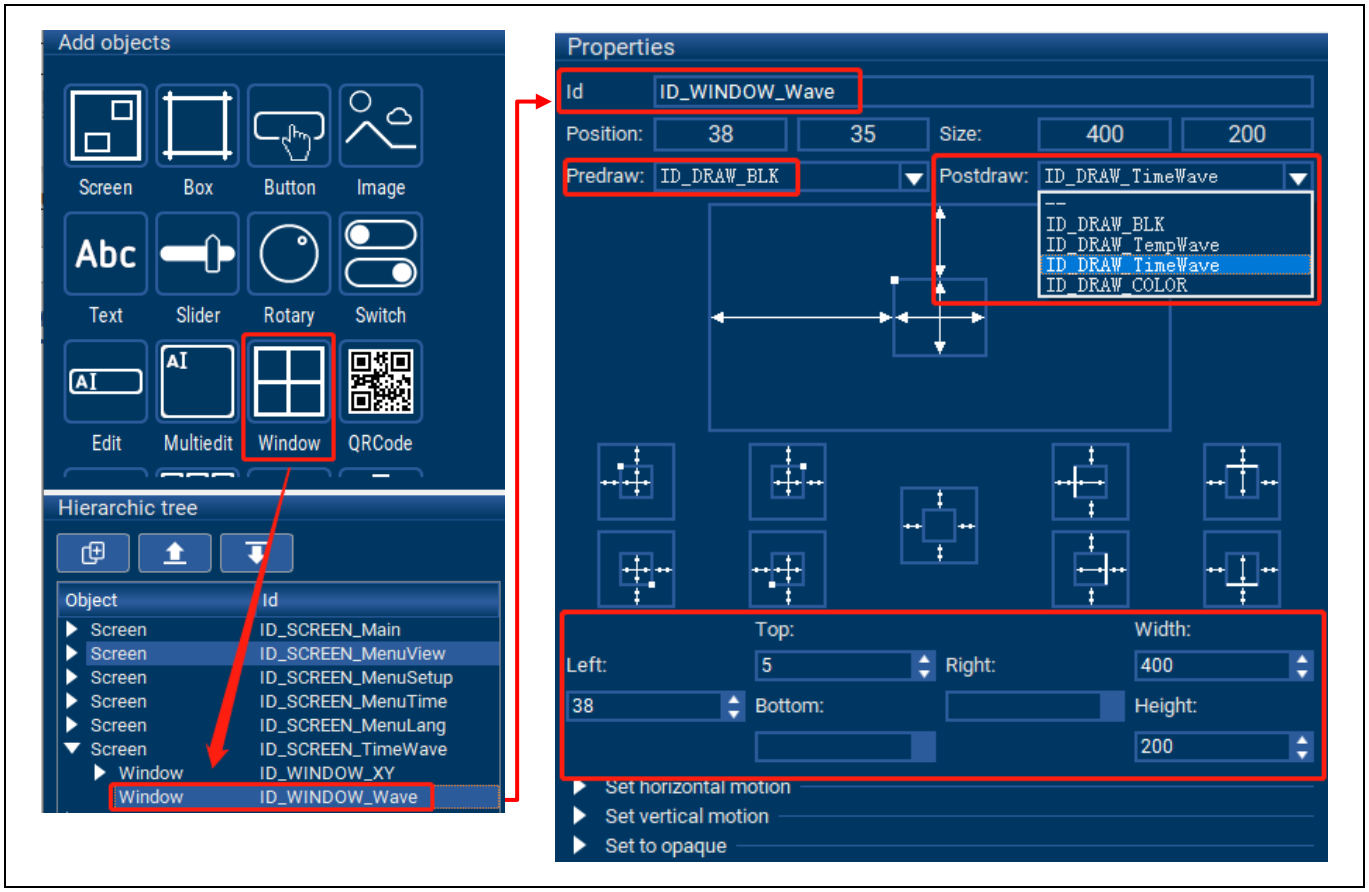


图 1-35 添加时间 - 温度曲线画图窗口

4. 添加定时器

添加一个定时器，用于定时刷新曲线。在 Add Objects 里点击 Timer 添加一个定时器。在 Properties 里设置 timer period 定时时间为 1000 ms，Timer 不存在任何显示，仅在后台运行。

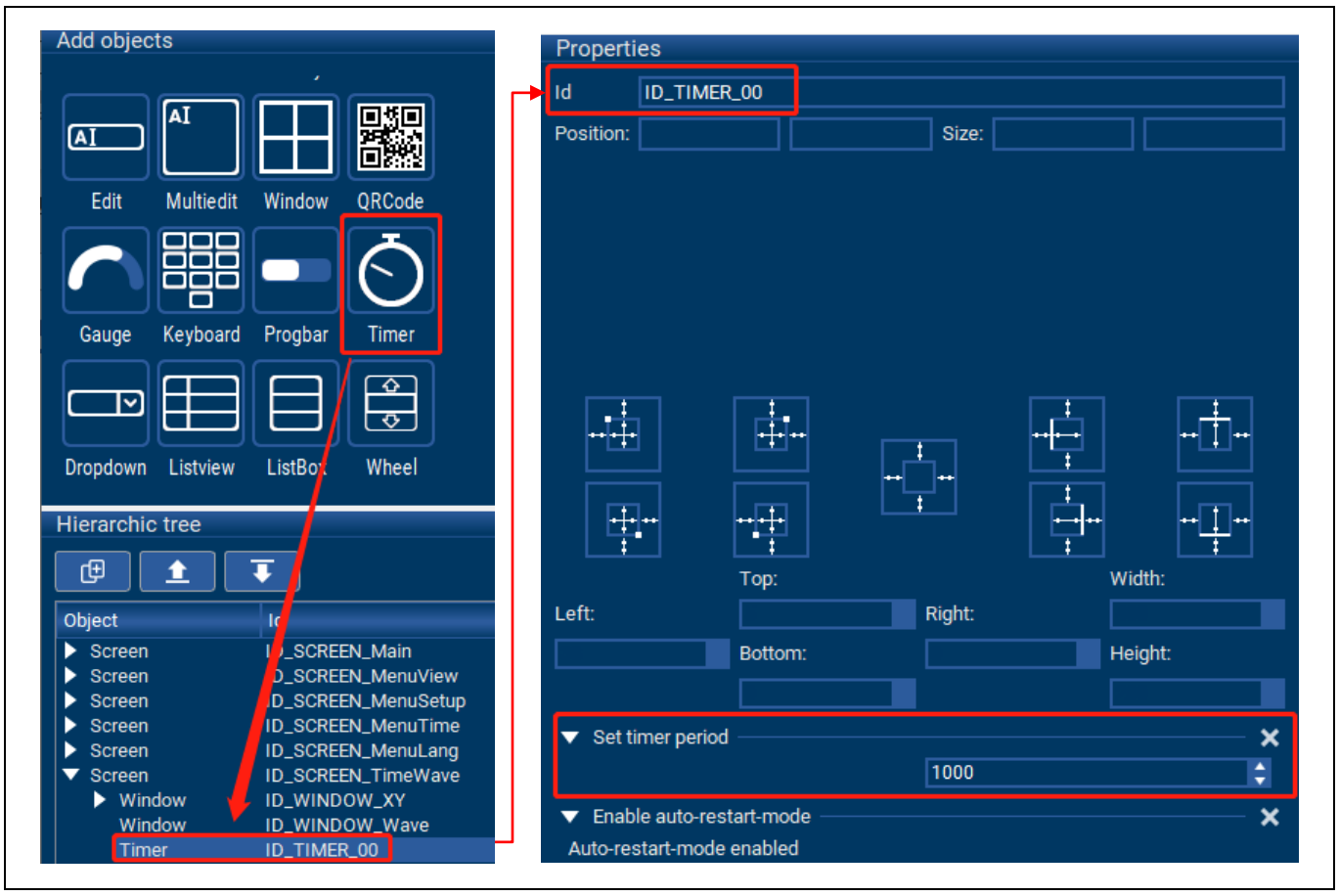

Add objects
Edit
Multiedit
Window
QRCode
Gauge
Keyboard
Progbar
Timer
Dropdown
Listview
ListBox
Wheel
Hierarchic tree
Object
Screen
Screen
Screen
Screen
Screen
Screen
Window
Window
Timer
ID_SCREEN_Main
ID_SCREEN_MenuView
ID_SCREEN_MenuSetup
ID_SCREEN_MenuTime
ID_SCREEN_MenuLang
ID_SCREEN_TimeWave
ID_WINDOW_XY
ID_WINDOW_Wave
ID_TIMER_00
Properties
Id
ID_TIMER_00
Position:
Size:
Top:
Width:
Left:
Right:
Bottom:
Height:
Set timer period
1000
Enable auto-restart-mode
Auto-restart-mode enabled

图 1-36 添加定时器

5. 设计完成后的时间-温度曲线画面

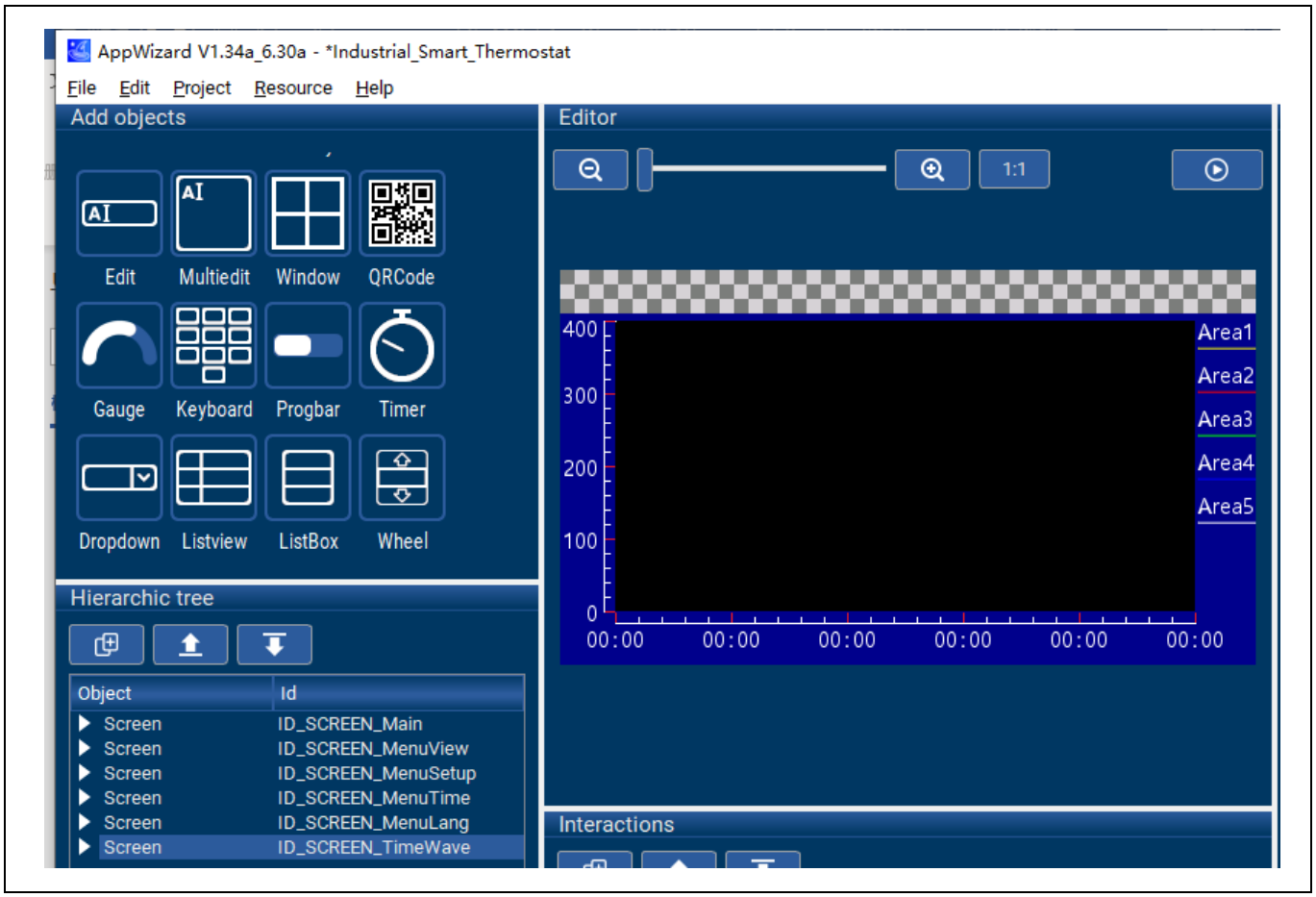


图 1-37 时间 - 温度曲线画面

1.2.2.11 创建区域-温度曲线画面

1. 添加画面

在 Add Objects 里点击 Screen 添加一个画面，在 Hierarchic tree 里重命名为 ID_SCREEN_TempWave。在 Properties 里设置坐标为 (0, 30, 480, 242)。

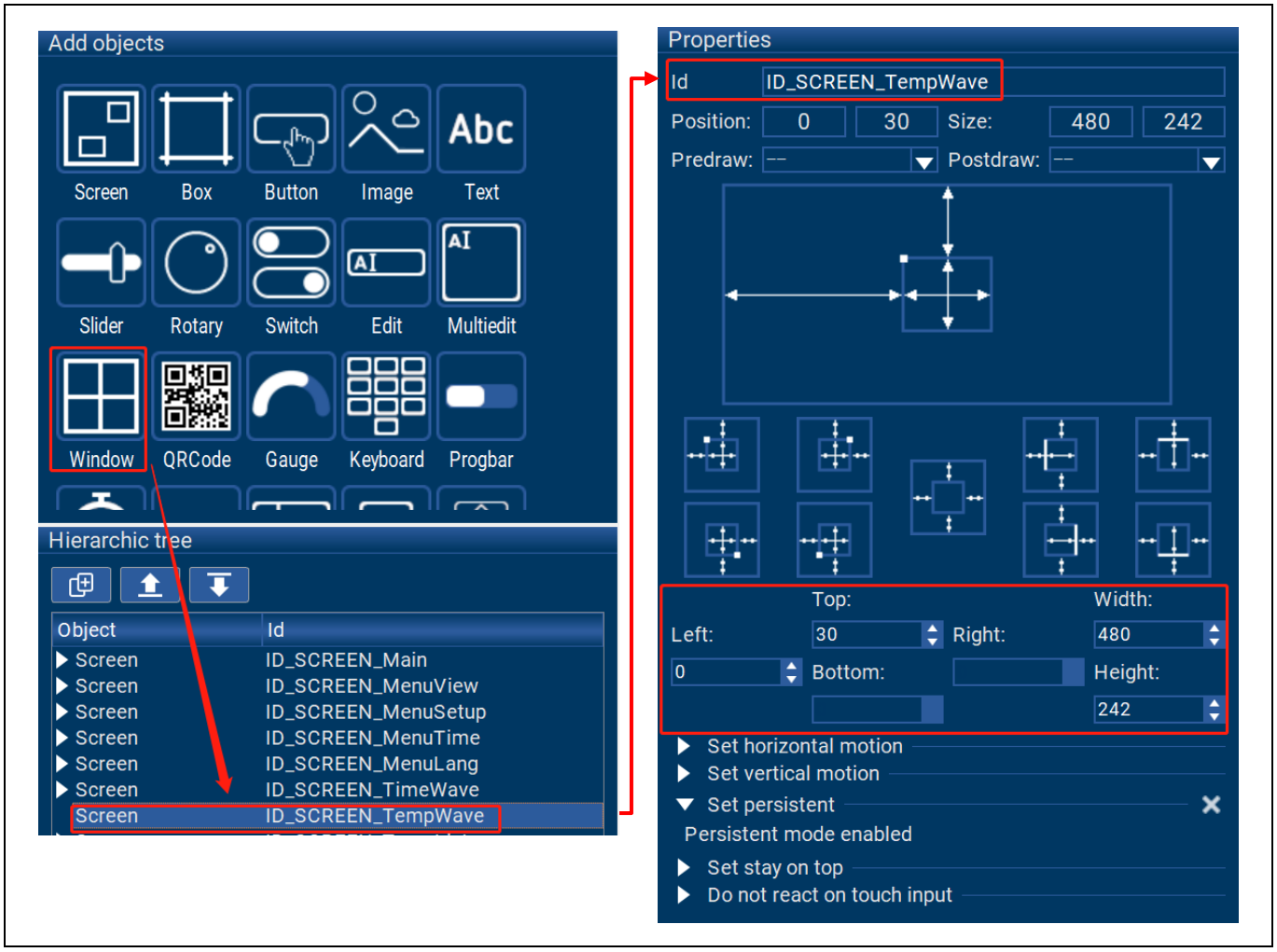


图 1-38 添加区域 - 温度曲线画面

2. 添加坐标显示

主要用于显示区域-温度曲线的坐标。

1) 添加窗口

在 Add Objects 里点击 Window 添加一个窗口，在 Hierarchic tree 里重命名为 ID_WINDOW_XY。在 Properties 里设置坐标为 (0, 0, 480, 242)。

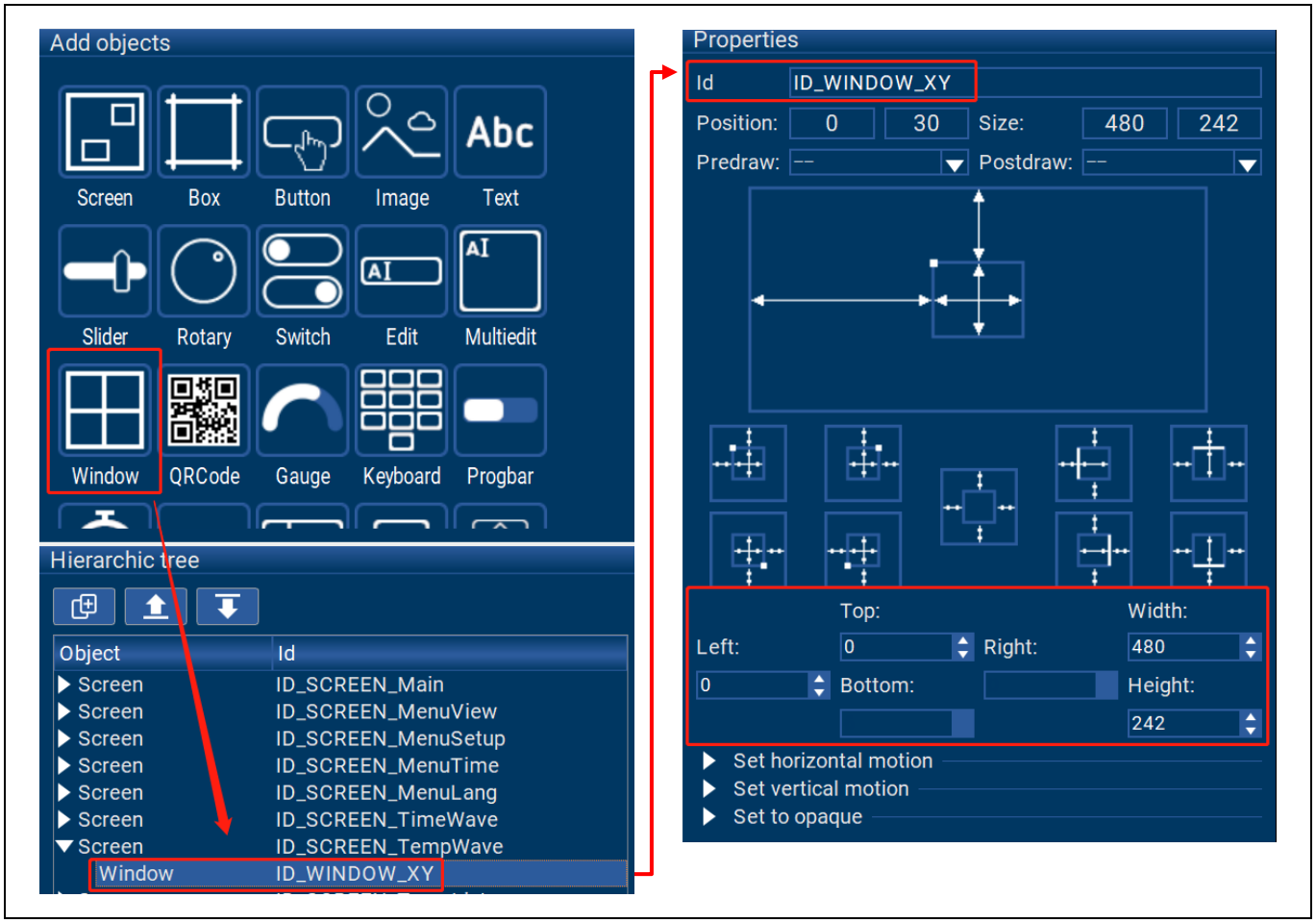


图 1-39 创建窗口 - 区域-温度曲线的坐标

2) 添加 Image

在 Add Objects 里点击 Image，为窗口添加一个 Image，并在 Hierarchic tree 里重命名为 ID_IMAGE_BG。在 Properties 里设置坐标为 (0, 0, 400, 242)，bitmap 里选择图片 BKG_Coordinate_480x242.png 作为区域-温度曲线画面的背景图片，实际设置项目如图 1-40 和图 1-41 所示。

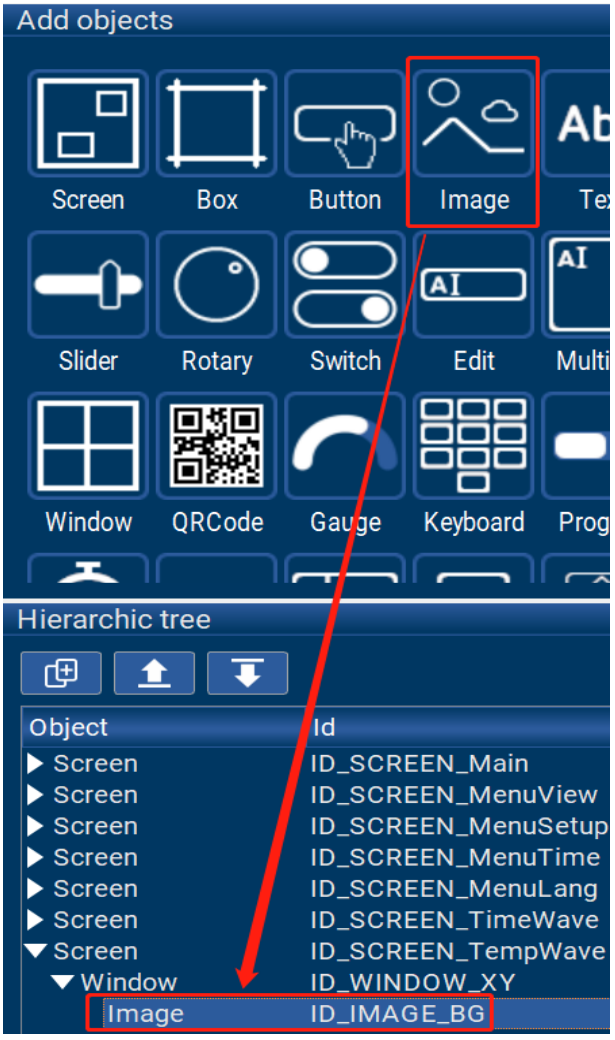


图 1-40 添加 Image - 区域温度坐标背景

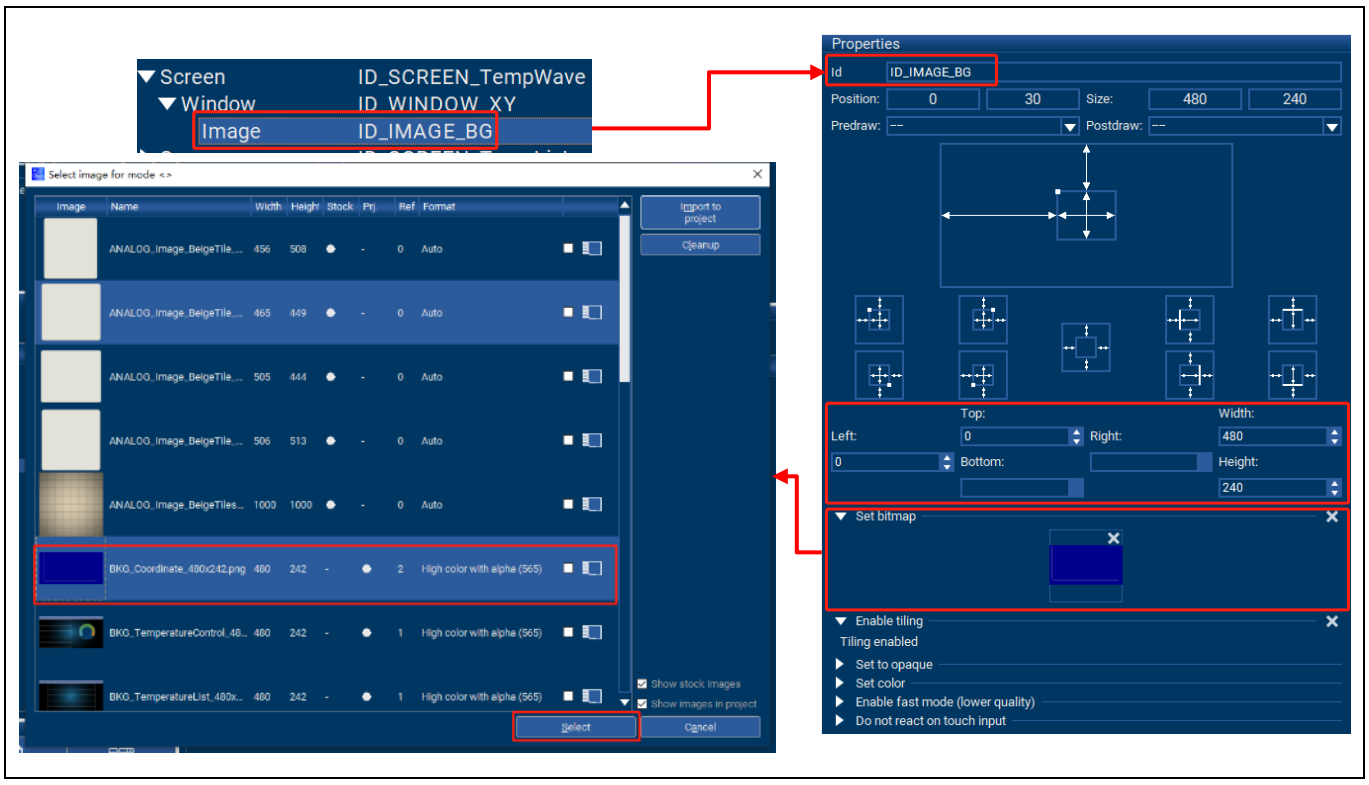


图 1-41 Image 属性参数设置

3) 添加 Text

为 ID_WINDOW_XY 窗口添加 Text，用于显示曲线的纵坐标（温度）和横坐标（区域名称）。在 Add Objects 中点击 Text，添加 10 个 Text，并在 Hierarchic tree 按图 1-42 重新命名。

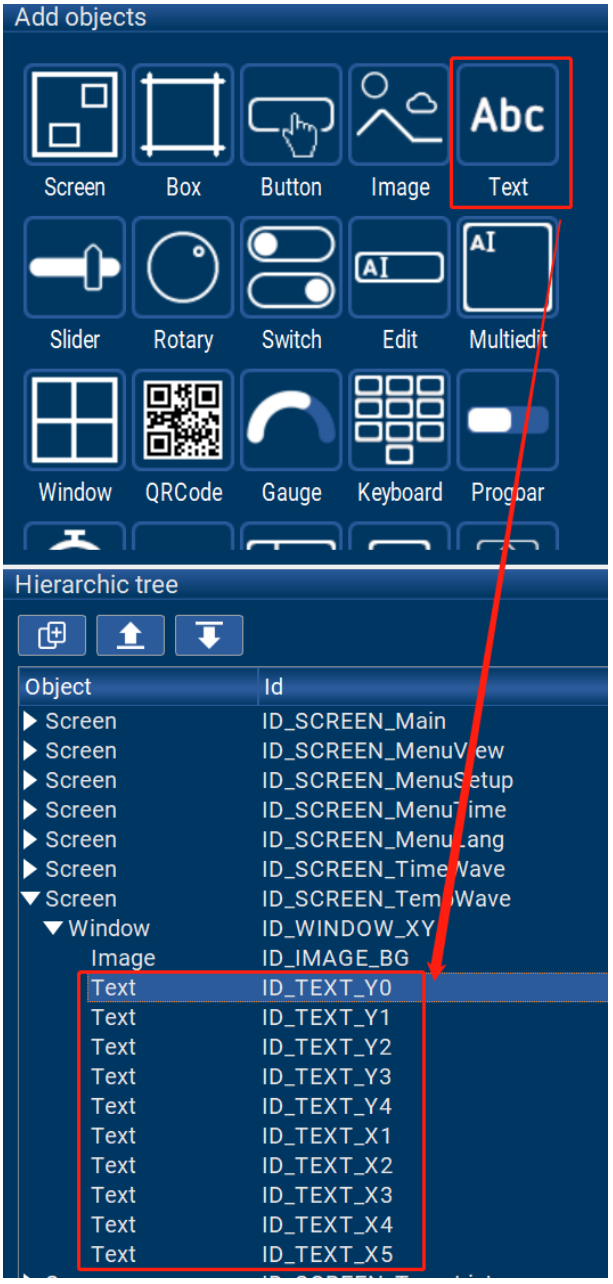


图 1-42 添加 Text - 坐标值显示

其中 5 个 Text 用于显示文本。在 Properties 里设置 text color 为 RGB (255, 255, 255)，text alignment 的 offset x = offset y = 0、Alignment 为居中，font 为 MicrosoftYaHeiUI_20_Normal_EXT_AA4，按表 1-18 设置坐标、text。实际设置项目如图 1-43。

ID	坐标 (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

表 1-18 Text 属性参数设置说明 - 横坐标值显示

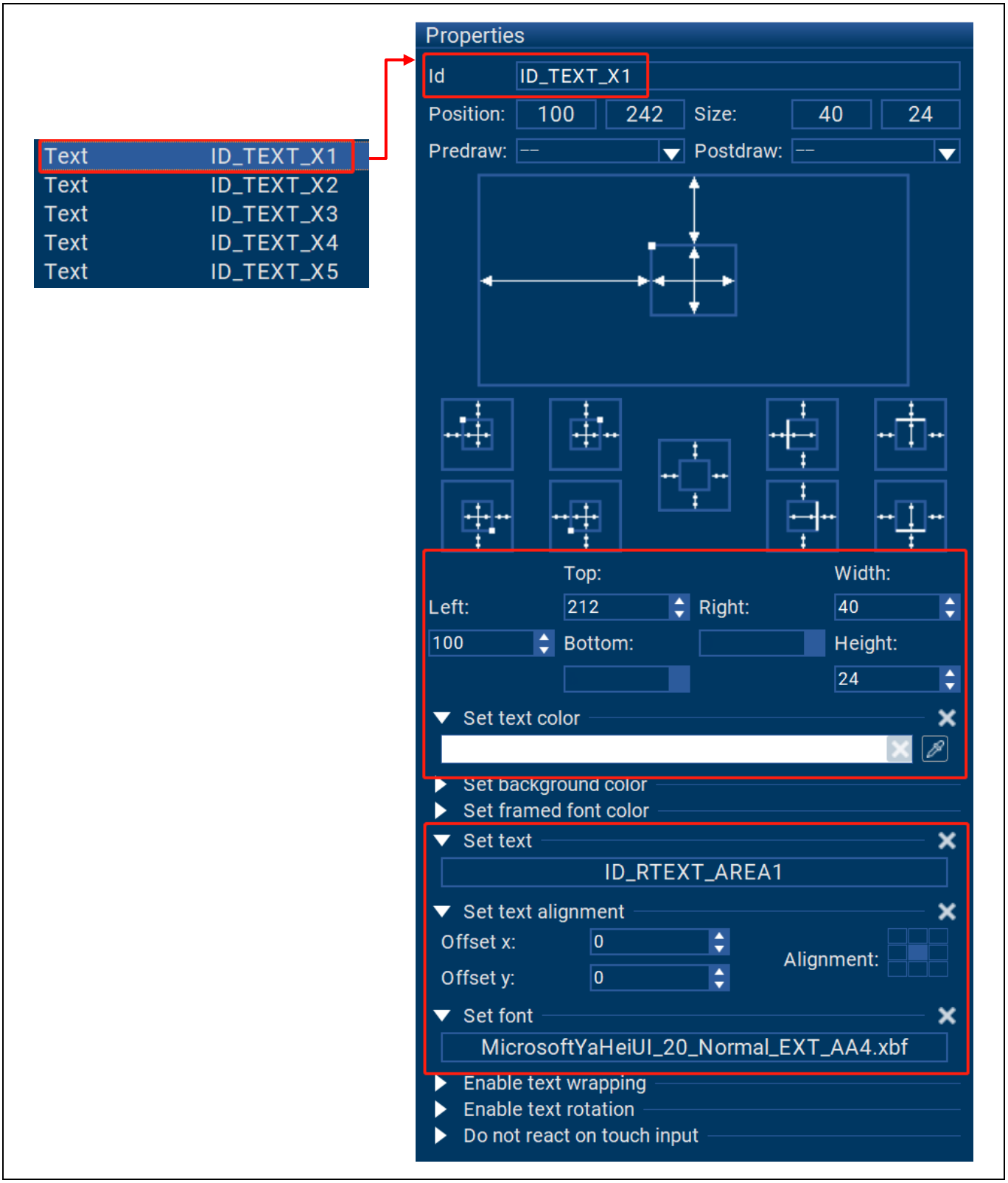


图 1-43 Text 属性参数设置 - 横坐标值显示

其中 5 个 Text 用于显示数字。在 Properties 里设置 text color 为 RGB (255, 255, 255)，font 为 MicrosoftYaHeiUI_20_Normal_EXT_AA4，按表 1-19 设置坐标、decimal mode、rang、text alignment，实际设置项目如图 1-44。

ID	坐标 (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, 居右)
ID_TEXT_Y1	(2, 146, 24, 20)	Mask:	(100, 100)	(0, 0, 居右)
ID_TEXT_Y2	(2, 96, 24, 20)	Mask:	(200, 200)	(0, 0, 居右)
ID_TEXT_Y3	(2, 46, 24, 20)	Mask:	(300, 300)	(0, 0, 居右)
ID_TEXT_Y4	(2, 0, 24, 20)	Mask:	(400, 400)	(0, 0, 居右)

表 1-19 Text 属性参数设置说明 - 纵坐标显示

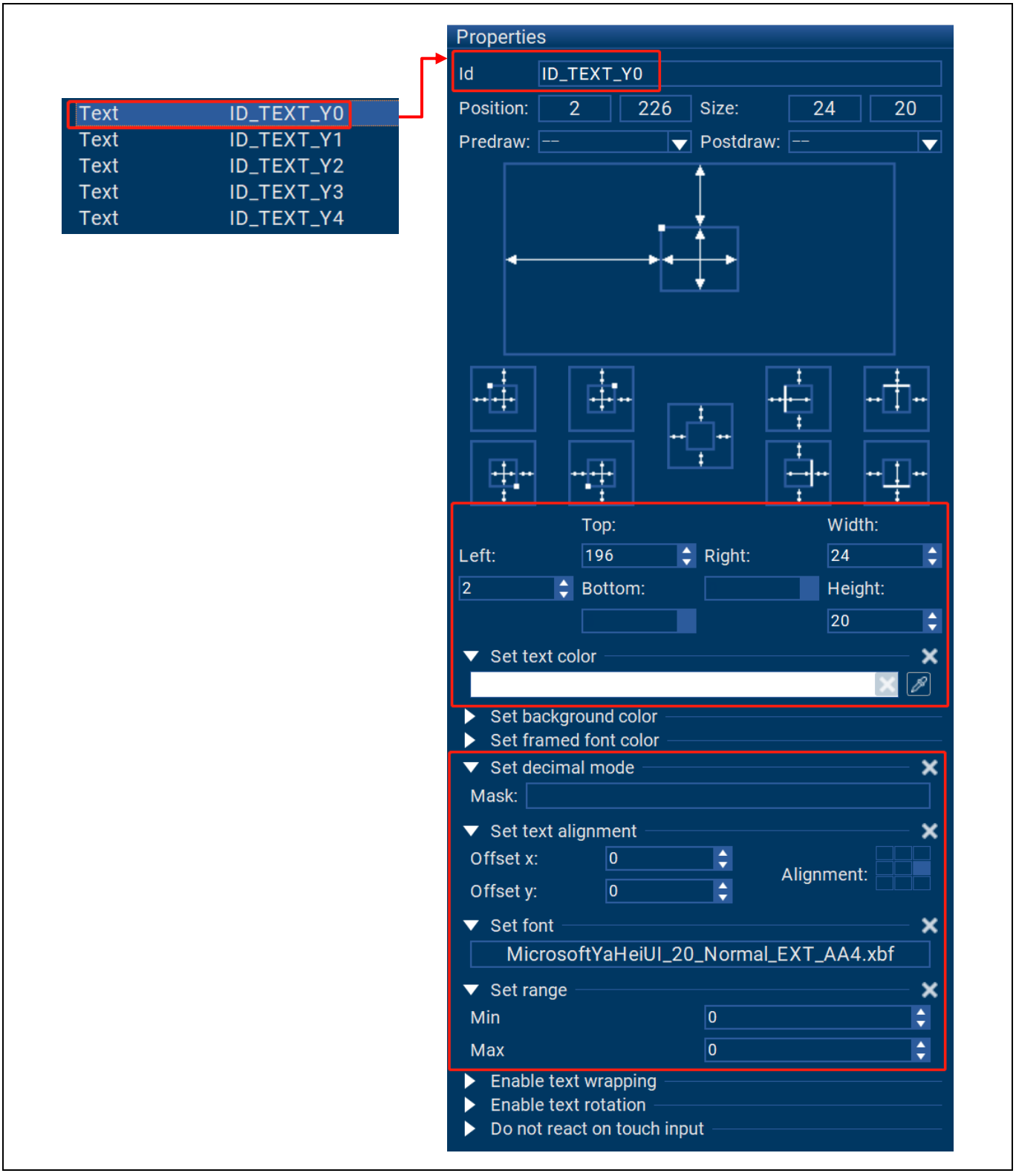


图 1-44 Text 属性参数设置 - 纵坐标显示

3. 添加窗口

添加一个窗口用于绘制区域-温度曲线。首先选中 ID_SCREEN_TempWave，在 Add Objects 里点击 Window 添加一个窗口，并在 Hierarchic tree 里重命名为 ID_WINDOW_Wave。在 Properties 里设置坐标为 (38, 5, 400, 200)，Predraw 的下拉菜单里选择预画图动作为 ID_DRAW_BLK，Postdraw 的下拉菜单选择画图动作为 ID_Draw_TempWave。

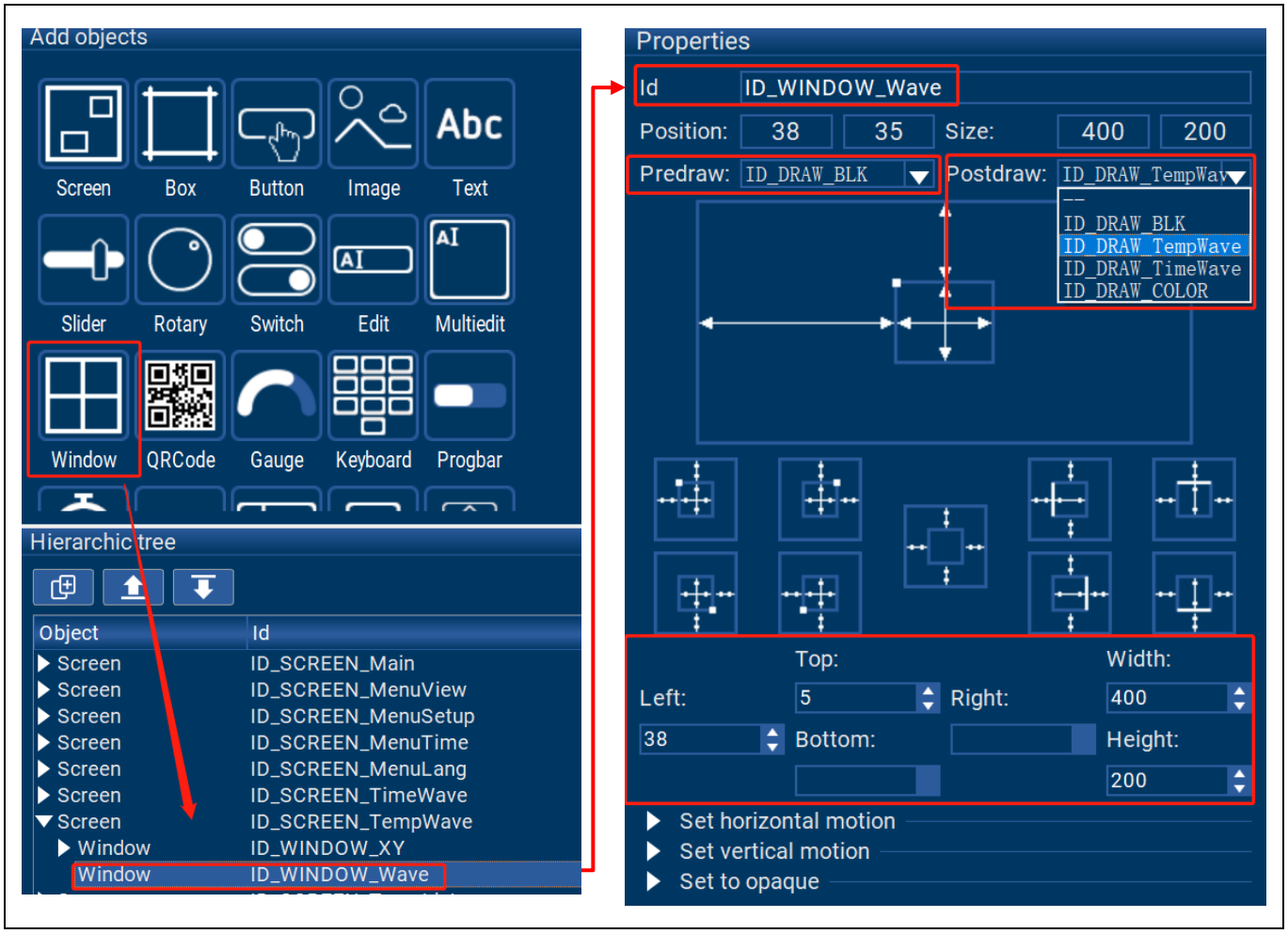


图 1-45 添加窗口 - 区域-温度曲线画图绘制

4. 设计完成后的区域-温度曲线画面

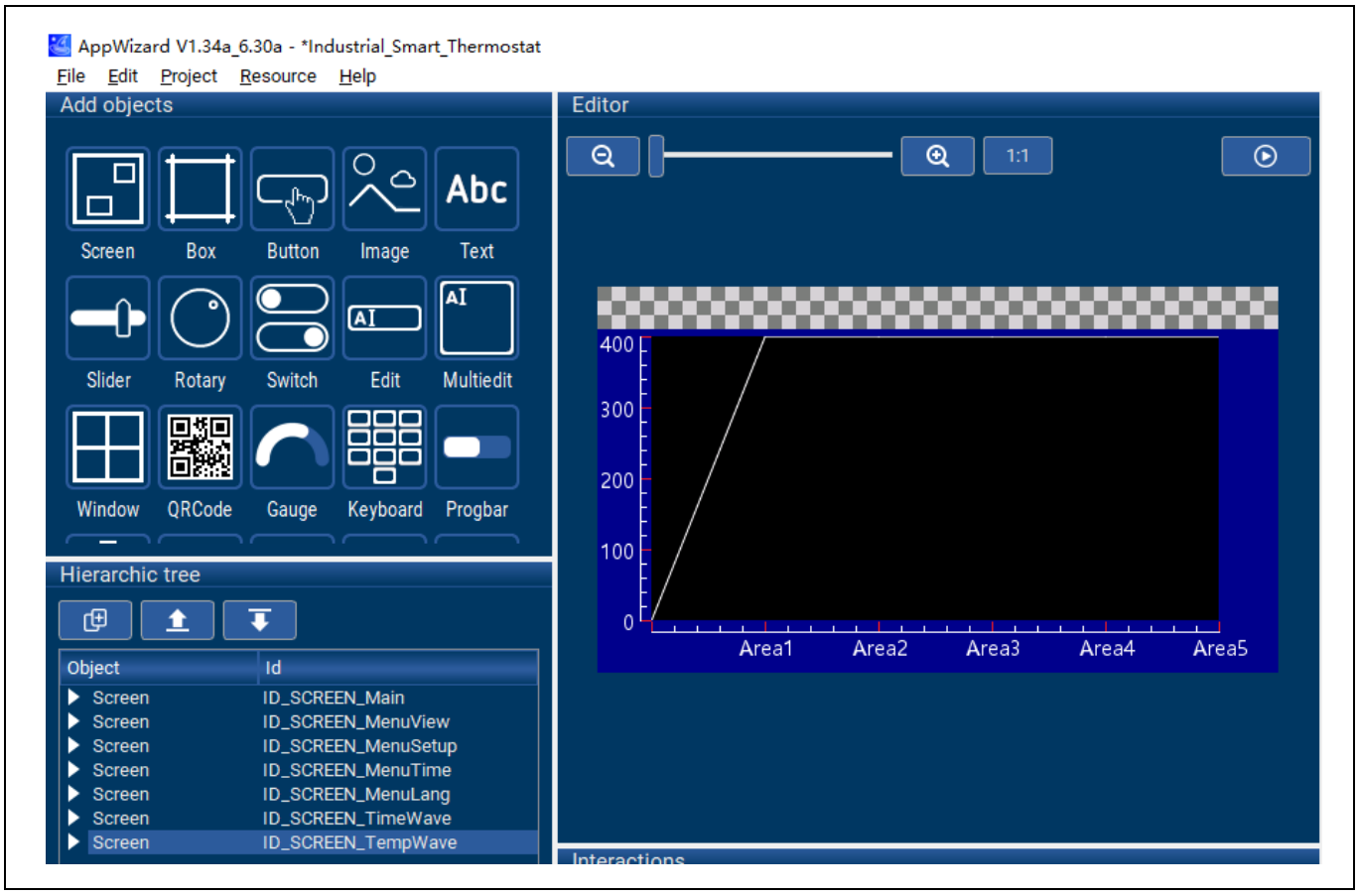


图 1-46 区域 - 温度曲线画面

1.2.2.12 创建温度列表画面

1. 添加画面

在 Add Objects 中点击 Screen 添加一个画面，在 Hierarchic tree 里重命名为 ID_SCREEN_TempList。在 Properties 里设置坐标为 (0, 30, 480, 242)。

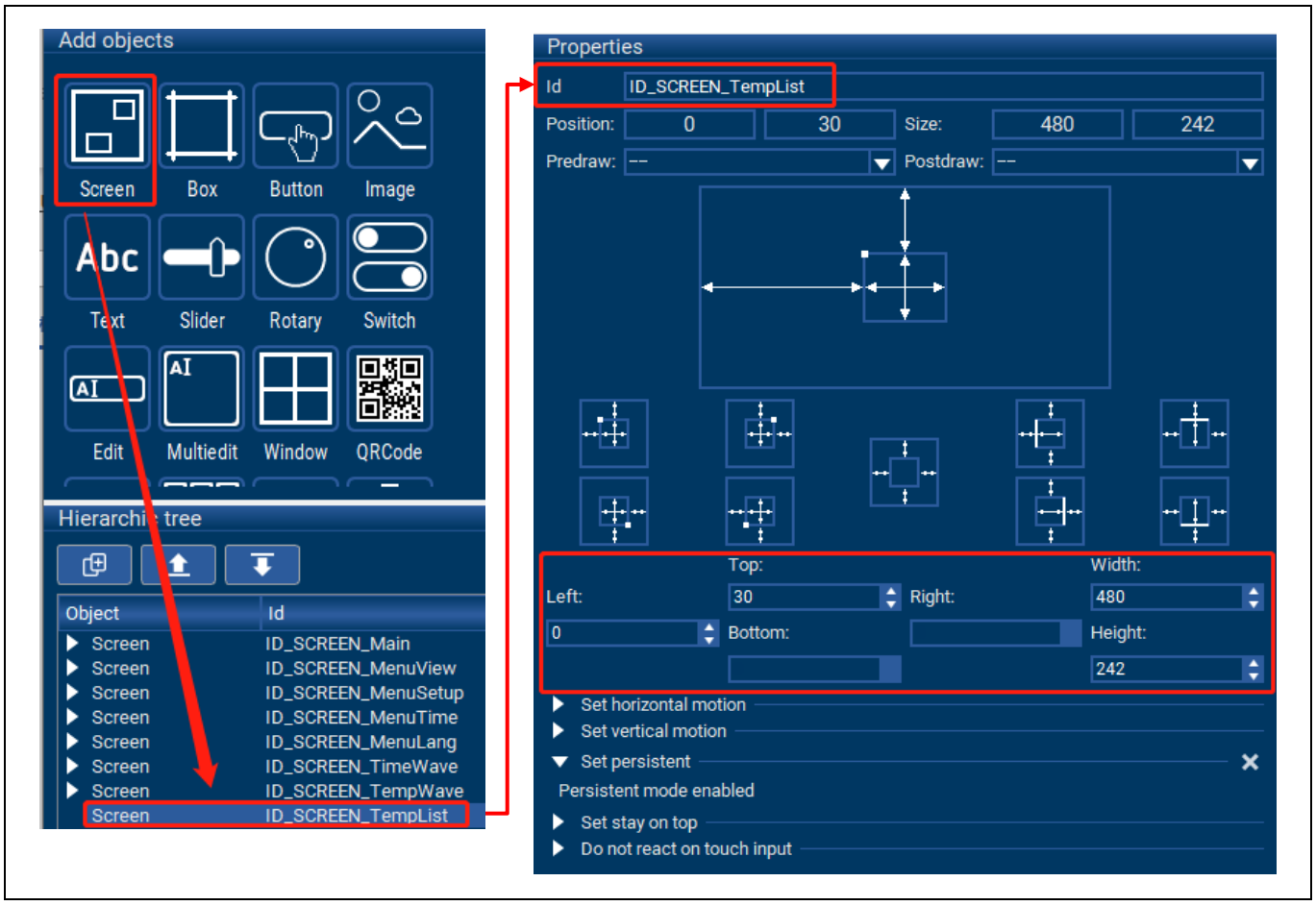


图 1-47 创建温度列表画面

2. 添加窗口

在 Hierarchic tree 里选中 ID_SCREEN_TempList。在 Add Objects 中点击 Window 为画面添加一个窗口，在 Hierarchic tree 里重命名为 ID_WINDOW_TempList。在 Properties 里设置坐标为 (0, 0, 480, 242)。

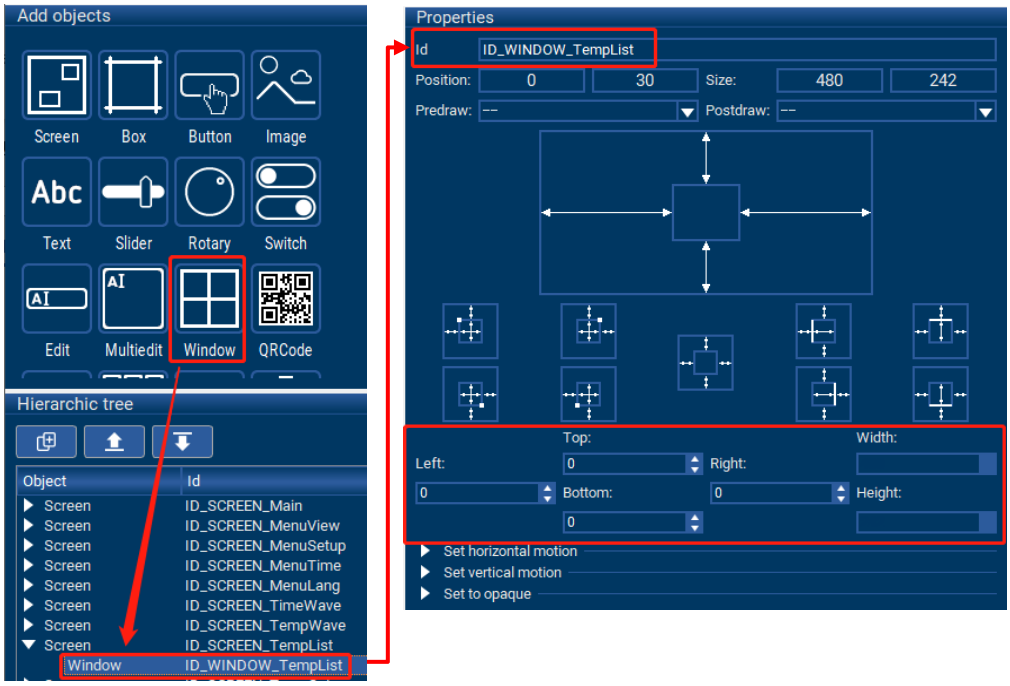


图 1-48 添加 Window - 区域温度列表显示

3. 添加 Image

如图 1-49 所示，在 Add Objects 中点击 Image，为窗口添加一个 Image，并在 Hierarchic tree 里重命名为 ID_IMAGE_BG。在 Properties 里设置坐标为 (0, 0, 400, 242)， bitmap 选择图片 BKG_TemperatureList_480x242.png 作为背景图片，实际设置项目如图 1-50。

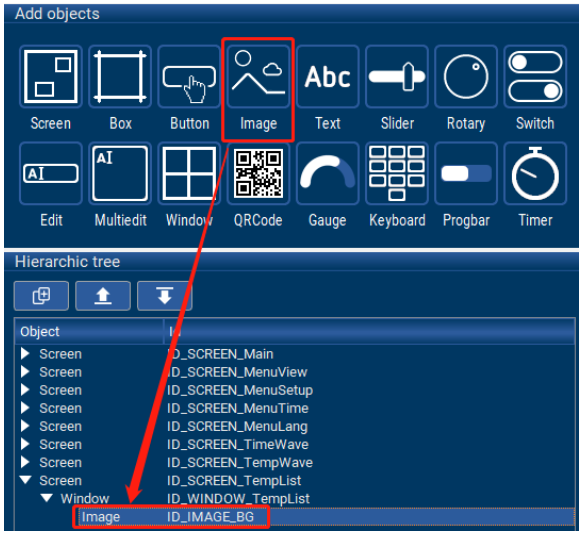


图 1-49 添加 Image - 背景显示

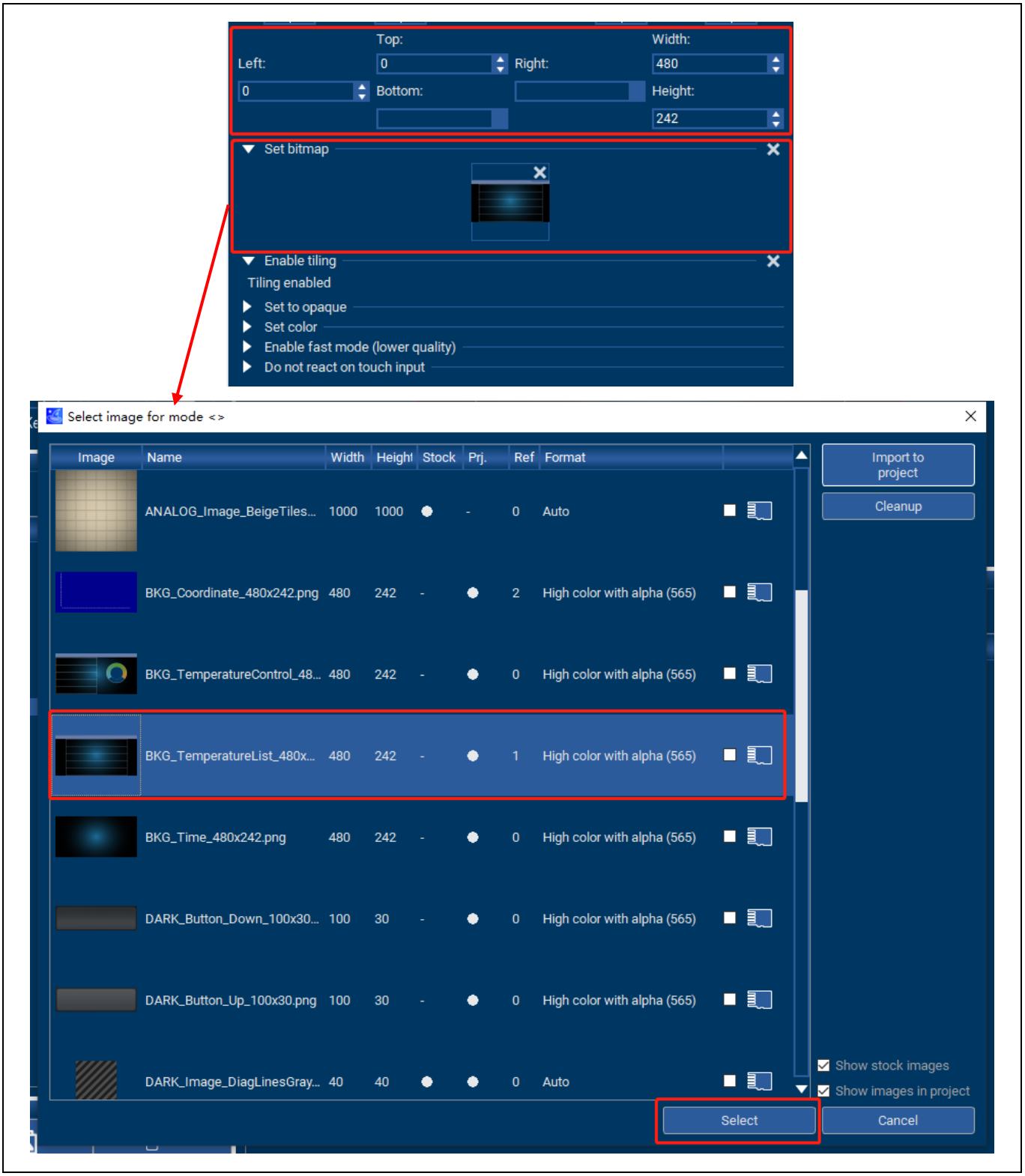


图 1-50 Image 属性参数设置

4. 添加 Text

为窗口添加 Text，用于显示所有区域的名称、设定温度和实际温度。

1) 添加 Text

在 Add Objects 中点击 Text，添加 28 个 Text，并在 Hierarchic tree 按图 1-51 重命名。

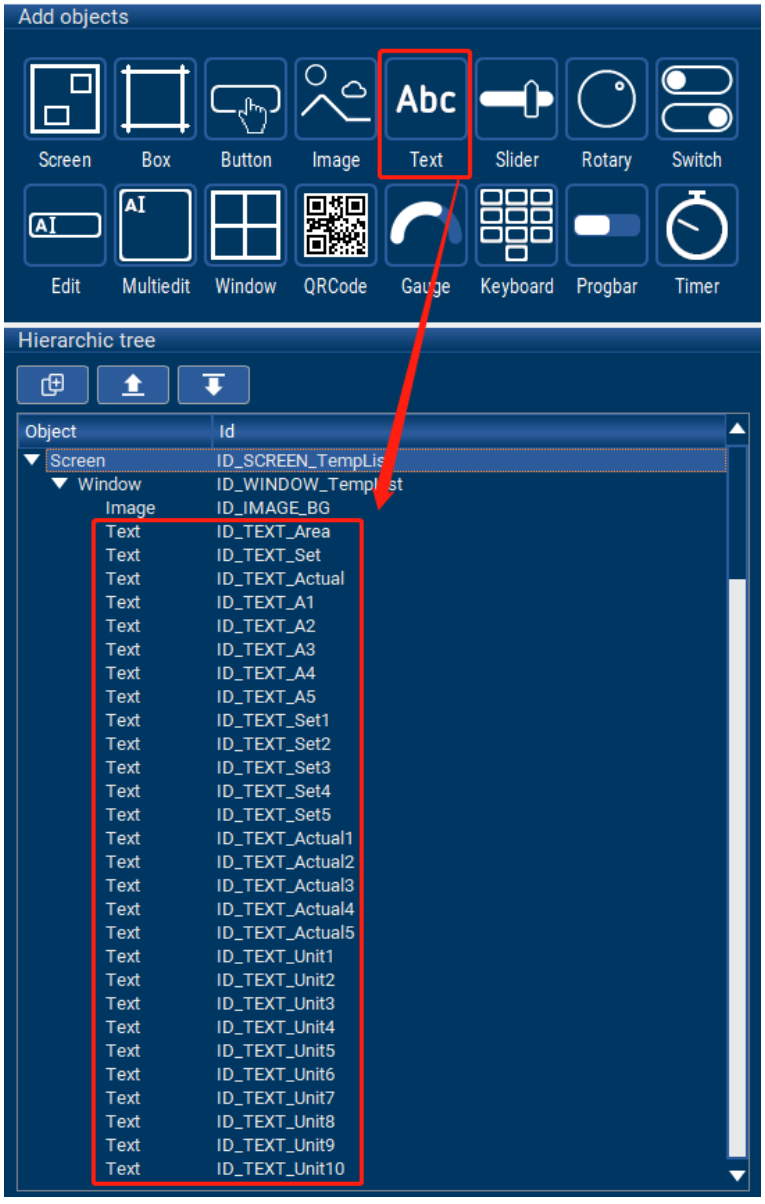


图 1-51 添加 Text - 区域设定温度和实际温度的显示

2) 文本显示的参数设置

其中 18 个 Text 用于显示文本。在 Properties 里设置 text color 为 RGB (192, 192, 192)，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4，按表 1-20 设置坐标、text、text alignment，实际设置项目如图 1-52。

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

ID	坐标 (left, top, width, height)	text	text alignment (offset x, offset y, Alignment)
ID_TEXT_Unit9	(372, 163, 20, 24)	ID_RTEXT_DEG	(0, 0, 居左)
ID_TEXT_Unit10	(372, 206, 20, 24)	ID_RTEXT_DEG	(0, 0, 居左)

表 1-20 Text 属性参数设置说明 - 文本显示

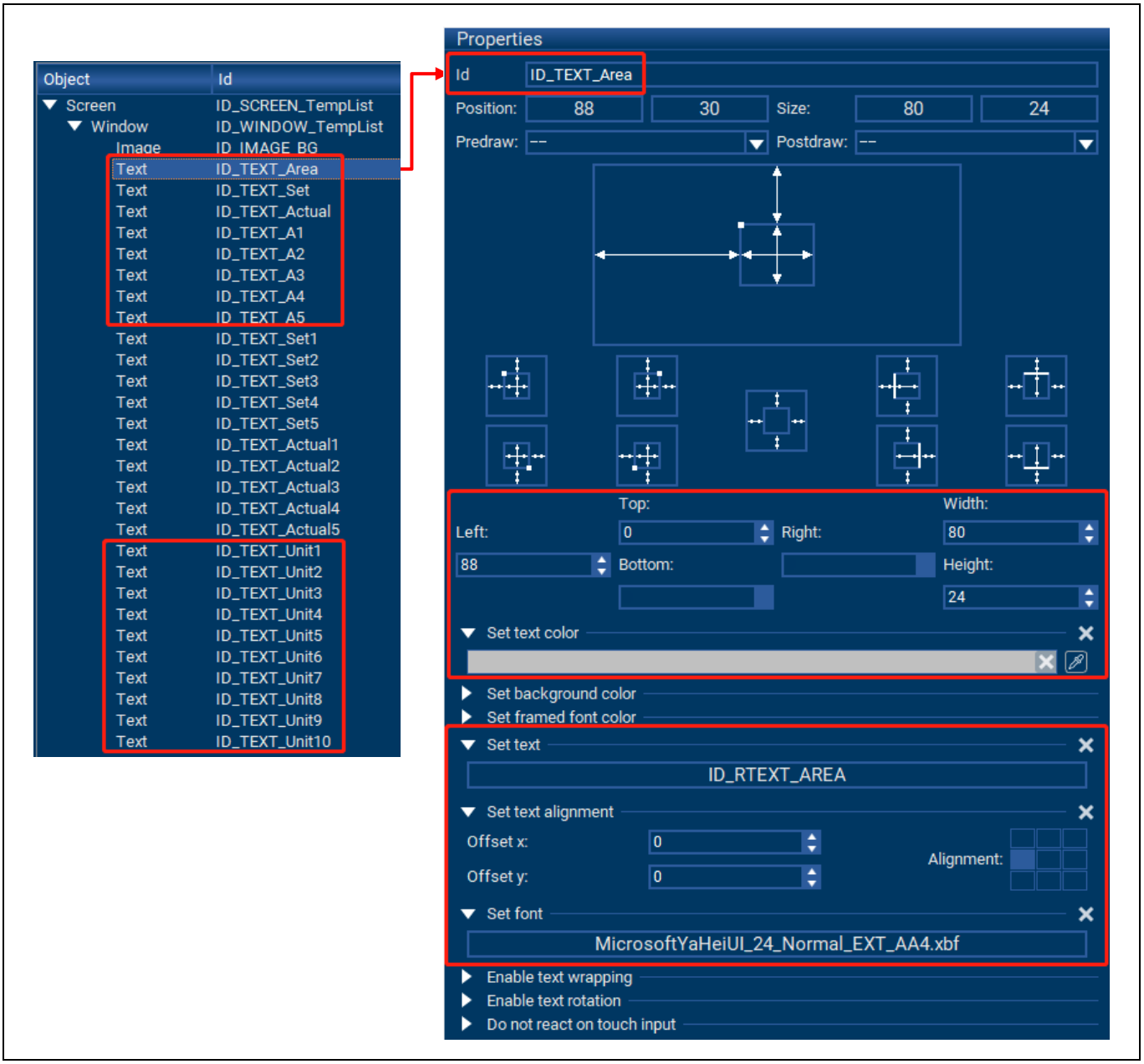


图 1-52 Text 属性参数设置 - 文本显示

3) 数字显示的参数设置

其中 10 个 Text 用于显示数字。在 Properties 里设置 text color 为 RGB (192, 192, 192)，decimal mode，text alignment 为 offset x = offset y = 0、Alignment 居右，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4，range 为 Min = 0、Max = 400，坐标按表 1-21 设置，实际设置项目如图 1-53。

ID	**坐标 (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)

表 1-21 Text 的坐标 - 温度显示

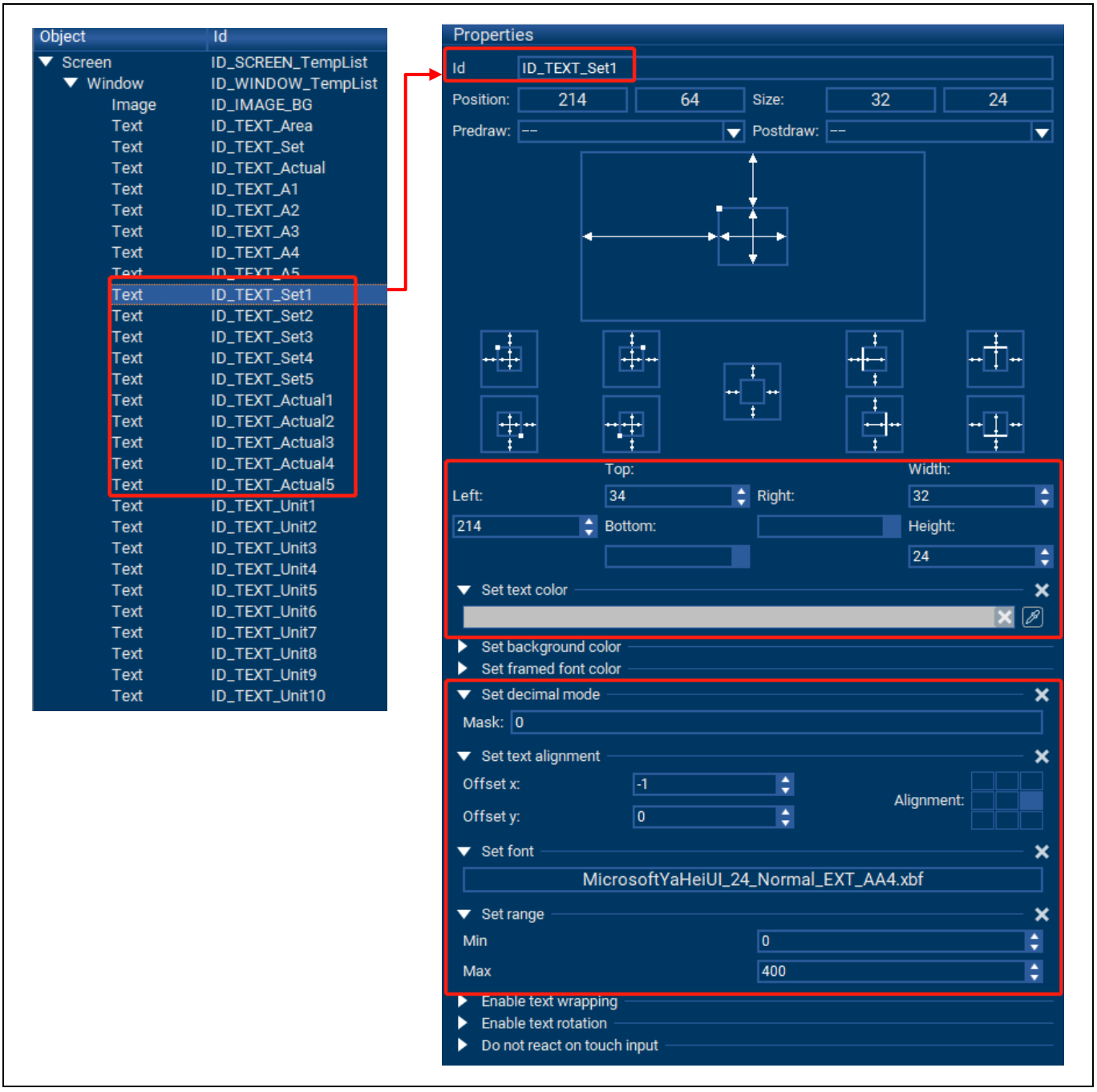


图 1-53 添加 Text - 设定温度和实际温度显示

5. 设计完成后的温度列表画面

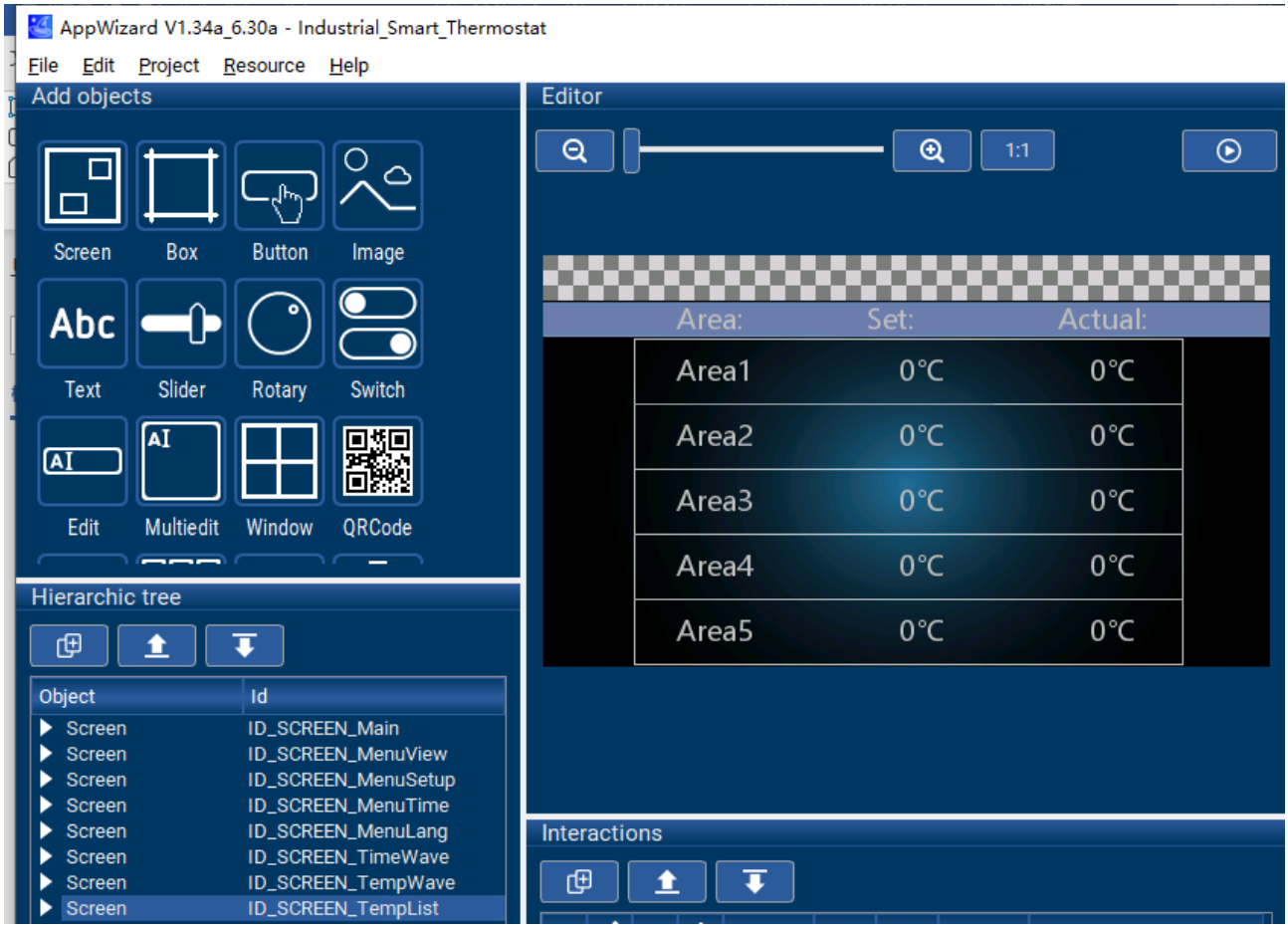


图 1-54 温度列表画面

1.2.2.13 创建温度设置画面

1. 添加画面

在 Add Objects 中点击 Screen 添加一个画面，在 Hierarchic tree 里重命名为 ID_SCREEN_TempSetup。在 Properties 里设置坐标为 (0, 30, 480, 242)。

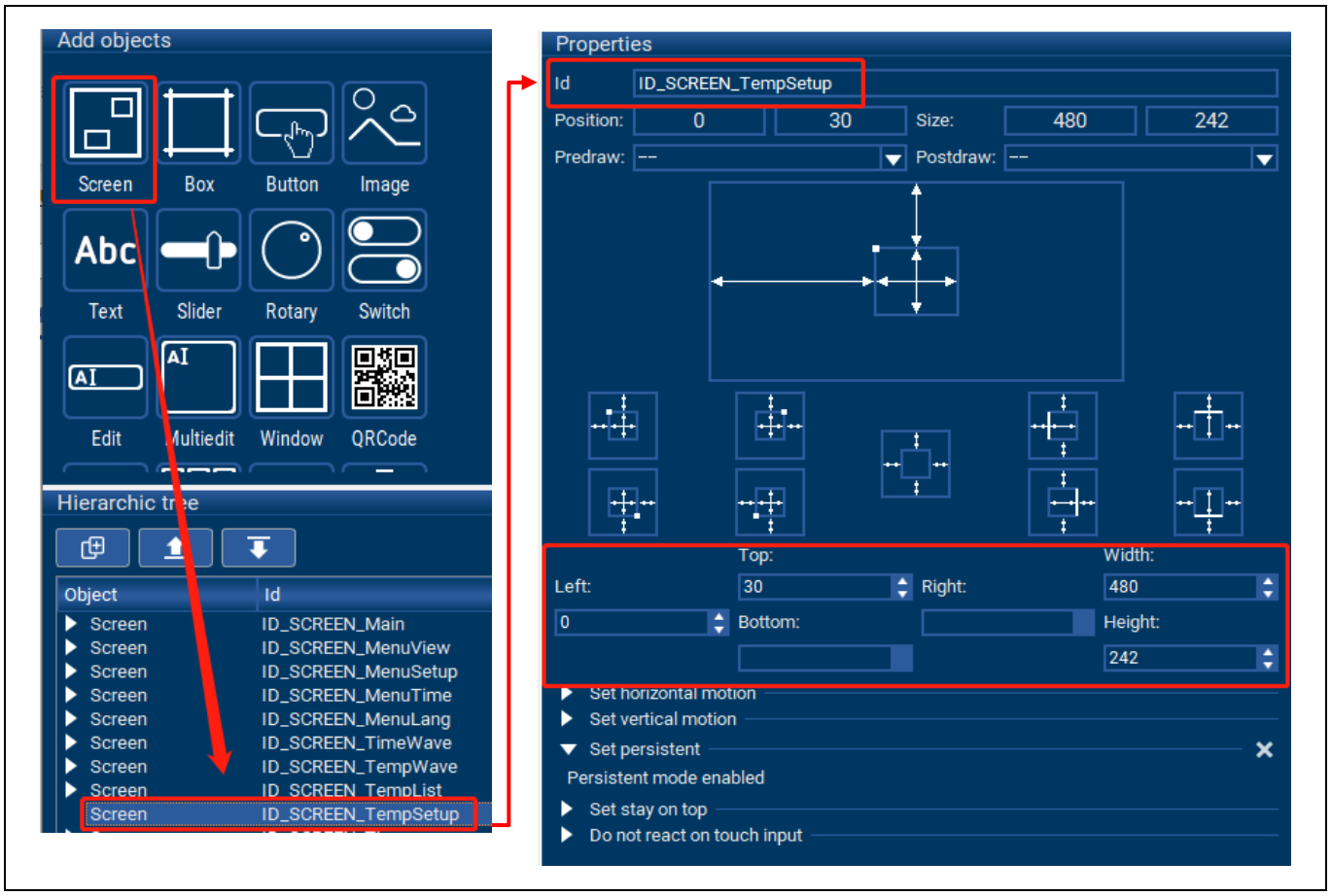


图 1-55 创建温度设置画面

2. 添加窗口

在 Add Objects 中点击 Window 为画面添加一个窗口，在 Hierarchic tree 里重命名为 ID_WINDOW_TempSetup。在 Properties 里设置坐标为 (0, 0, 480, 242)。

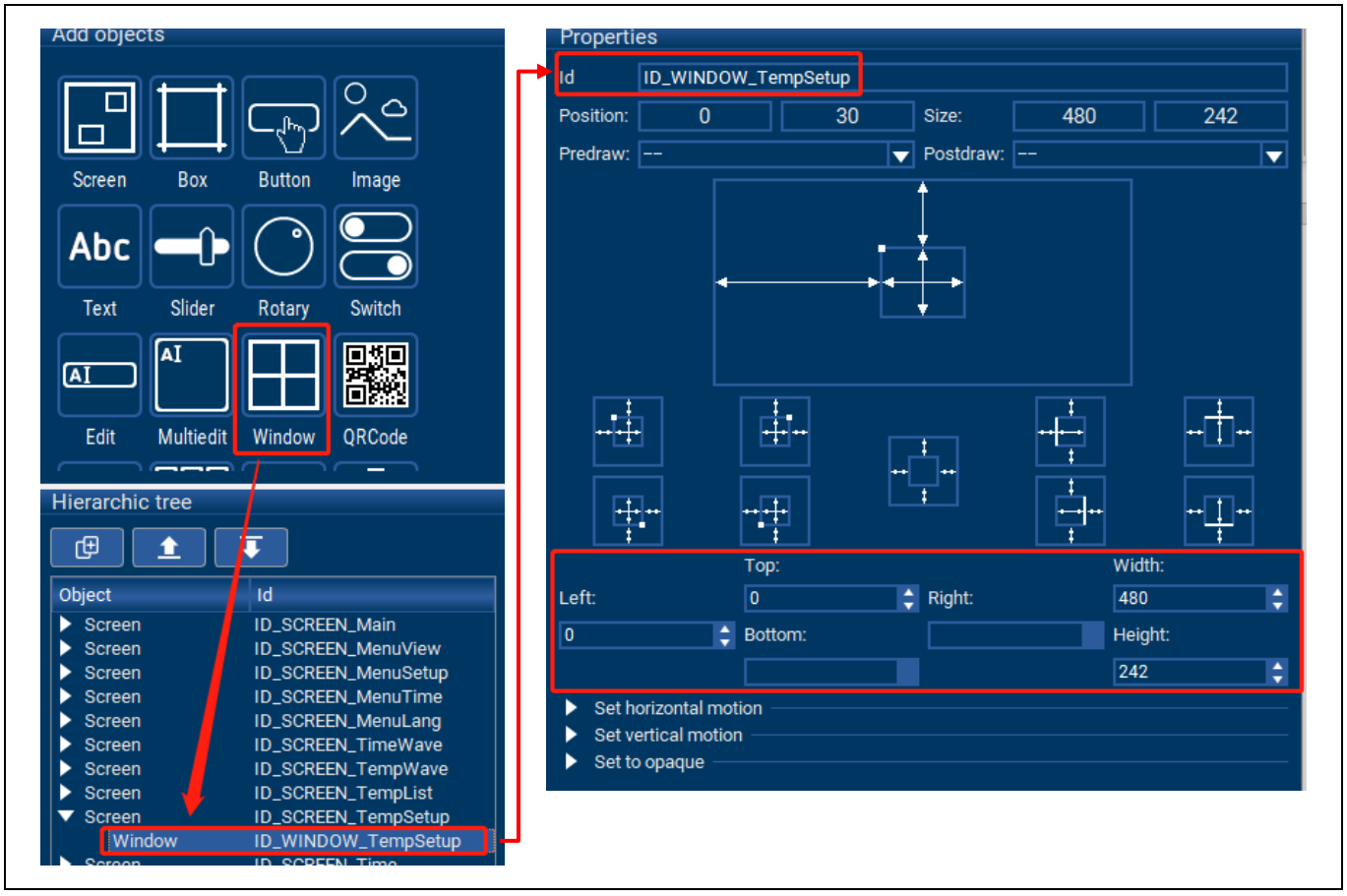


图 1-56 添加 Window - 温度列表显示

3. 添加 Image

如图 1-57 和图 1-58 所示，在 Add Objects 中点击 Image，为窗口添加一个 Image，并在 Hierarchic tree 里重命名为 ID_IMAGE_BG。在 Properties 里设置坐标为 (0, 0, 400, 242)，bitmap 选择图片 BKG_TemperatureControl_480x242.png 作为背景图片。

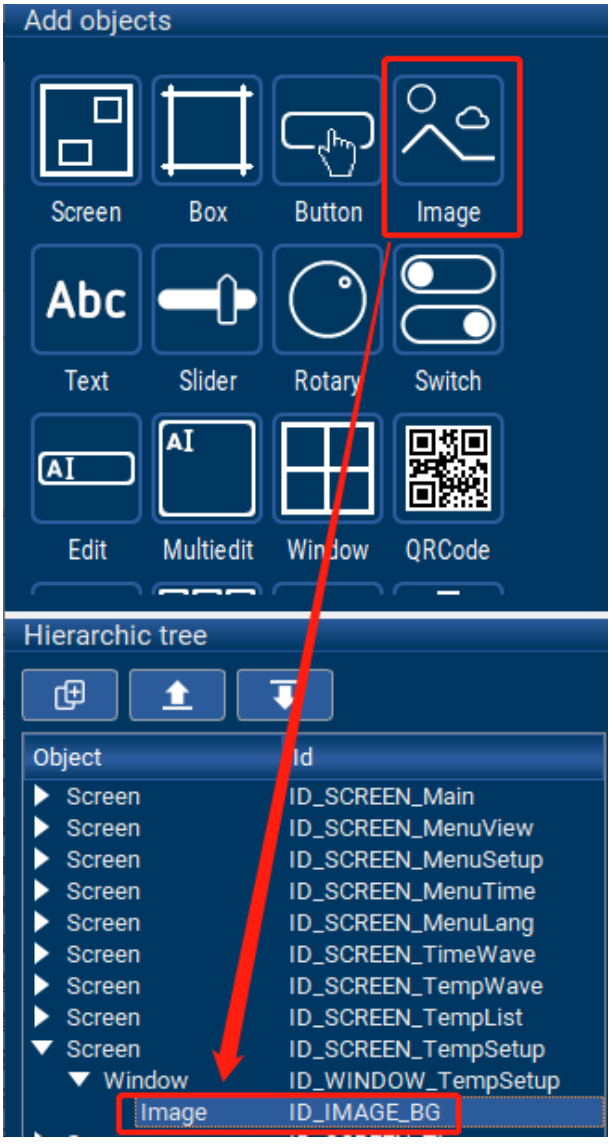


图 1-57 添加 Image - 温度设置画面的背景

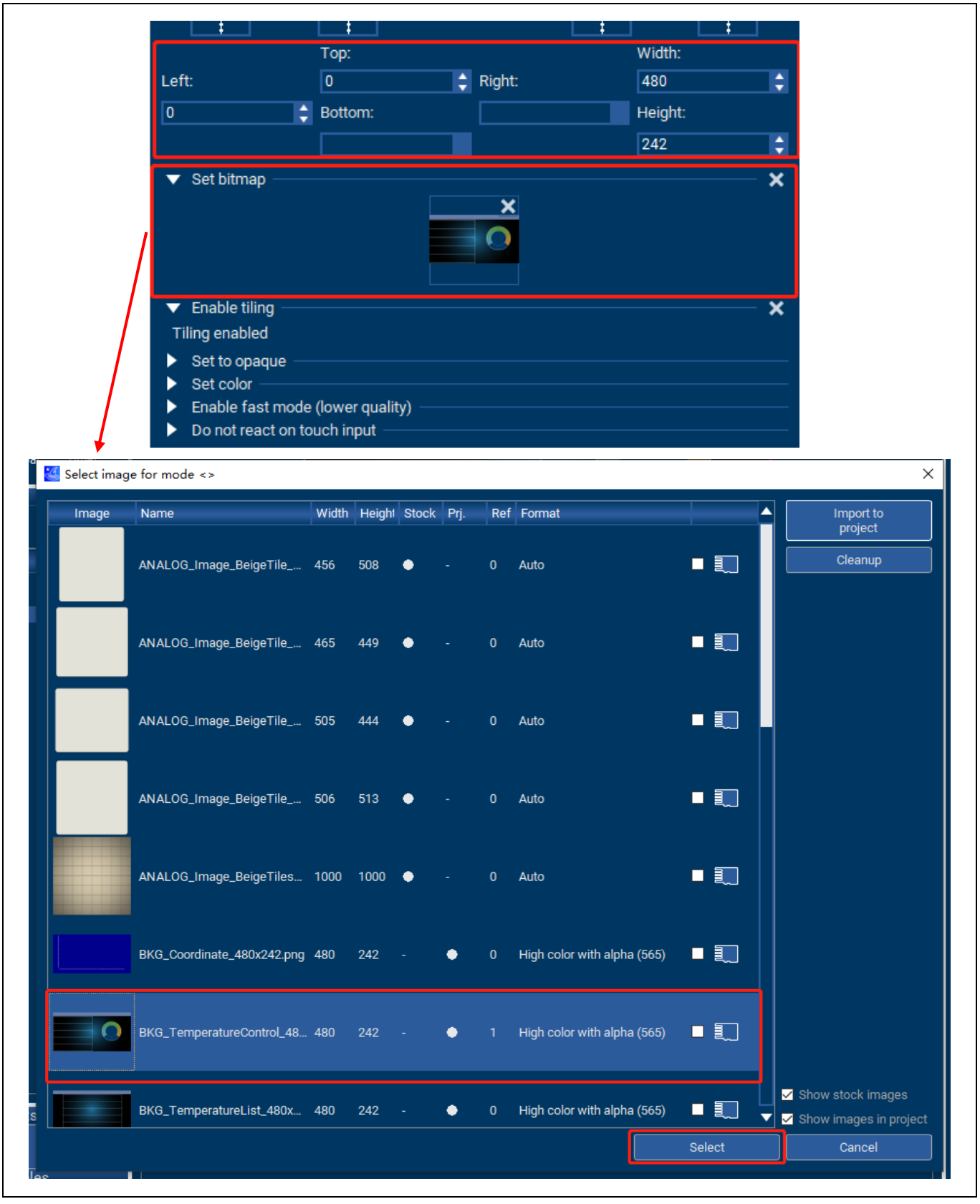


图 1-58 Image 属性参数设置

4. 添加选中区域高亮显示

为窗口添加一个 Box，用于高亮显示选中区域。

在 Add Objects 中点击 Box，并在 Hierarchic tree 里重新命名为 ID_BOX_IndexBar。在 Properties 里设置坐标为 (3, 25, 241, 41)， color 为 RGB (51, 94, 128)。

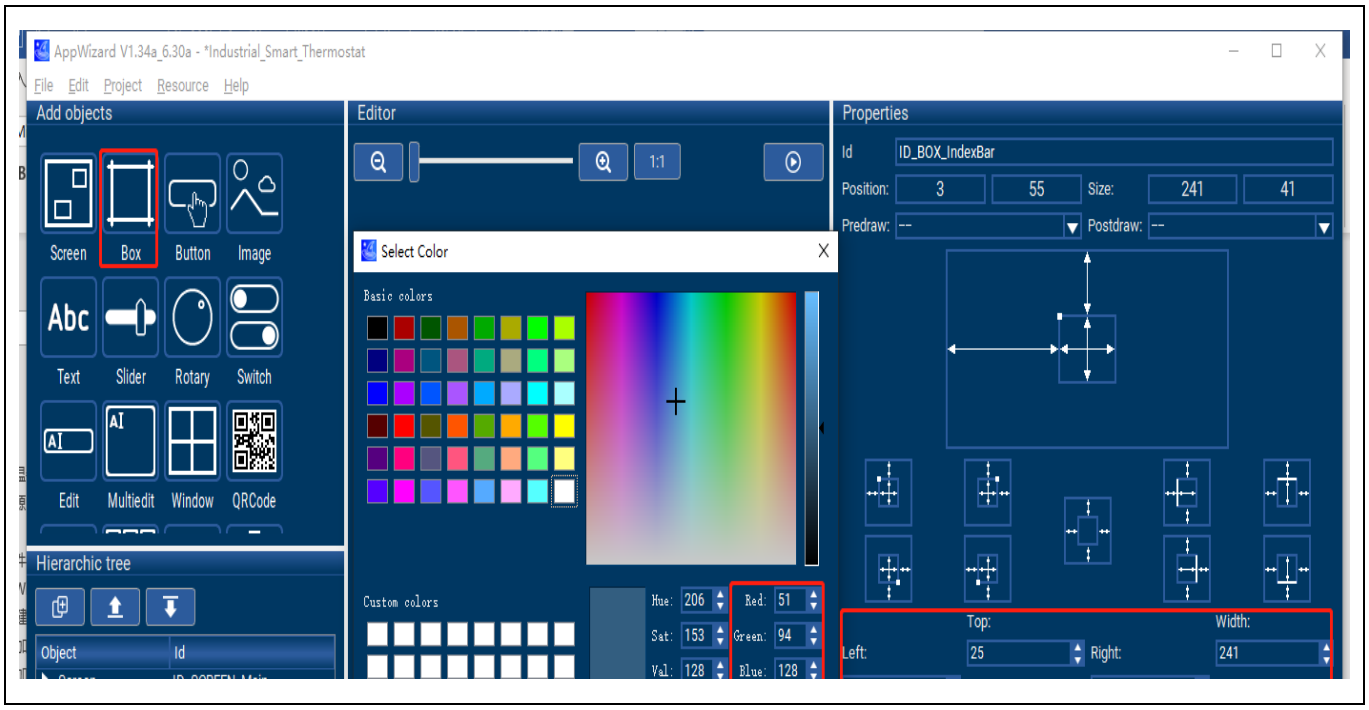


图 1-59 添加 Box - 选中区域高亮显示

5. 添加区域设置温度和实际温度的显示

为窗口添加 Text，用于显示所有区域的名称和温度。

1) 添加 Text

在 Hierarchic tree 里选中窗口 ID_WINDOW_TempSetup，在 Add Objects 中点击 Text，添加 28 个 Text，并在 Hierarchic tree 按图 1-60 重新命名。

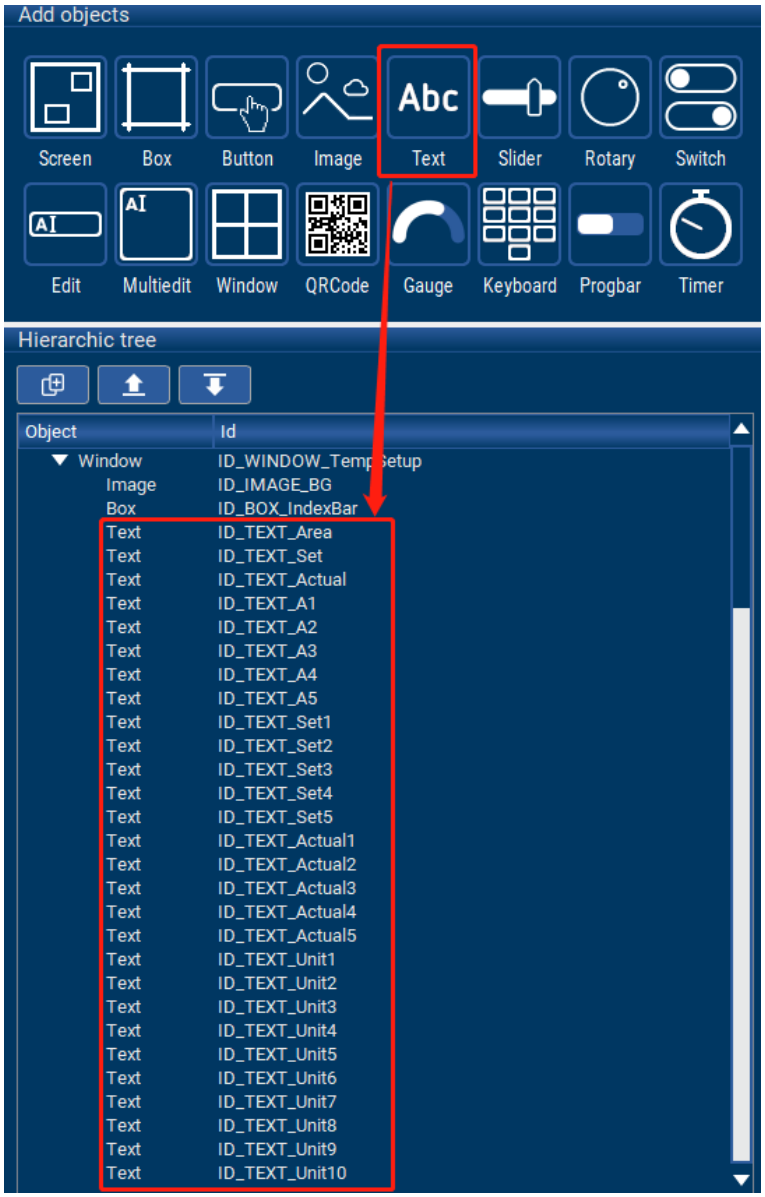


图 1-60 添加 Text - 区域设定温度和实际温度的显示

2) 文本显示的参数设置

其中 18 个 Text 用于显示文本。在 Properties 里设置 text color 为 RGB (192, 192, 192)，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4，按表 1-22 设置坐标、text、text alignment，实际设置项目如图 1-61。

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

ID	坐标 (left, top, width, height)	text	text alignment (offset x, offset y, Alignment)
ID_TEXT_Unit9	(126, 163, 20, 24)	ID_RTEXT_DEG	(0, 0, 居左)
ID_TEXT_Unit10	(126, 206, 20, 24)	ID_RTEXT_DEG	(0, 0, 居左)

表 1-22 Text 属性参数设置说明 - 文本显示

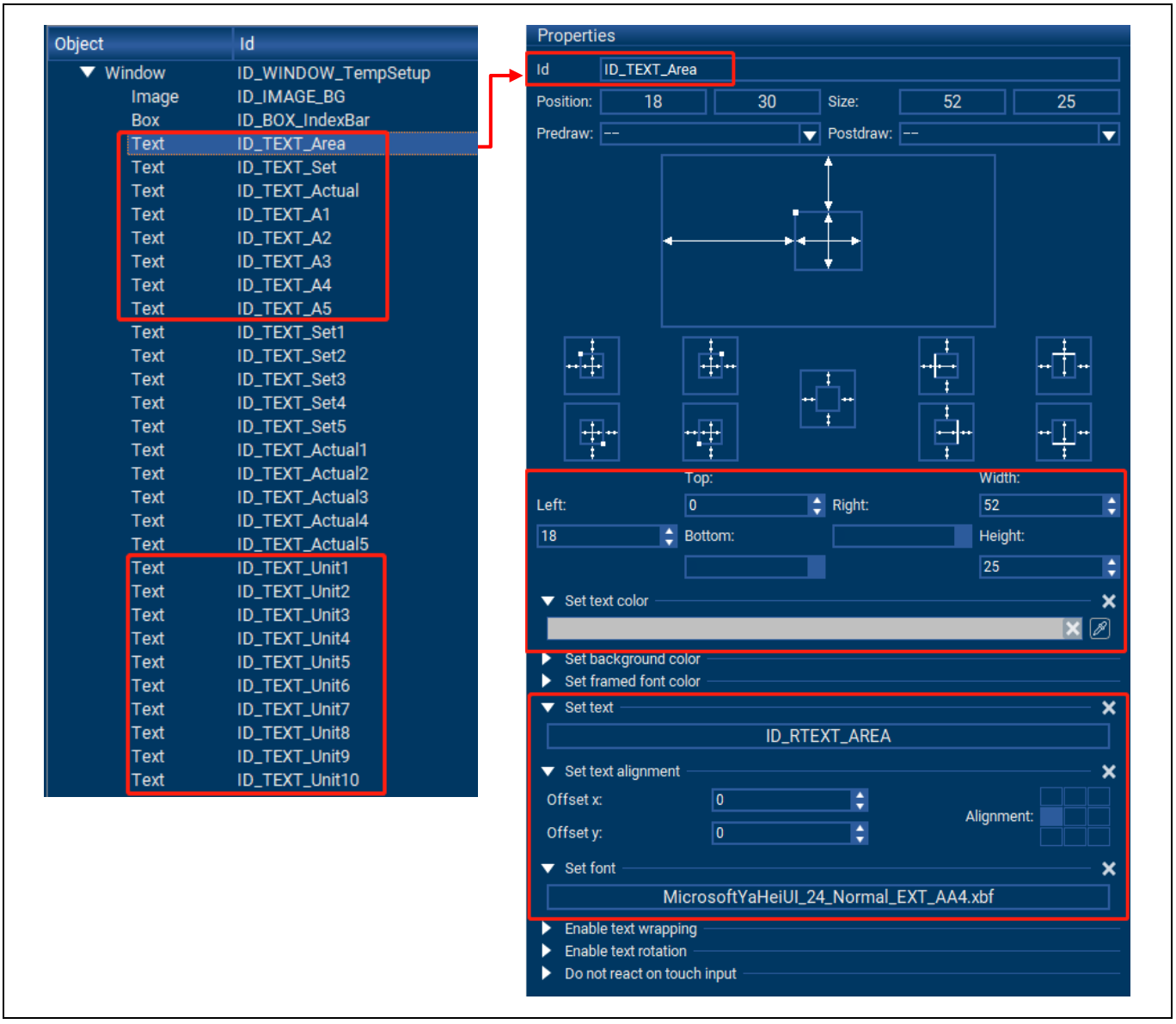


图 1-61 Text 属性参数设置 - 文本显示

3) 数字显示的参数设置

其中 10 个 Text 用于显示数字。在 Properties 里设置 text color 为 RGB (192, 192, 192)，decimal mode 为 Mask 不设置，text alignment 为 offset x = offset y = 0、Alignment 居右，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4，range 为 Min= 0，Max= 400。按表 1-23 设置坐标，实际设置项目如图 1-62。

ID	**坐标 (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)

表 1-23 Text 属性参数设置说明 - 显示温度

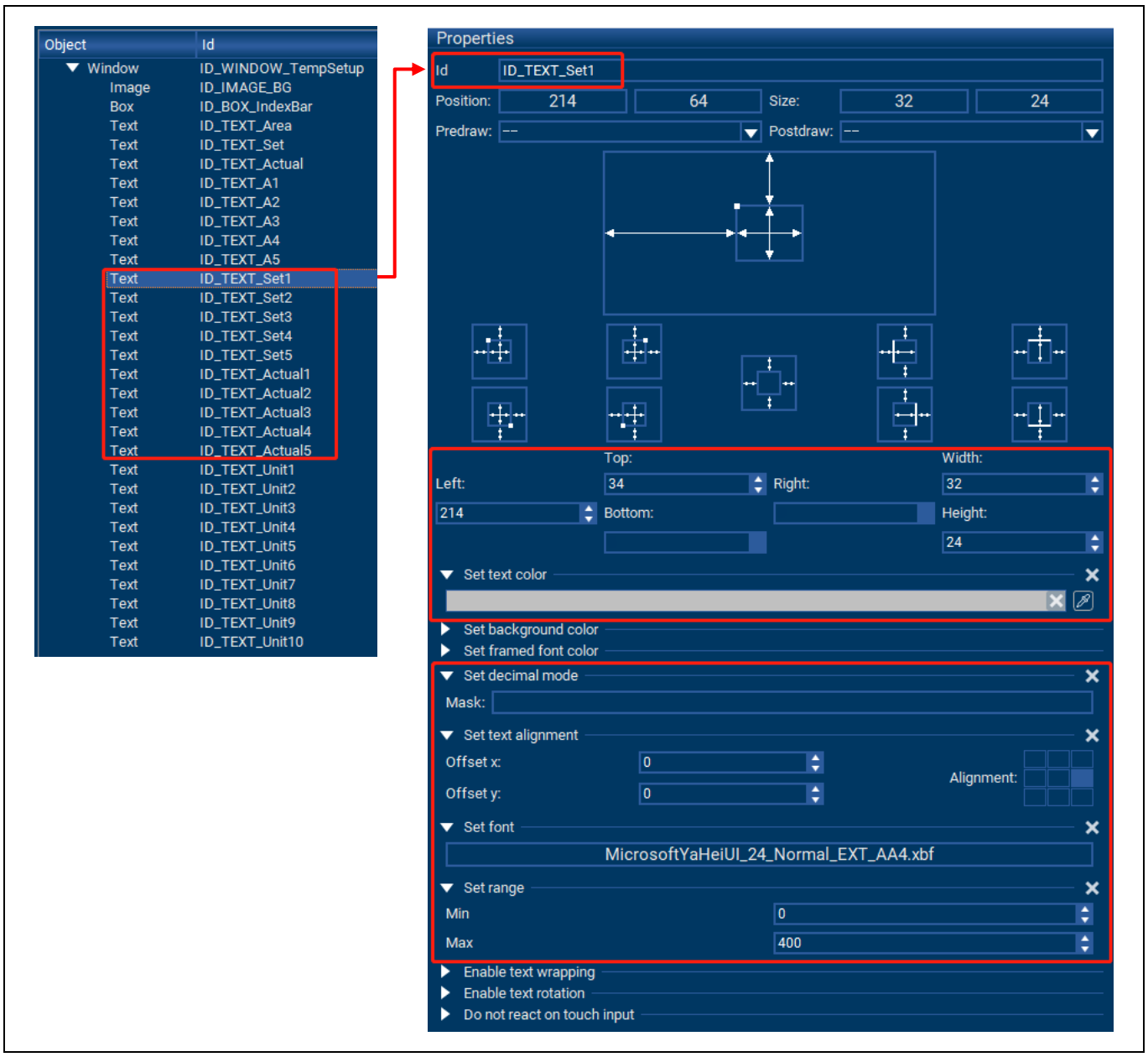


图 1-62 添加 Text - 设定温度和实际温度的显示

6. 添加区域选择按钮

为 ID_WINDOW_TempSetup 窗口添加 5 个按钮，用于区域选择。Button 的 ID 和坐标如表 1-24 所示，并在 bitmaps 里去掉背景图片，实际设置项目如图 1-63 所示。

ID	坐标 (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)

表 1-24 Button 属性参数设置说明 - 区域选择按钮

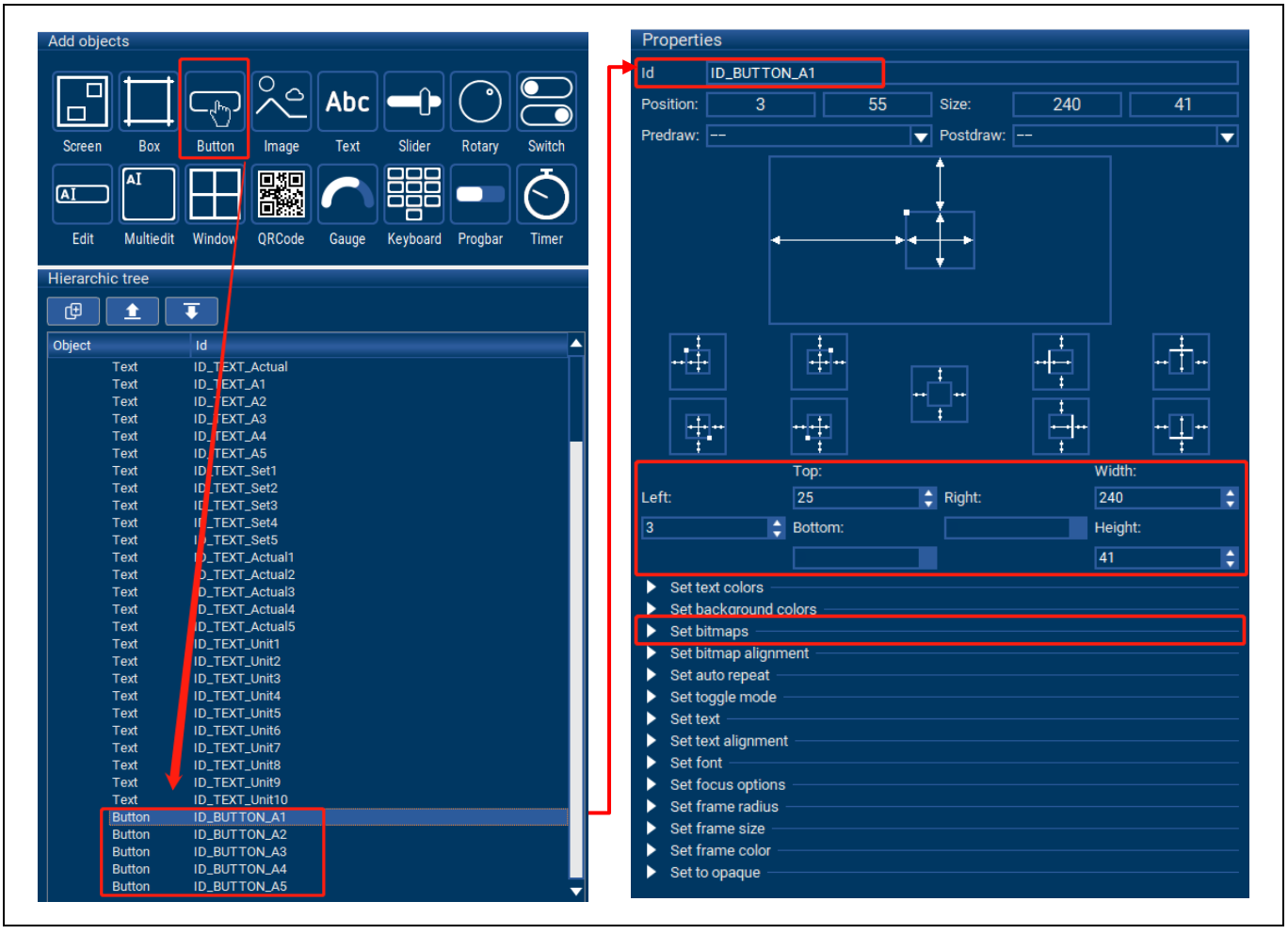


图 1-63 添加按钮 - 区域选择

7. 添加温度选择旋钮

1) 添加 Rotary

在 Hierarchic tree 里选中窗口 ID_WINDOW_TempSetup，在 Add Objects 里点击 Rotary 为窗口添加一个旋钮，并在 Hierarchic tree 里重命名为 ID_ROTARY_00。在 Properties 里设置坐标为(307, 66, 105, 105)，在 bitmaps 里设置 Backgroud 为 GUI_Knob_Icon_106x106.png，指针 Marker 为 GUI_KnobMark_Icon_16x9.png。

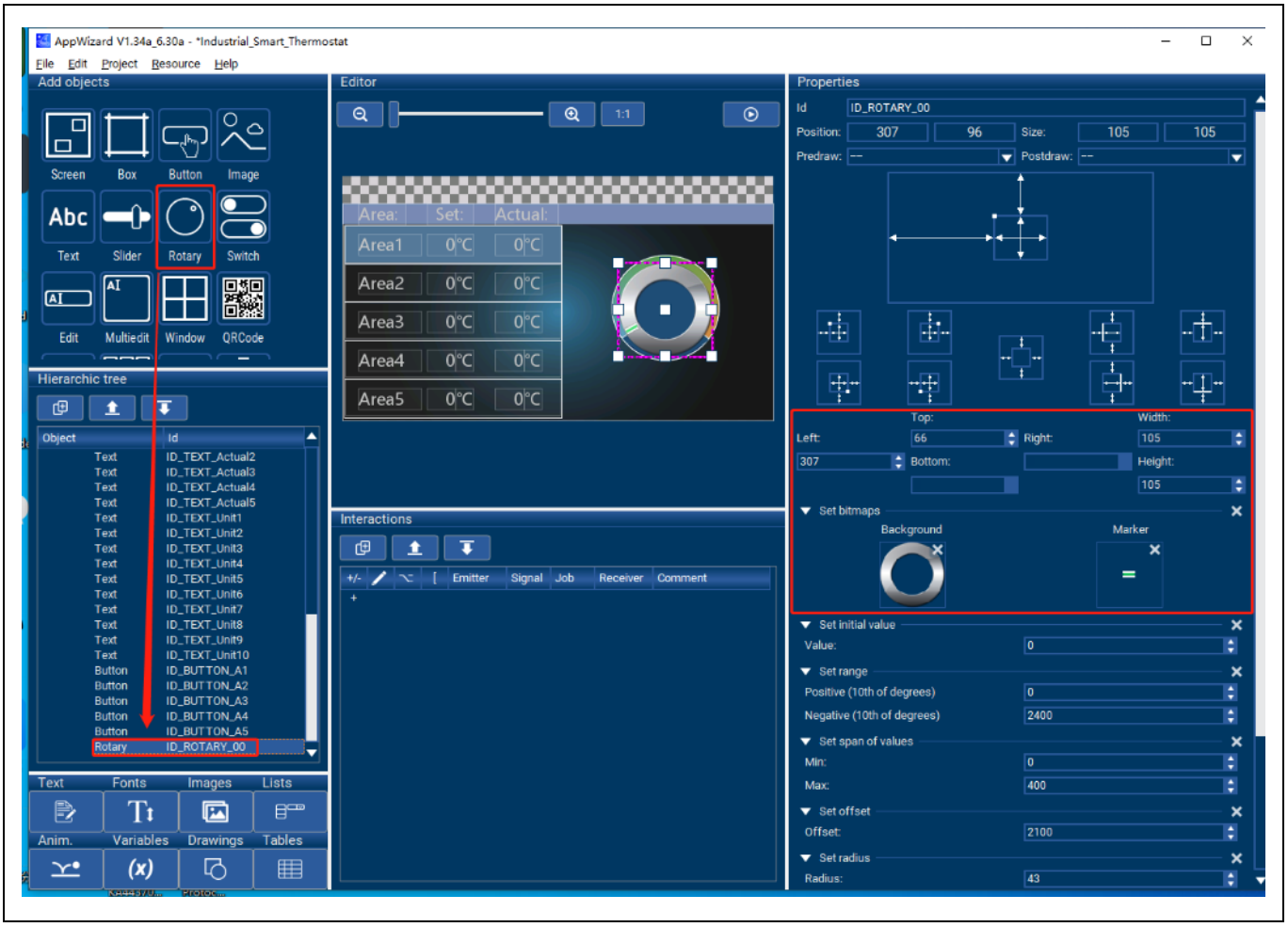


图 1-64 添加 Rotary - 温度调节旋钮

2) 设置 Rotary 的角度参数

因系统设定初始角度 0°位置如图 1-65 所示，为符合图片的旋转角度与旋转值，依照表 1-25 说明，设定 initial value、range、span of value、offset。实际设置项目如图 1-66。

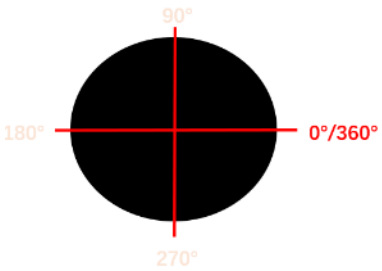


图 1-65 Rotary 系统初始角度

属性参数	设定值	说明
initial value	200	默认的数值
span of values	Min: 0	设置数值的范围
	Max: 400	
rang	Positive: 0	设置可调整的角度的起始范围 单位 0.1°
	Positive: 2400	
offset	2100	起点的偏移角度 (相对 0°位置) 单位 0.1°

表 1-25 Rotary 角度设置说明

3) 设置 Rotary 的指针的参数

为了对齐指针的旋转位置，按照表 1-26 说明，设定 marker alignment、radius，实际设置项目如图 1-66 所示。

属性参数	设定值	说明
radius	43	Rotary 的半径
Rotate marker	Marker rotation enabled	指针旋转 90°
Marker alignment	Offset x: 0. Offset y: 0. Alignment: 居中	指针的位置

表 1-26 Rotary 指针设置说明

图 1-66 Rotary 属性参数设置

8. 添加显示选择的温度

需要添加 2 个 Text，分别用来显示所选择温度的数值和单位。

1) 添加 Text

如图 1-67 在 Hierarchic tree 里选中窗口 ID_WINDOW_TempSetup，在 Add Objects 中点击 Text，添加 2 个 Text，并在 Hierarchic tree 中分别重新命名为 ID_TEXT_Unit 和 ID_TEXT_Temp。

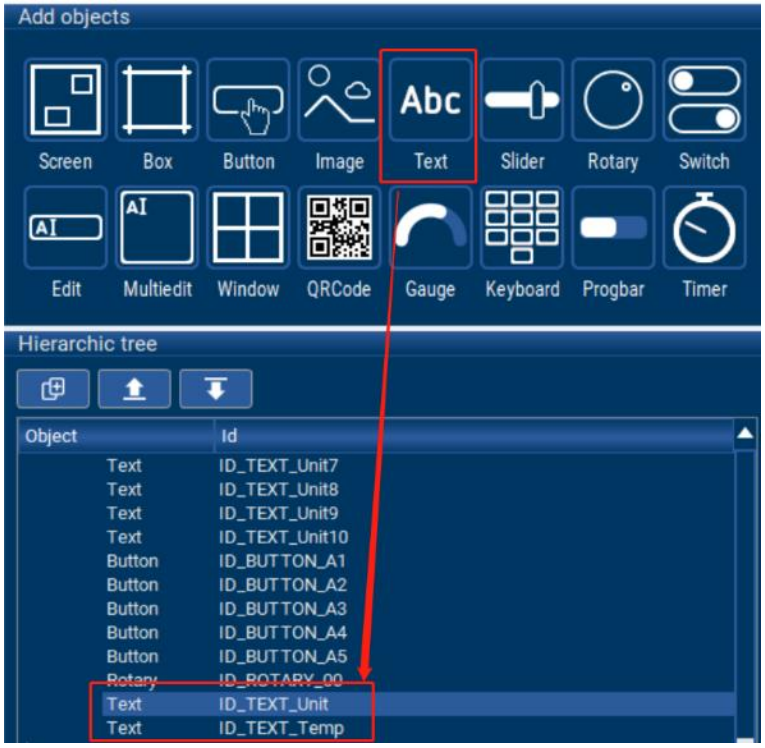


图 1-67 添加 Text - 设定温度的显示

2) 设置 Text 的属性参数

按照表 1-27 依次设置 ID_TEXT_Unit 和 ID_TEXT_Temp 的属性参数，实际设置项目如图 1-68 所示。

参数	ID		说明
	ID_TEXT_Unit	ID_TEXT_Temp	
坐标	Left: 368 Top: 108 Width: 20 Height: 24	Left: 334 Top: 108 Width: 32 Height: 24	设置 text 的坐标
text color	RGB (192, 192, 192)	RGB (192, 192, 192)	设置文本的颜色
Decimal mode		Mask:	设置以数字格式显示
Text	ID_RTEXT_DEG		设置以文本格式显示时显示的内容
text alignment	Offset x: 0. Offset y: 0. Alignment: 居左	Offset x: 0. Offset y: 0. Alignment: 居右	文本显示的位置
font	MicrosoftYaHeiUI_24_Normal_EXT_AA4	MicrosoftYaHeiUI_24_Normal_EXT_AA4	文本字体
rang		Min: 0 Max: 400	设置数字范围

表 1-27 Text 属性参数设置说明 - 温度显示

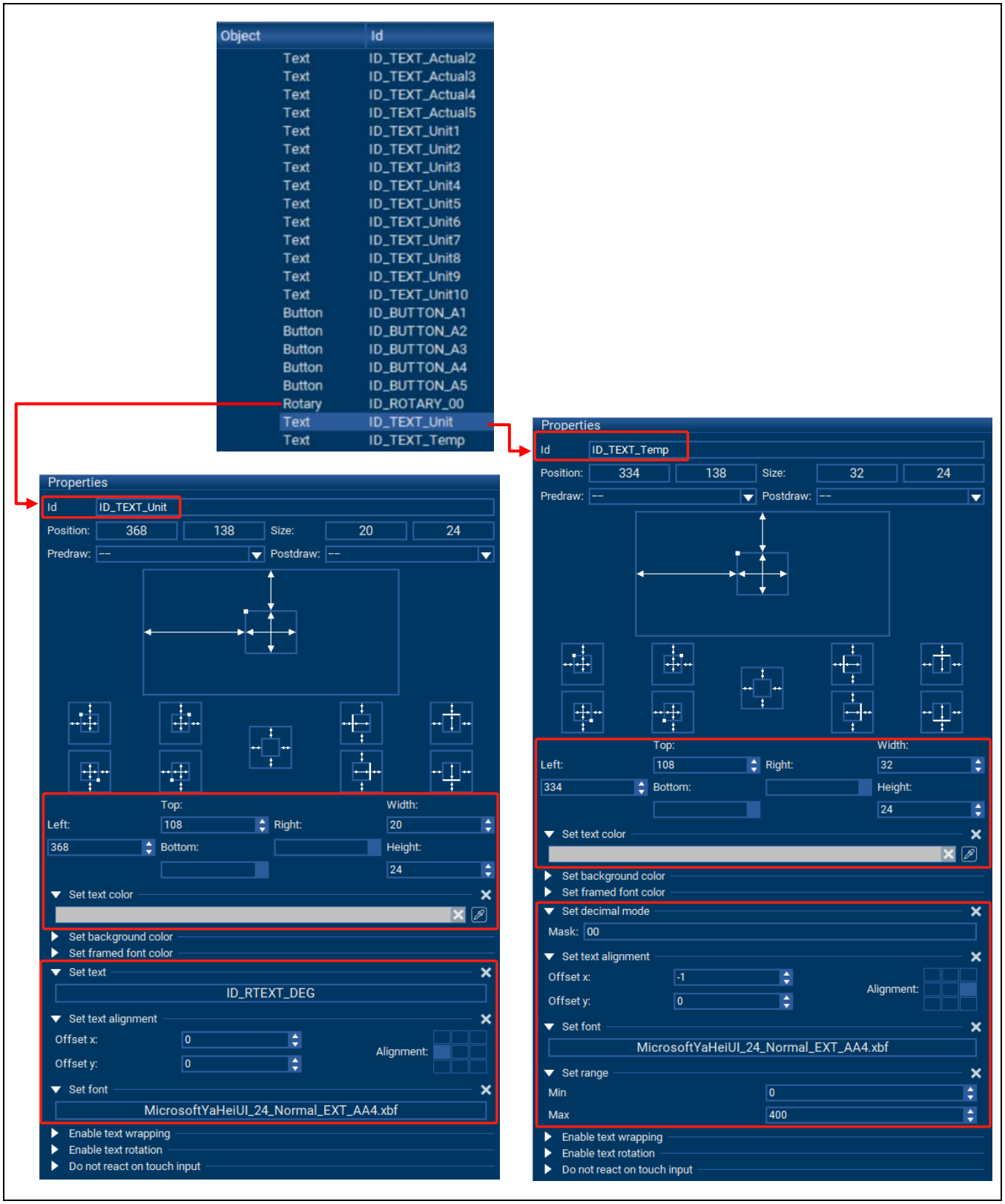


图 1-68 Text 属性参数设置 - 温度显示

9. 设定完成后的温度设置画面

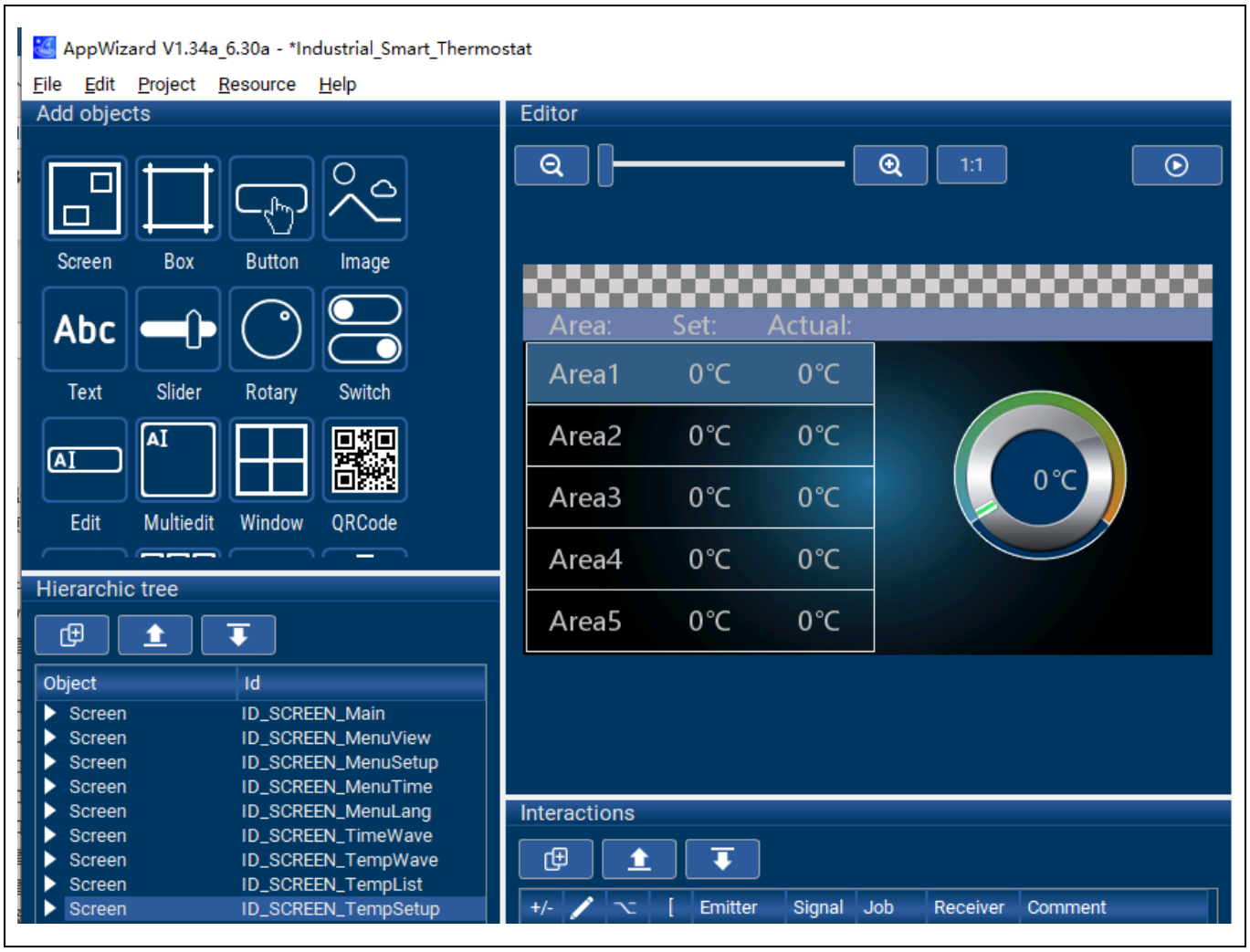


图 1-69 温度设置画面

1.2.2.14 创建时间设置画面

1. 添加画面

在 Add Objects 中点击 Screen 添加一个画面，在 Hierarchic tree 里重命名为 ID_SCREEN_Time。在 Properties 里设置坐标为 (0, 30, 480, 242)。

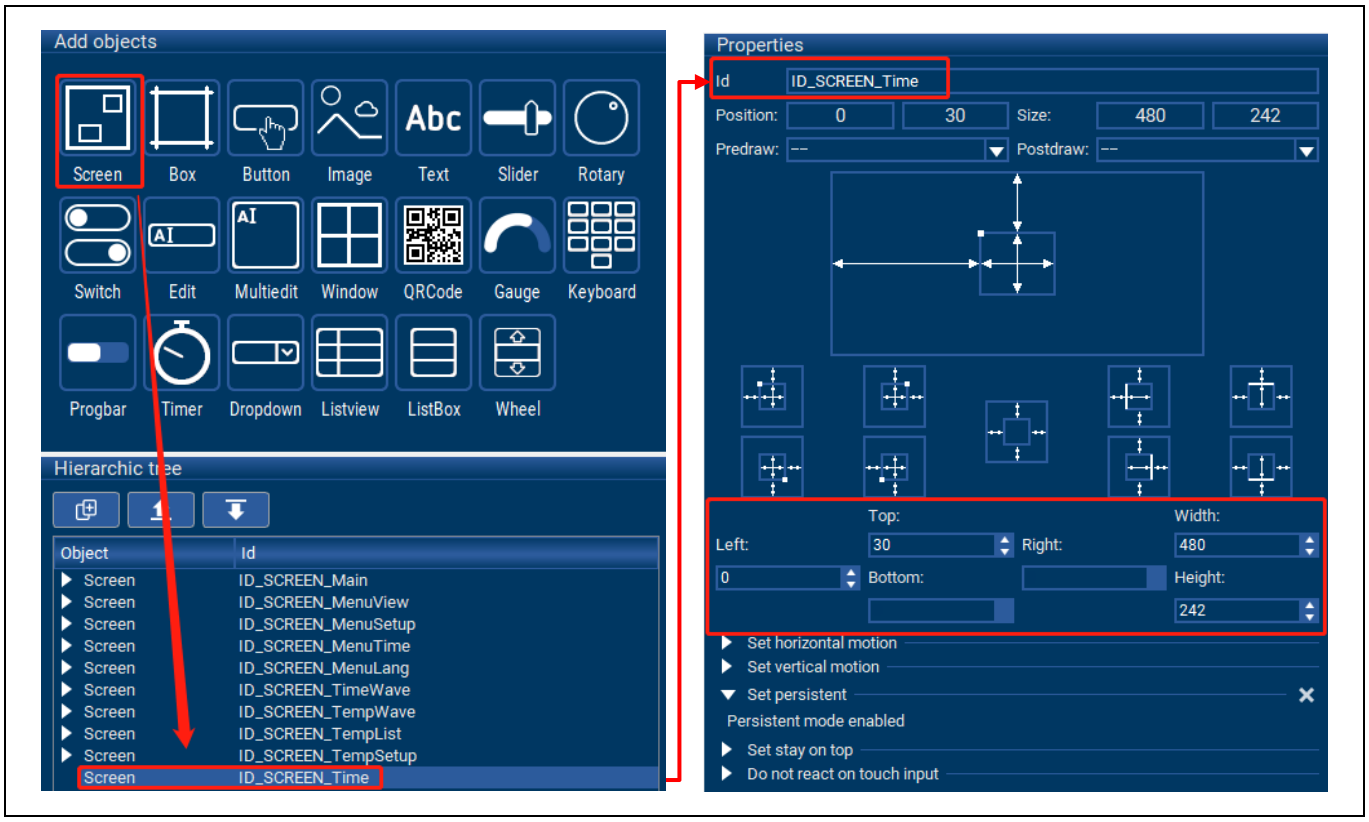


图 1-70 创建时间设置画面

2. 添加窗口

在 Add Objects 中点击 Window 为画面添加一个窗口，在 Hierarchic tree 里重命名为 ID_WINDOW_Time。在 Properties 里设置坐标为 (0, 0, 480, 242)。

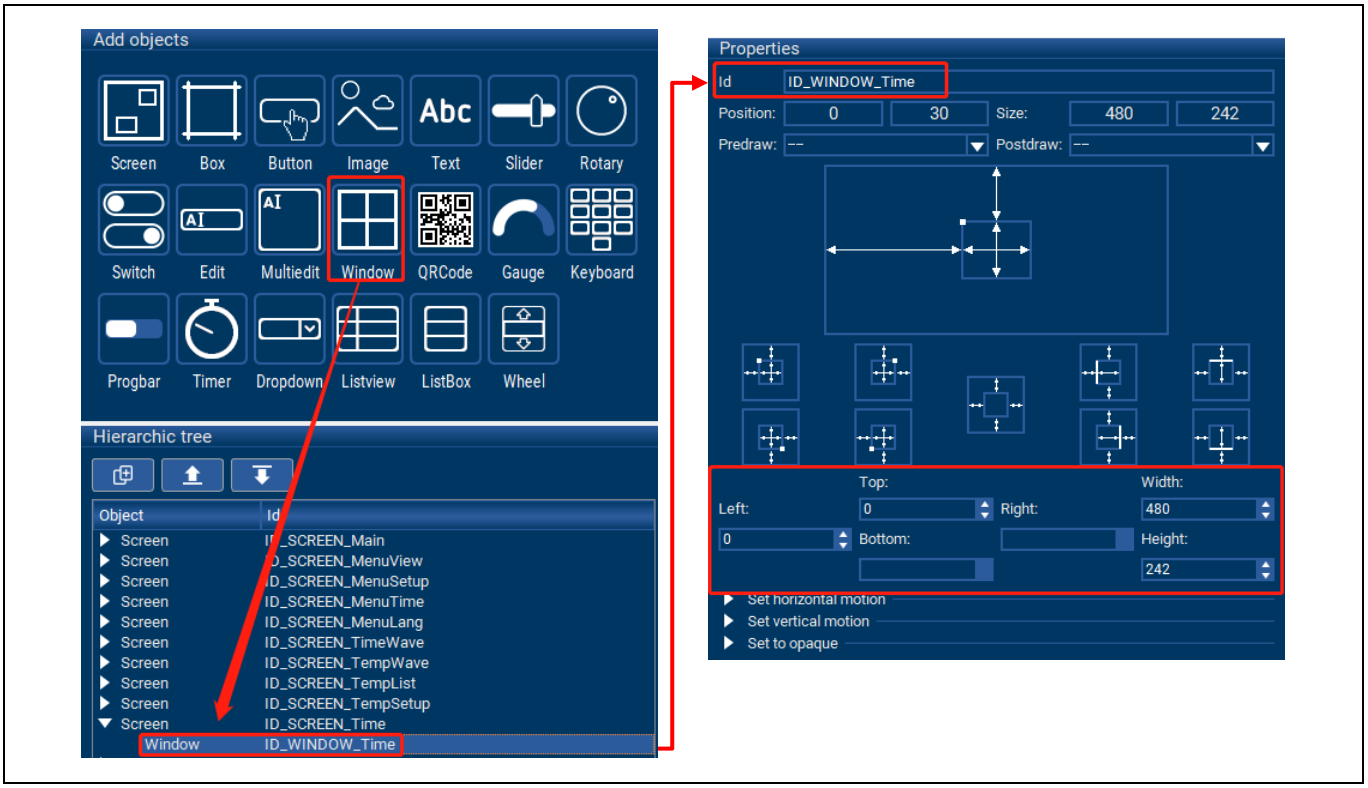


图 1-71 添加 Window - 时间设置

3. 添加 Image

添加 Image 用于背景和画面的功能显示。在 Add Objects 中点击 Image，为窗口添加 2 个 Image。如表 1-28 所示，在 Hierarchic tree 里重命名，并在 Properties 里设置坐标和 bitmap。实际操作如图 1-72。

ID	坐标 (left, top, width, height)	bitmap	说明
ID_IMAGE_BG	(0, 0, 480, 242)	BKG_Time_480x242.png	背景显示
ID_IMAGE_Type	(19, 19, 37, 39)	GUI_Time_Icon_37x38.png	用于指示画面的功能是时间设置

表 1-28 Image 的属性参数设置说明

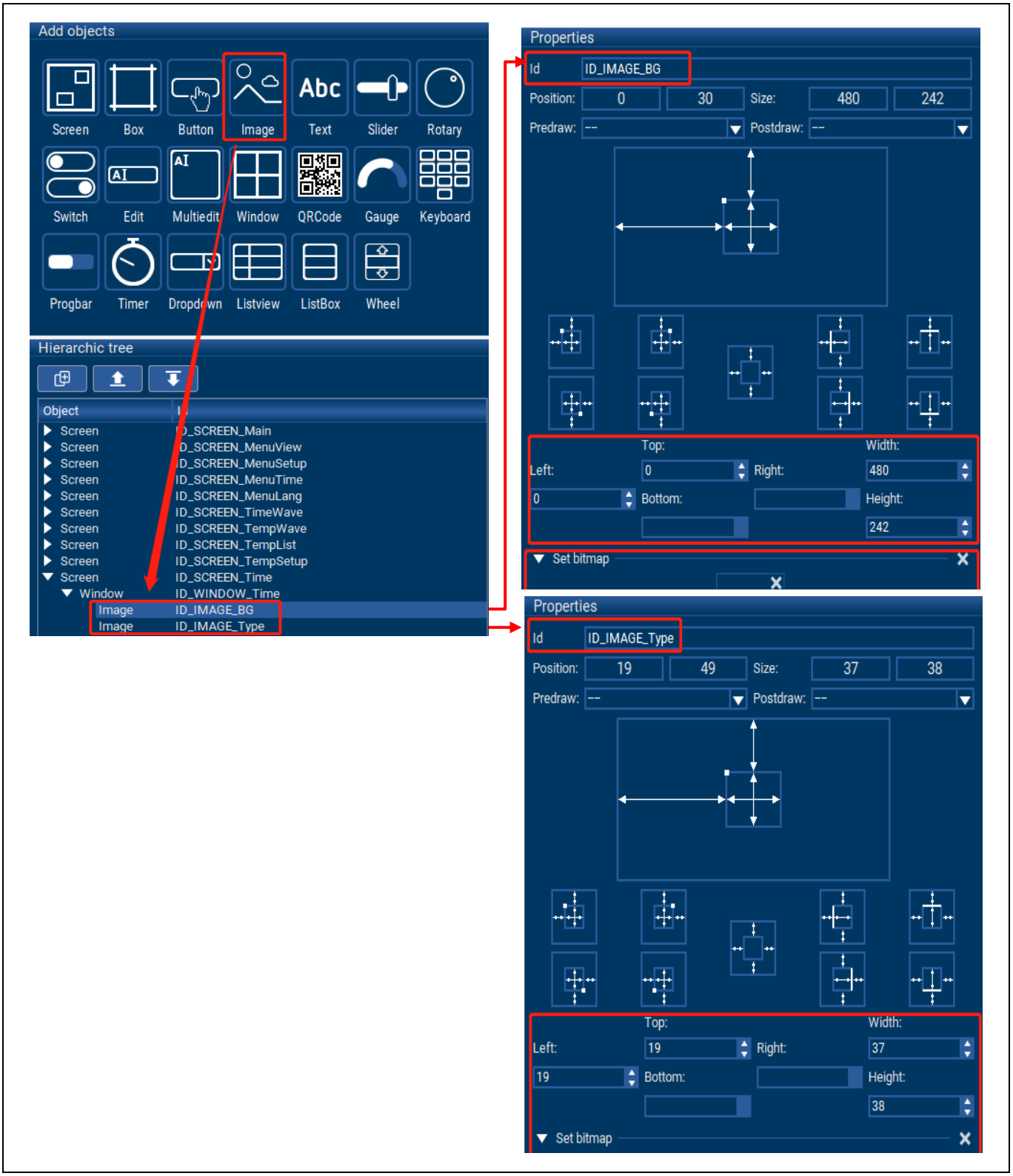


图 1-72 添加 Image - 背景和类型显示

4. 添加按钮

按钮用于保存当前设置和退出当前画面。在 Add Objects 里点击 Button，添加 2 个按键。如表 1-29 所示，在 Hierarchic tree 里重命名，并在 Properties 里设置坐标和 bitmap。实际设置项目如图 1-73 所示。

Button	坐标 (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

表 1-29 Button 属性参数设置说明 - 时间设置画面

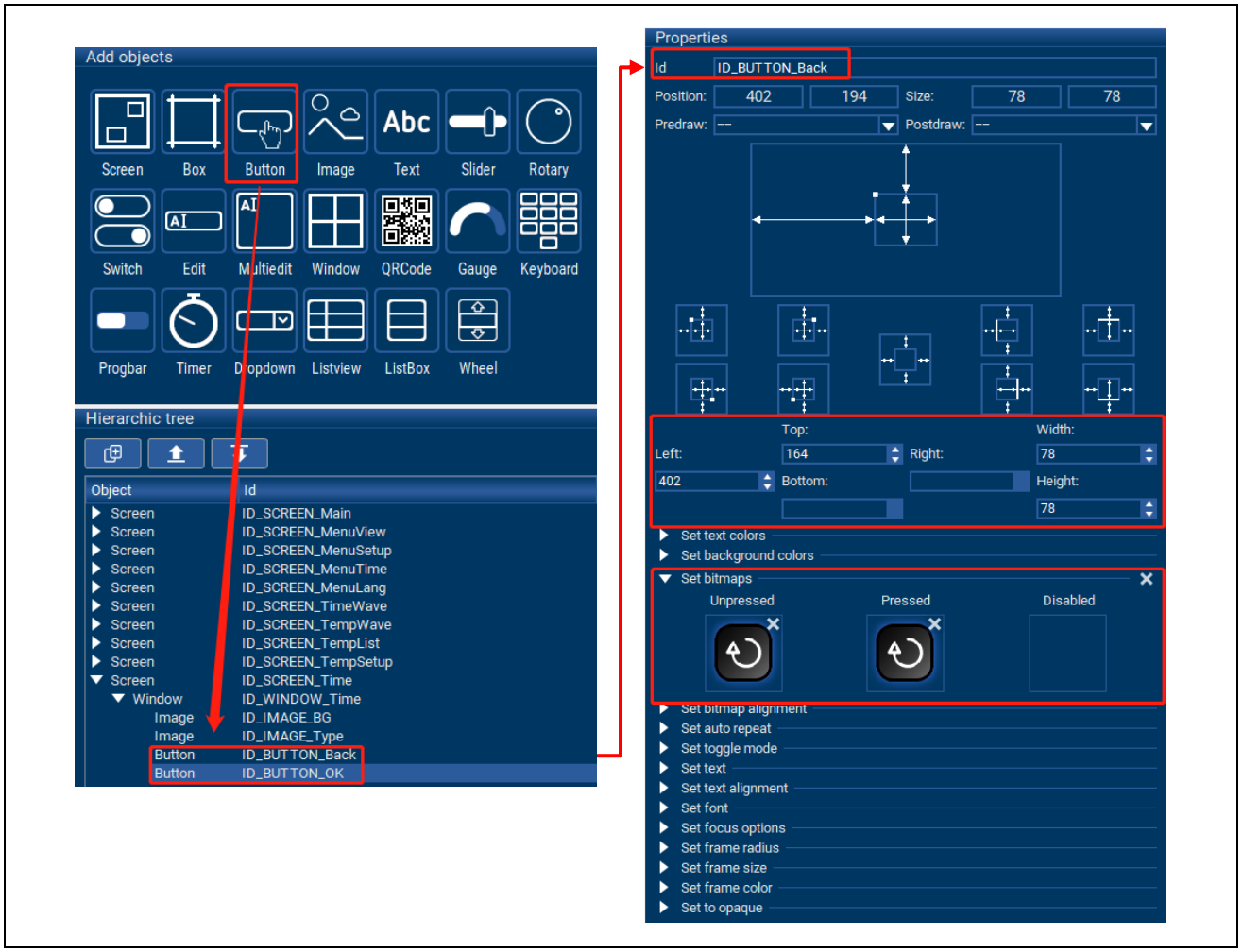


图 1-73 添加 Button - 确定、返回按钮

5. 添加 Box

使用 Box 作为时间设置选项的背景。在 Add Objects 里点击 Box 添加一个 Box，并在 Hierarchic tree 里重命名为 ID_BOX_00。在 Properties 里设置坐标为(120, 21, 240, 200)，color 为(255, 255, 255)，radius 为 25。

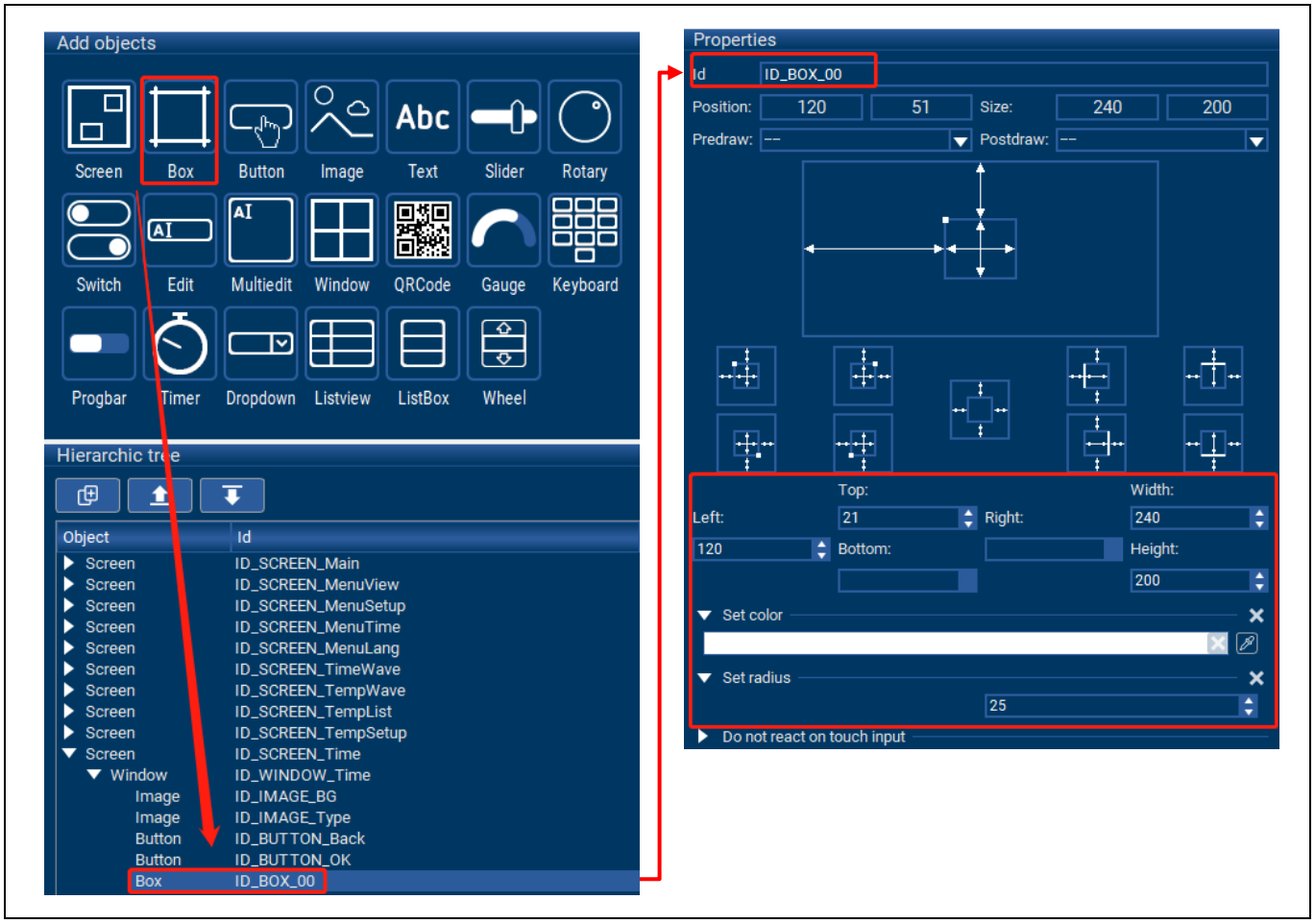


图 1-74 添加 Box - 时间设置选项背景

6. 添加 Text

为窗口添加 Text，用于显示文本时、分、秒和冒号。

在 Add Objects 中点击 Text，添加 5 个 Text，并在 Hierarchic tree 里按表 1-30 重新命名。在 Properties 里设置 text color 为 RGB(0, 0, 0)，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4。在 Properties 里按表 1-30 设置坐标、text、text alignment。实际设置项目如图 1-75。

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

表 1-30 Text 参数设置说明 - 文本显示

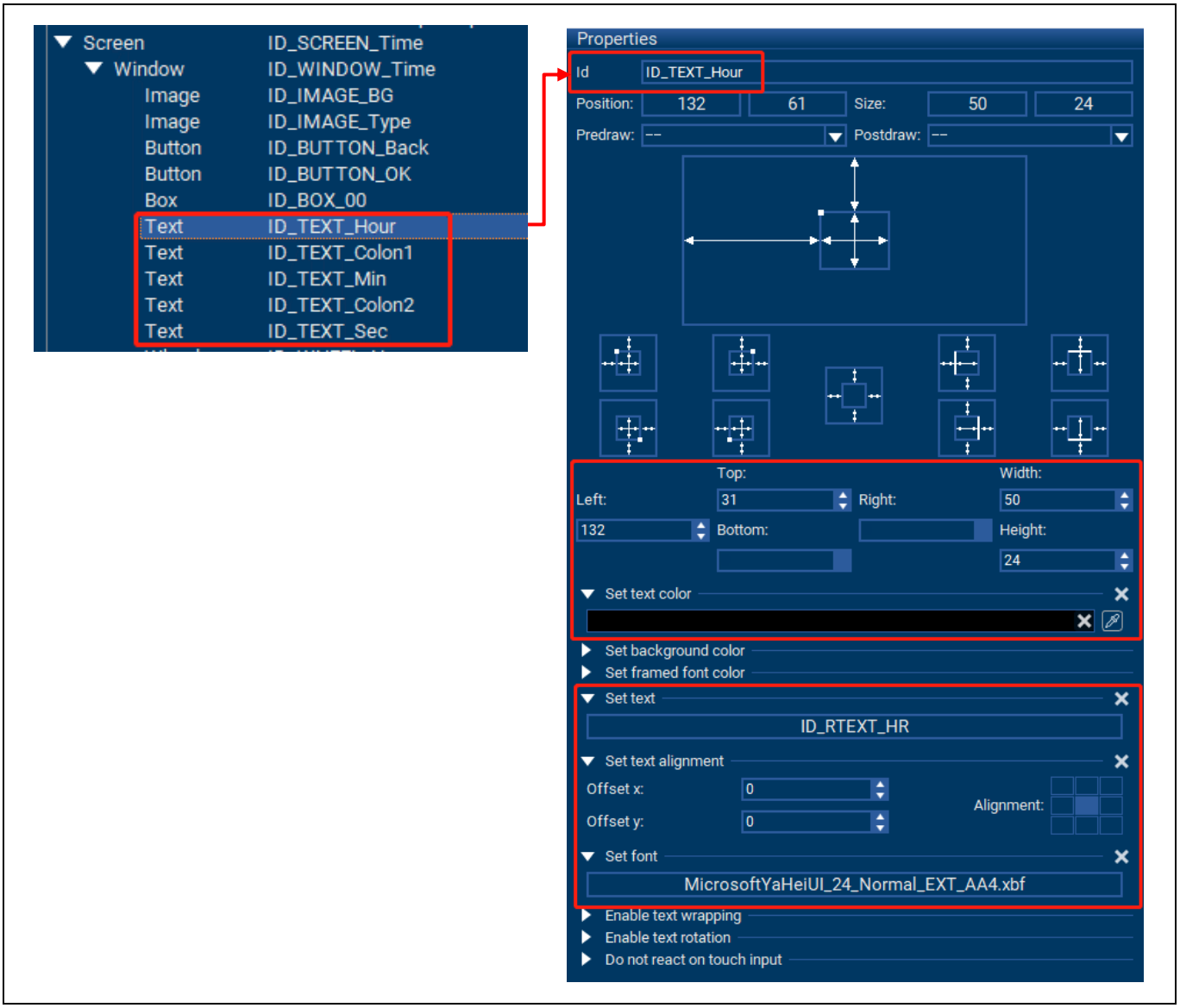


图 1-75 Text 属性参数设置 - 文本显示

7. 添加 Wheel

添加 Wheel 用于小时、分、秒的选择。

1) 添加 Wheel

在 Add Objects 里点击 Wheel，添加 3 个 Wheel，并按表 1-31 在 Hierarchic tree 里重命名。Properties 里设置坐标，实际设置项目如图 1-76。

ID	坐标 (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)

表 1-31 Wheel 坐标参数

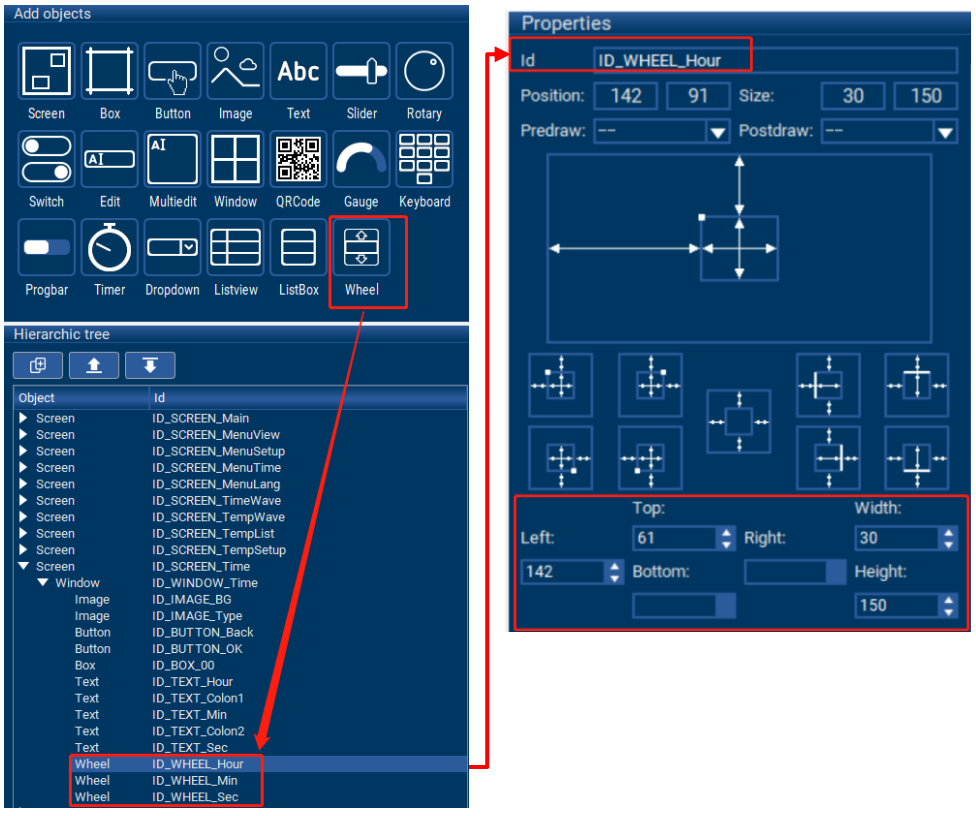


图 1-76 添加 Wheel - 时间设置选项

2) 设置 Wheel 的单元格参数

在 Properties 里按表 1-32 设置 endless mode、Cell and center size 和 color，实际设置项目如图 1-77。

属性	参数	说明
endless mode		选项首尾相连，循环显示
Cell and center size	cell: 32 center: 32	设置未选中和选中单元格的大小
color	Background cell：RBG (255, 255, 255) Background Center：RBG (255, 255, 255) Background：RBG (255, 255, 255)	设置未选中和选中单元格的背景色

表 1-32 单元格属性参数设置

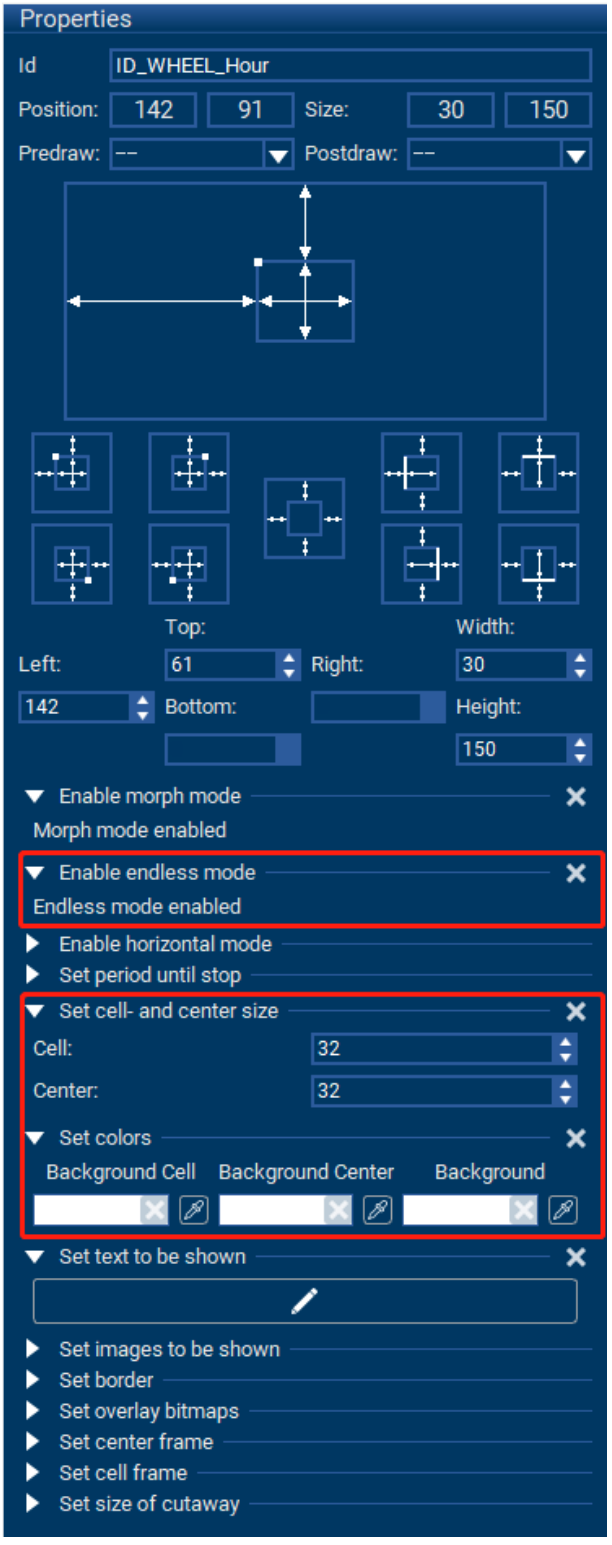


图 1-77 设置单元格属性参数

3) 设置显示 Text

在 Properties 里，按表 1-33，在 text to be show 里设置 Wheel 显示的文本内容。实际设置项目如图 1-78。

参数	ID			说明
	ID_WHEEL_Hour	ID_WHEEL_Min	ID_WHEEL_Sec	
Content	ID_CONTENT_HOUR	ID_CONTENT_MIN	ID_CONTENT_SEC	设置显示的文本列表
Cell alignment	Offset x：0 Offset y：0 Alignment：居中			未选中单元格的对齐
Cell color (R, G, B)	(170, 170, 170)			未选中单元格字体的颜色
Cell Font	MicrosoftYaHeiUI_24_Normal_EXT_AA4			字体
Center alignment	Offset x：0 Offset y：0 Alignment：居中			被选中单元格的对齐
Center Color (R, G, B)	(0, 0, 0)			被选中单元格的字体颜色

表 1-33 Wheel 显示内容设置

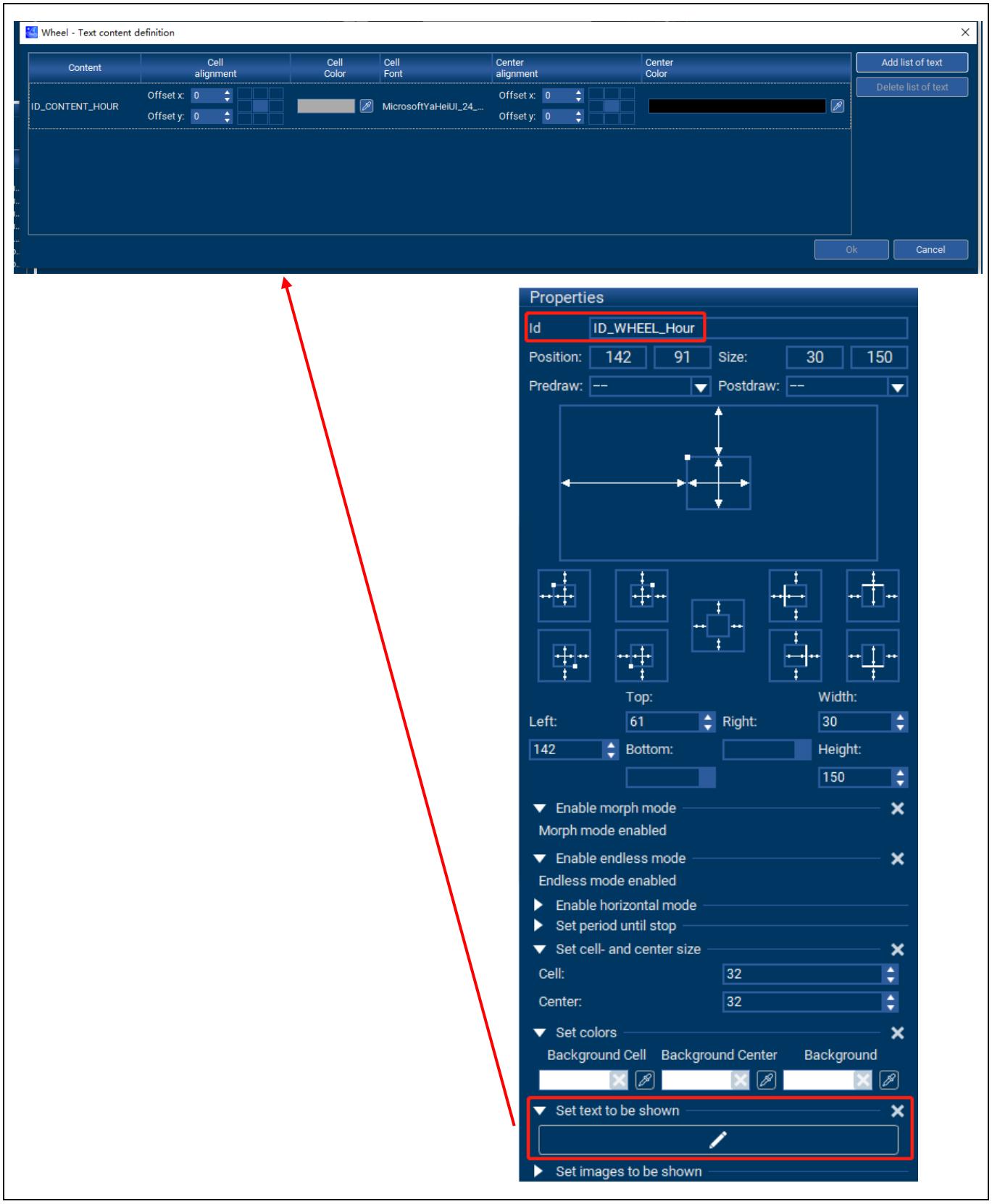


图 1-78 text to be shown 的设置

8. 设置完成的时间设置画面

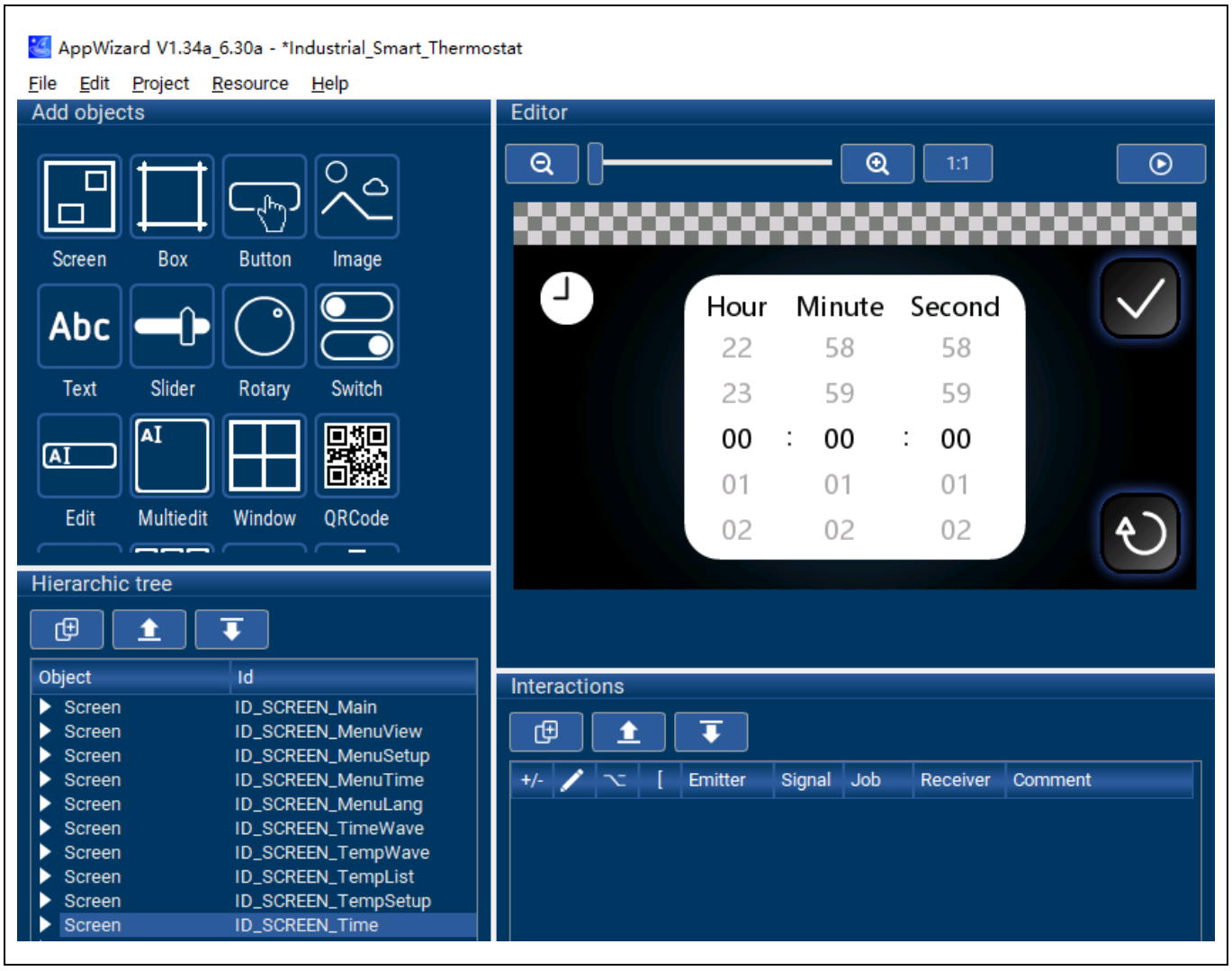


图 1-79 时间设置画面

1.2.2.15 建日期设置画面

1. 添加画面

在 Add Objects 中点击 Screen 添加一个画面，在 Hierarchic tree 里重命名为 ID_SCREEN_Date。在 Properties 里设置坐标为 (0, 30, 480, 242)。

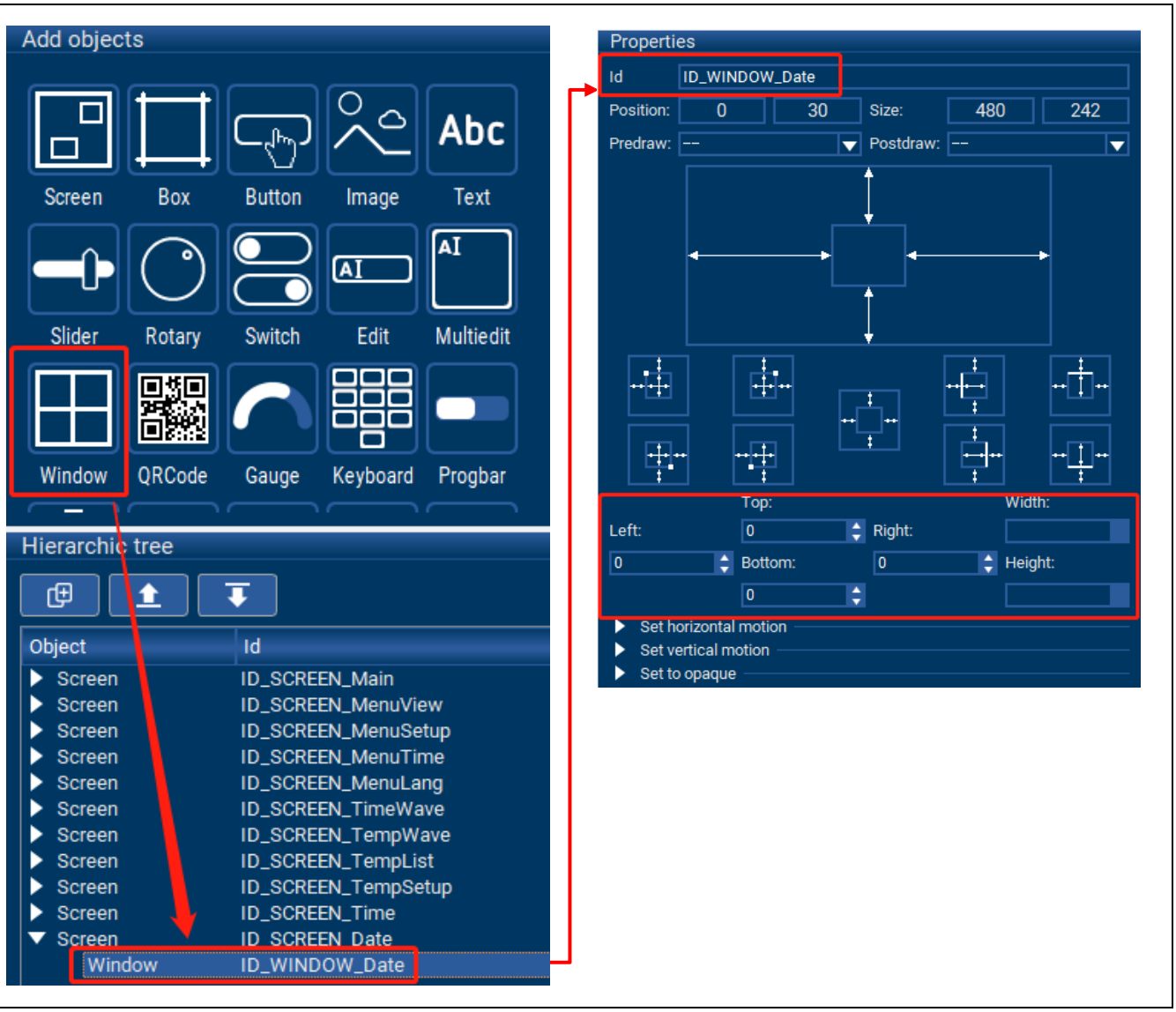


图 1-80 创建日期设置画面

2. 添加窗口

在 Add Objects 中点击 Window 为画面添加一个窗口，在 Hierarchic tree 里重命名为 ID_WINDOW_Date。在 Properties 里设置坐标为 (0, 0, 480, 242)。

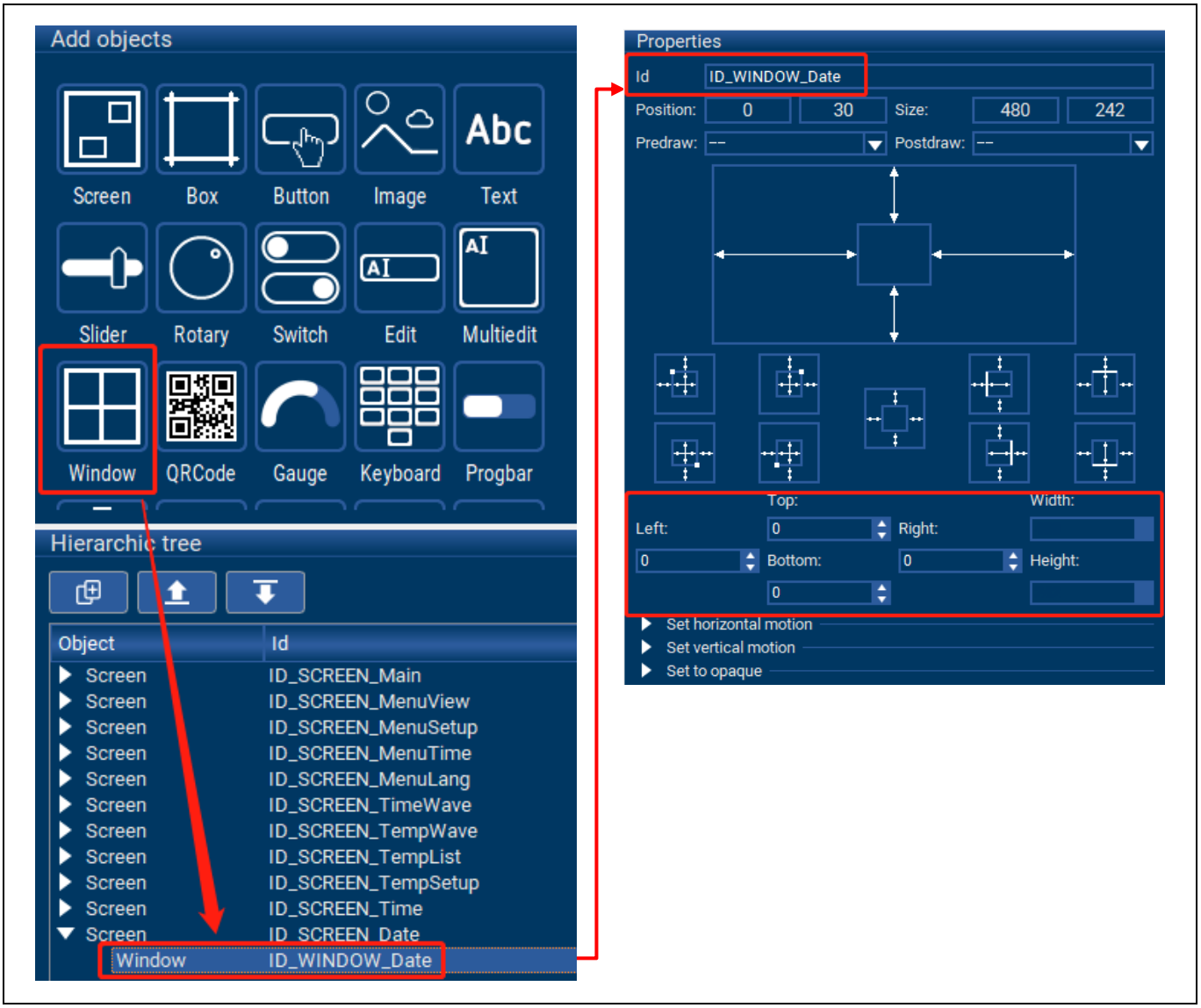


图 1-81 添加 Window - 显示日期设置

3. 添加 Image

添加 Image，用于背景和画面功能的显示。在 Add Objects 中点击 Image，为窗口添加 2 个 Image。按表 1-34 在 Hierarchic tree 里重命名，并在 Properties 里设置坐标和 bitmap。实际设置如图 1-82。

ID	坐标 (left, top, width, height)	bitmap	说明
ID_IMAGE_BG	(0, 0, 480, 242)	BKG_Time_480x242.png	背景显示
ID_IMAGE_Type	(19, 19, 37, 39)	GUI_Date_Icon_37x38.png	用于指示画面的功能是日期设置

表 1-34 Image 的属性参数设置说明

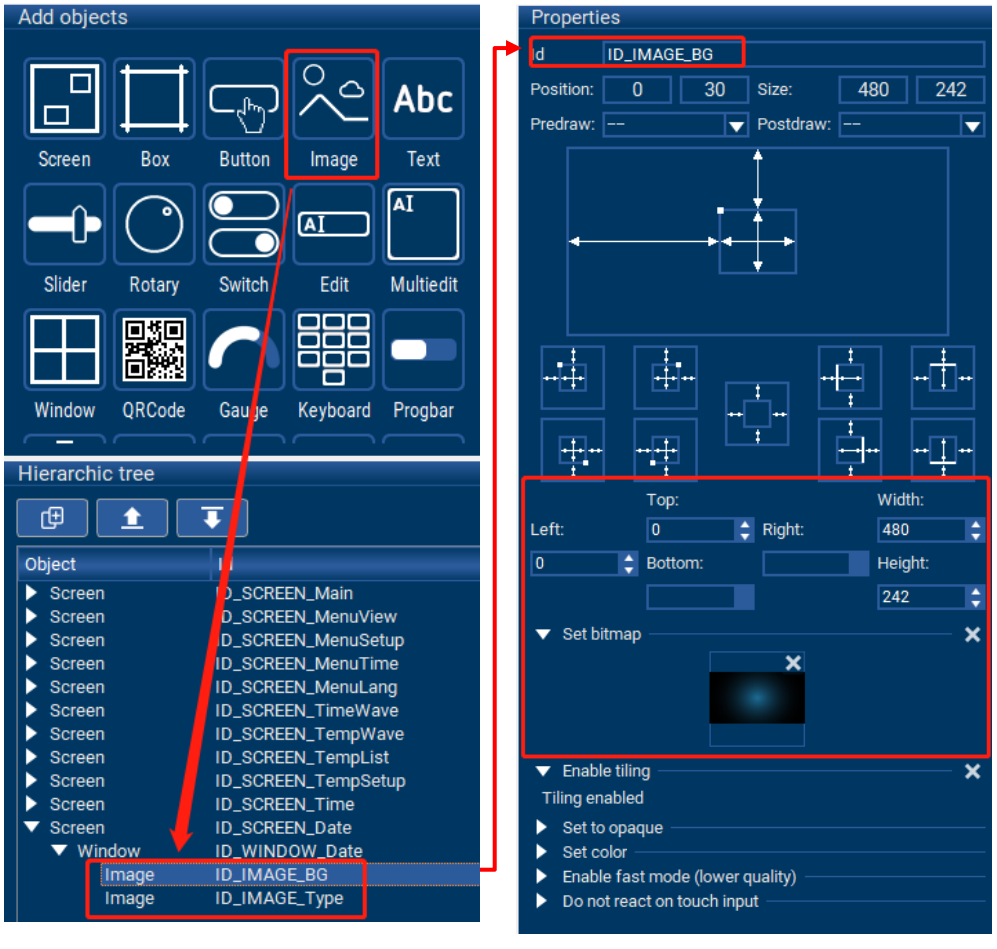


图 1-82 添加 Image - 背景和类型显示

4. 添加按钮

按钮用于保存当前设置和退出当前画面。在 Add Objects 里点击 Button，添加 2 个按键。按照表 1-35 在 Hierarchic tree 里为 Button 重命名，并在 Properties 里设置坐标和 bitmap。实际设置项目如图 1-83 所示。

Button	**坐标 (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

表 1-35 Button 的属性参数设置说明 - 日期设置画面

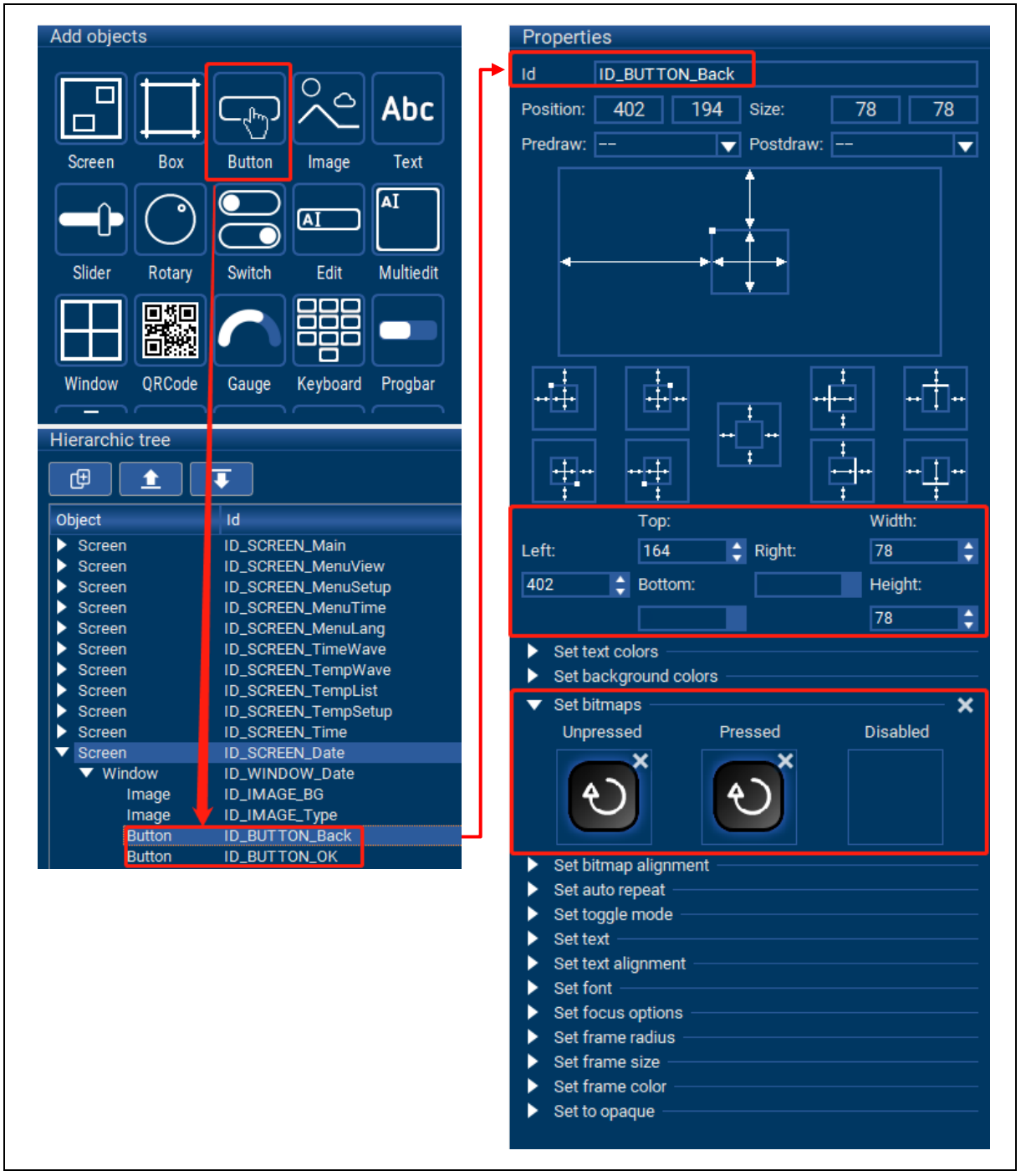

Add objects
Screen
Box
Button
Image
Text
Slider
Rotary
Switch
Edit
Multiedit
Window
QRCode
Gauge
Keyboard
Progbar
Hierarchic tree
Object
Id
Screen ID_SCREEN_Main
Screen ID_SCREEN_MenuView
Screen ID_SCREEN_MenuSetup
Screen ID_SCREEN_MenuTime
Screen ID_SCREEN_MenuLang
Screen ID_SCREEN_TimeWave
Screen ID_SCREEN_TempWave
Screen ID_SCREEN_TempList
Screen ID_SCREEN_TempSetup
Screen ID_SCREEN_Time
Screen ID_SCREEN_Date
Window ID_WINDOW_Date
Image ID_IMAGE_BG
Image ID_IMAGE_Type
Button ID_BUTTON_Back
Button ID_BUTTON_OK
Properties
Id ID_BUTTON_Back
Position: 402 194 Size: 78 78
Predraw: --- Postdraw: ---
Top:
Width:
Left: 164 Right: 78
402 Bottom:
Height:
78
Set text colors
Set background colors
Set bitmaps
Unpressed
Pressed
Disabled
Set bitmap alignment
Set auto repeat
Set toggle mode
Set text
Set text alignment
Set font
Set focus options
Set frame radius
Set frame size
Set frame color
Set to opaque

图 1-83 添加按钮

5. 添加 Box

使用 Box 显示时间设置选项的背景。在 Add Objects 里点击 Box 添加 1 个 Box，并在 Hierarchic tree 里重命名为 ID_BOX_00。在 Properties 里设置坐标为 (100, 21, 280, 200)，color 为 (255, 255, 255)，radius 为 25。

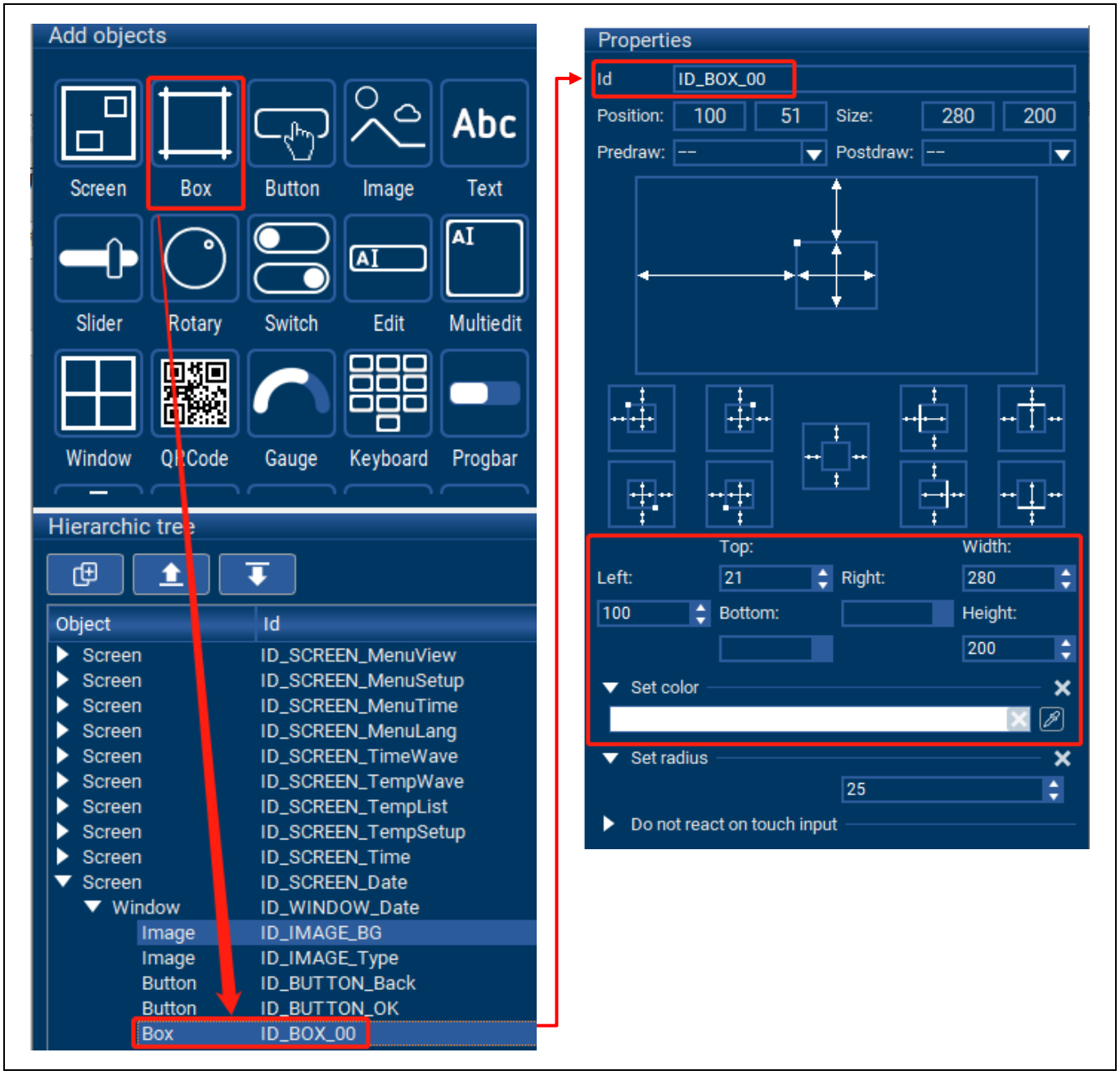


图 1-84 添加 Box - 日期设置选项背景

6. 添加 Text

为窗口添加 Text，用于显示年、月、日。

在 Add Objects 中点击 Text，添加 3 个 Text，并在 Hierarchic tree 按表 1-36 重新命名。在 Properties 里设置 text color 为 RGB (0, 0, 0)，font 为 MicrosoftYaHeiUI_24_Normal_EXT_AA4。按照表 1-36 设置坐标、text、text alignment。实际设置项目如图 1-85。

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

表 1-36 Text 参数设置说明 - 文本显示

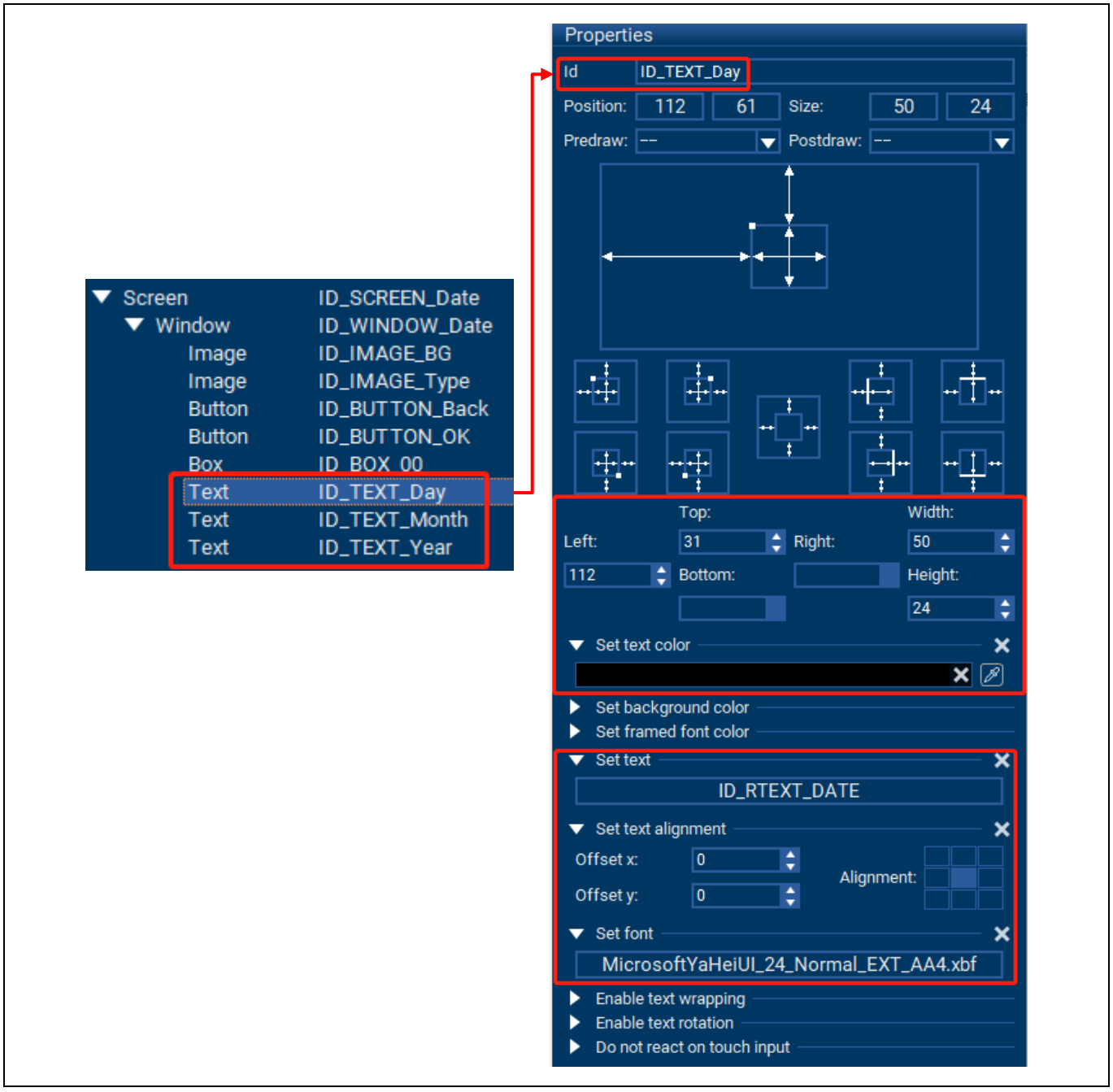


图 1-85 Text 属性参数设置 - 文本显示

7. 添加 Wheel

添加 3 个 Wheel，分别用于年、月、日的选择。

1) 添加 Wheel

在 Add Objects 里点击 Wheel，添加 3 个 Wheel，并按表 1-37 在 Hierarchic tree 重命名，在 Properties 里设置坐标。

ID	**坐标 (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)

表 1-37 Wheel 坐标参数

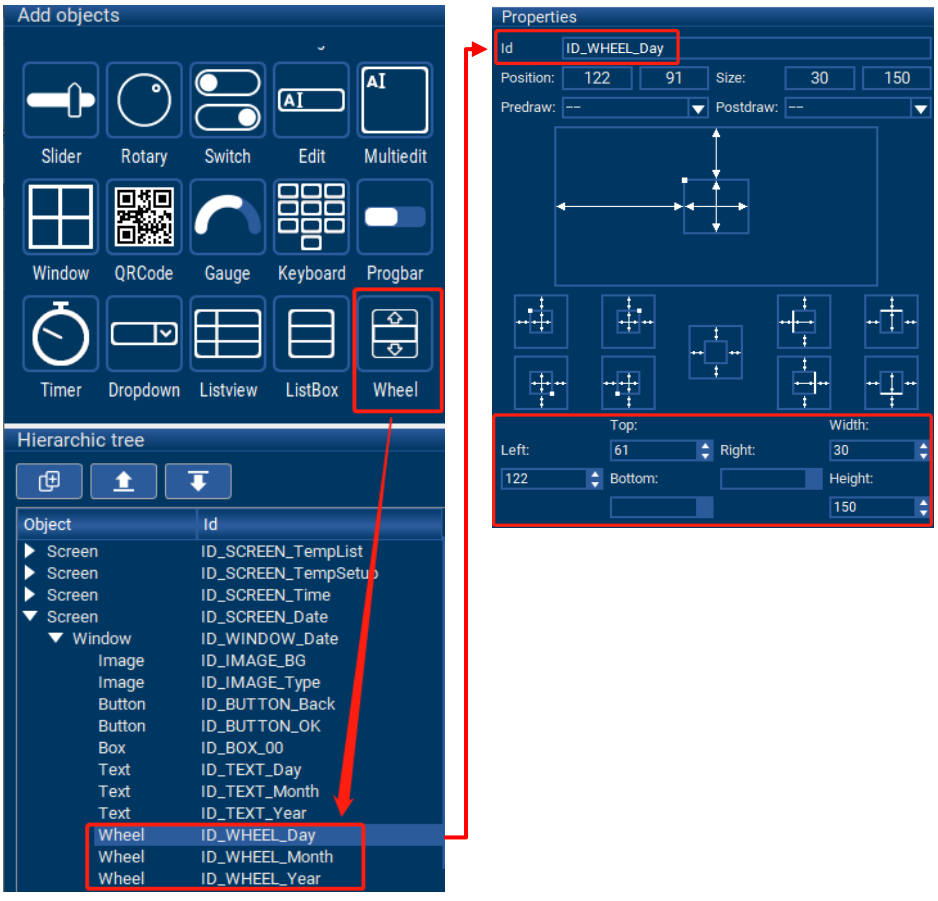


图 1-86 添加 Wheel - 日期设置选项

2) 设置 Wheel 的单元格参数

按照表 1-38 在 Properties 里设置 endless mode、Cell and center size 和 color。实际设置项目如图 1-87。

属性	参数	说明
endless mode		选项首尾相连，循环显示
Cell and center size	cell: 32 center: 32	设置未选中和选中单元格的大小
color	Background cell：RGB (0, 0, 0) Background Center：RGB (0, 0, 0) Background：RGB (0, 0, 0)	设置未选中和选中单元格的背景色

表 1-38 单元格属性参数设置说明

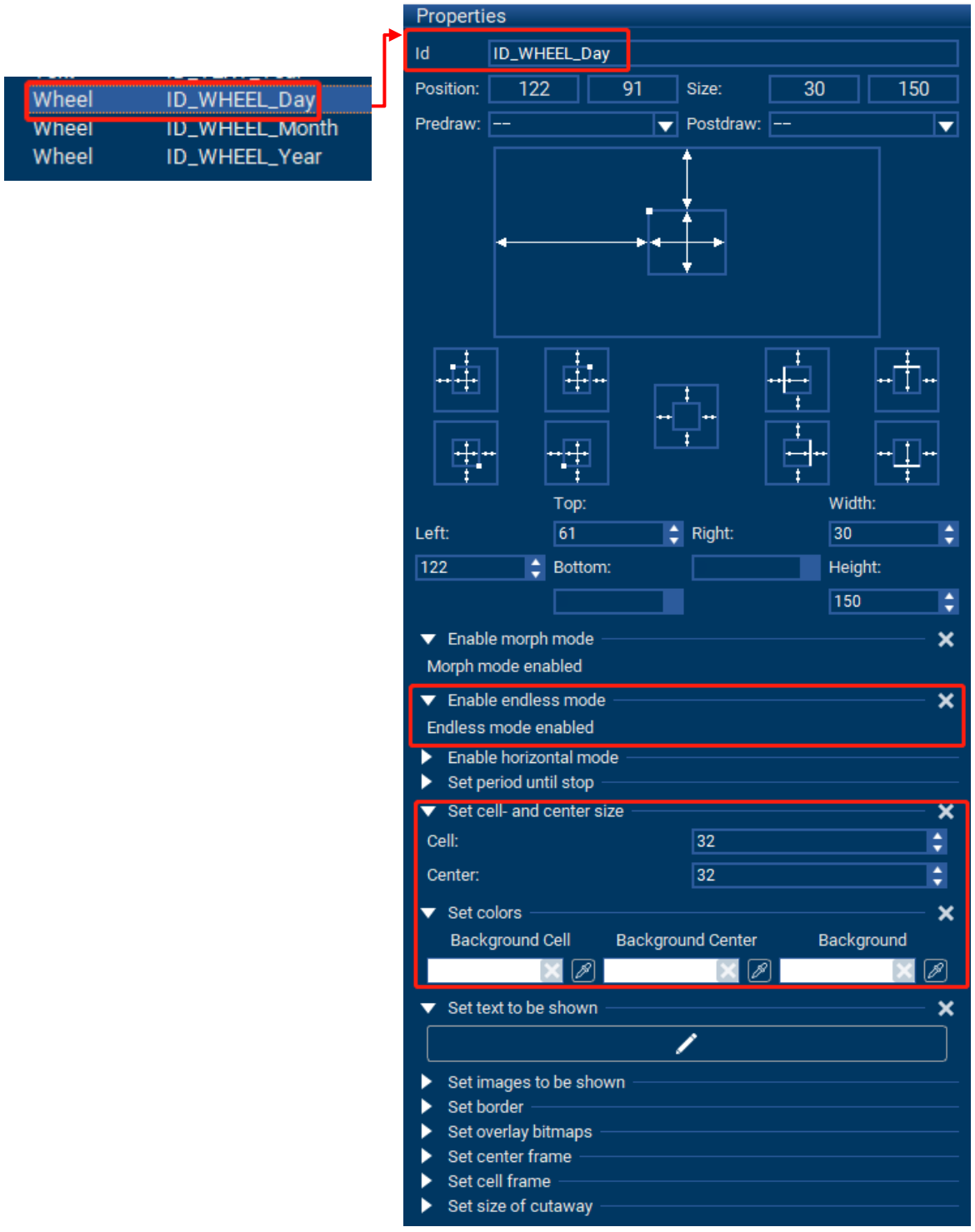


图 1-87 单元格属性参数设置

3) 设置显示 Text

在 Properties 里 text to be show 设置 Wheel 显示的文本内容。按表 1-39 设置 text to be show，实际设置项目如图 1-88。

参数	ID			说明
	ID_WHEEL_Day	ID_WHEEL_Month	ID_WHEEL_Year	
Content	ID_CONTENT_DAY	ID_CONTENT_MONTH	ID_CONTENT_YEAR	设置显示的文本列表
Cell alignment	Offset x：0 Offset y：0 Alignment：居中			未选中单元格的对齐
Cell color (R, G, B)	(170, 170, 170)			未选中单元格字体的颜色
Cell Font	MicrosoftYaHeiUI_24_Normal_EXT_AA4			字体
Center alignment	Offset x：0 Offset y：0 Alignment：居中			被选中单元格的对齐
Center Color (R, G, B)	(0, 0, 0)			被选中单元格的字体颜色

表 1-39 Wheel 显示内容设置

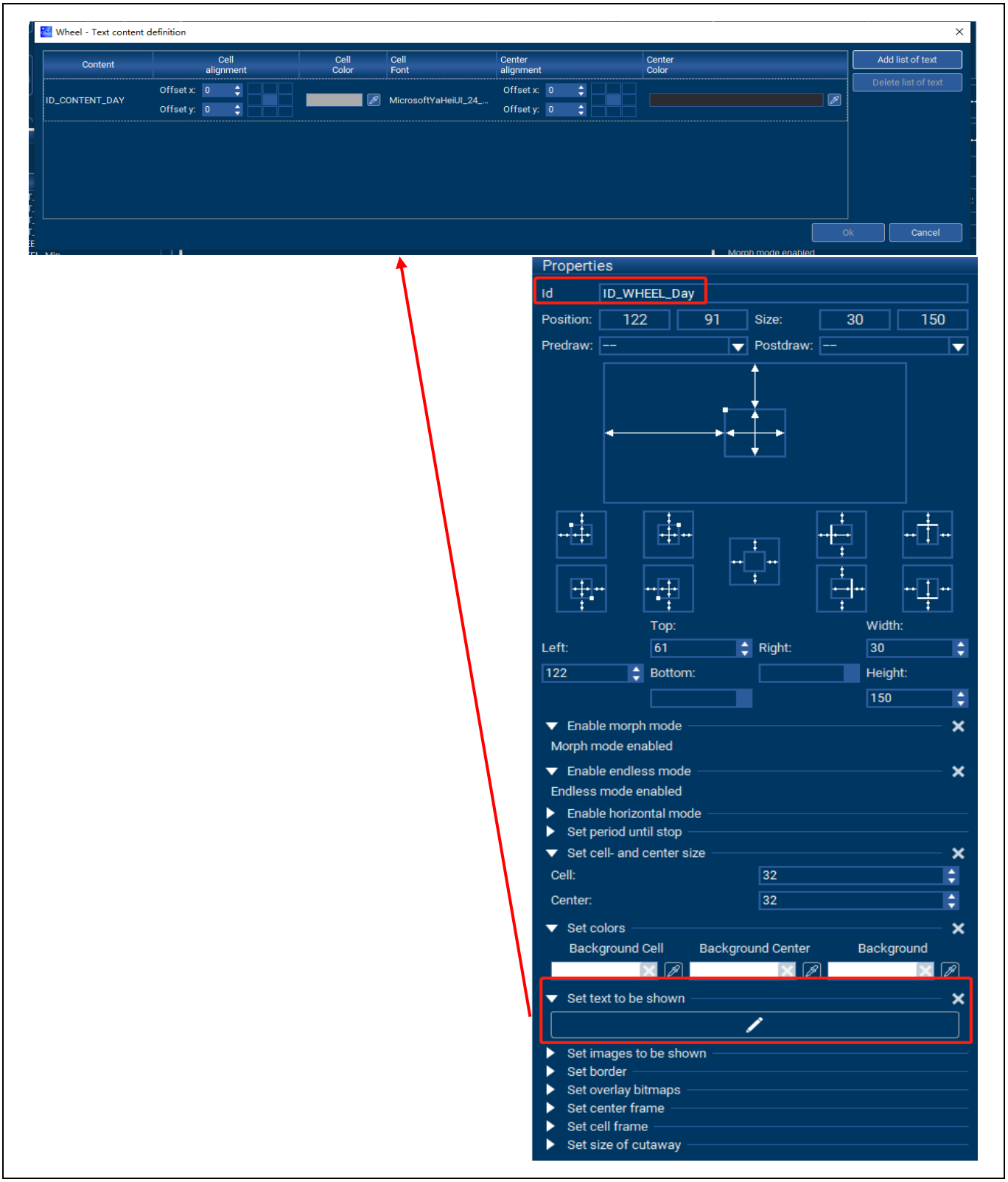


图 1-88 text to be shown 的设置

8. 设置完成的时间设置画面

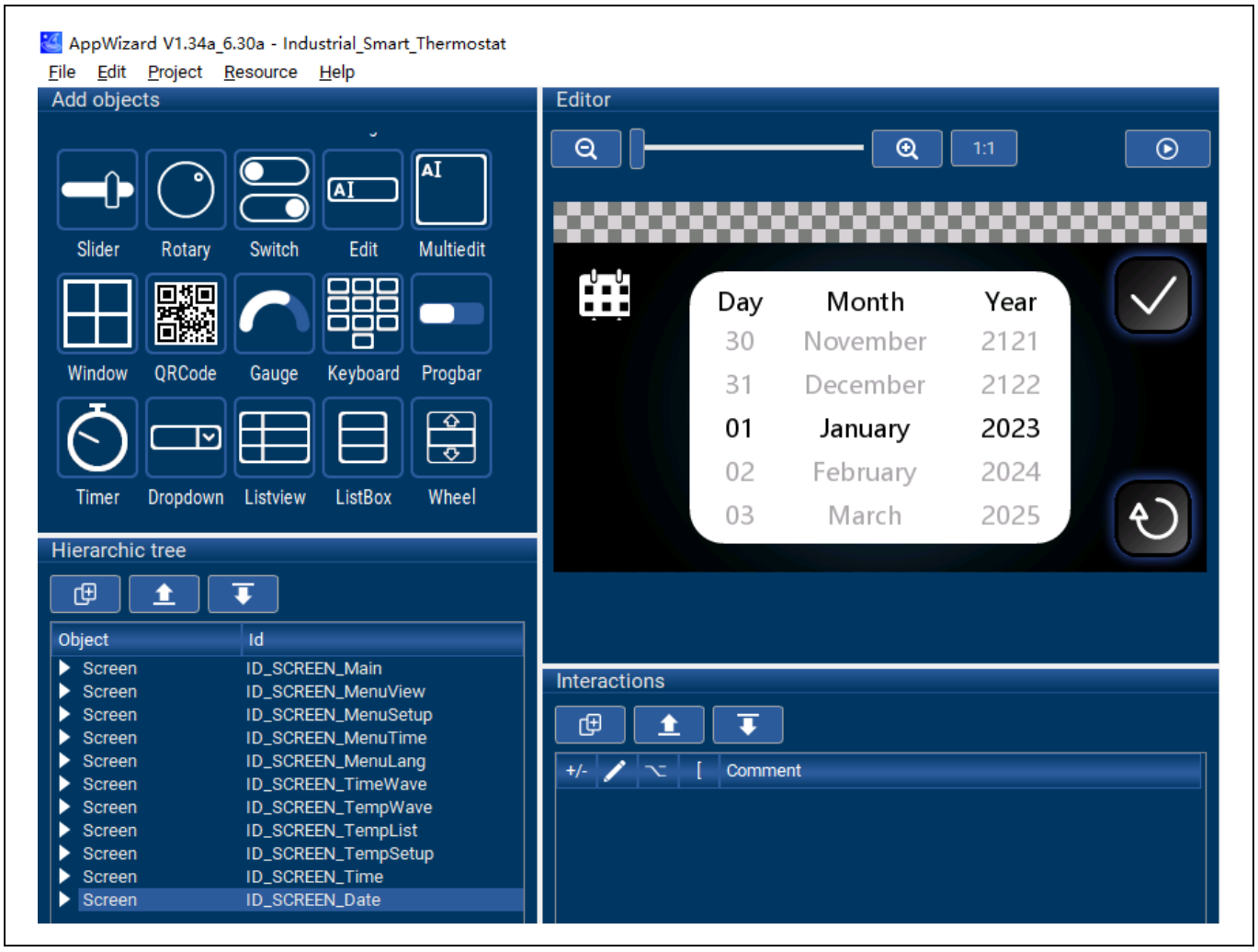


图 1-89 日期设置画面

1.2.2.16 互动设置 (interaction)

互动主要实现：当 **Emitter A** 做 **Signal**，**Receiver B** 做 **Job**。

或当 **Emitter A** 做/发生 **Signal**，对 **Receiver B** 做 **Job**。

完整设置项目如图 1-90，设置项的说明如表 1-40。

图 1-90 互动设置

内容	说明
	添加或删除互动动作
	设置互动动作的参数
	发生互动动作的条件，非必须
	组设置，可以将几个互动动作设置成一组
Emitter	发送者
Signal	发送者发生的事件
Job	接受者要发生的事件
Receiver	接受者
Comment	注释

表 1-40 Interaction 设置说明

1. 主菜单的互动设置

1) 开机同时显示主菜单和温度列表

开机时需要同时显示主菜单和温度列表。这里通过 ID_SCREEN_Main 画面初始化时触发显示温度列表画面 ID_SCREEN_TempList 来实现，实际设置如图 1-91。

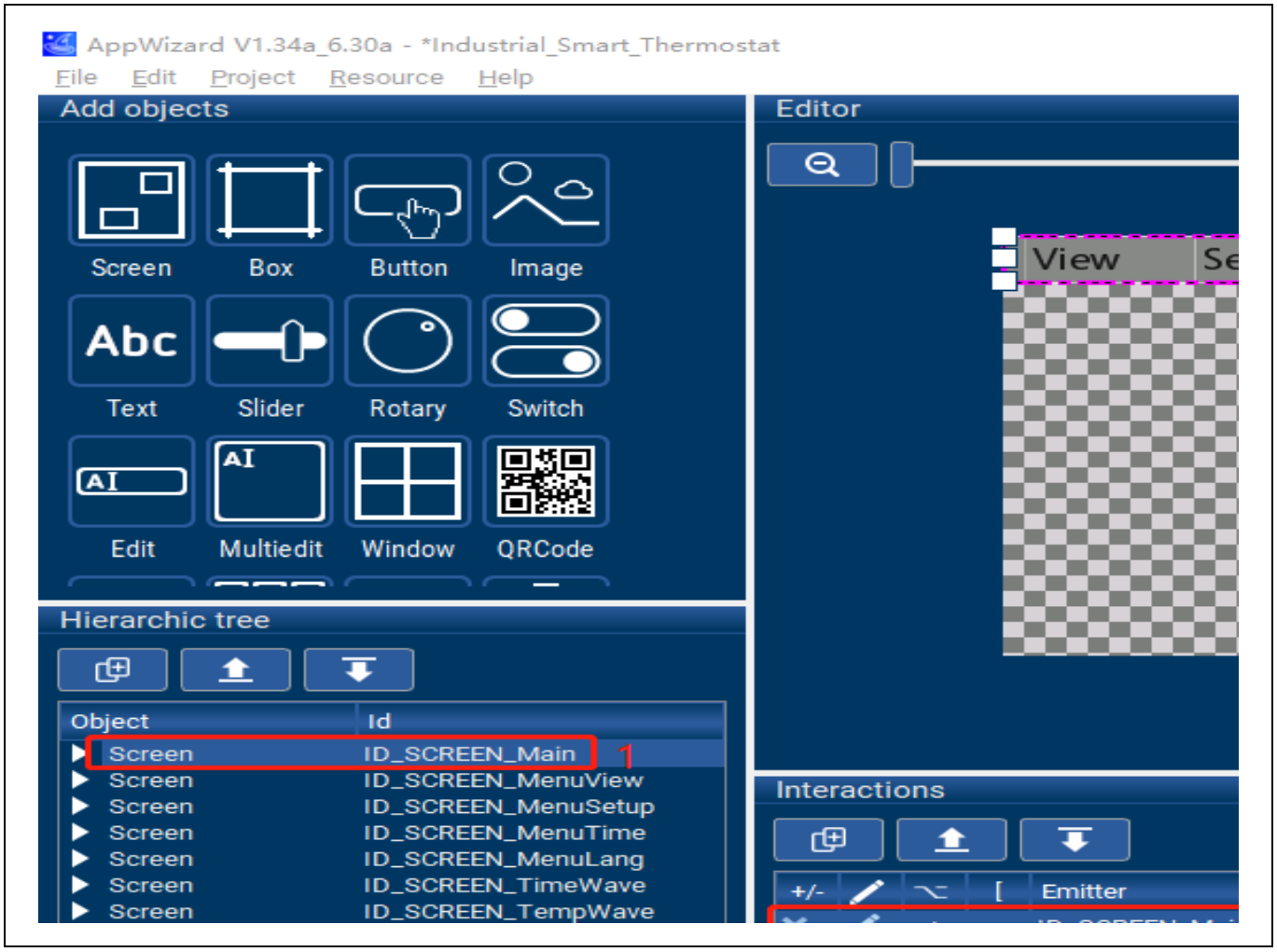
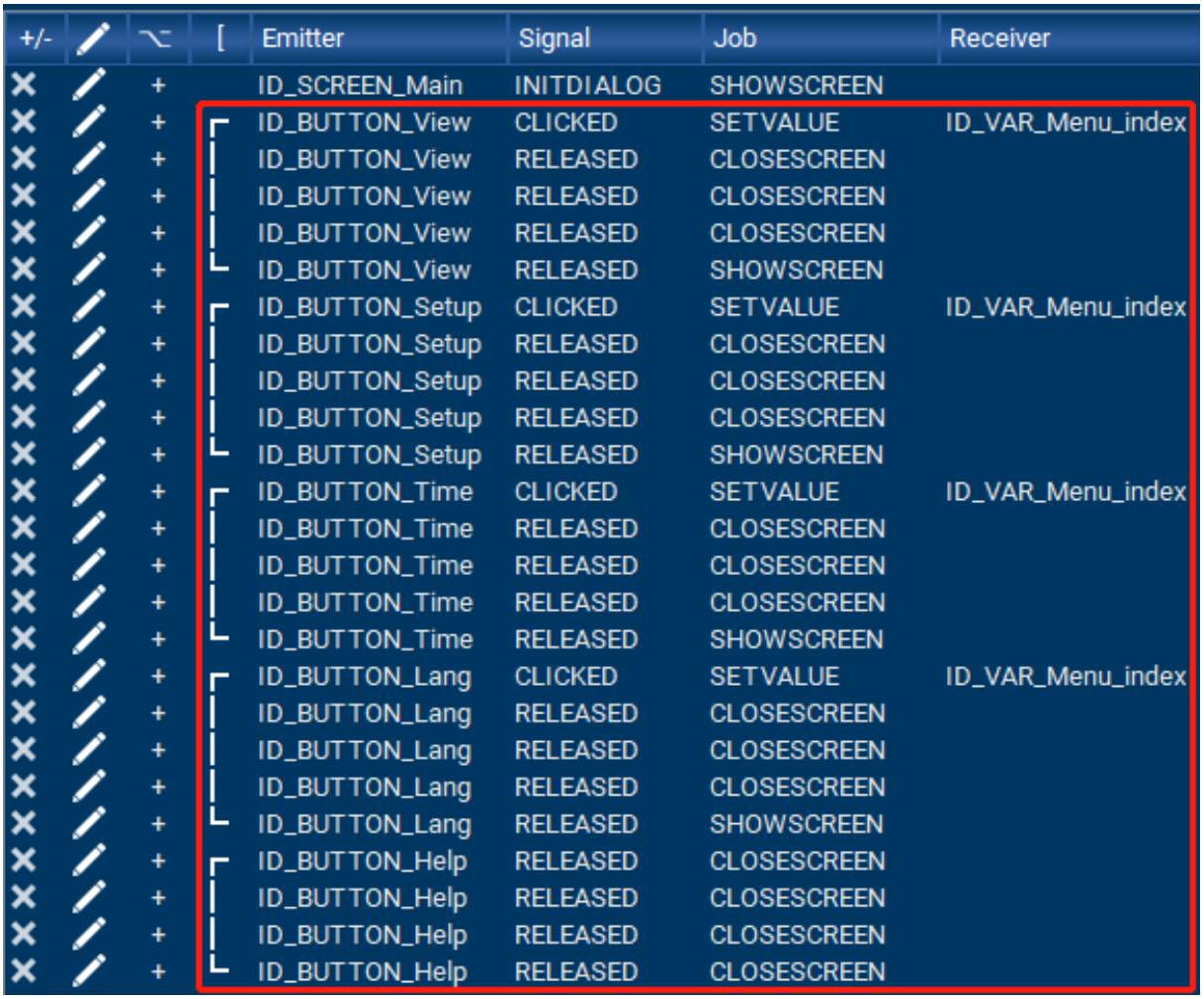


图 1-91 互动设置 - 主菜单初始化时显示温度列表

2) 菜单中的按钮按下

当主菜单里的按钮按下时，需要显示对应的下拉菜单和隐藏其他下拉菜单。注意 ID_BUTTON_Help 按钮没有对应的下拉菜单，当它按下时只会隐藏其他下拉菜单。实际设置项目如图 1-92，动作具体说明表 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	

图 1-92 互动设置 - 主菜单按钮

参数	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

参数	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			

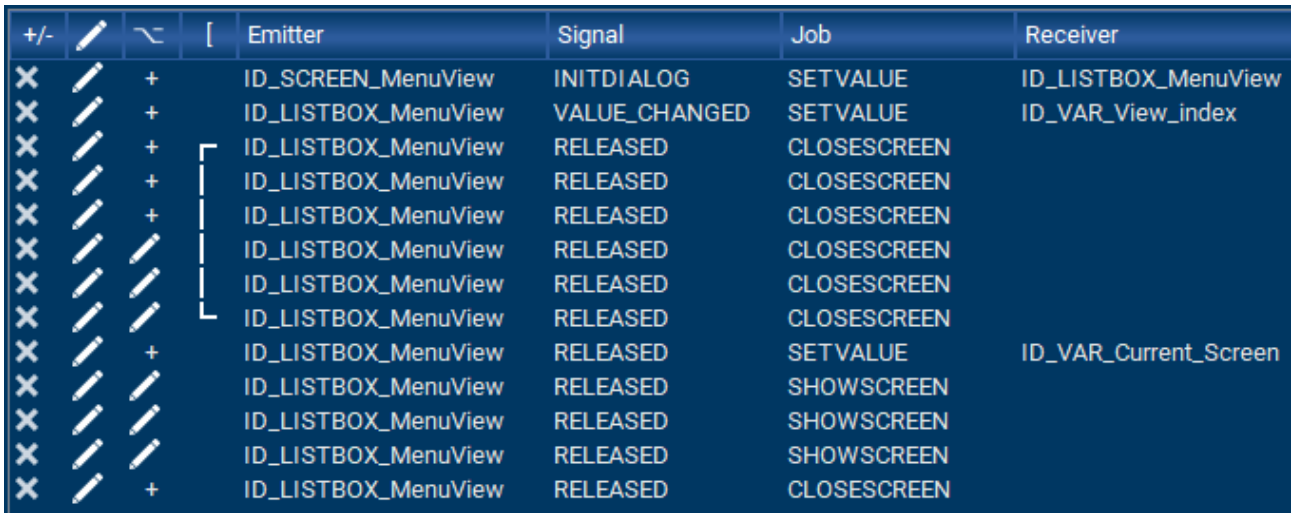
表 1-41 互动说明 - 主菜单按钮

2. 下拉菜单的互动设置

1) View 下拉菜单的互动设置

当 ID_SCREEN_MenuView 画面显示时，默认焦点在第一个选项上。当选择某个选项后需要显示对应的画面以及保存焦点位置。定义一个变量 ID_VAR_View_index 用来

保存 index。当画面初始化时设置变量值为 0；当 ID_LISTBOX_Menuview 值发生变化时将选项的 index 保存到变量里。当松开按住的选项后，根据变量值显示对应的画面，并保存画面的 index。实际设置项如图 1-93，动作的说明如表 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	

图 1-93 互动设置 - View 下拉菜单

参数	条件	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_in				

参数	条件	Emitter	Signal	Job	Receiver
	dex ≠ 1				
ID_SCREEN_TimeWave	ID_VAR_View_index ≠ 2				
ID_VAR_View_index				SETVALUE	ID_VAR_Current_Screen
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	

表 1-42 互动说明 - View 下拉菜单

2) Setup 和 Time 下拉菜单的互动设置和 View 下拉菜单类似，如图 1-94 和图 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	

图 1-94 互动设置 - Setup 下拉菜单

+/-	⌥	[	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	

图 1-95 互动设置 - Time 下拉菜单

3) 语言选择下拉菜单的互动设置

当语言选择下拉菜单选中时，需要设置相应的语言为中文或英文，并隐藏下拉菜单。另外如果当前显示画面是日期设置画面时，需要重新显示画面，语言设置才能生效，实际设置如图 1-96。

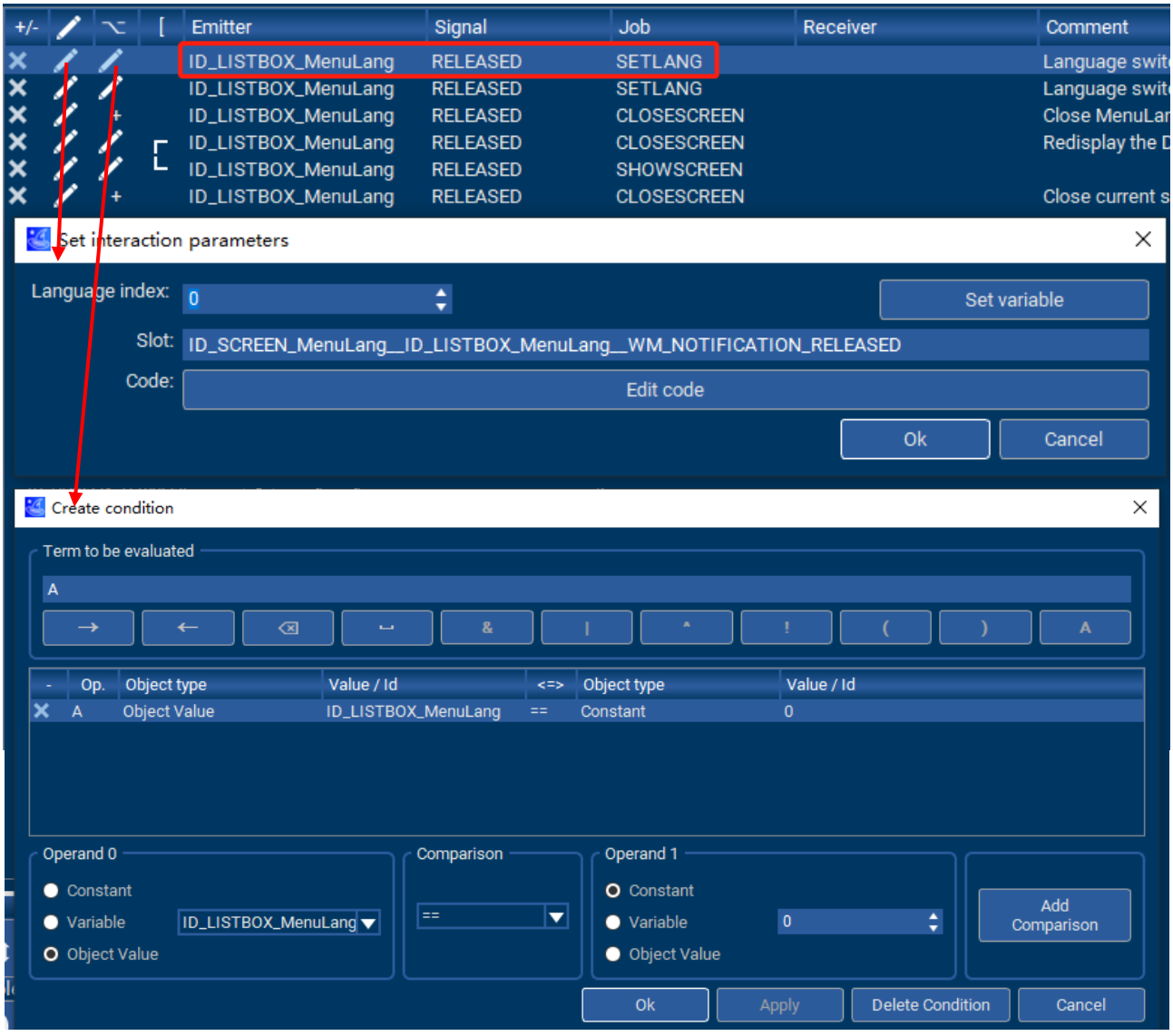


图 1-96 互动设置 - 语言切换下拉菜单

3. 时间-温度曲线画面的互动设置

波形绘制区域大小定义为 400x200，分成 100 个点，刷新时间设置为 60 s，也就是刷新 100 个点需要 100 min。如图 1-97 横坐标刻度分成 5 格，每一大格的时间间隔是 20 min。

图 1-97 横坐标 (时间)

从左边开始显示，每次刷新增加一个点；当达到 100 个点后，每次刷新将原波形向左移动一个点，最后一个点始终显示最新的数据。

当画面初始显示时，当前时间是第一个点的时间，显示满 100 个点后，当前时间对应最后一个点。

1) 定时器启动

将 1 s 定时器作为节拍定时器，通过对定时器的超时事件计数来控制波形刷新间隔。初始化时启动定时器，画面关闭时停止定时器。

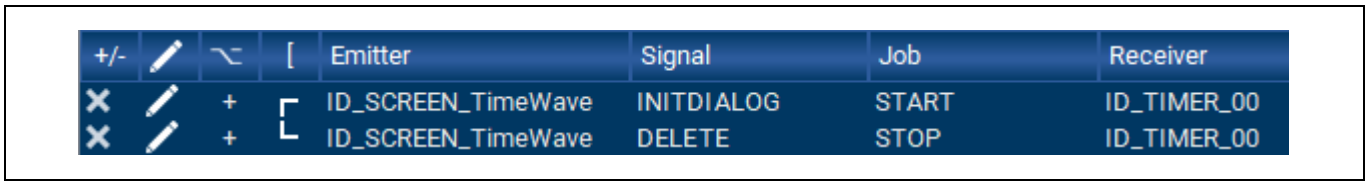


图 1-98 互动设置 - 定时器启停设置

2) 计算像素增量、时间增量

当设置 Job = CALC，触发事件发生时会调用计算公式计算 Receiver 的值。计算公式是变量定义时定义的如图 1-99 所示。

两点间的像素增量 = X 轴像素 ÷ 点数 = 400 ÷ 100 = 4 像素

时间增量 = 刷新间隔 × 点数 ÷ 刻度格数 = 60 s × 100 ÷ 5 = 20 min

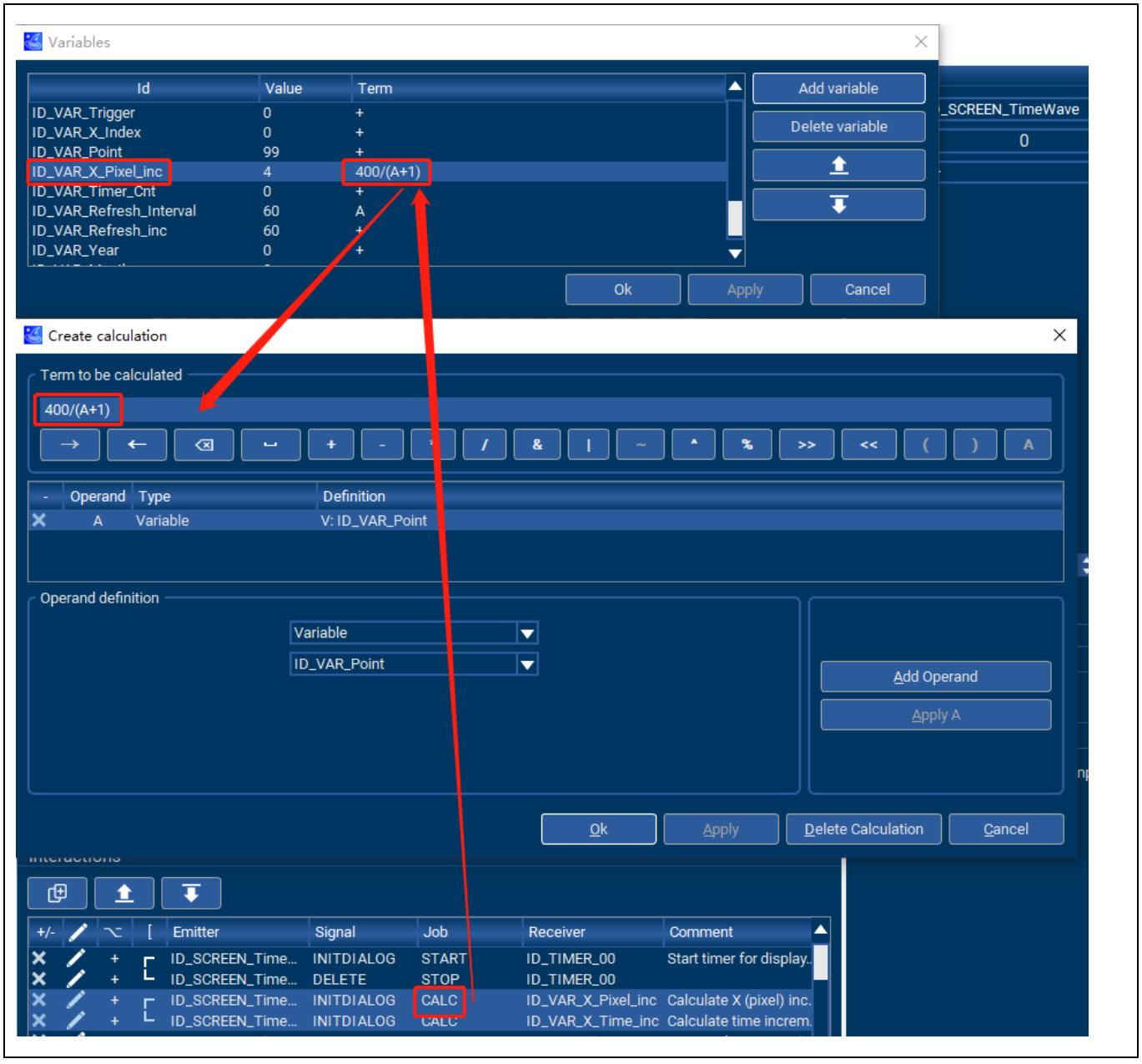


图 1-99 互动设置 - 计算像素和时间增量

3) 初始化横坐标显示

初始时只显示一个点，横坐标第一个刻度对应当前时间。将当前时间换算以秒，计算每个刻度对应的时间，然后再换算成小时和分钟，显示在对应刻度的位置。

+/-		⌥	[	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

图 1-100 互动设置 - 初始化横坐标显示

4) X 和 Y 的初始值

这里 X 和 Y 值都是指的像素。初始时将 100 个点的 X 坐标值都设置为 0，如图 1-101 所示；将 A5 区域 100 个点的 Y 坐标值都设置为 0，如图 1-102 所示，A1~A4 区域 Y 坐标值按照同样的方法设置。

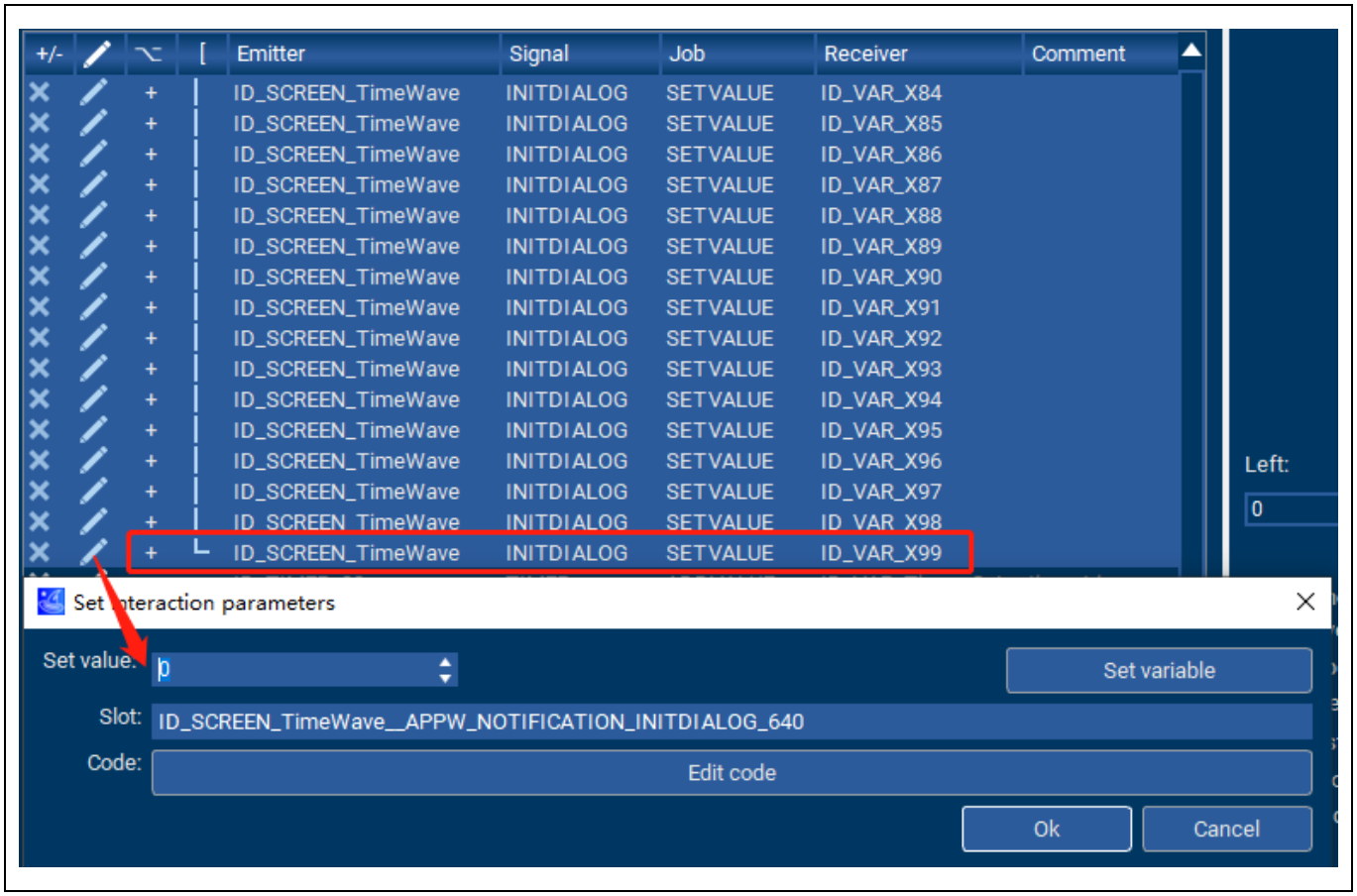


图 1-101 互动设置 – 设置 X 的初始值

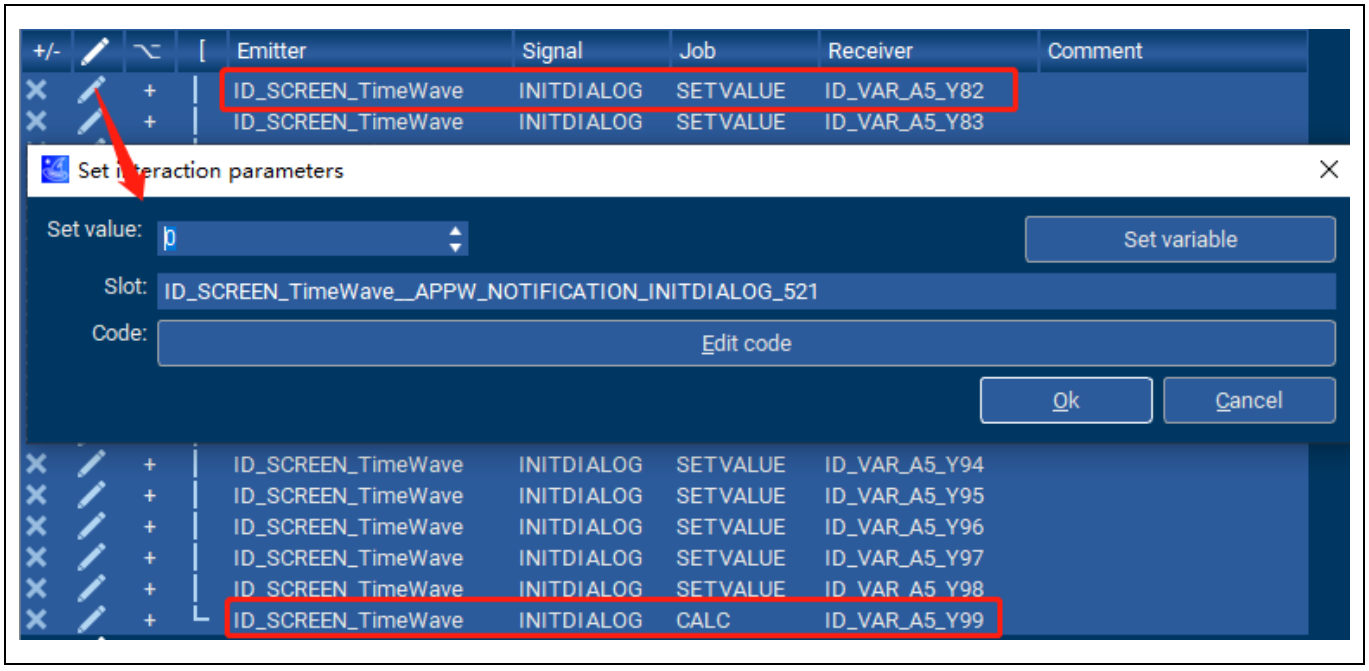


图 1-102 互动设置 – 设置 Y 的初始值

5) 产生刷新事件

初始化时设置了 1 s 的定时器作为节拍定时器。要实现 60s 刷新一次的目的，需要再对定时器超时事件计数，当计数次数达到 60 次后产生刷新事件。如图 1-103 当定时器事件发生时，对 ID_VAR_Timer_CNT 加 1。当 ID_VAR_Timer_CNT 等于 60 时，对 ID_VAR_Trigger 加 1，并清除 ID_VAR_Timer_CNT 计数值。再用 ID_VAR_Trigger 值的改变触发显示刷新。

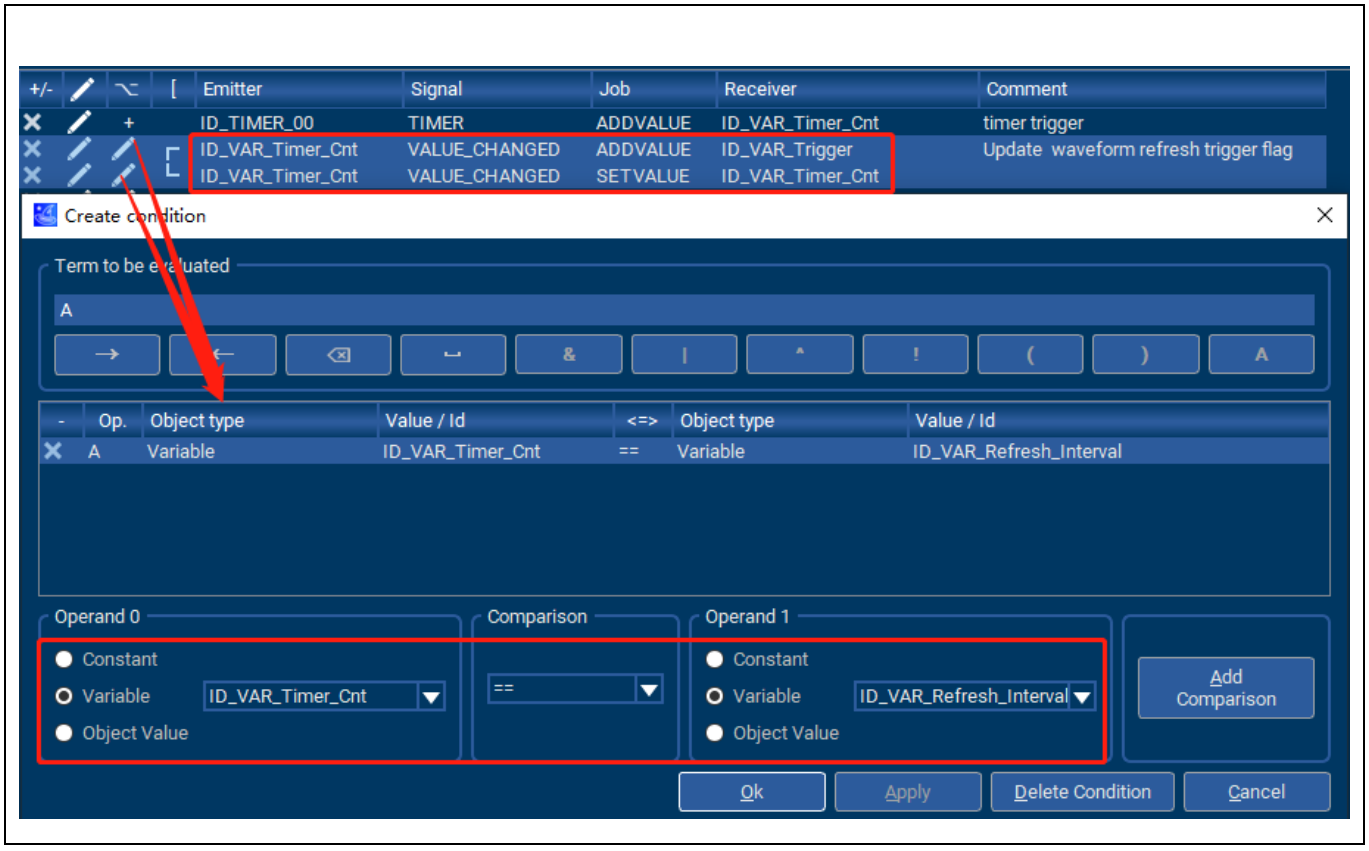


图 1-103 互动设置 - 产生波形刷新事件

6) 更新时间刻度显示

当前显示的点数达到 100 个点后，每次刷新事件发生时，需要更新时间刻度的显示。读取当前时间作为最后一个刻度的值显示。再减去时间增量 (20 min) 作为倒数第二个刻度的值，以此类推....

+/-	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

图 1-104 互动设置 - 更新时间刻度显示

7) 更新 X 和 Y 值

更新 X 值

当刷新事件发生时，如果显示的点数小于 100 个点，设置 $X_{n-1} = X_n$，$X_{99} = X_{99} +$ 增量。如果显示的点数达到 100 个点后，不再更新 X 值，如图 1-105。

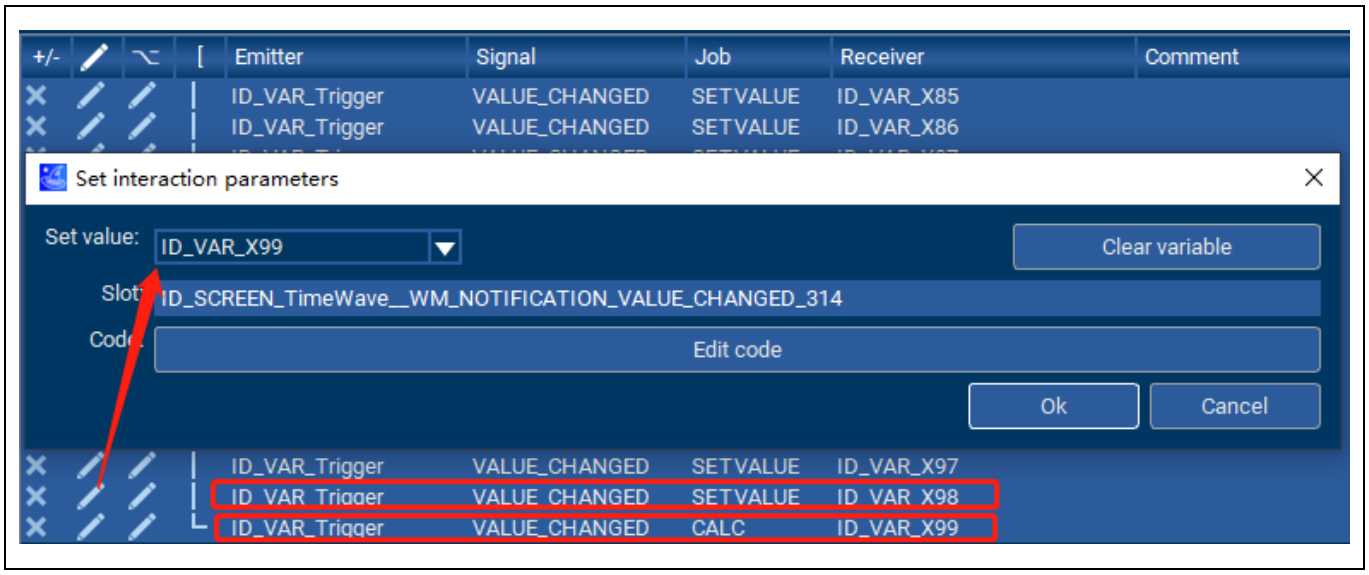


图 1-105 互动设置 - 更新 X 值

更新 Y 值

这里有 5 组 Y 值，分别对应区域 A1~A5。当刷新事件发生时，设置 $Y_{n-1} = Y_n$，然后将当前区域温度换算成像素后赋值给 Y_{99}, 如图 1-106。

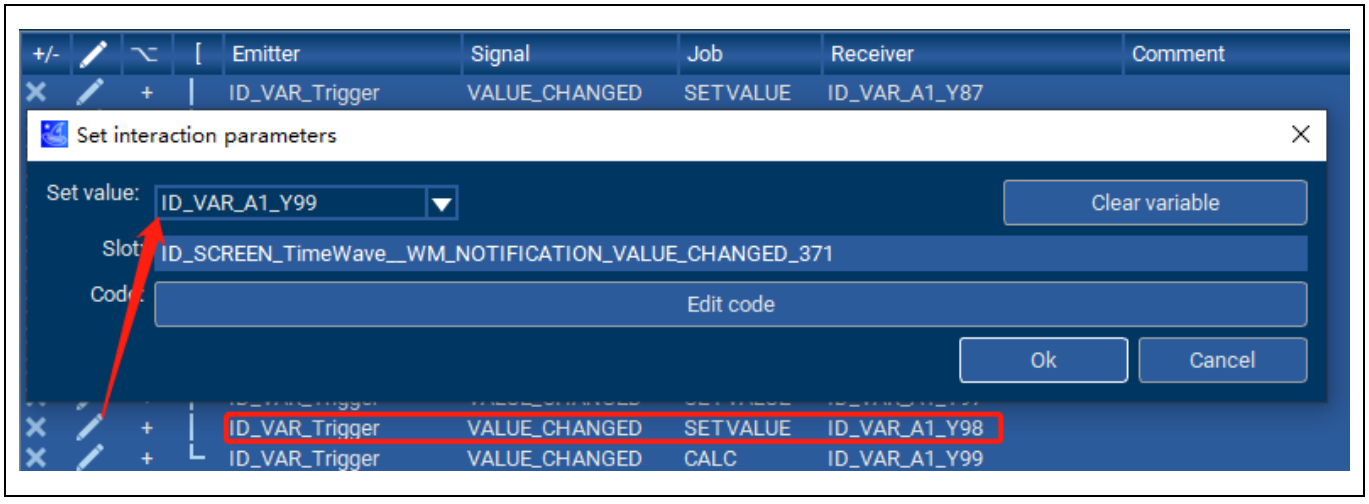


图 1-106 互动设置 - 更新 Y 值

8) 更新点数和重画曲线

当显示的点数小于 100 时，需要记录当前已显示的点数。当发生刷新事件时，需要重新绘制曲线，如图 1-107。

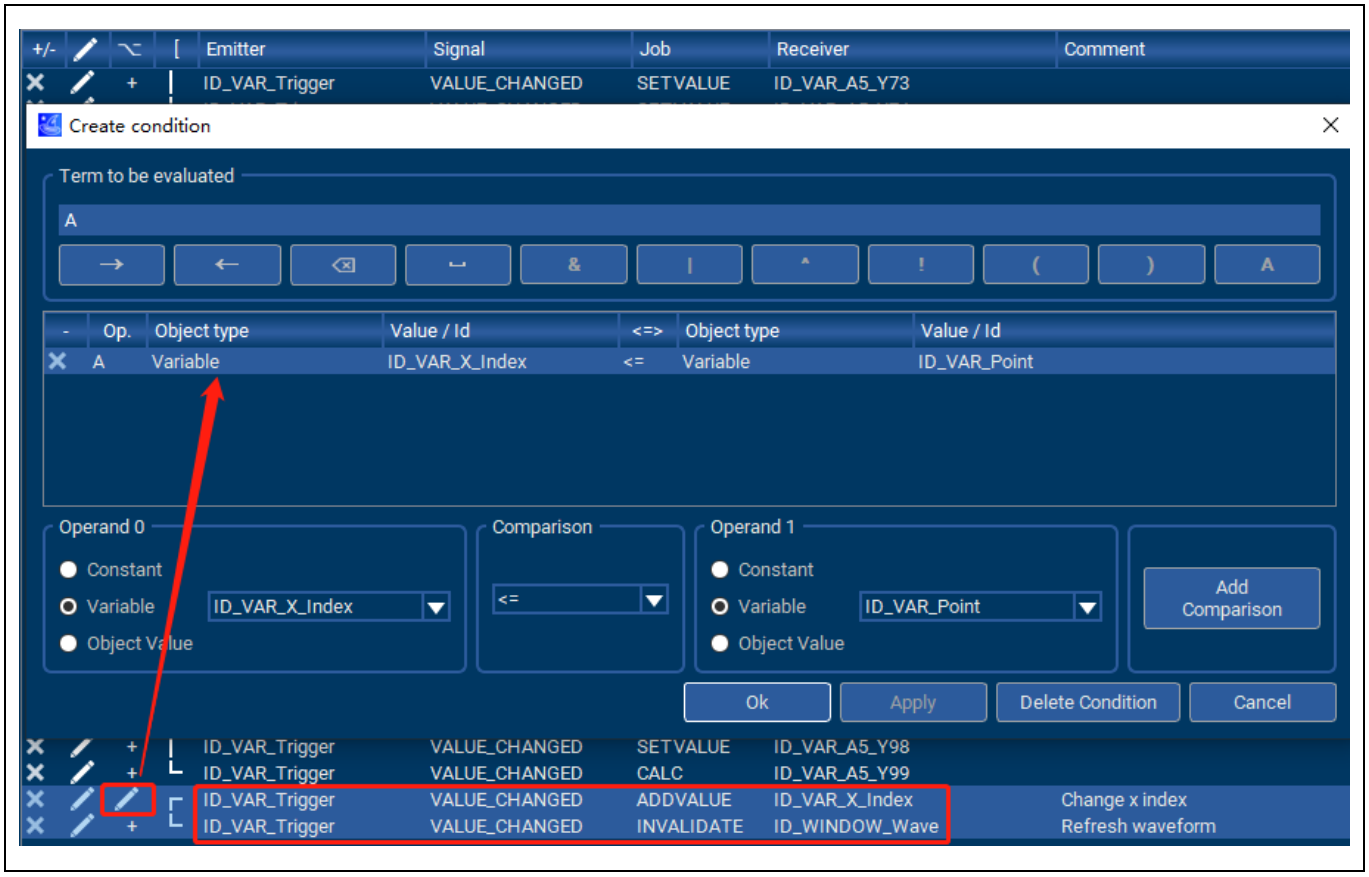


图 1-107 互动设置 – 更新点数、重画曲线

4. 区域-温度曲线互动设置

区域-温度曲线里 5 个 X 坐标值是固定的。当温度发生变化时更新 Y 值并重画曲线。

初始显示时，按照各区域的实际温度计算 Y 值。当实际温度值发生变化时触发计算 Y 值；当 Y 值发生变化时触发重画曲线。如图 1-108 所示。

+/-		⌥	[	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

图 1-108 互动设置 - 区域-温度曲线

5. 温度列表画面的互动设置

温度列表画面用于显示 5 个区域的设定温度和实际温度。将设定温度和实际温度变量的值显示在 Text 里。当实际温度的变量值发生变化时更新显示。如图 1-109 所示。

+/-		⌥	[	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

图 1-109 互动设置 - 温度列表画面

6. 温度设置画面的互动设置

温度设置画面除了和温度列表画面一样显示温度列表外，还添加了区域选择。当选中区域后，选中区域高亮显示，旋钮以及旋钮中心位置的文本框里都显示选中区域的设定温度。当旋钮滑动后，旋钮中心的文本框以及列表里对应区域的设定温度跟随旋钮变化。

1) 初始化显示

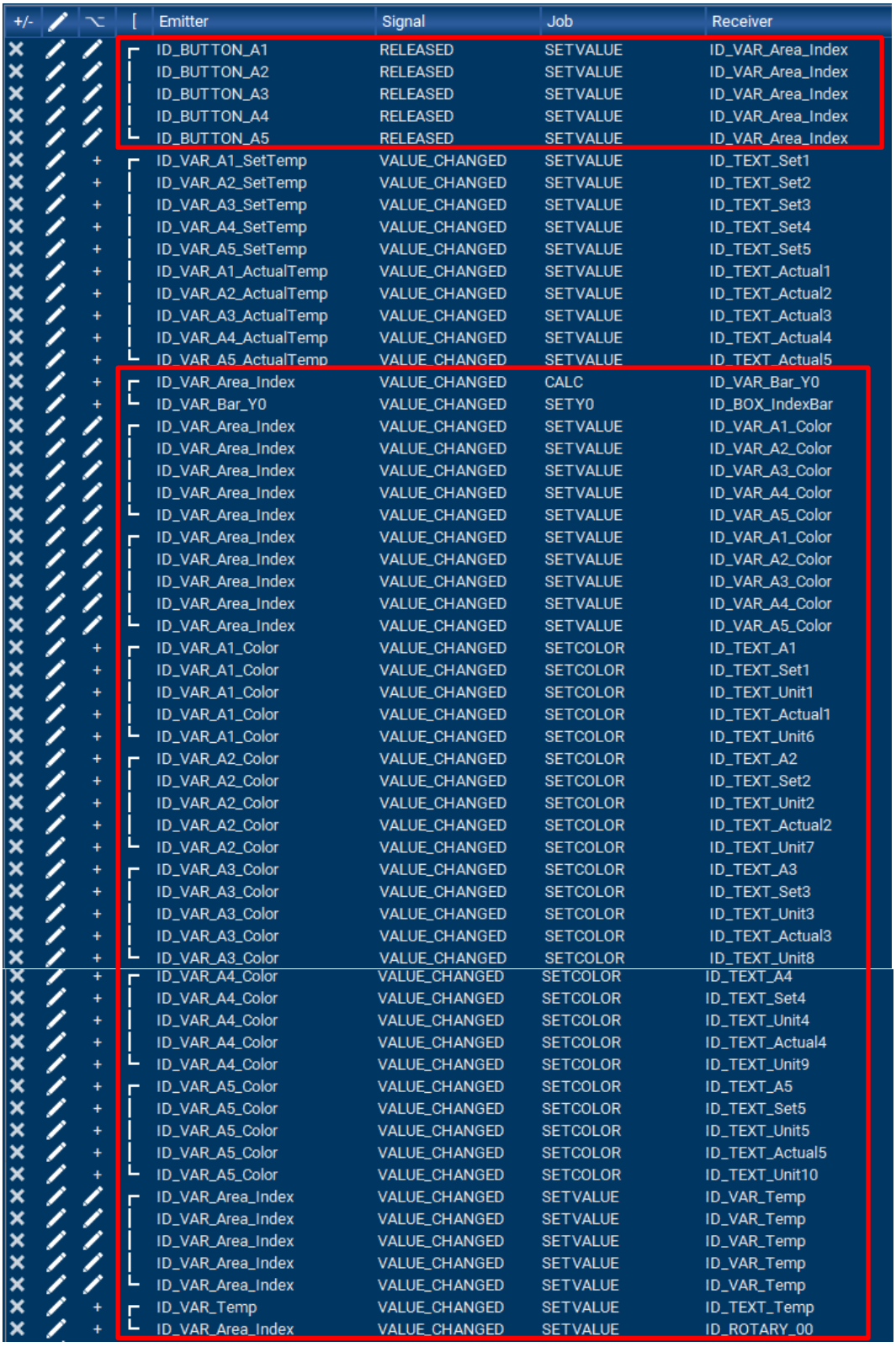
初始显示时设置默认选中区域为区域 1，并用 BOX 纯色背景和 Text 文字颜色来表示高亮。将 5 个区域的设定温度和实际温度赋值给列表里的 Text。在旋钮和旋钮里的 Text 里都显示区域 1 的设定温度。如图 1-110 所示。

+/-		⌥	[	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

图 1-110 互动设置 - 温度设置画面初始化

2) 区域选择按钮互动设置

当某个区域按钮被按下表示区域被选中。需要设置选中区域 index 变量，设置高亮 Box 的坐标，将选中区域的文本颜色设置为黄色，其他区域的设置为白色。再将选中区域的设定温度赋值给旋钮以及旋钮中心的 Text。如图 1-111 所示。

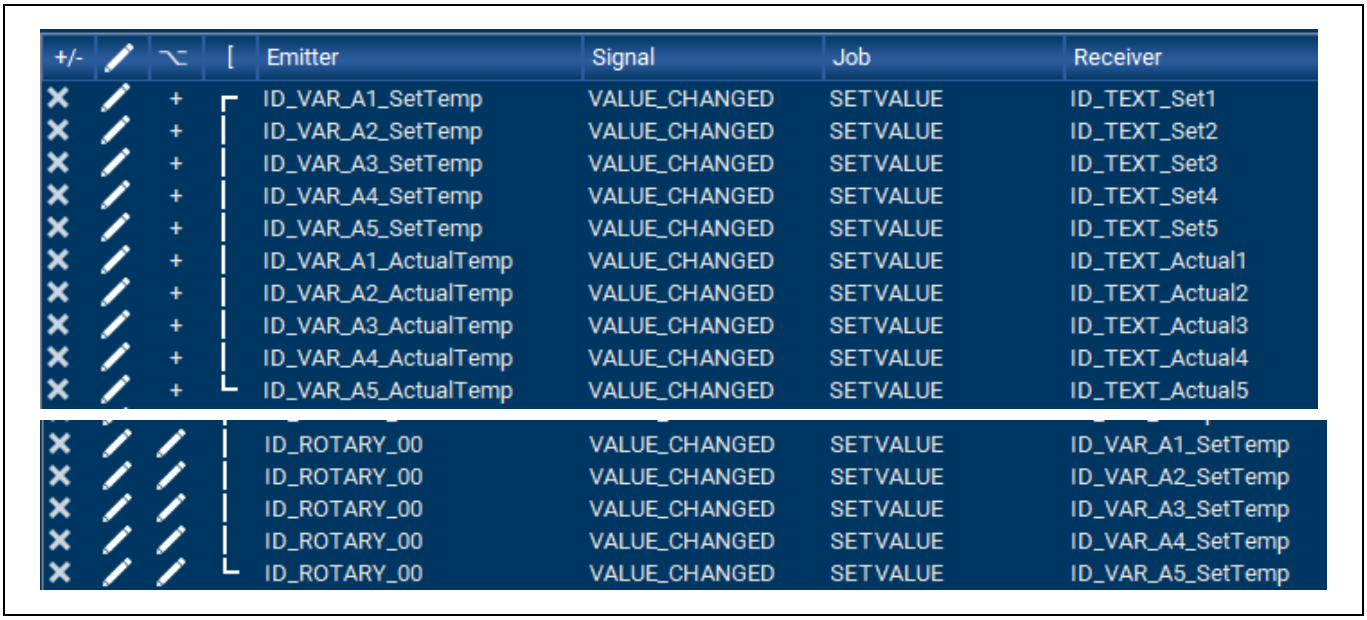


Emitter	Signal	Job	Receiver
ID_BUTTON_A1	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A2	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A3	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A4	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_BUTTON_A5	RELEASED	SETVALUE	ID_VAR_Area_Index
ID_VAR_A1_SetTemp	VALUE_CHANGED	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

图 1-111 互动设置 - 区域按钮

3) 旋钮互动设置

旋动旋钮时，列表里的设定温度、旋钮中心的 Text 显示内容随旋钮变化。如图 1-112 所示。



+/-	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

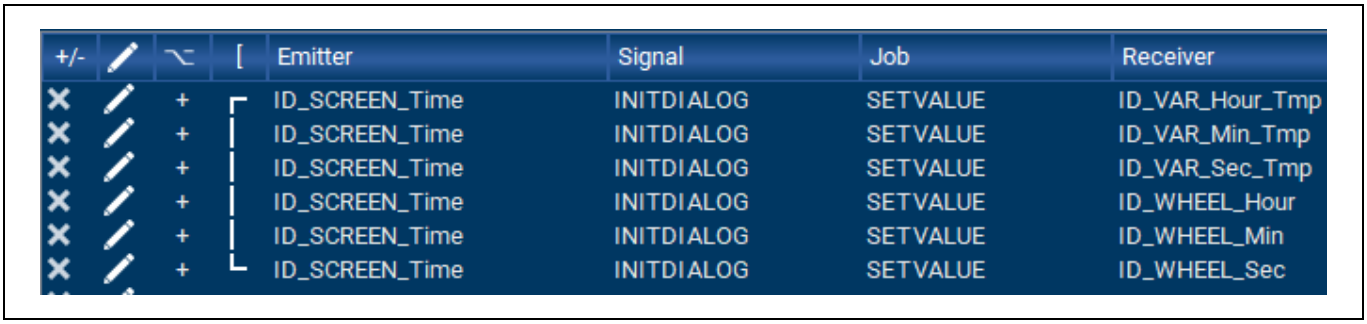
图 1-112 互动设置 - 旋钮

7. 时间设置画面的互动设置

时间设置画面需要实现显示和修改当前时间、保存设置时间、退出当前画面的功能。

1) 初始显示

当显示画面时需要将当前时间显示到 Wheel 上。在画面初始化时，将实际时间赋值给 Wheel。如图 1-113 所示。

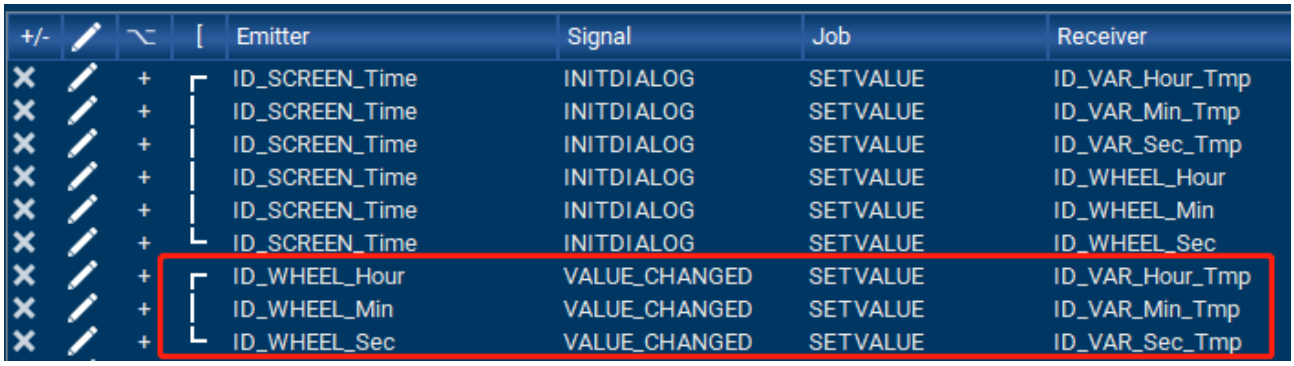


+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Hour_Tmp
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Min_Tmp
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Sec_Tmp
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Hour
+	ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Min
+	ID_SCREEN_Time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

图 1-113 互动设置 - Wheel 初始显示值

2) Wheel 里设置时间

当在 Wheel 里选择了时、分、秒后，将选择的时间保存到临时变量里。如图 1-114 所示。

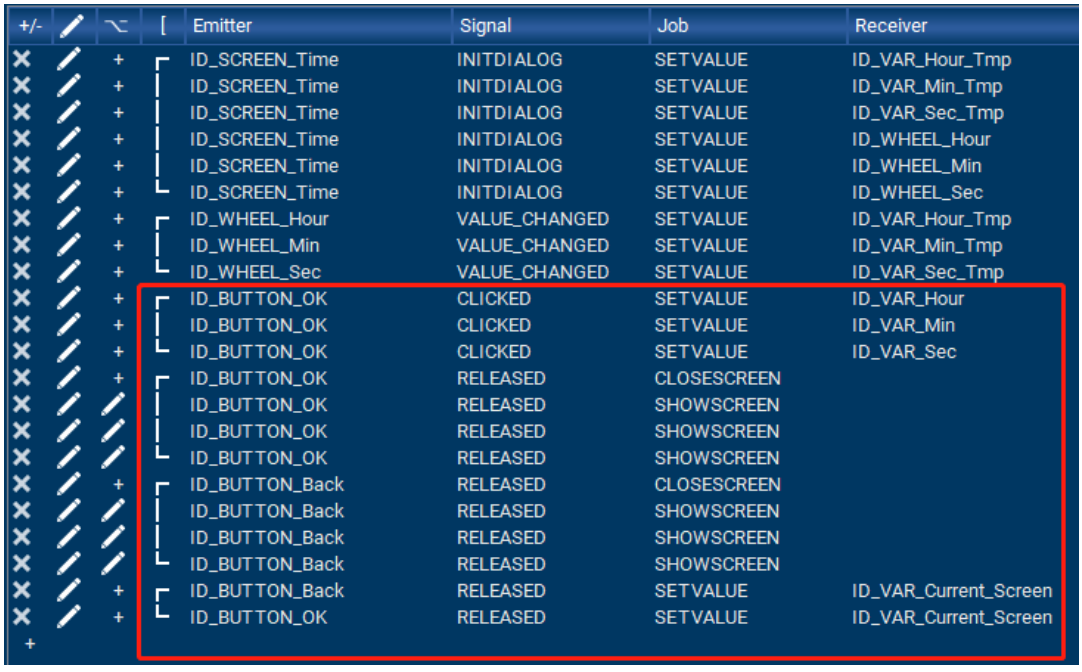


Emitter	Signal	Job	Receiver
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Hour_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Min_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_VAR_Sec_Tmp
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Hour
ID_SCREEN_Time	INITDIALOG	SETVALUE	ID_WHEEL_Min
ID_SCREEN_Time	INITDIALOG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

图 1-114 互动设置 – 设定时间保存到临时变量里

3) 确定和返回按钮

当按下确定按钮会保存设定时间，并退出当前画面，根据 index 显示温度列表、时间-温度曲线或者区域-温度曲线画面。当按下返回按钮时不保存设定时间，其他处理和确定按钮相同。如图 1-115 所示。



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

图 1-115 互动设置 - 确定和返回按钮

8. 日期设置画面互动

日期设置画面需要实现显示和修改当前日期、保存设置日期、退出当前显示画面的功能。实现方式和时间设置画面相同。如图 1-116 所示。

+/-		⌥	[	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
+							

图 1-116 互动设置 - 日期设置画面

1.2.2.17 MCU 和 AppWizard 间交换数据

1. 为了将 MCU 里采样的温度和当前时间传递给 AppWizard，需要使用 Variable 功能。并在程序代码里给 Variable 赋值，再使用 Interaction 功能将 Variable 的值设置给 Object，如 Text 和 Wheel，流程如图 1-117 所示。

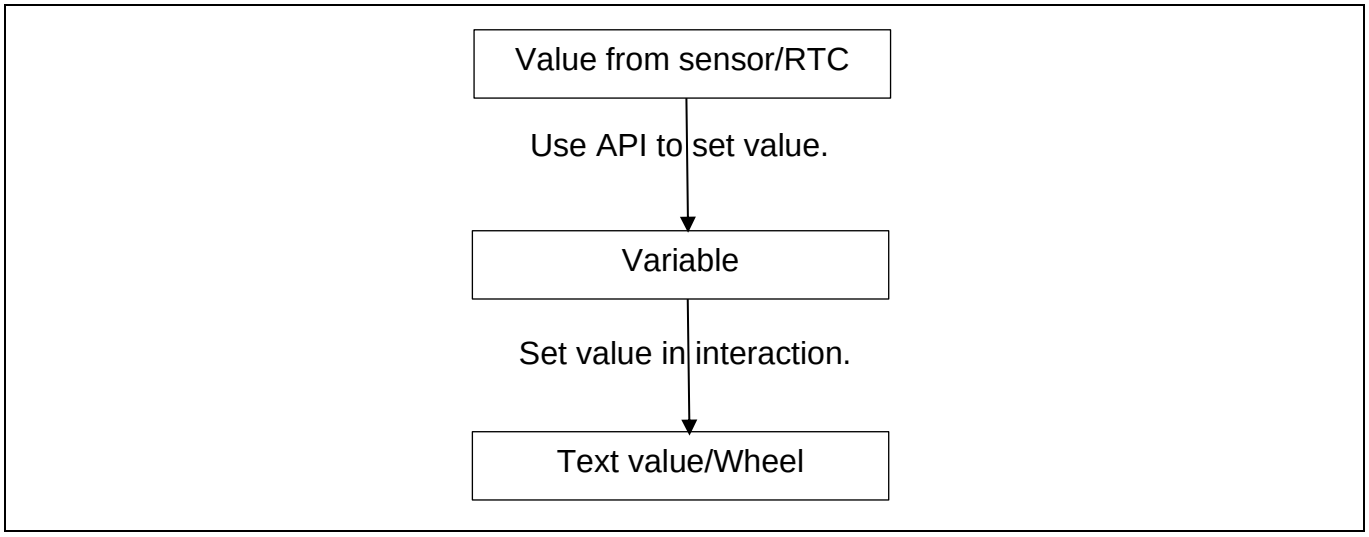


图 1-117 将温度和时间从 MCU 传递到 AppWizard 上显示的流程

2. 为了将 AppWizard 设置的日期和时间传递给 MCU，需要将 Wheel 的值赋给 Variable 变量。再在代码里读出 Variable 的值，去设置 RTC。流程如图 1-118 所示，变量的定义参考 1.2.2.6 章节，Interaction 的设置参考图 1-115 和图 1-116，程序代码参考 2 章节。

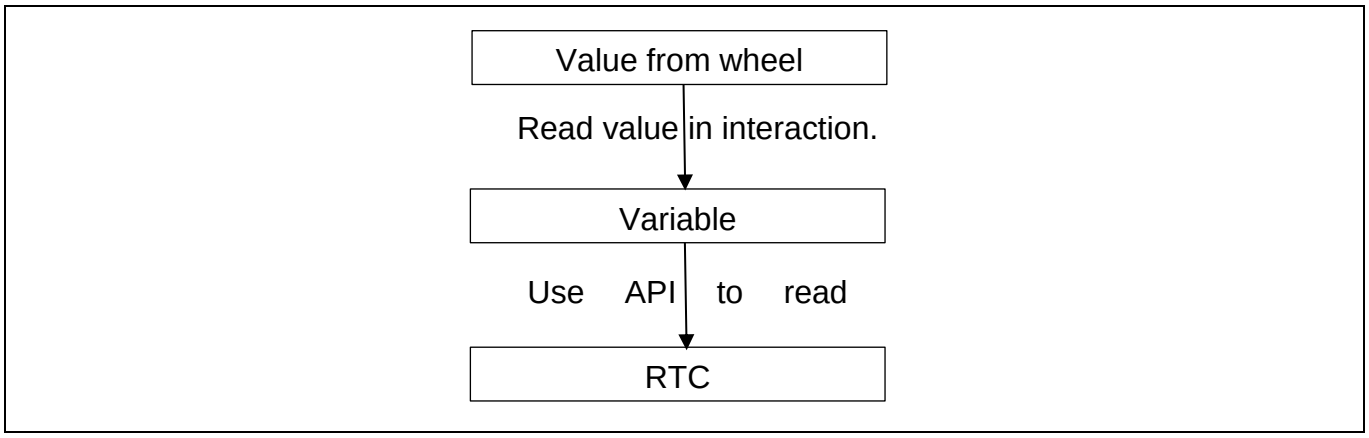


图 1-118 将 AppWizard 里设置的时间设置到 MCU 的 RTC 里的流程

3. 变量在 1.2.2.6 章节已经定义过了，需要的变量如图 1-119 所示。

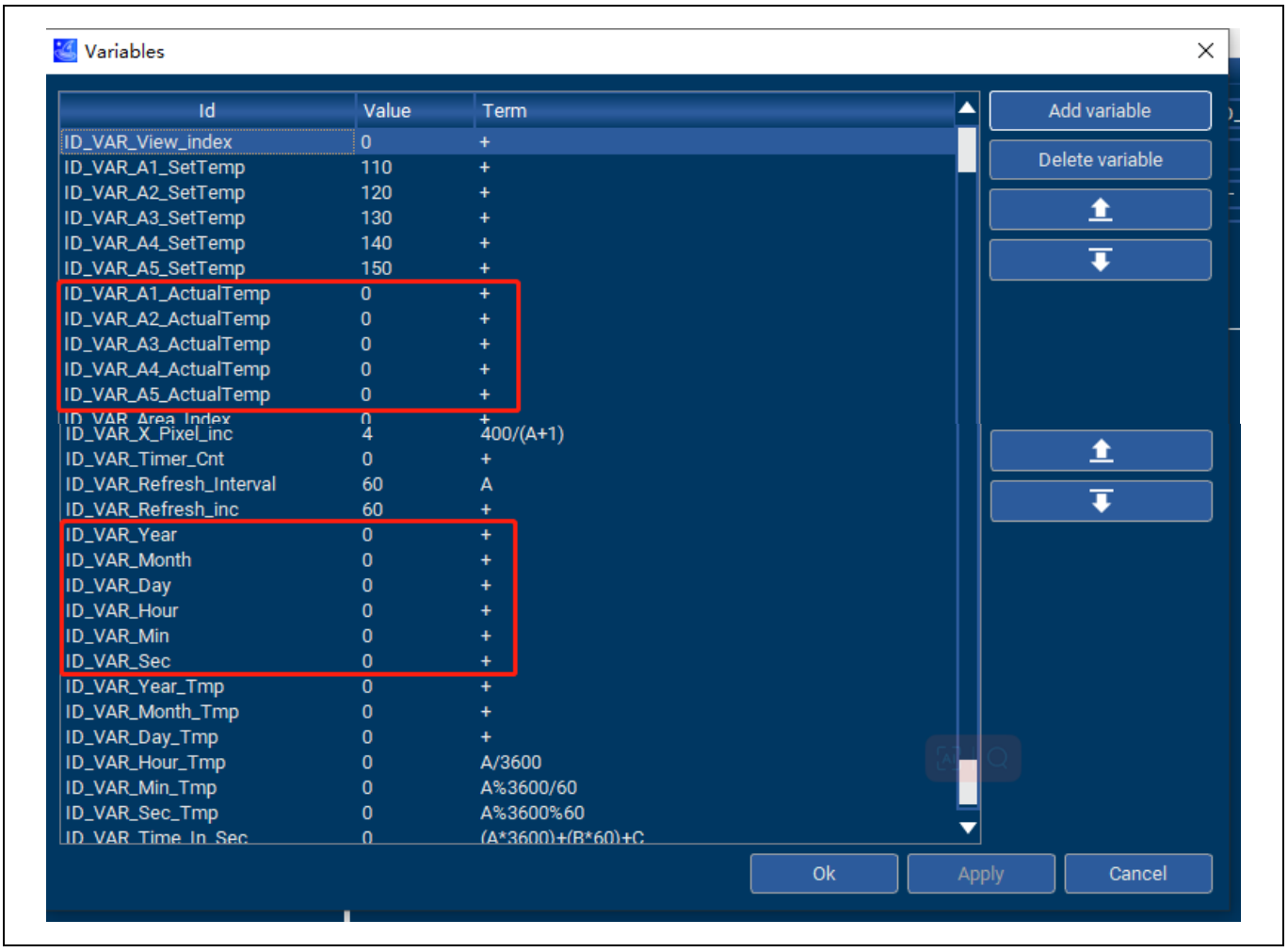


图 1-119 MCU 和 AppWizard 参数交换的 Variable 定义

4. Interaction 的设置如图 1-120。使用的地方比较多无法一一列举，请参考 1.2.2.16 章节里关于温度列表画面、温度设置画面、时间-温度曲线画面、区域-温度曲线画面、时间设置画面、日期设置画面互动的设置。

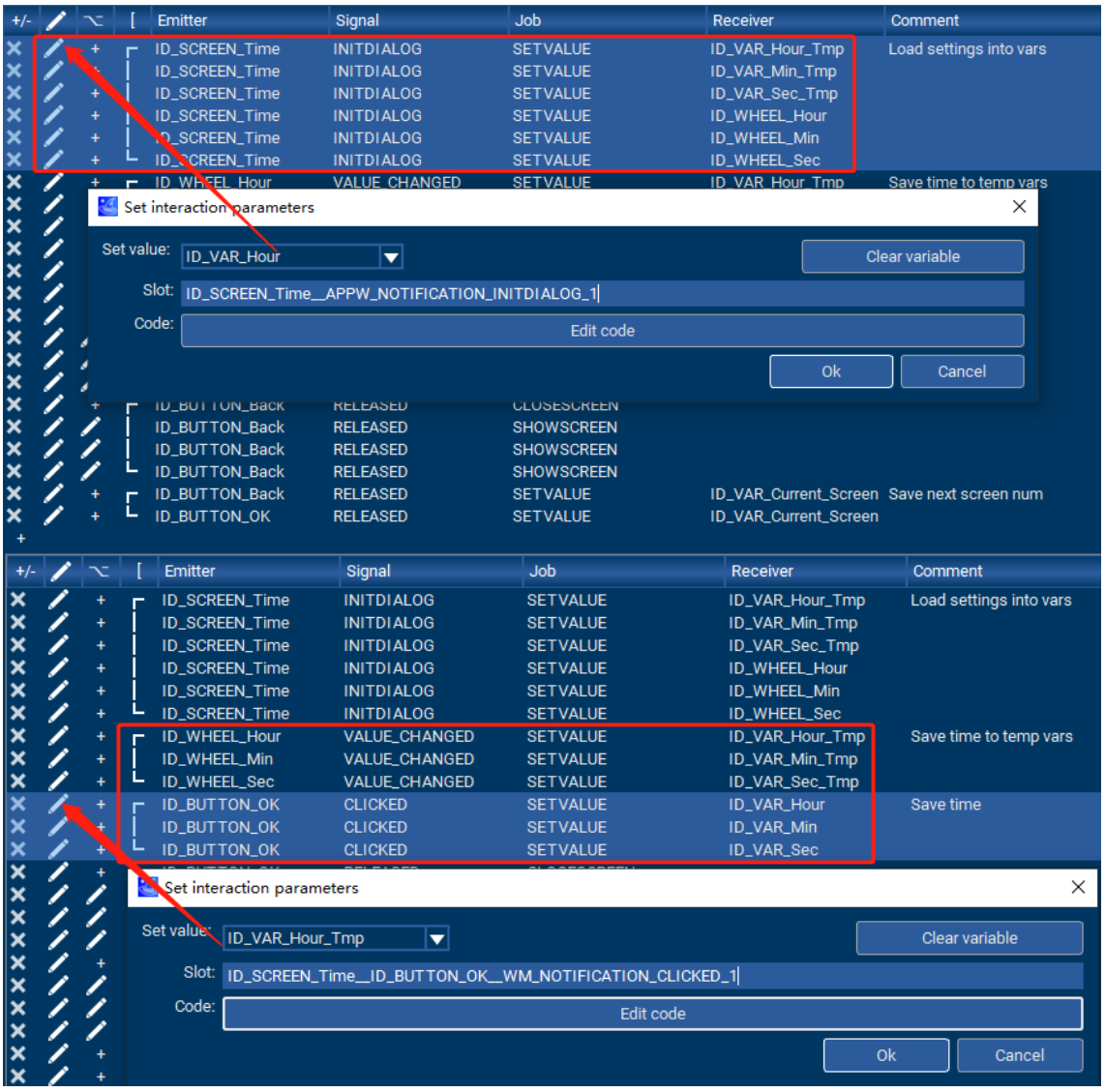


图 1-120 Interaction 设定 - Variable

5. 程序代码参考代码介绍。

1.2.3 程序代码编辑

1. 导出程序代码

设计完成后及可用 File -> Export&Save 输出程序代码，如图 1-121。

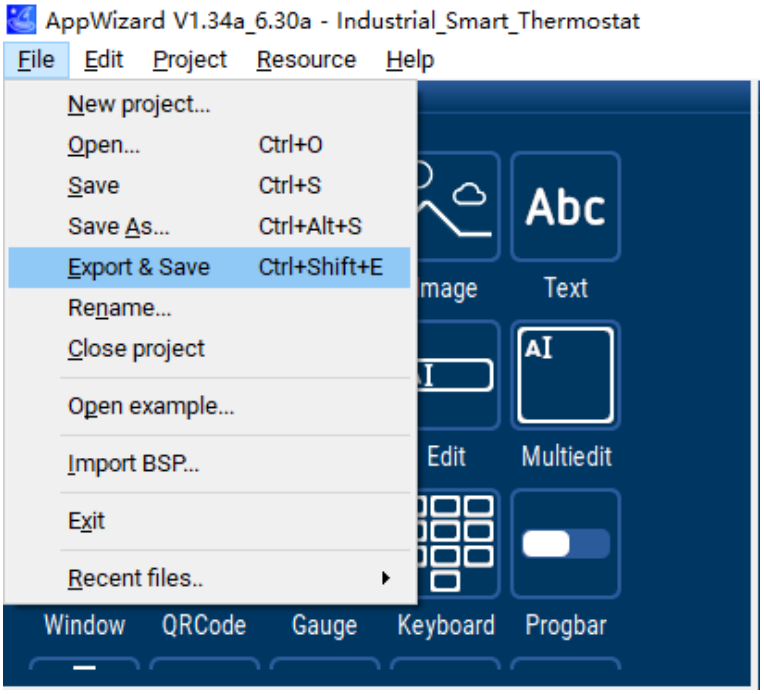


图 1-121 输出程序代码

2. 拷贝导出的代码

点击 Project -> Open containing folder 打开 AppWizard 工程文件夹，复制 Resource 与 Source 文件夹，到路径 EC_M467_AppWizard_ISMThermostat_V1.00\SampleCode\M467_AppWizard_ISMT\Application 中覆盖原本内容，步骤如图 1-122 和图 1-123 所示。

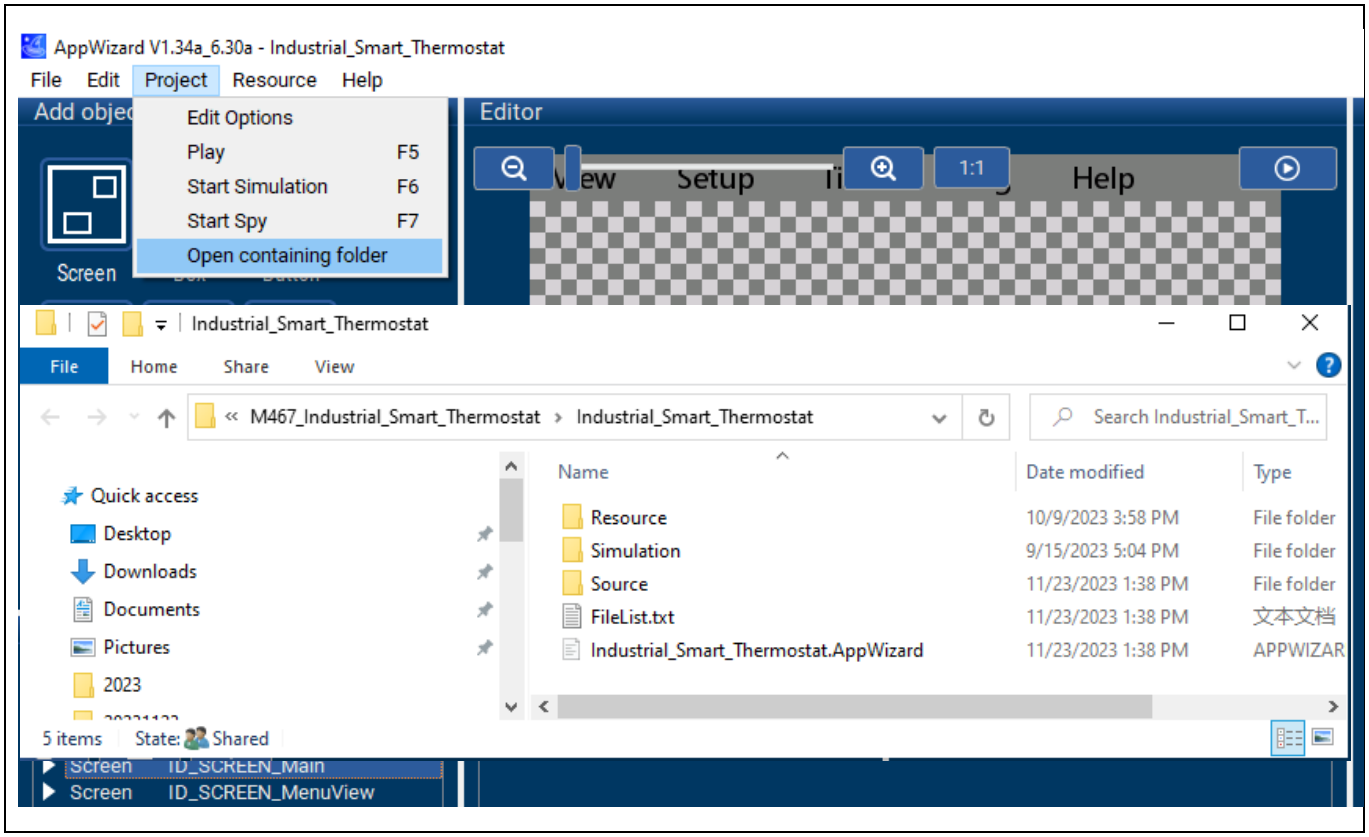


图 1-122 打开 AppWizard 工程文件夹

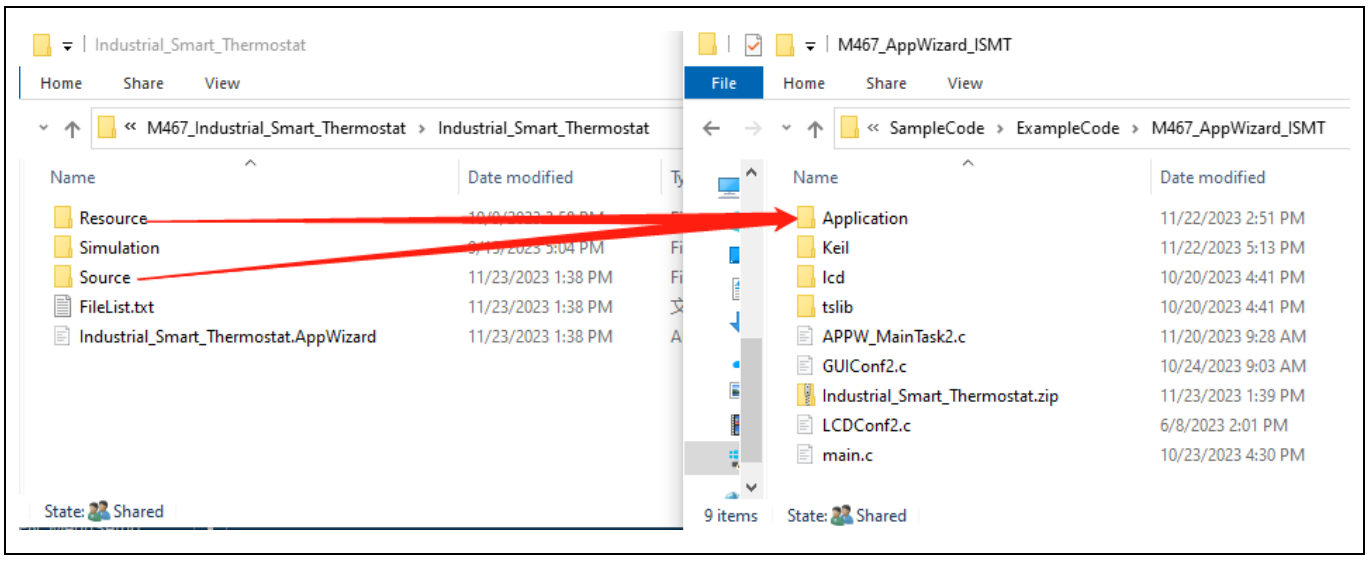


图 1-123 拷贝到程序代码工程中

3. 将 AppWizard 生成代码导入 Keil 工程

打开 Keil 工程，在 Application 上点击右键，选择 Add Existing Files to Group Application，找到上一步里的 Application 文件夹，将 Resource 与 Source 文件夹下所有*.c 文件都加载到工程中。

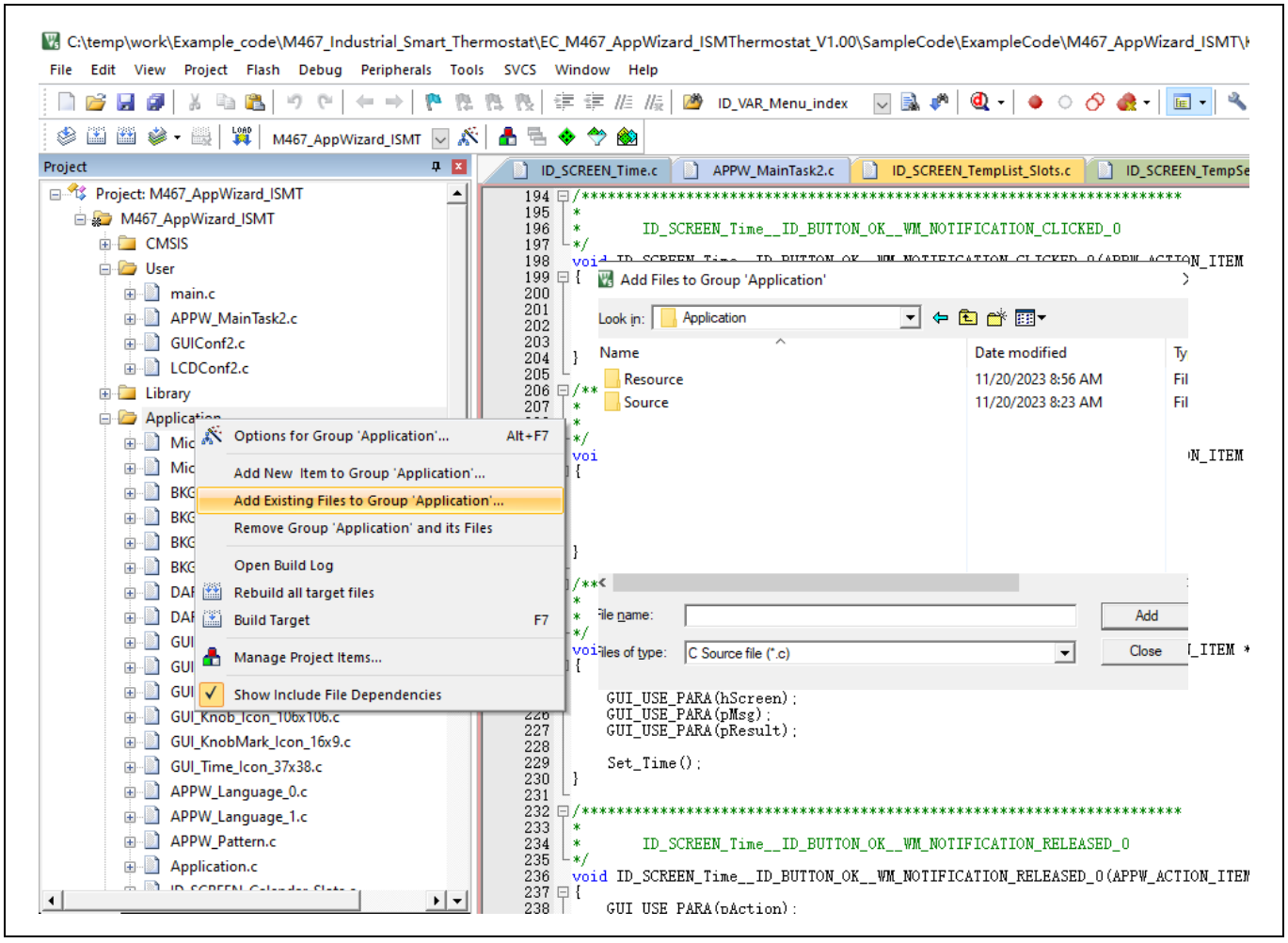


图 1-124 将生成代码导入 Keil 工程

4. 添加与 AppWizard 交互的代码

在工业智能温控器的应用中，GUI 上显示的时间来源于 MCU 的 RTC，温度由 MCU 的 ADC 采样 NTC 得到，再传递给 AppWizard 显示。GUI 上设置的时间也需要写入 RTC 才能生效。因此程序里需要添加 ADC 采样和 RTC 读写相关代码、MCU 向 AppWizard 传递温度和当前时间、AppWizard 向 MCU 传递设定时间和日期。相关函数在 Application.c 里定义，在 Applicatio.h 里声明。AppWizard 和 MCU 交互的代码涉及到 MainTask2.c、ID_SCREEN_Date_Slots.c 和 ID_SCREEN_Time_Slots.c 文件。当 AppWizard 重新生成源代码后，将 Resource 与 Source

拷贝到 Keil 工程里时注意不要覆盖 Application.c、 Applicatio.h、ID_SCREEN_Date_Slots.c 和 ID_SCREEN_Time_Slots.c 四个文件里的用户代码。

```
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();
        }
    }
}
```

图 1-125 MCU 将温度和时间传递给 AppWizard

```
void ID_SCREEN_Time__ID_BUTTON_OK__WM_NOTIFICATION_RELEASED_2(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();
}
```

图 1-126 OK 键释放时将 AppWizard 设定时间写到 MCU 的 RTC 里

1.2.4 在开发板上烧录和运行程序

在 Keil 里编译并将程序下载到 NuMaker-HMI-M467 开发板上运行。

1.3 执行结果

本范例代码执行编译和烧录的结果如图 1-127 和图 1-128 所示。

图 1-127 执行结果 - 英文

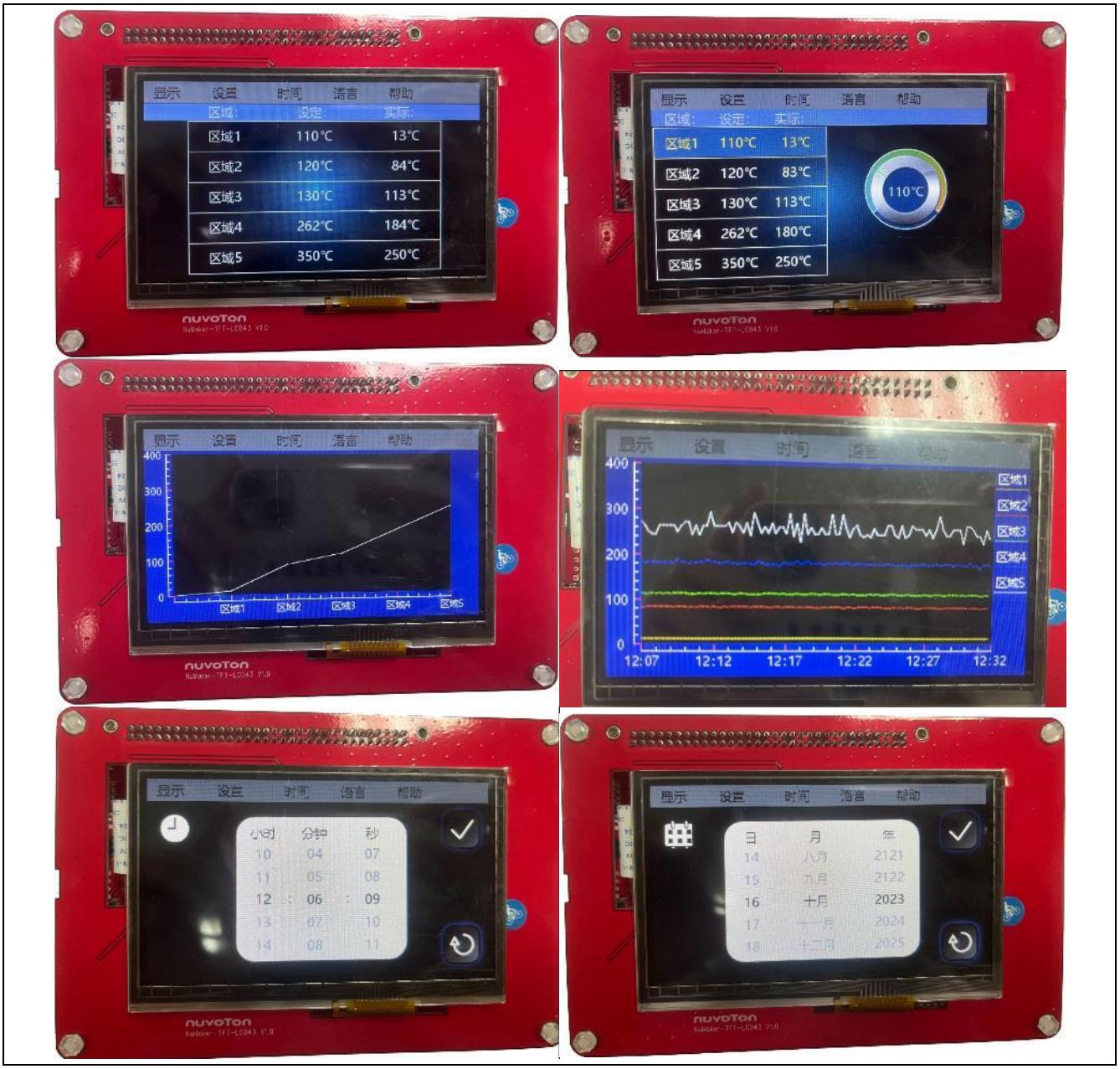


图 1-128 执行结果 - 中文

2. 代码介绍

2.1 main 函数

主函数由 *main.c* 进入，包含 3 个部分如下：

- 系统时钟和引脚初始化：sys_init ()
- 初始化：UART、SPIM、Timer、Touch、ADC、RTC 等
- 应用主程序：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;
}
```

其中，使用 SPIM 接口，是为了将占较大资源的图片、文本文件保存到外部 SPI Flash，利用 SPI Synchronous Serial Interface Controller (SPI Master Mode)，在 Direct Memory Mapping Mode(DMM)的模式下，SPI Flash 可被视为 ROM Module，直接到指定地址读取 SPI Flash，并 build-in cache 帮忙下，以及使用四线模式(quad read)，提升整体读取效率。

初始化 SPIM 的程序代码如下，需包含 SPIM 速度、SPI flash detection、开启 SPIM cache mode、开启 SPIM DMM mode、开启 fast quad read mode。相关参数设定可参考 W25Q32JVTBIQ 说明。

```
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);
}
```

除了 SPIM 初始化，还须编辑 Scatter file，让编译器知道 SPI Flash 的位置与内容。在 KEIL 环境中，依照路径 Option-> Linker->Scatter File-> Edit，开启.sct 档案并编辑内容如下：

```
;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)
    }
}
```

程序进入应用主程序 MainTask2()后，将执行 AppWizard 相关函数，在 while 循环中会执行画图的动作，此范例程序同时新增 Read_Time()函数和 Read_Temperature()函数，这两个函数定义在 Application.c 文件里。在函数里通过调用 APPW_SetVarData()函数将值从 MCU 传出给 AppWizard 中的 Variable，再通过如步骤 1.2.2.17 所述之方法，数值便会呈现在画面上。

```
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 添加用户处理

AppWizard 里执行的画面内容会在 ID_Screen_xxx_Slots.c 中读取。在 Interaction 里设置的每个 action，在 ID_Screen_xxx_Slots.c 文件里都有对应的处理函数，函数命名规则如图 2-1。每个画面都有一个 cbID_SCREEN_xxx 回调函数。用户可以把相关动作的代码添加到这两个函数中的任意一个里。

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

图 2-1 Slot 命名规则

下面分别用两种方法添加用户处理：

1. Interaction 的 action 处理函数里添加用户代码

在时间设置画面里 OK 按钮释放会触发关闭时间设置画面，在这个 action 的函数里添加将设置时间写入 RTC 的代码。

```
/*********************************************************************
*
*       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. 在画面的回调函数里添加用户处理

日期设置画面的回调函数里，想要实现 OK 按键释放时将设置的日期写入 MCU 的 RTC 中。需要添加判断 OK 键释放，和将设定日期写入 RTC 的代码。

```
/*********************************************************************
*
*       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 文件

AppWizard 生成的 Application.c 是一个空的文件，主要用于添加用户代码。注意在 AppWizard 里修改了 GUI，重新生成代码时，不要将用户代码覆盖了。这里主要包含了 RTC 的读写和温度的采样，以及 MCU 和 AppWizard 的交互。

```
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. 软件与硬件需求

3.1 软件需求

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

3.2 硬件需求

- 电路组件
  - NuMaker-HMI-M467HJ V1.0

4. 目录信息

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	

图 4-1 目录信息

5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *M467_AppWizard_ISMT.uvprojx*。
2. 进入编译模式界面
   - 编译
   - 下载代码至内存
   - 进入 / 离开仿真模式
3. 进入仿真模式界面
   - 执行代码

6. 修订纪录

Date	Revision	Description
2023.11.15	1.00	初始发布。

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.