以 AppWizard 設計充電樁 GUI 專案

NuMicro® 32 位元系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例代碼使用 emWin AppWizard 工具，結合 M467 系列開發平臺與開發資源，完成充電樁圖形化介面設計。
BSP 版本	M460_Series_BSP_CMSIS_V3.00.002 emWin Software Package for M460 V3.0
開發平臺	NuMaker-M467HJ 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

本範例代碼將示範如何使用 SEGGER emWin AppWizard 工具，搭配 NuMaker-M467HJ V1.0 開發平臺，逐步完成一個充電樁專案的畫面設計、基本功能演示。

1.1 原理

1.1.1系統方塊圖

此範例代碼直接使用新唐 GUI 開發平臺 - NuMaker-M467HJ V1.0 開發板，開發時會使用到的元件包括解析度 480x272 且螢幕大小 4.3 吋的 TFT-LCD 電容式觸控面板、主控板上 NuMicro M467HJHAN 微控制器、以及容量大小 4Mbyte 的 SPI Flash。其系統方塊圖如圖 1-2 所示。

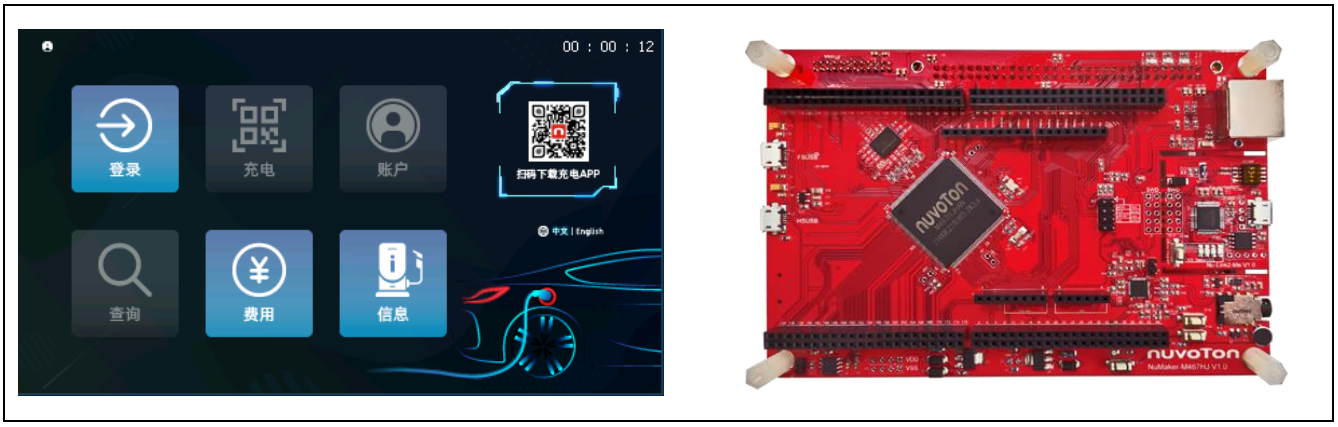


圖 1-1 NuMaker-M467HJ V1.0 開發板

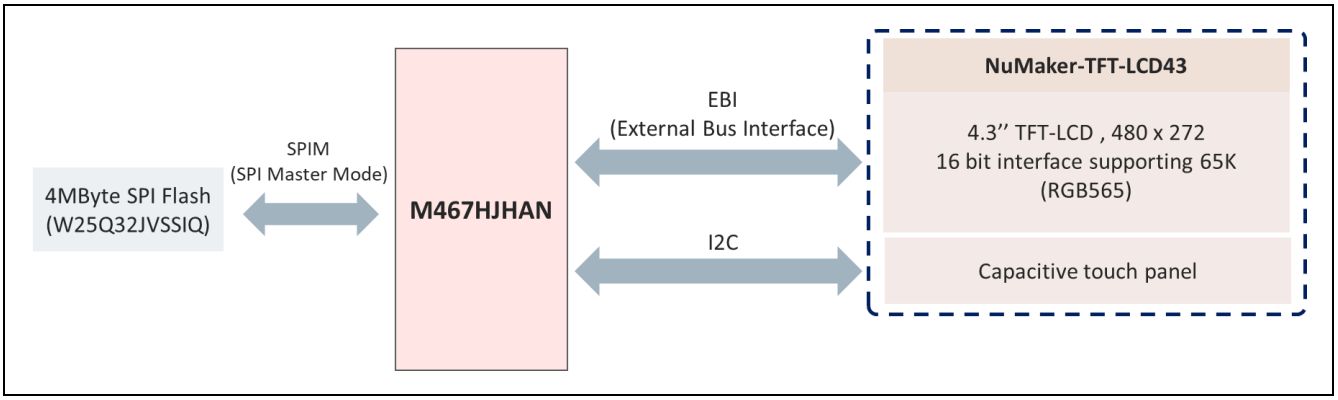
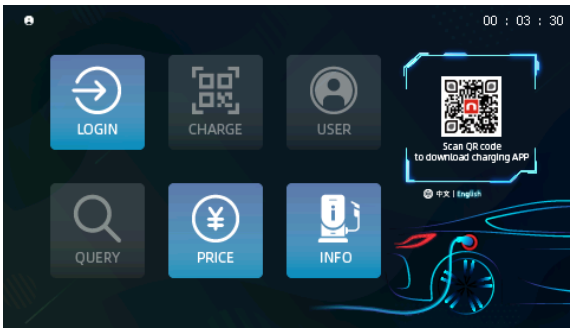
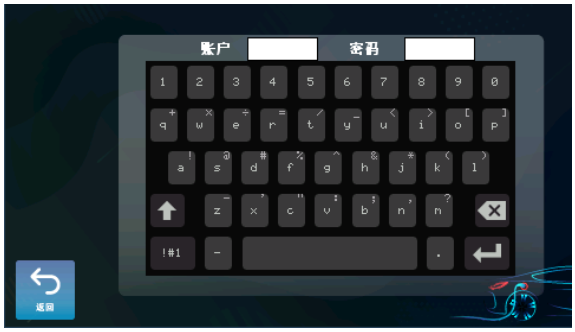
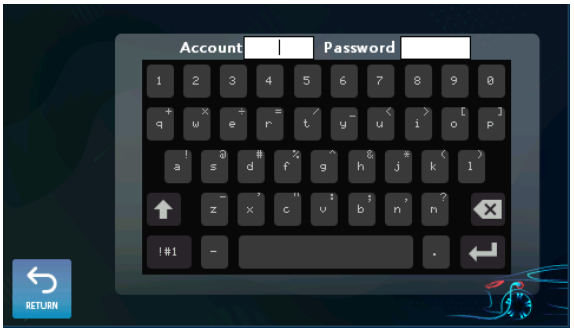
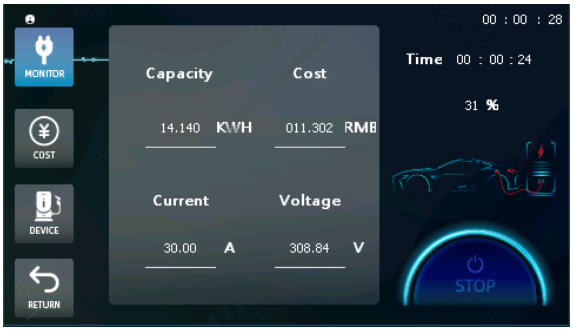
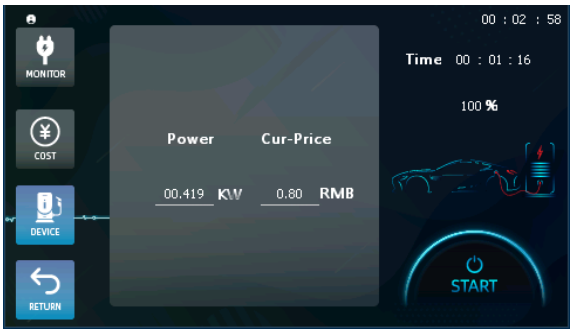
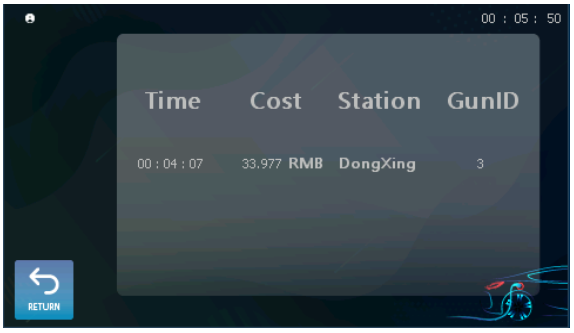


圖 1-2 系統方塊圖

1.1.2 充電樁畫面

此範例設計畫面如下：

	中文畫面	英文畫面
螢幕一		
螢幕二		
螢幕三		

螢幕三		
螢幕三		
螢幕四		
螢幕五		

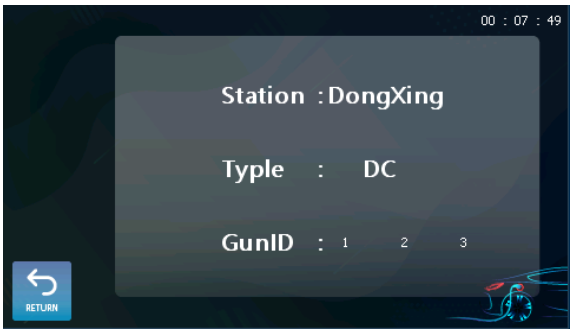
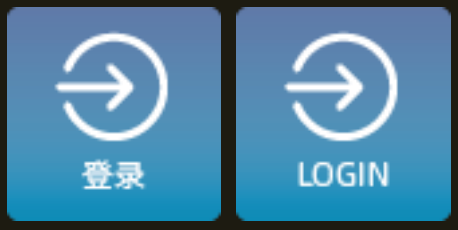
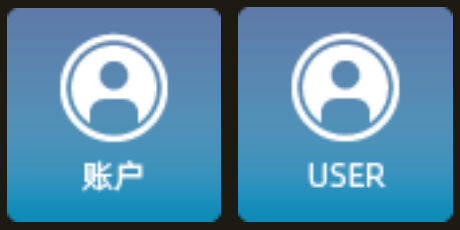
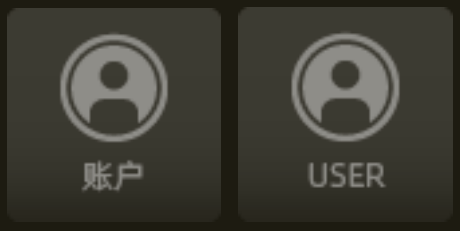
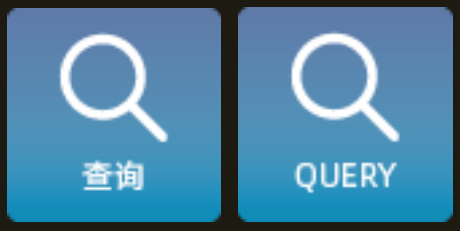
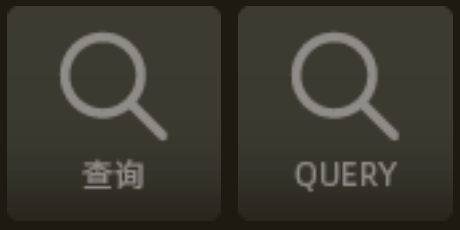
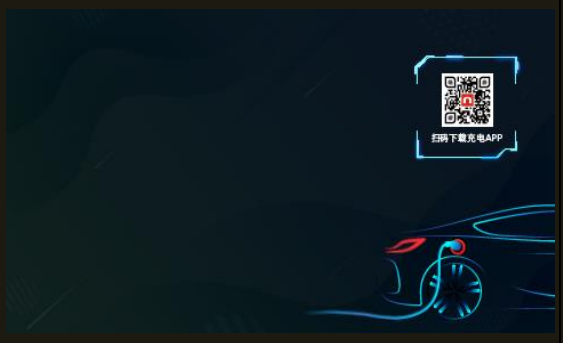
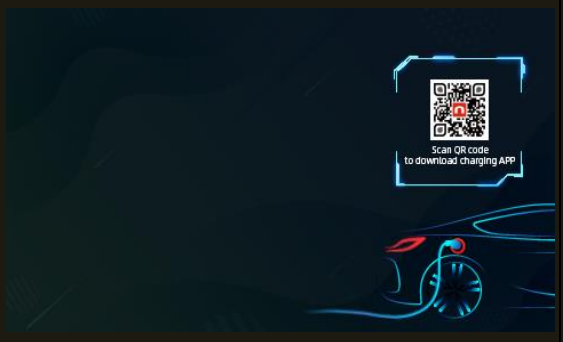
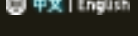
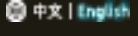
螢幕六		
螢幕七		

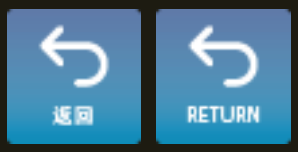
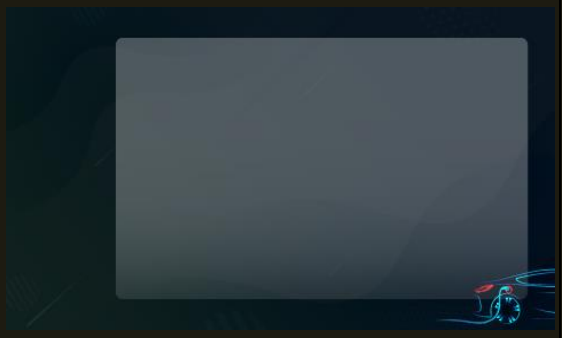
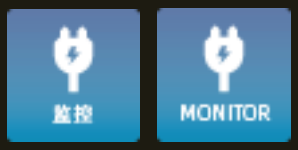
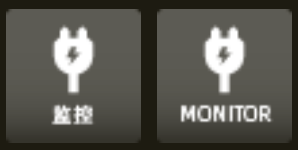
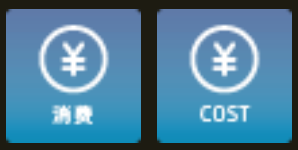
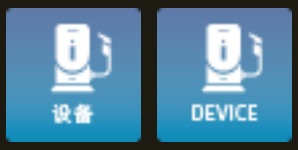
表 1-1 充电桩画面

1.1.3 充電樁畫面使用元素總覽

ChargePile App Des	Icon Name	ChargePile App GUI	Object
登錄鍵有效	Login_Pressed_Ini、Login_Pressed_Ini_En		Image
充電鍵有效	Charge_Pressed_Ini、Charge_Pressed_Ini_En		

充電鍵無效	Charge_Lock_Ini、 Charge_Lock_Ini_En	
帳戶鍵有效	Account_Pressed_Ini、 Account_Pressed_Ini_En	
帳戶鍵無效	Account_Lock_Ini、 Account_Lock_Ini_En	
查詢鍵有效	Query_Pressed_Ini、 Query_Pressed_Ini_En	
查詢鍵無效	Query_Lock_Ini、 Query_Lock_Ini_En	
價格鍵有效	Price_Pressed_Ini、 Price_Pressed_Ini_En	

價格鍵無效	Price_Ini、 Price_Ini_En	
資訊鍵有效	Info_Pressed_Ini、 Info_Pressed_Ini_En	
資訊鍵無效	Info_Ini、Info_Ini_En	
初始螢幕背景	Bg2、Bg2_En	 
中文	Lan_Cn	
英文	Lan_En	

用戶	Head_Ini	
返回鍵有效	Return1、Return1_En	
返回鍵無效	Return0、Return0_En	
其他螢幕背景	Bg3	
監控鍵有效	Mon、Mon_En	
監控鍵無效	Mon_0、Mon_0_En	
消費鍵有效	Cost、Cost_En	
消費鍵無效	Cost_0、Cost_0_En	
設備鍵有效	Device、Device_En	

設備鍵無效	Device_0、Device_0_En	
啟動充電	Start	
停止充電	Stop_Pressed	
0%電量	che0	
0%~19%電量	che1	
20%~39%電量	che2	
40%~59%電量	che3	
60%~79%電量	che4	
80%~100%電量	che5	

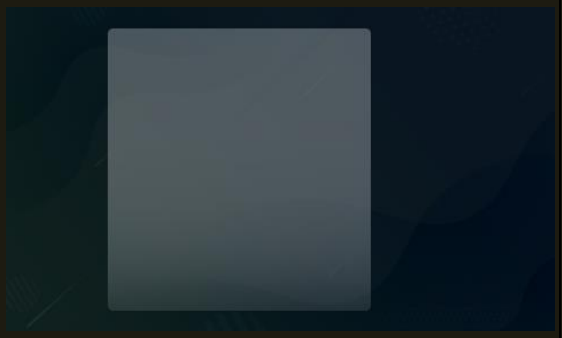
充電螢幕背景	Bg_Cha		
連接線	Connect		
短數據底線	ExpenseLine		
長數據底線	MonLine		

表 1-2 畫面元素

1.1.4 相關資源

- SEGGER emWin 官方論壇：Segger emWin Forum
- 新唐官方論壇：Nuvoton emWin Forum
- SEGGER emWin AppWizard 工具使用手冊下載：AppWizard User Manual

1.2 執行步驟

使用 AppWizard 工具完成一項 GUI 專案的執行步驟可分為五大階段，第一階段需準備硬體與軟體下載，第二階段則於 AppWizard 中完成對應的畫面設計、object 設定，第三階段則於 AppWizard 中完成 Interactions 設定、動畫設定，第四階段使用 AppWizard 匯出代碼並同時加入客制化代碼，以完成更多應用功能，第五階段則是實際燒錄到板端。

以下於 1.2.1-1.2.5 就各項階段詳細說明執行步驟。

1.2.1 準備相關硬體與軟體資源

1. 硬體使用 NuMaker-M467HJ V1.0，相關資源可參考 輔助線路、開發板使用手冊。

2. 軟體資源需下載 AppWizard 軟體工具

   1) 請點擊以下連結 emWin Software Package for M460 進入下載頁面，點選 emWin Software Package for M460 專案即可開始下載，下載前需註冊與登入新唐官網會員。

   2) 下載完成後，解壓縮，開啟 M460_emWin_NonOS 資料夾。

   3) 依表 1-3 路徑，點擊 AppWizard_V134a_630a_Install.exe 安裝 AppWizard 工具，需注意若有舊的 AppWizard 版本，需將舊版本完全移除後，才能安裝新版本。

M460_emWin_NonOS emWin Tool AppWizard_V134a_630a_Install.exe

表 1-3 AppWizard 工具安裝檔路徑

1.2.2 在 AppWizard 工具中編輯與設定 GUI

1. 新建工程，並添加專案所需螢幕。

   1) 于桌面快捷圖示開啟 AppWizard 工具。

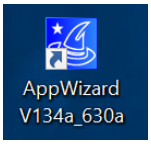


圖 1-3 AppWizard 桌面快捷方式

2) 點選 “Create new project” 以創建新專案。

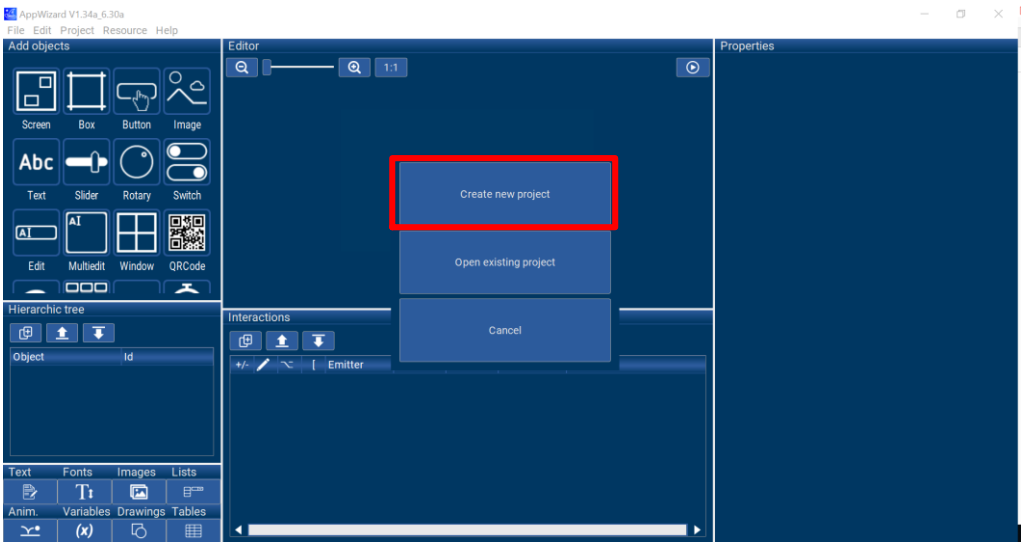


圖 1-4 創建專案頁面

3) 依下圖設定專案路徑、名稱，符合使用螢幕的顏色格式：16 Bit, GUICC_M565，勾選“Enable Multibuffering”，點選 “ok”。

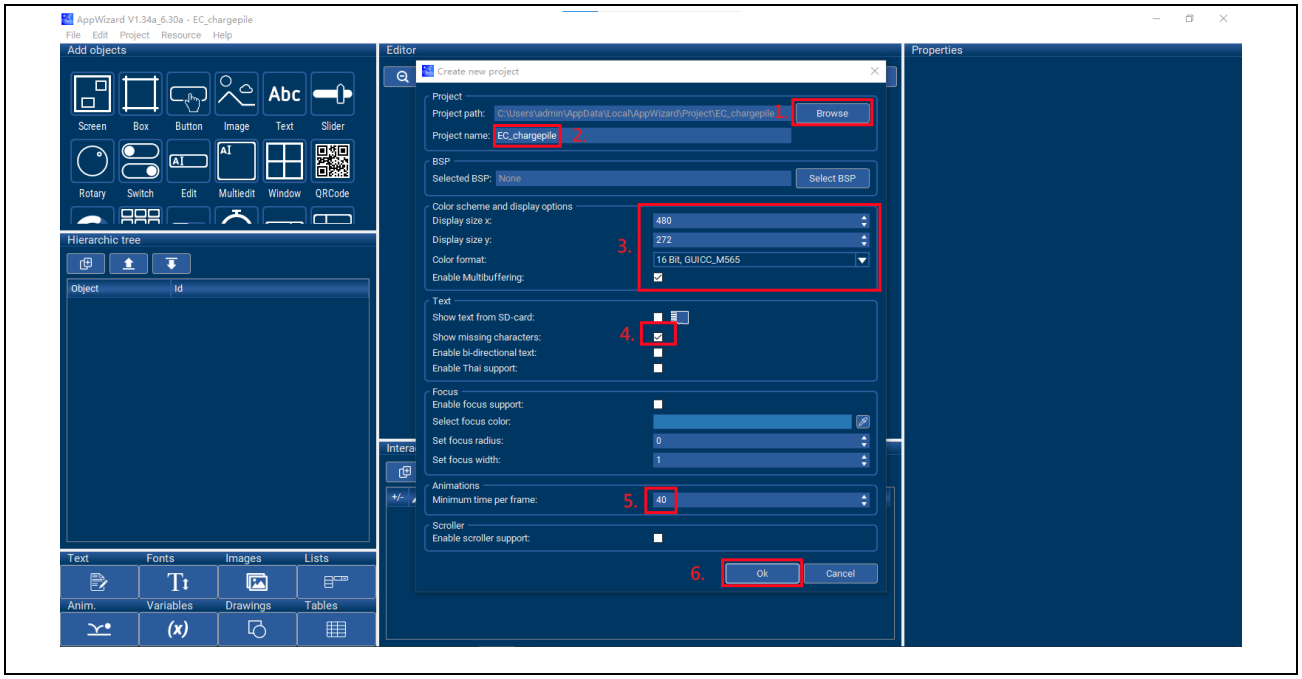


圖 1-5 新專案設定項目

4) 在“Add objects”下點選七次“Screen”，以創建七個螢幕，並在“Hierarchic tree”中，為 Screen 分別命名為 ID_SCREEN_Init、ID_SCREEN_Login、ID_SCREEN_Charge、ID_SCREEN_Account、ID_SCREEN_Query、ID_SCREEN_Price、ID_SCREEN_Info 以利辨認。

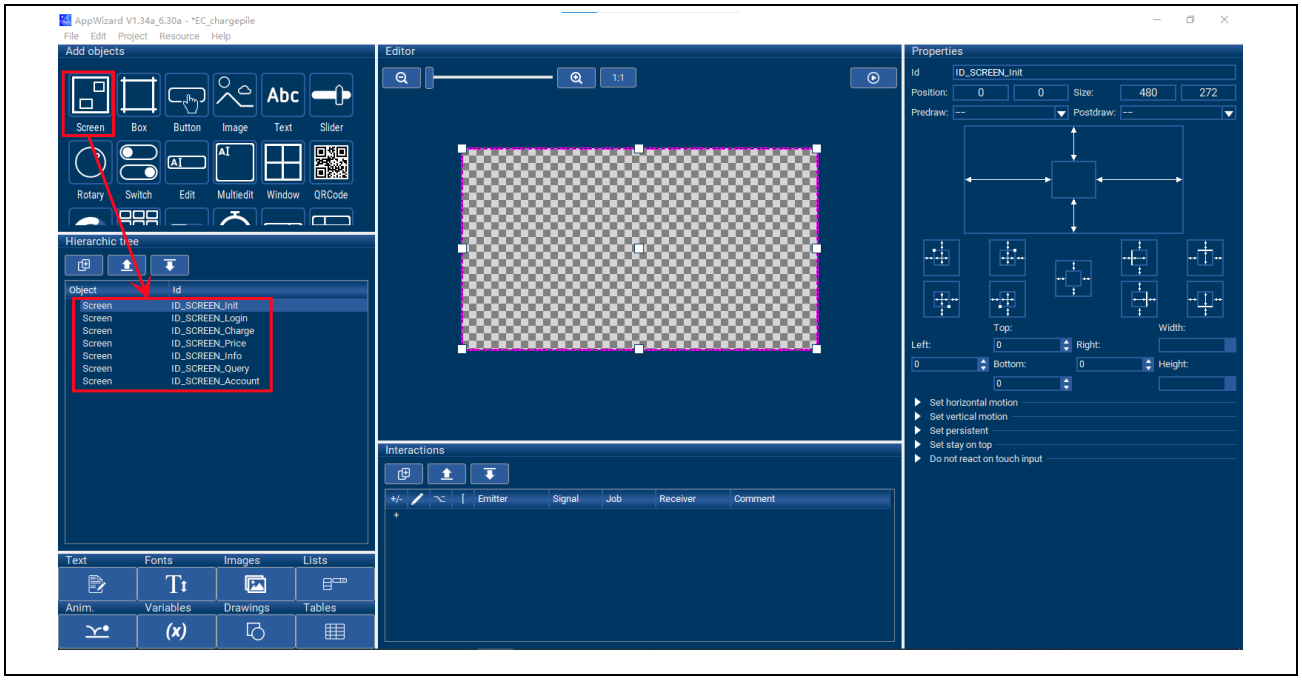


圖 1-6 創建 7 個螢幕

2. Init 螢幕設計。包括背景、使用者圖案；用戶名、即時時間顯示；六個螢幕切換鍵及語言切換鍵。

1) 點選 ID_SCREEN_Init 以編輯第一個畫面，在“Add objects”下面點選“Image”，以創建一個背景，將名稱改為 ID_IMAGE_Bg_Ini，然後在 Properties 中設置它的位置和尺寸屬性。

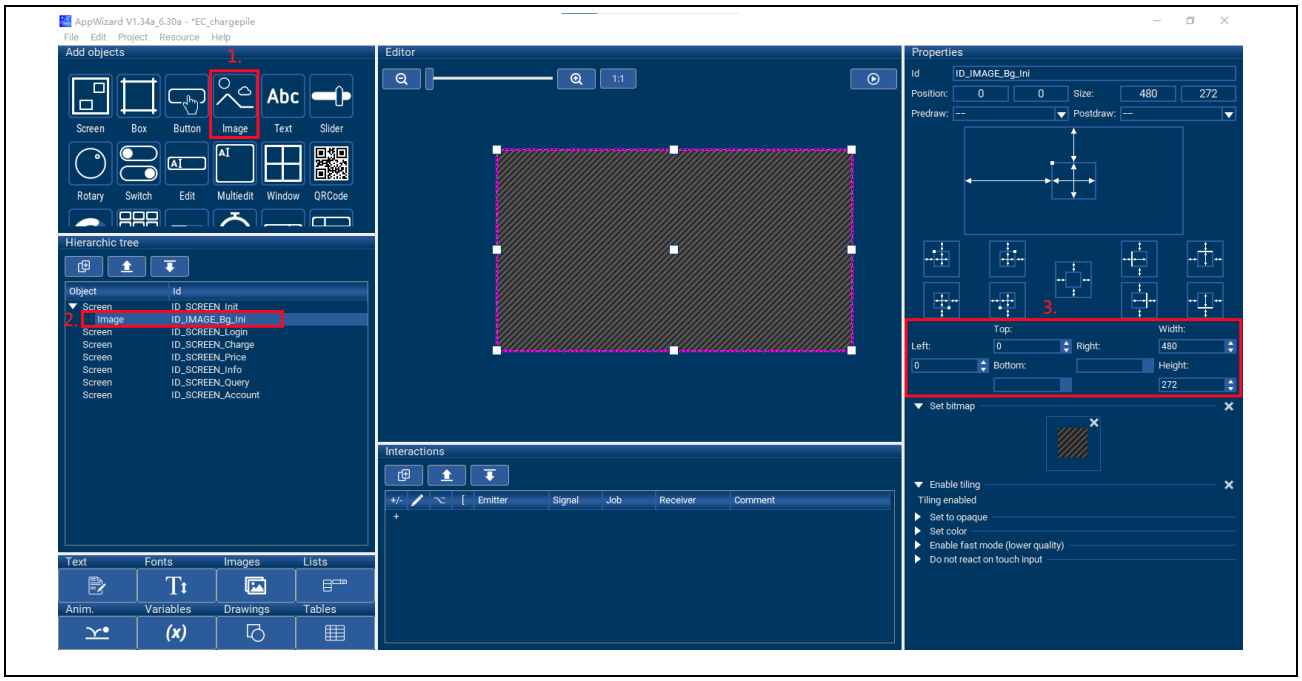


圖 1-7 以 Image 創建背景及設置位置、尺寸屬性

2) 在 Properties 中，為“Bg”設定 Bitmap 屬性。

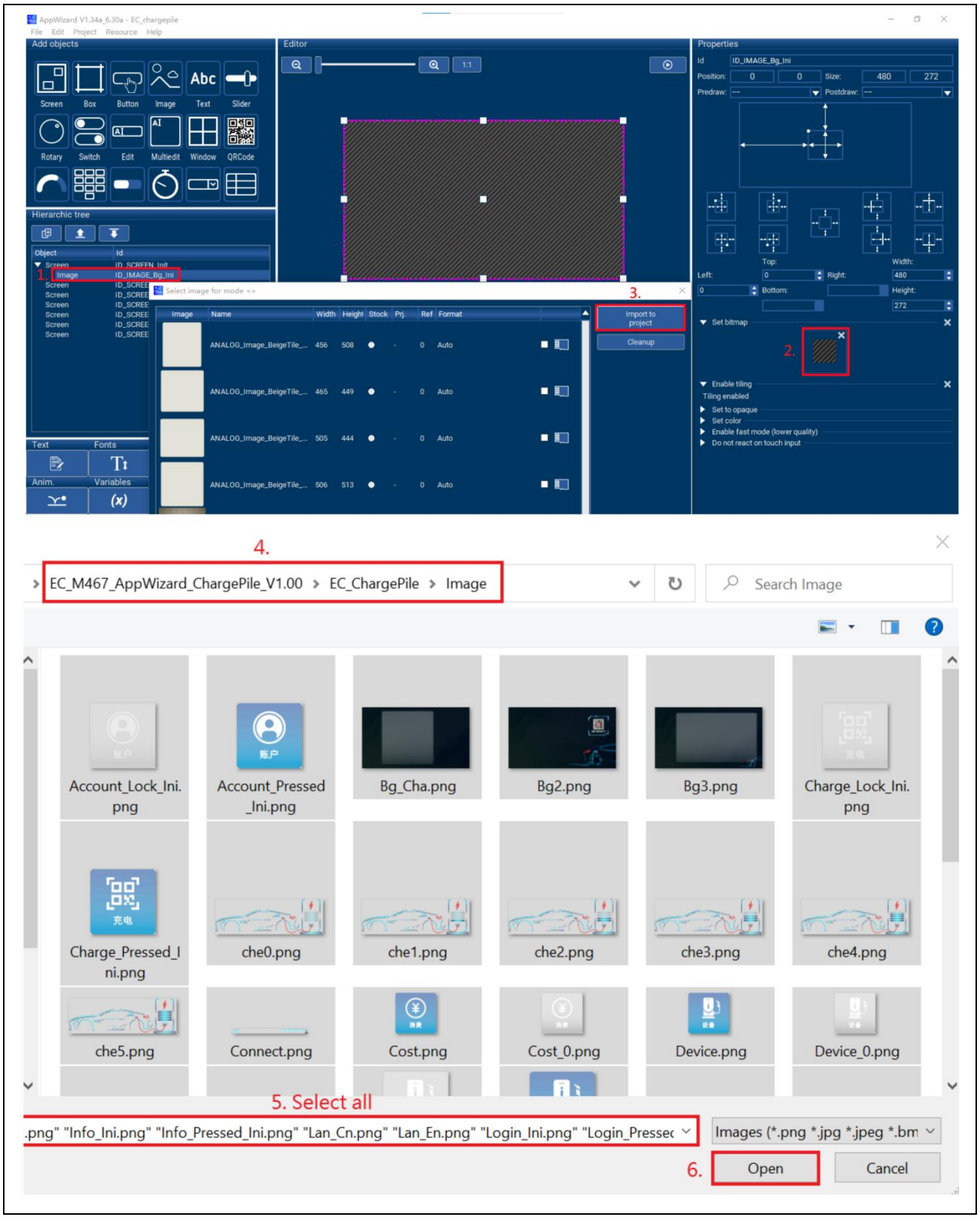


圖 1-8 導入專案所需 icon、為“Bg” 設置 Bitmap 屬性

3) 點選 ID_SCREEN_Init，為製作功能切換鍵，在 “Add objects” 下七次點選 “Button”，並於 “Hierarchic tree” 中分別重新命名為 ID_BUTTON_Login_Ini 、ID_BUTTON_Charge_Ini 、ID_BUTTON_Acc_Ini 、ID_BUTTON_Query_Ini 、ID_BUTTON_Price_Ini 、ID_BUTTON_Info_Ini 、ID_BUTTON_Lan_Ini 以利辨認，並對他們的位置、尺寸屬性進行相關設置。

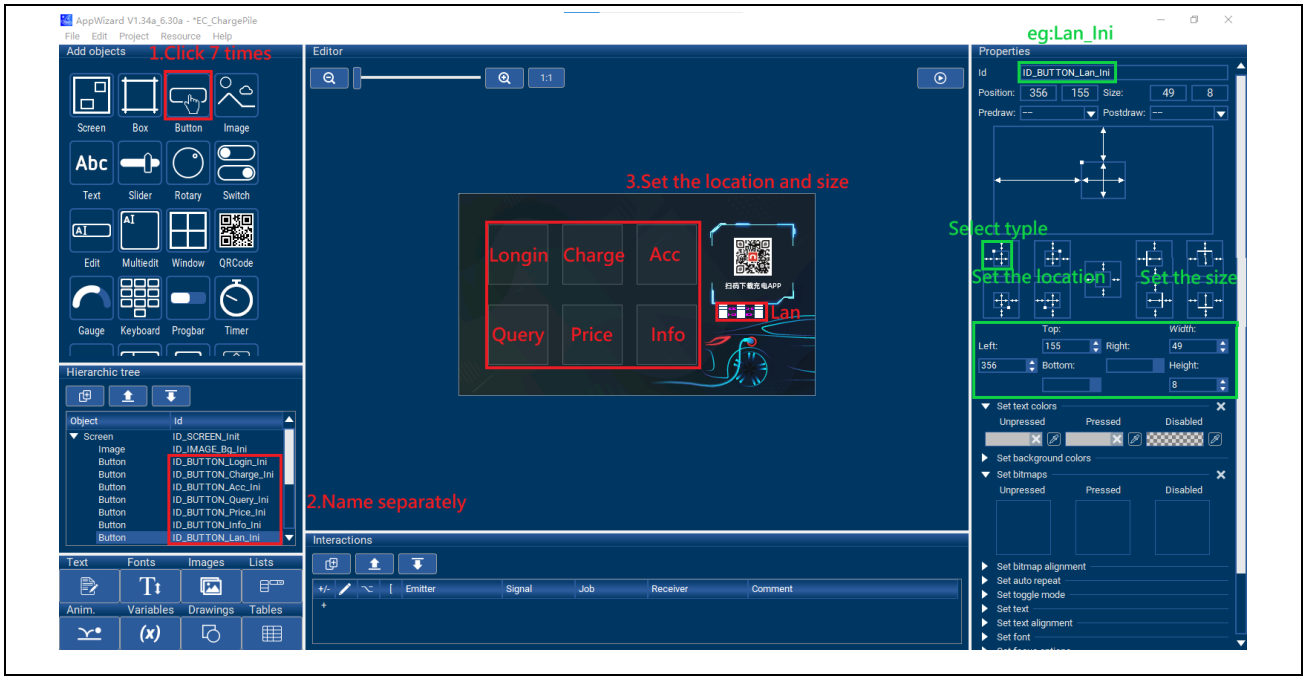


圖 1-9 以 Button 創建螢幕切換鍵

4) 在 Properties 中，為各 Button 設定 Bitmaps 屬性（參考 1.2.2 章 2 節 2 步驟，忽略其中第 3~6 子步驟）。

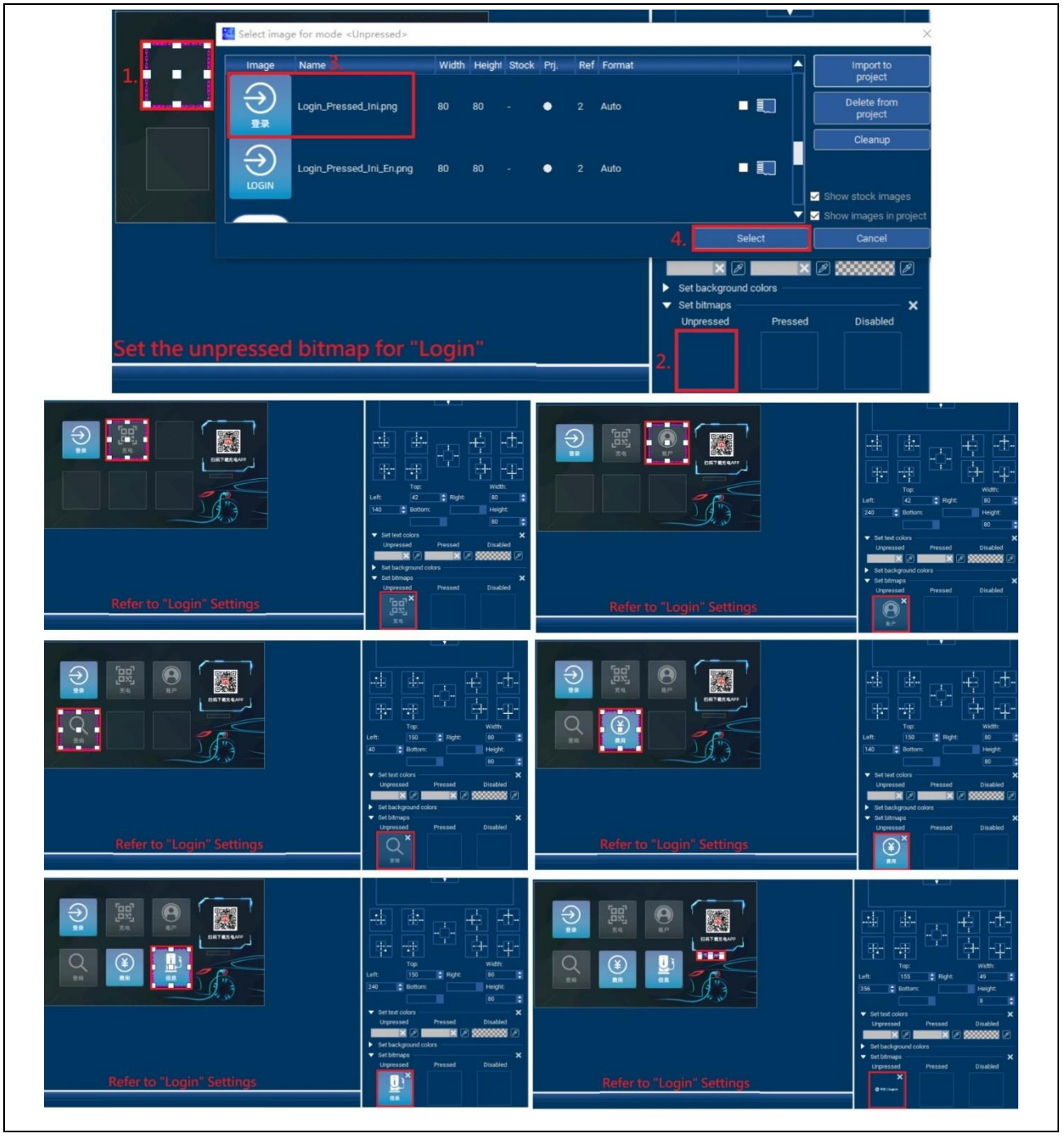
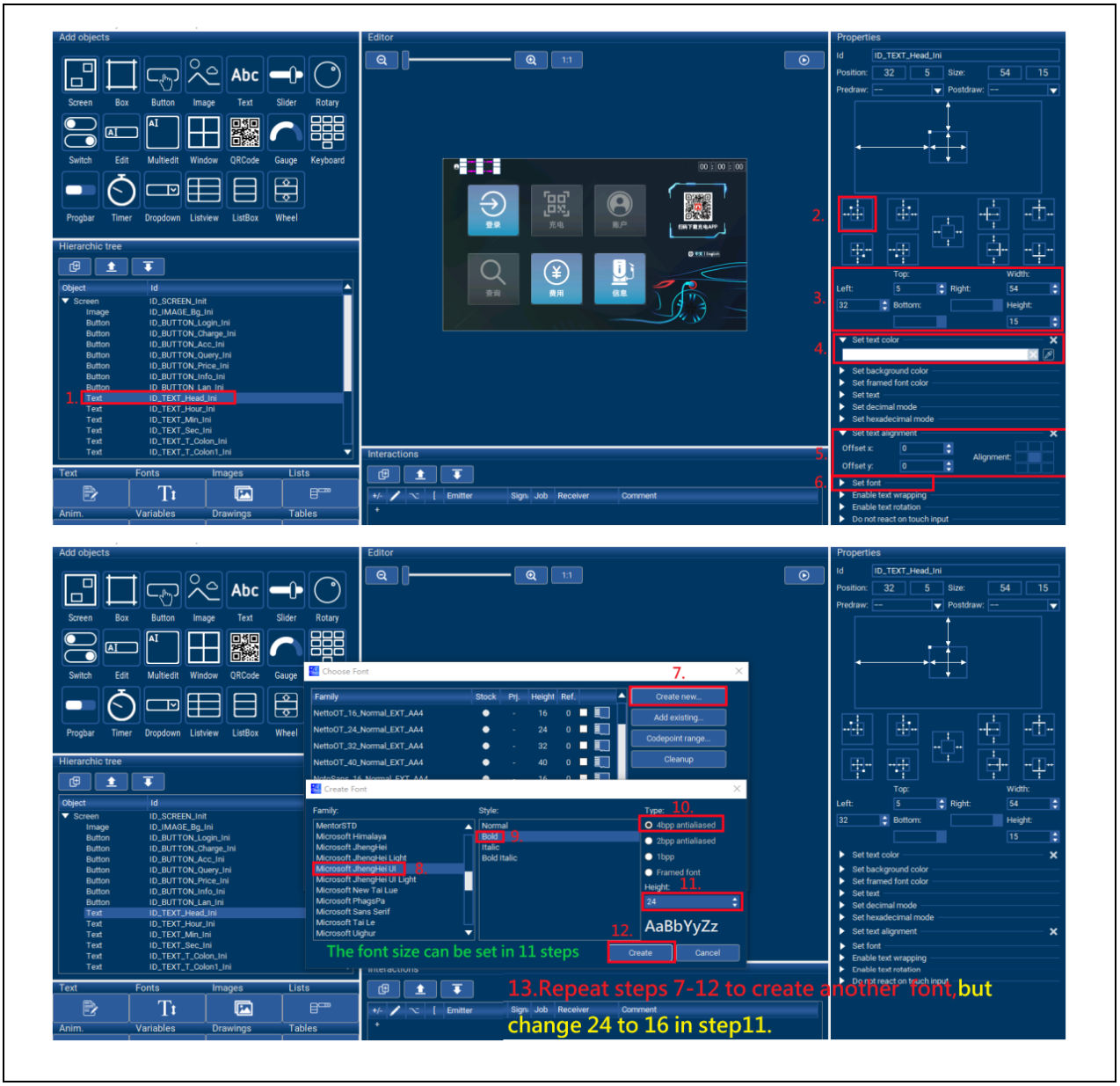


圖 1-10 設定各 Button Bitmaps 屬性

5) 點選 ID_SCREEN_Init，為製作顯示文本，在 “Add objects” 中六次點選 “Text”，並於 “Hierarchic tree” 中分別重新命名為 ID_TEXT_Head_Ini 、ID_TEXT_Hour_Ini 、ID_TEXT_Min_Ini 、 ID_TEXT_Sec_Ini 、 ID_TEXT_T_Colon_Ini 、 ID_TEXT_T_Colon1_Ini 以利辨認。

6) 設置 ID_TEXT_Head_Ini 物件的位置、尺寸、顏色、對齊模式、字體，用於顯示文本。注意子步驟 7~13 是創建 2 種字體。



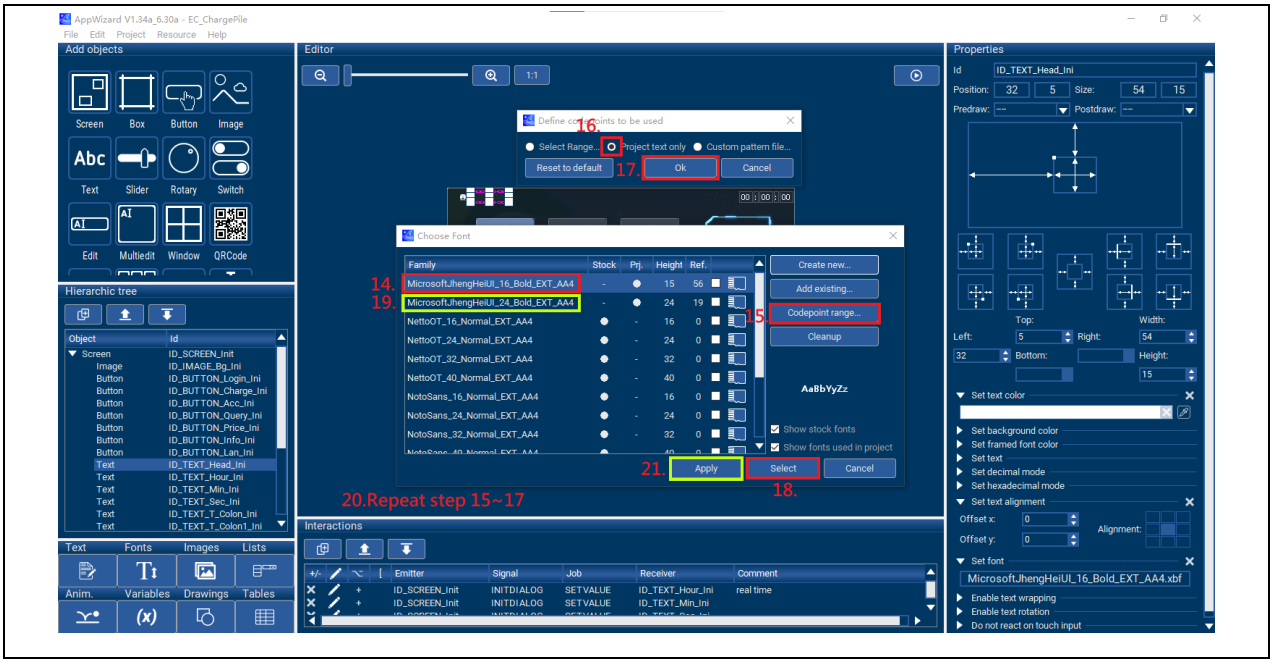


圖 1-11 設置 Text 以顯示文本的相關屬性

7) 設置 ID_TEXT_Hour_Ini 相關屬性以顯示 2 位元數值(ID_TEXT_Min_Ini、ID_TEXT_Sec_Ini 的設置和 ID_TEXT_Hour_Ini 類似)

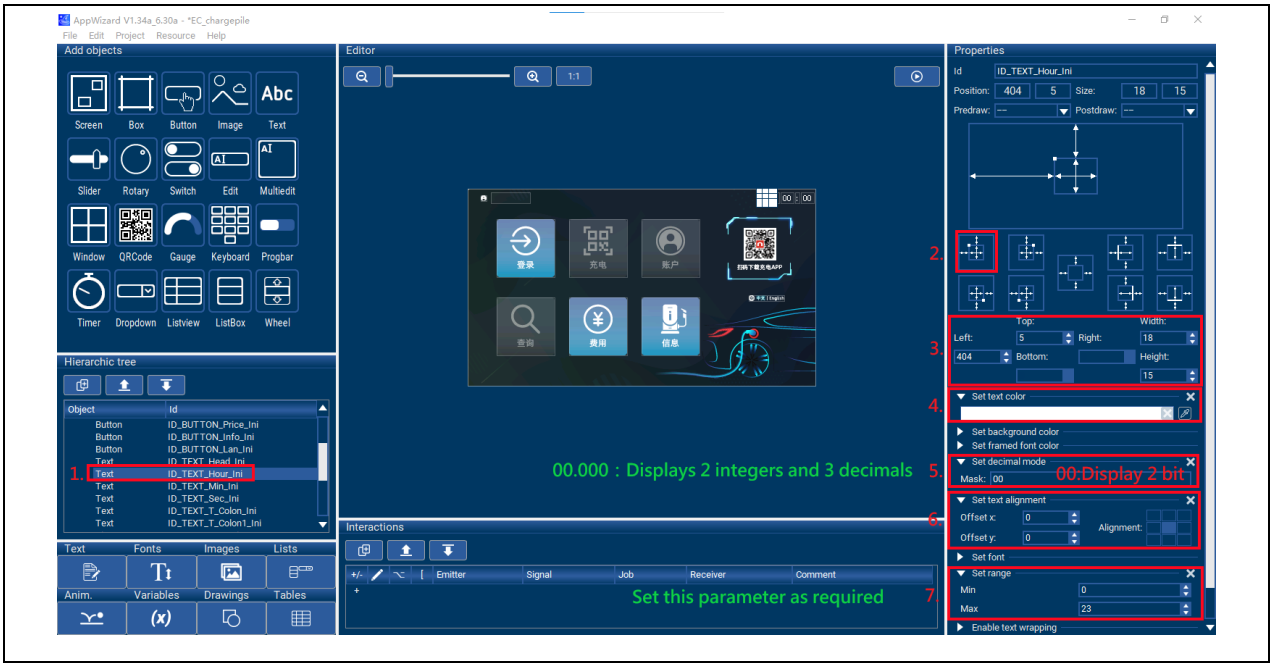


圖 1-12 設置 Text 以顯示數值的相關屬性

8) 設置 ID_TEXT_T_Colon_Ini 以顯示中英文時間分隔符號(ID_TEXT_T_Colon1_Ini 的設置與 ID_TEXT_T_Colon_Ini 類似)。注意子步驟 7~12 以創建中英文 2 種語言。

圖 1-13 設置 Text 以顯示中英文本相關屬性

9) 點選 ID_SCREEN_Init 以編輯第一個畫面，於 “Add objects”中點選”Image”，以創建使用者圖案，將名稱改為 ID_IMAGE_User_Ini，並設置位置、尺寸、Bitmaps 屬性。

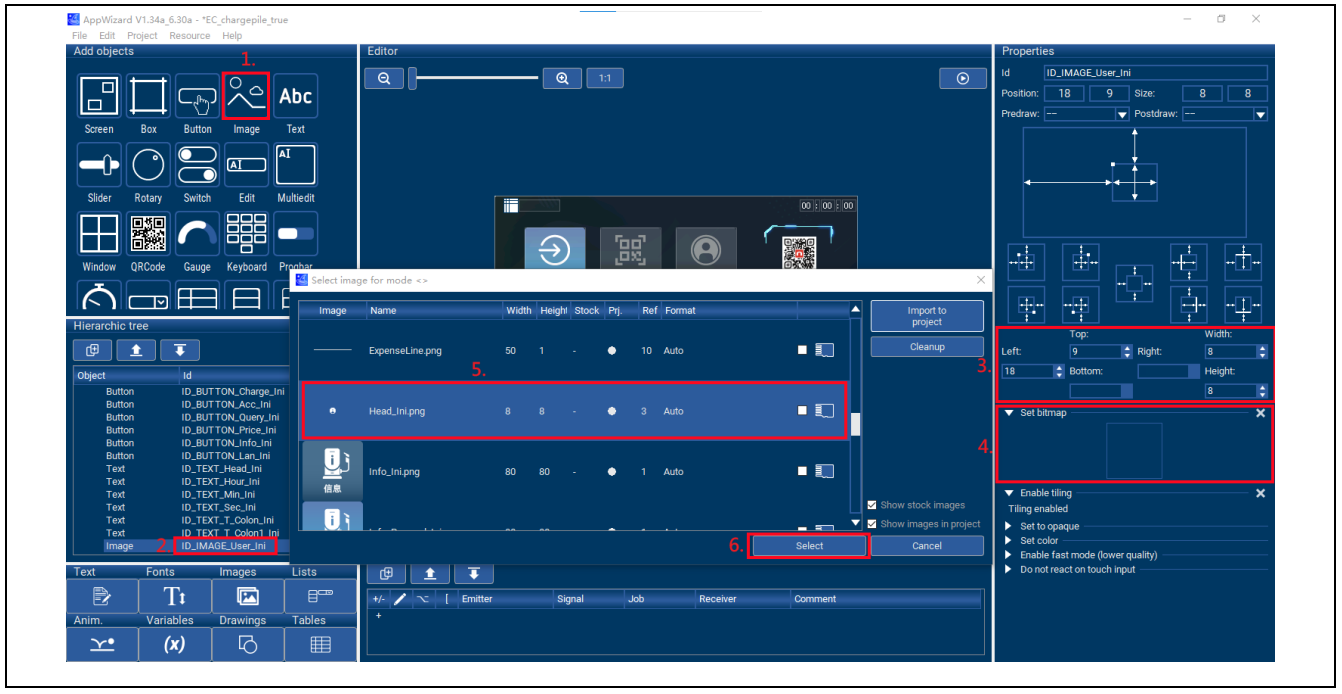


圖 1-14 設置“User”位置、尺寸及 Bitmaps 屬性

10) 點選 ID_SCREEN_Init 以編輯第一個畫面，在 “Add objects”中點選“Timer”，以創建一個計時器，將名稱改為 ID_TIMER_1S_Ini。並設置定時週期及使能自動重啟模式。

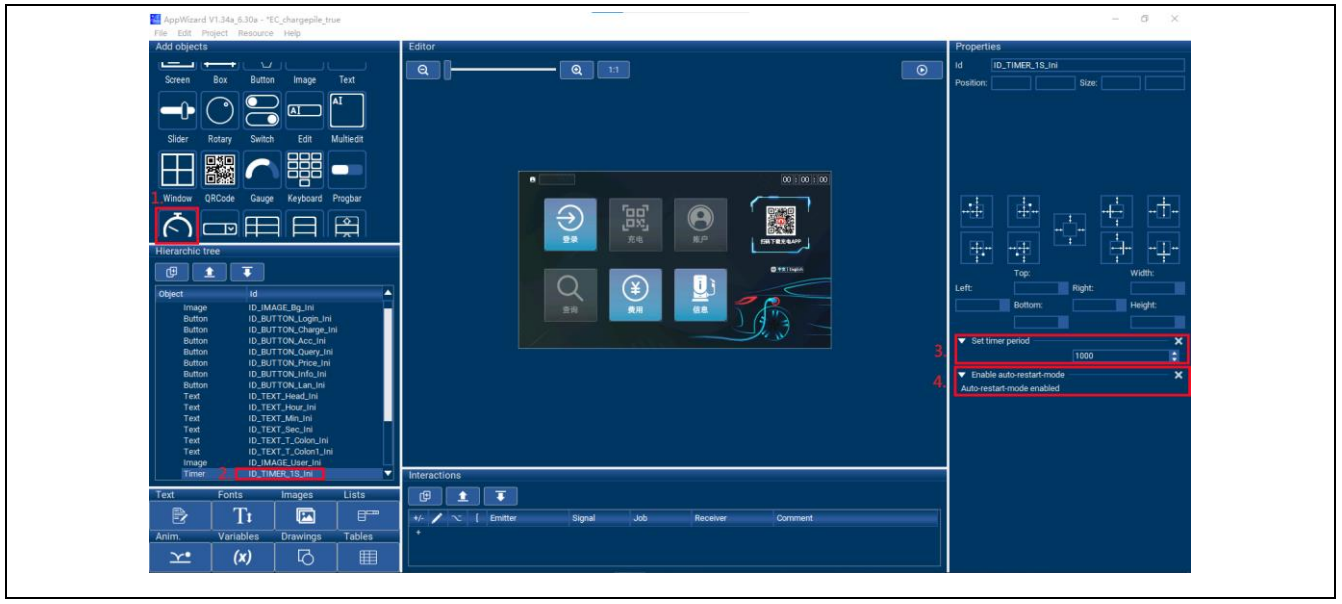


圖 1-15 設置 Timer 相關屬性

3. Charge 螢幕設計。包括主體部分、監控視窗、消費視窗、設計視窗。

1) Charge 螢幕主體。參考 Init 螢幕設計，添加各物件並設置他們的相關屬性。包括 3 個切換鍵 1 個返回鍵及 1 個啟動/停止充電鍵；即時時鐘、充電時間、用戶名顯示；背景、使用者圖案載入；充電百分數及圖形化顯示。

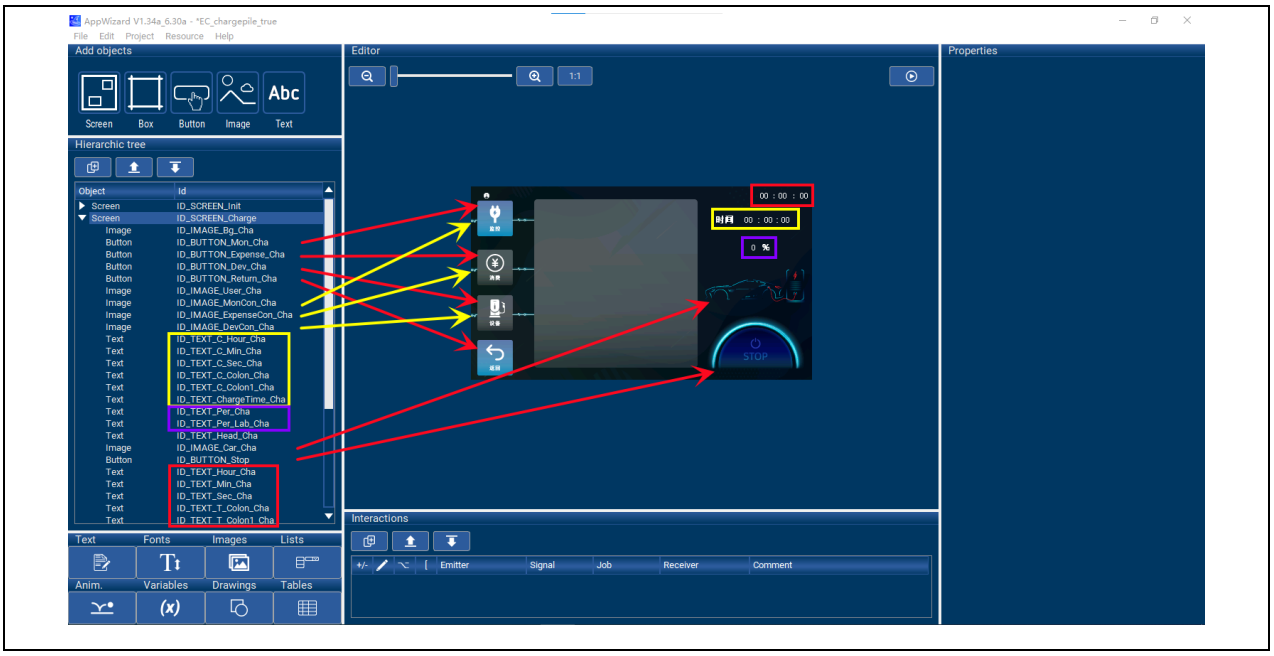


圖 1-16 添加所有物件並設置它們的相關屬性

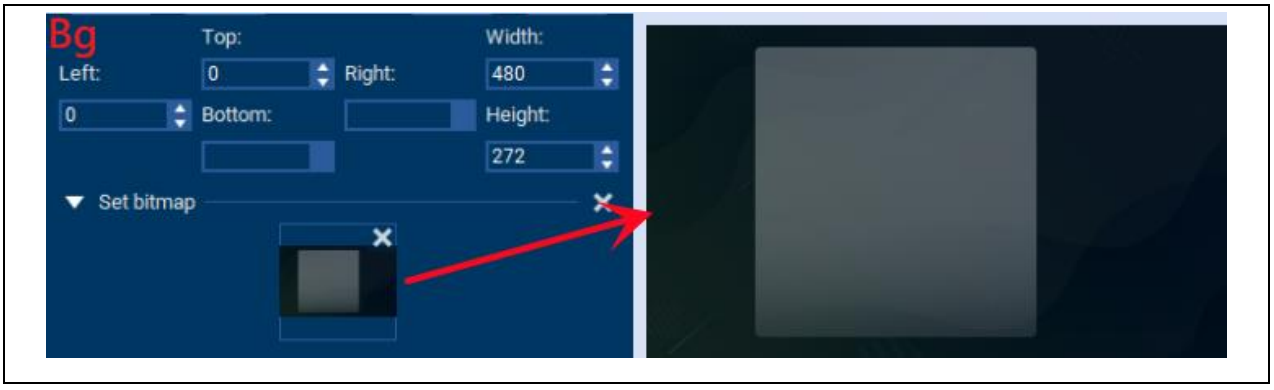


圖 1-17 設置“Bg”的屬性

圖 1-18 設置“Mon”、 “Dev”、 “Expense”、 “Return” 的相關屬性

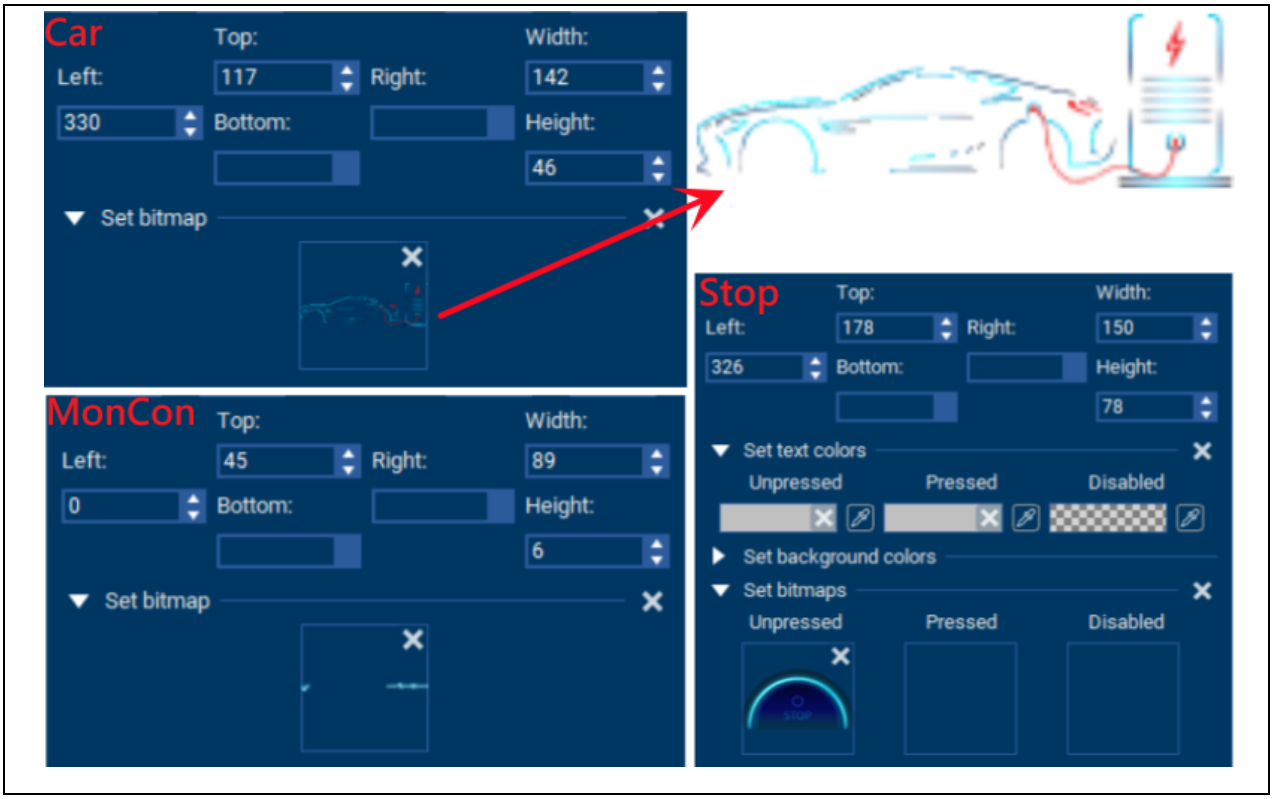


圖 1-19 設置“Car”、“MonCon”、“Stop”的相關屬性

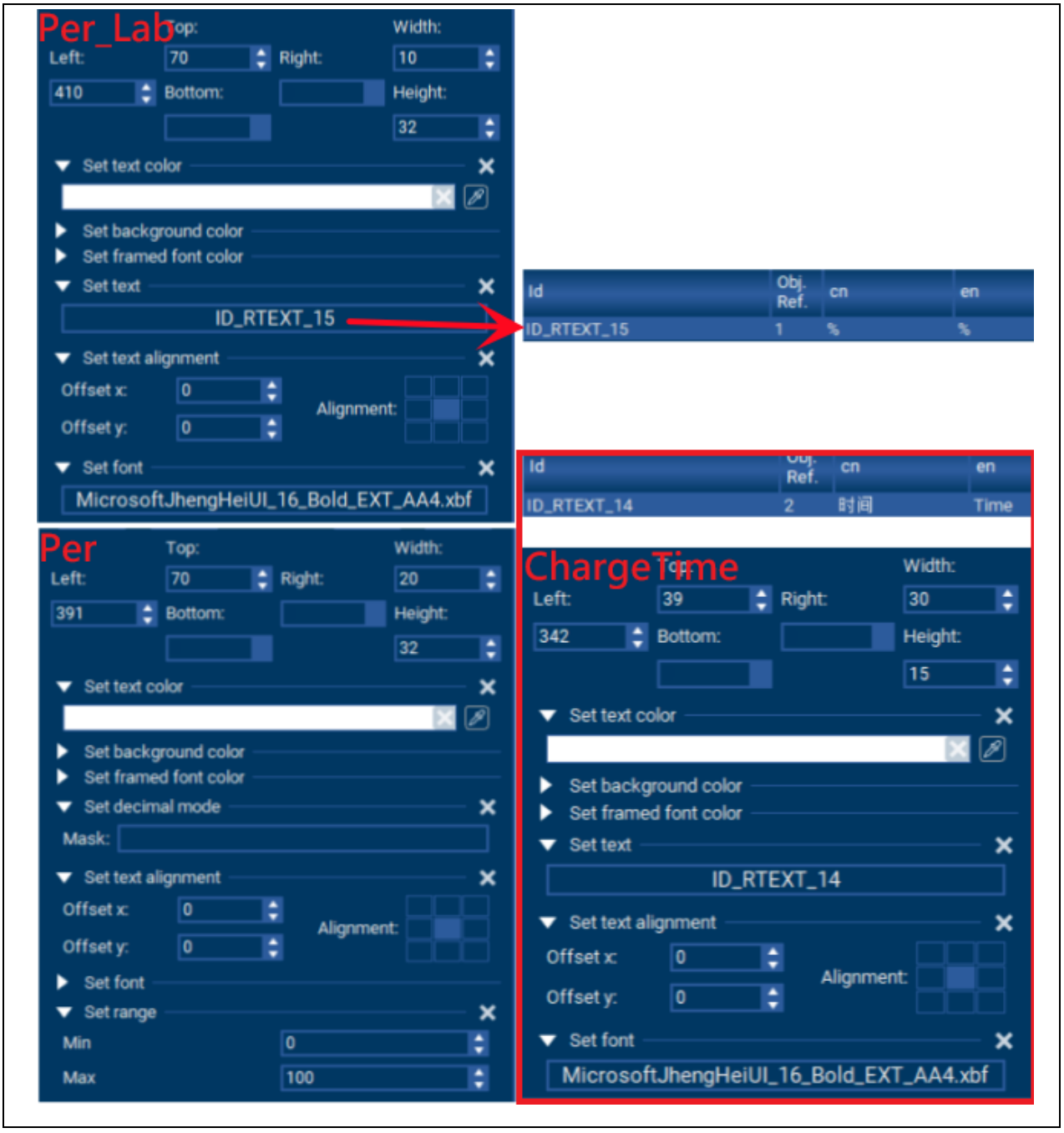


圖 1-20 設置“Per_Lab”、“Per”、“ChargeTime” 相關屬性

2) Charge 螢幕監控視窗。包括累計電量、累計消費顯示和充電電流、充電電壓顯示。

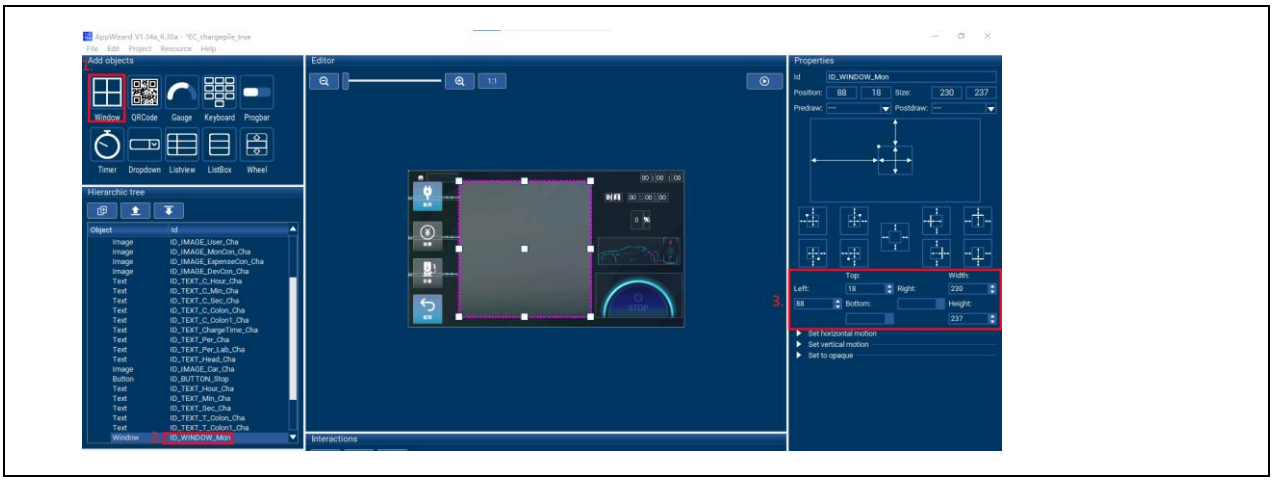


圖 1-21 新建“Mon”窗口

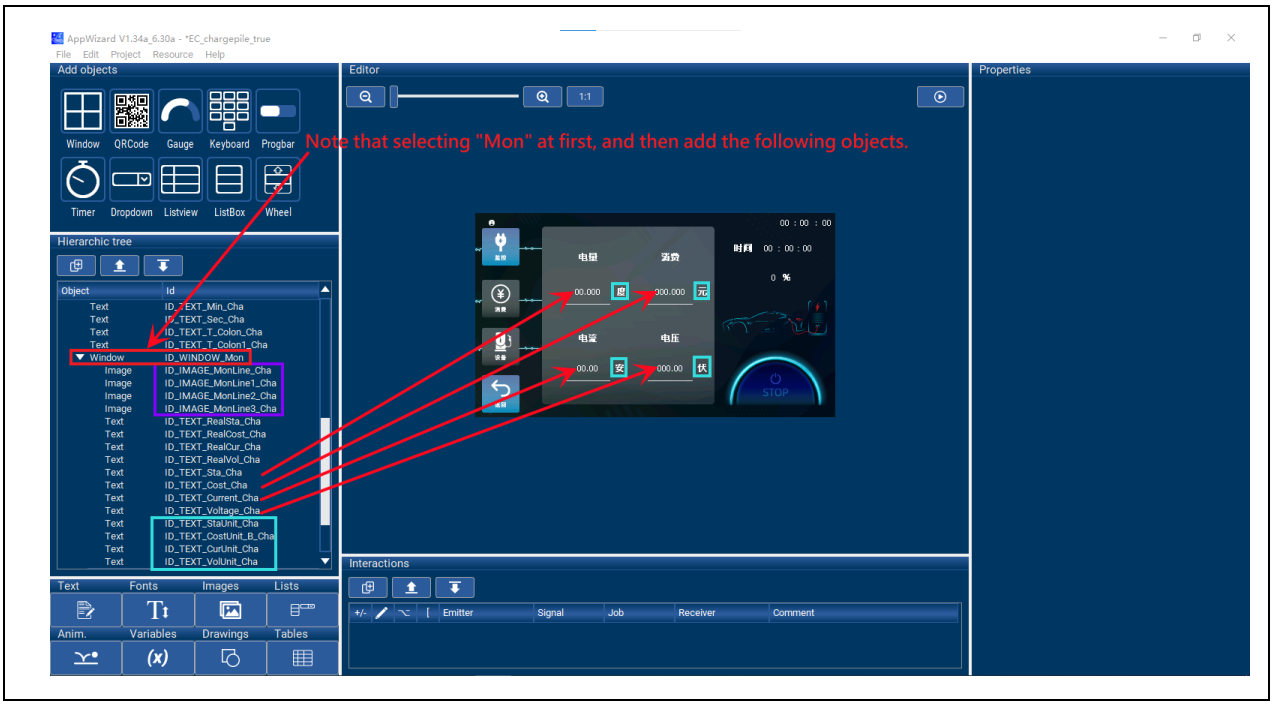


圖 1-22 添加視窗所有物件並設置它們的相關屬性

圖 1-23 設置“RealSta”、“Realcost”、“RealCur”、“RealVol”相關屬性

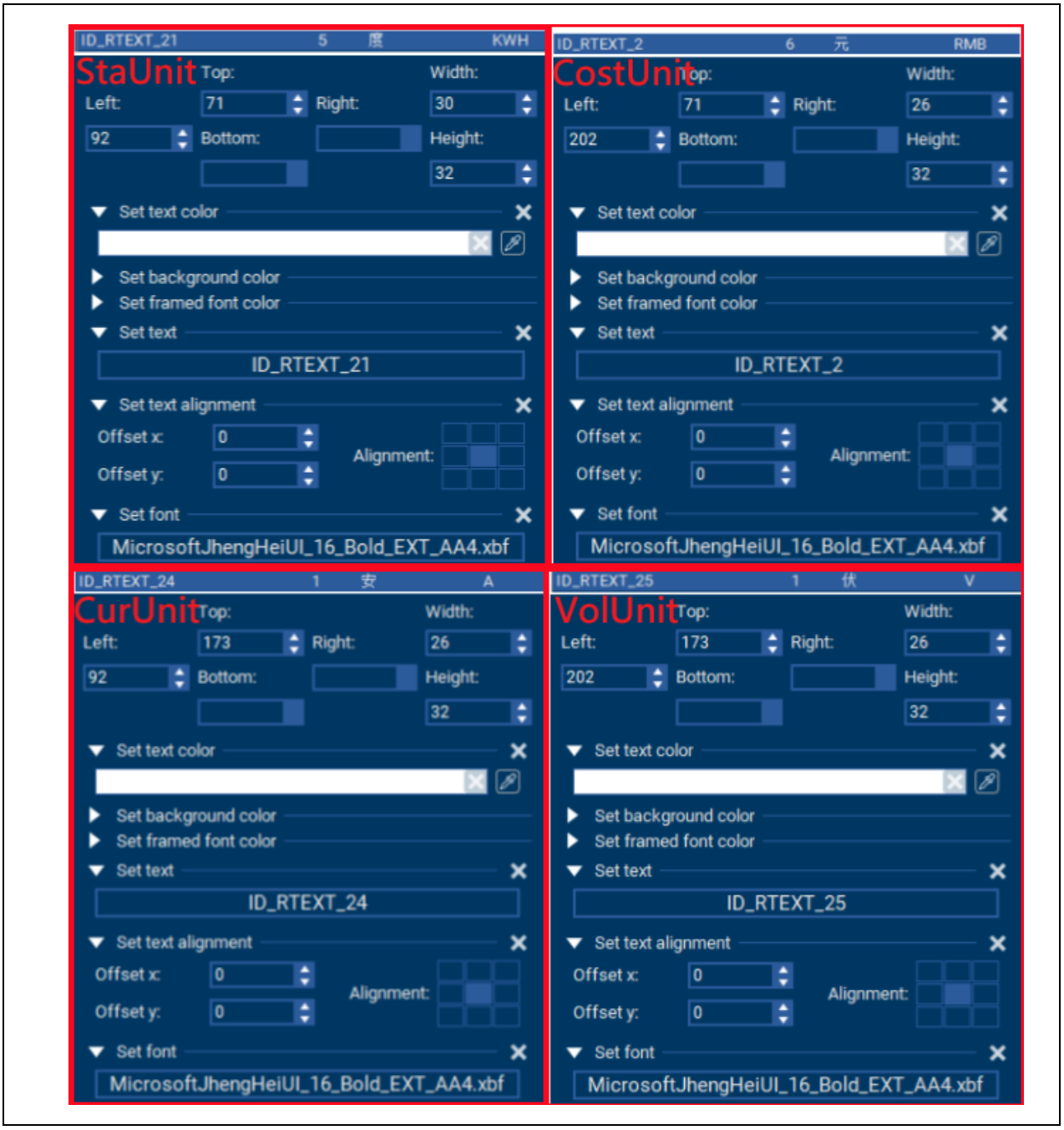


圖 1-24 設置“StaUnit”、“CostUnit”、“CurUnit”、“VolUnit”相關屬性

圖 1-25 設置“Sta”、“Cost”、“Current”、“Voltage” 相關屬性

3) Charge 螢幕消費視窗，參考 1.2.2 章 3 節 2 步驟。包括顯示尖、峰、平、谷各時間段所消費的電量及費用。

圖 1-26 添加視窗所有物件並設置它們的相關屬性

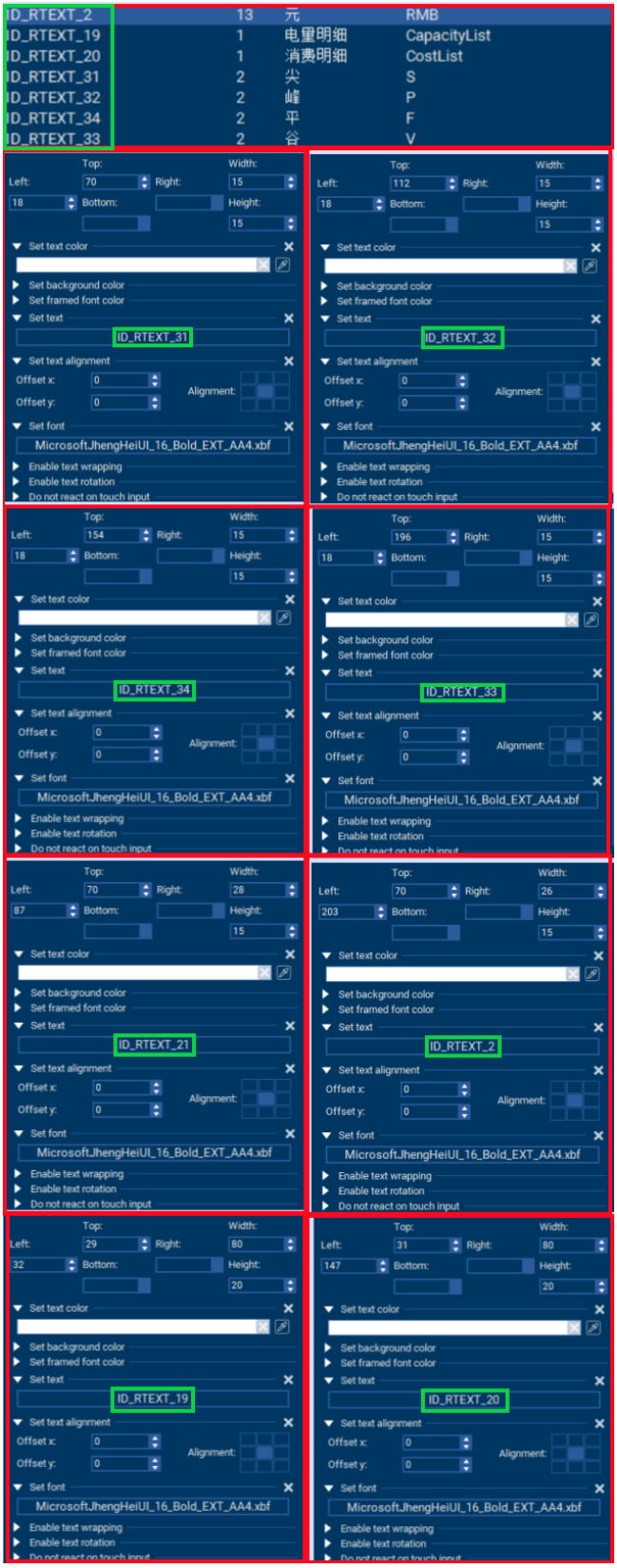


圖 1-27 設置所有 Text 文字屬性

4) Charge 螢幕設備視窗，參考 1.2.2 章 3 節 2 步驟。包括功率、當前價格顯示。

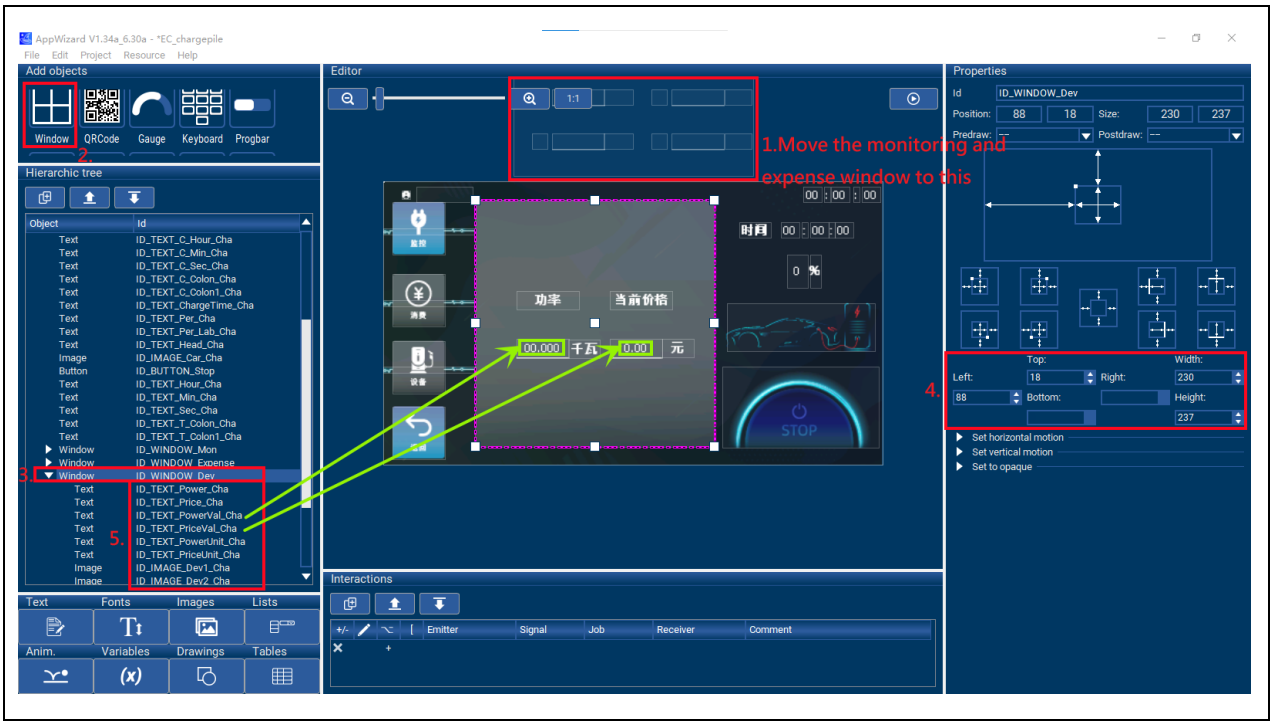


圖 1-28 添加視窗所有物件並設置它們的相關屬性

圖 1-29 設置“Power”、“Price”、“PowerUnit”、“PriceUnit”相關屬性

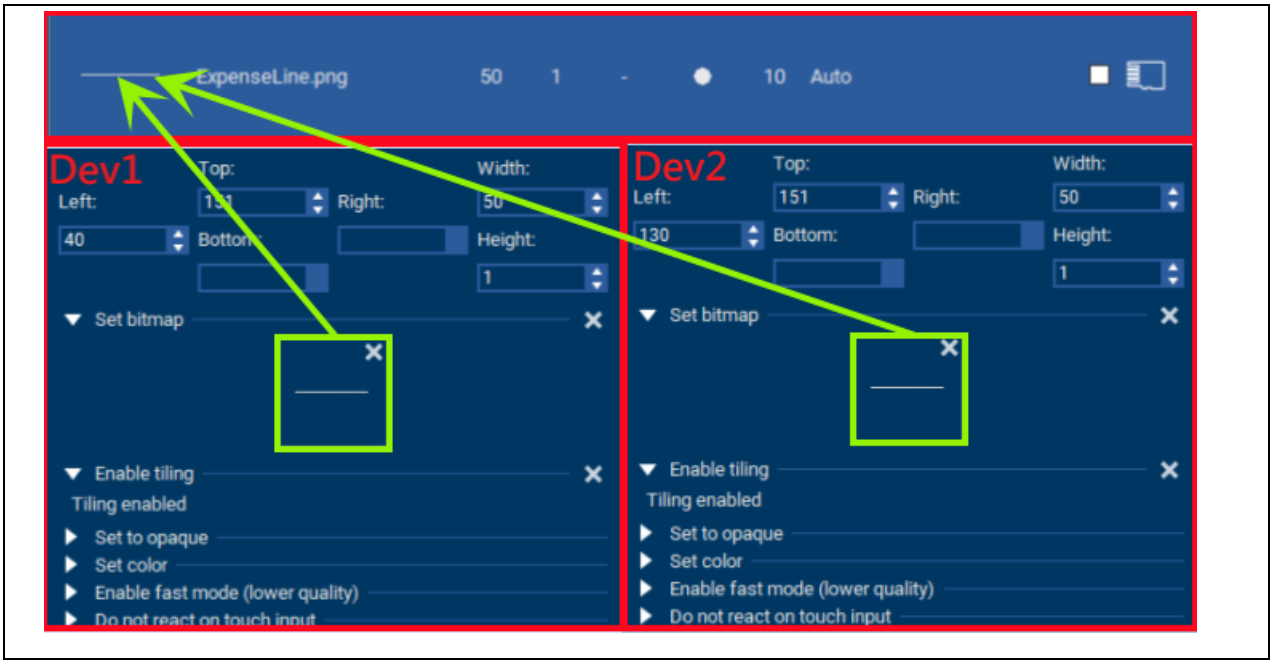


圖 1-30 設置“Dev1”、“Dev2”相關屬性

5) 添加 2 個計時器。分別用於即時時鐘顯示和充電計時顯示。

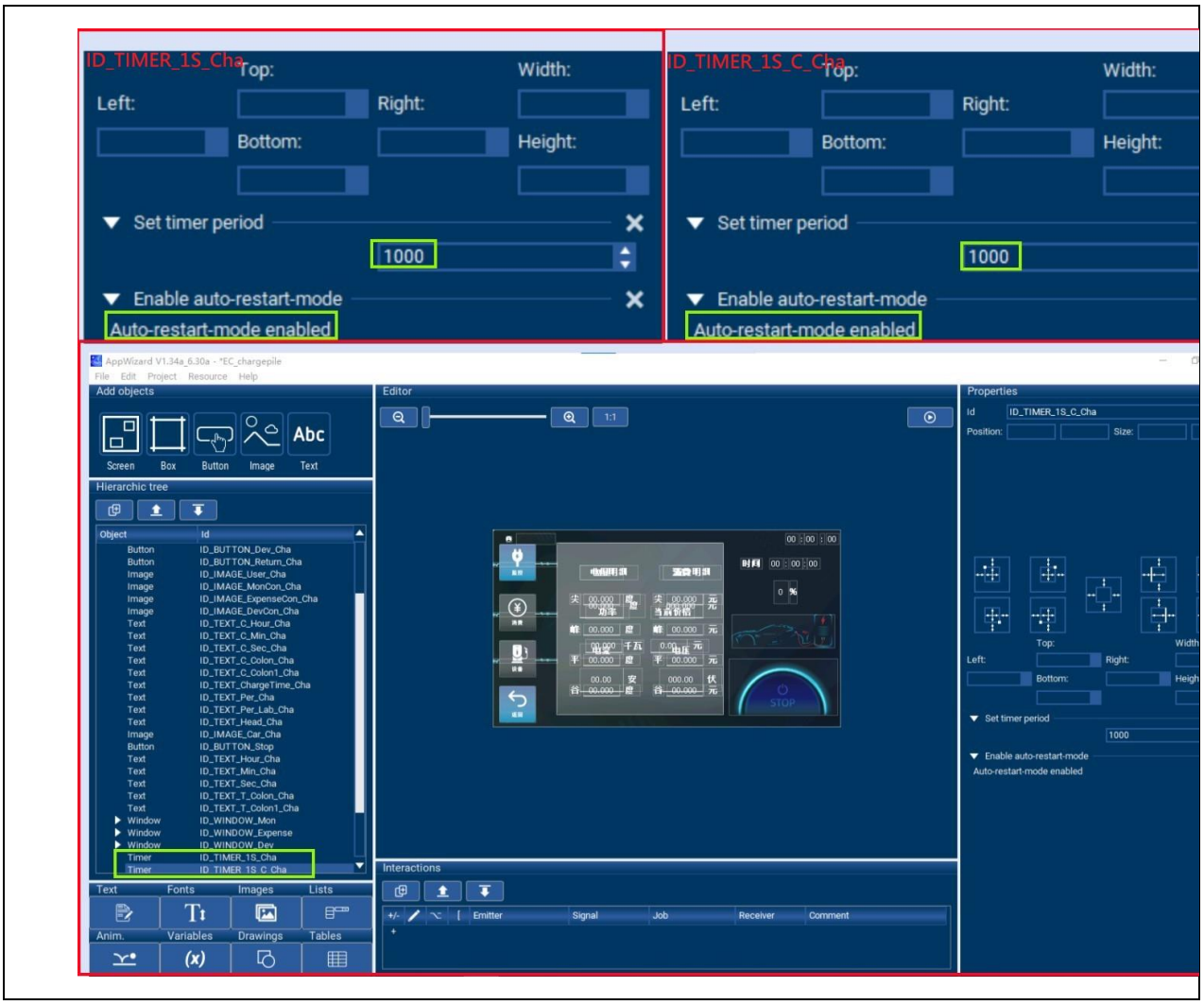


圖 1-31 設置計時器相關屬性

4. Price 螢幕設計。主要包括顯示一天中 6 個時間段的價格。

1) 各時間段價格顯示。參考 Init 螢幕設計，添加所有物件並設置它們的相關屬性。

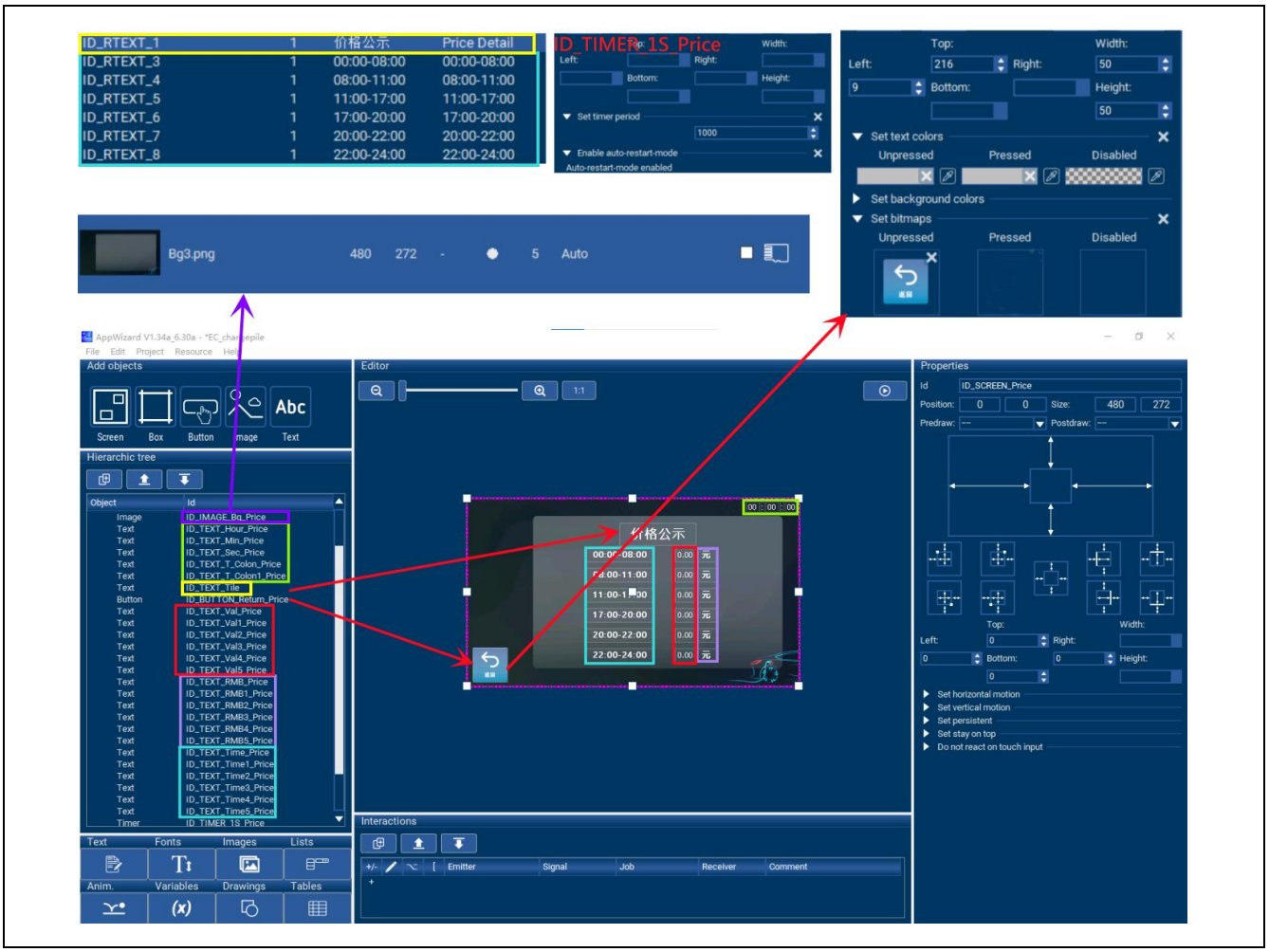


圖 1-32 添加所有物件並設置他們的相關屬性

5. Info 螢幕設計。主要包括顯示當前充電站名稱、充電類型及支持的充電槍號。

   1) 充電站、類型、槍號顯示。參考 Init 螢幕設計，添加所有物件並設置它們的相關屬性。

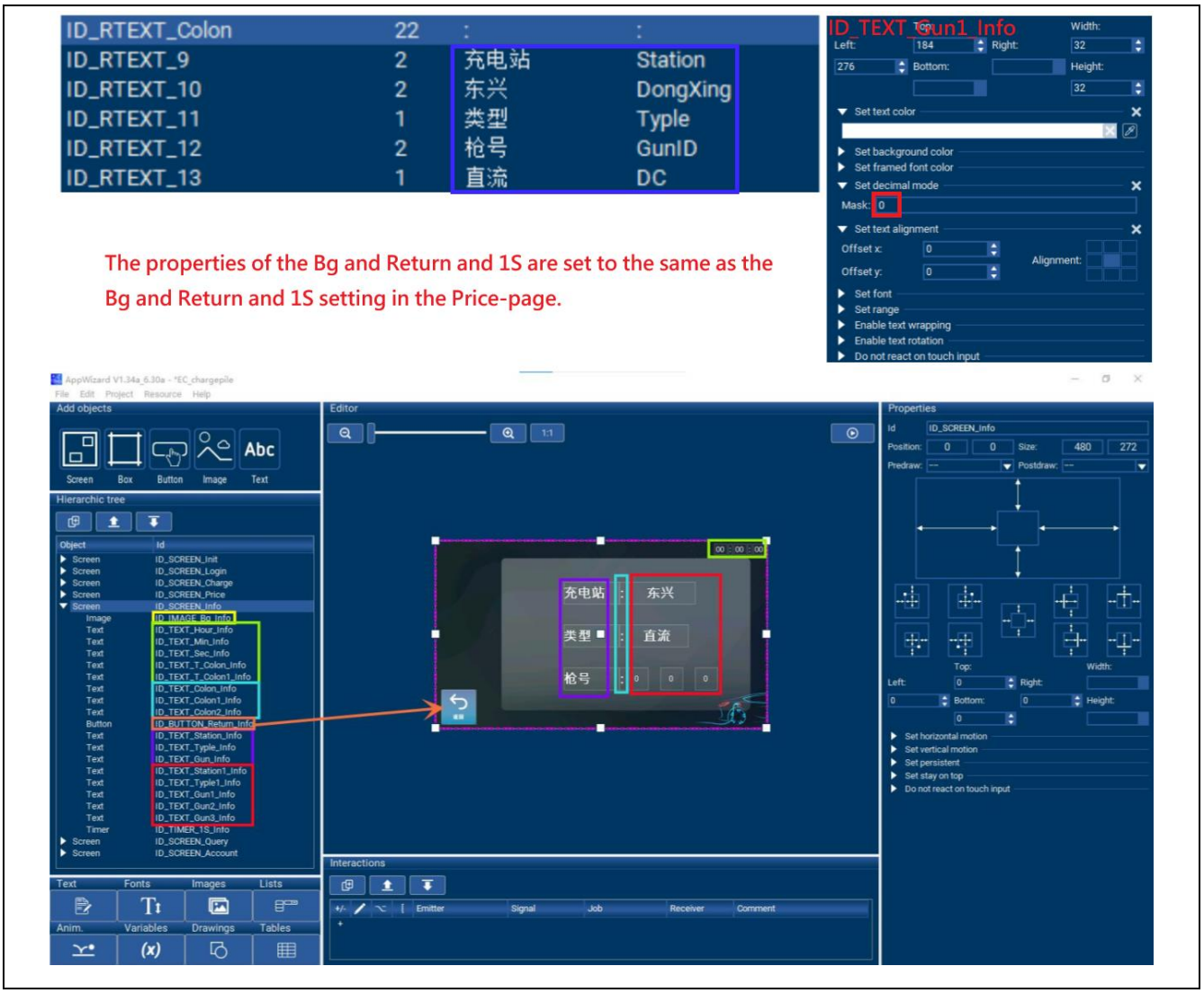


圖 1-33 添加所有物件並設置它們的相關屬性

6. Query 螢幕設計。主要包括最近一次充電時長、費用、充電站名稱及充電槍號顯示。

1) 充電資訊顯示。參考 Init 螢幕設計，添加所有物件並設置它們的相關屬性。

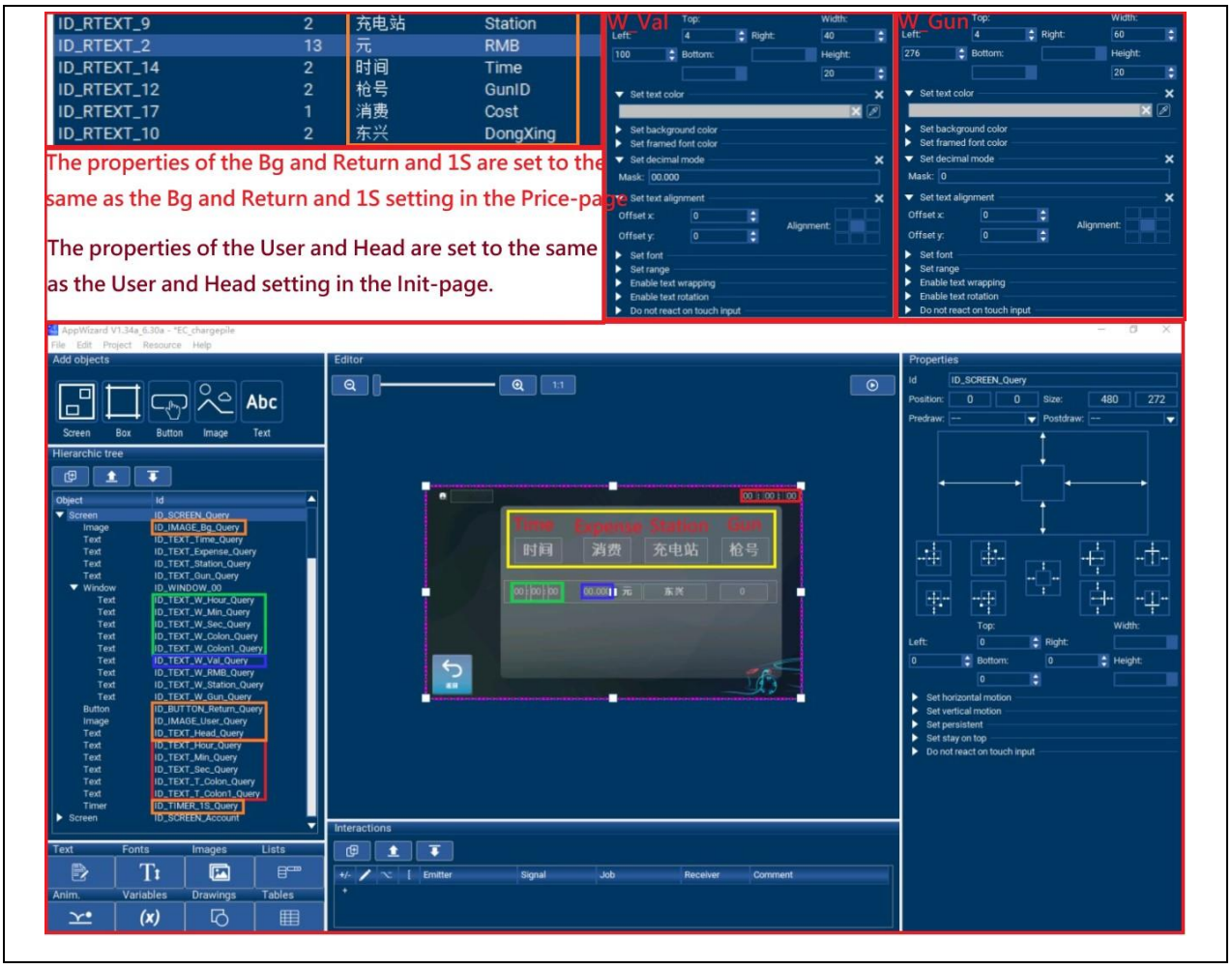


圖 1-34 添加所有物件並設置它們的相關屬性

7. Account 螢幕設計。主要包括顯示帳戶姓名、餘額及手機號碼。

1) 帳戶姓名、餘額、手機號碼顯示。參考 Init 螢幕設計，添加所有物件並設置它們的相關屬性。

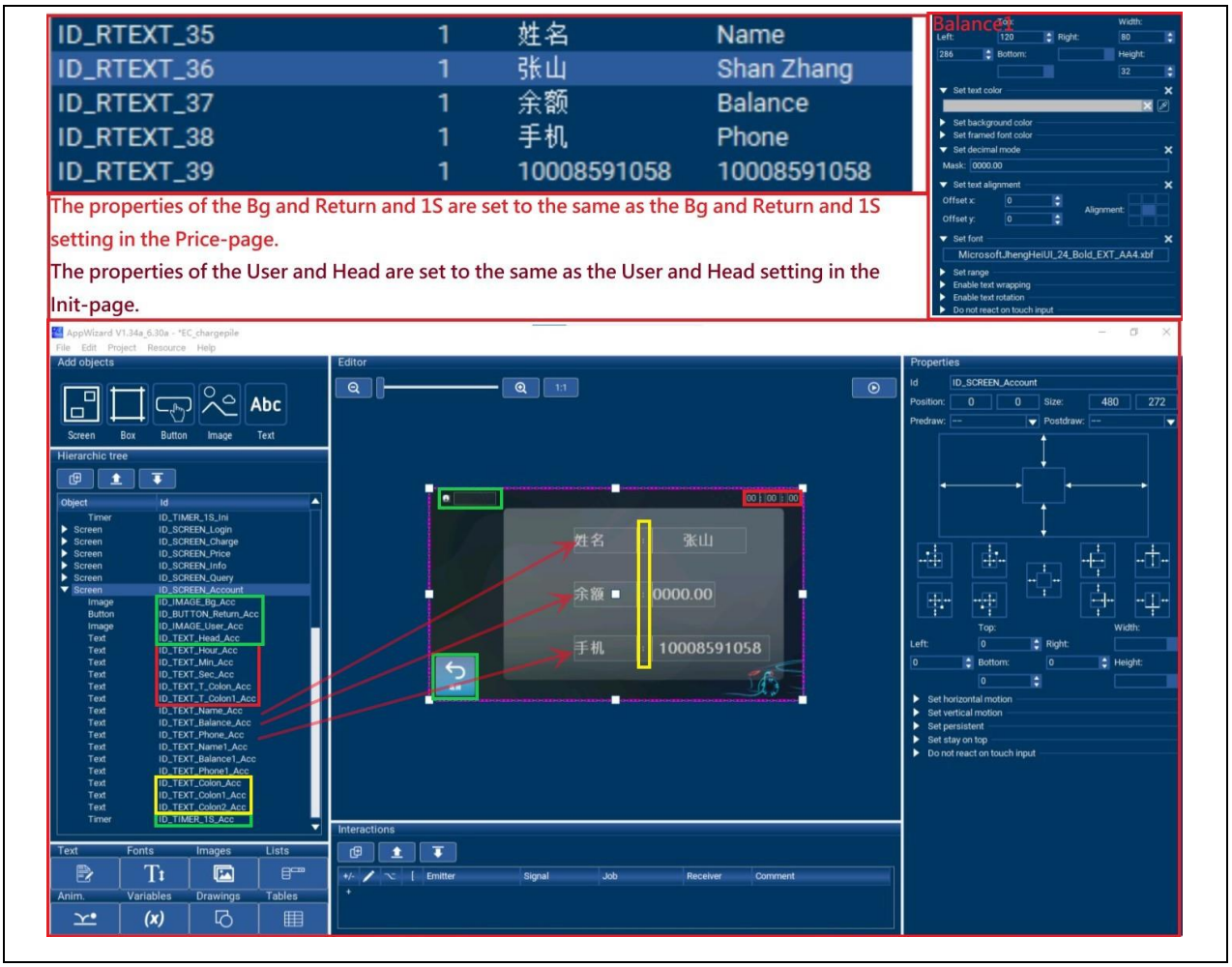


圖 1-35 添加所有物件並設置它們的相關屬性

8. Login 螢幕設計。主要包括接收用戶名、密碼輸入及顯示密碼狀態。

1) 用戶名、密碼輸入及密碼狀態顯示。參考 Init 螢幕設計，添加所有物件並設置它們的相關屬性。

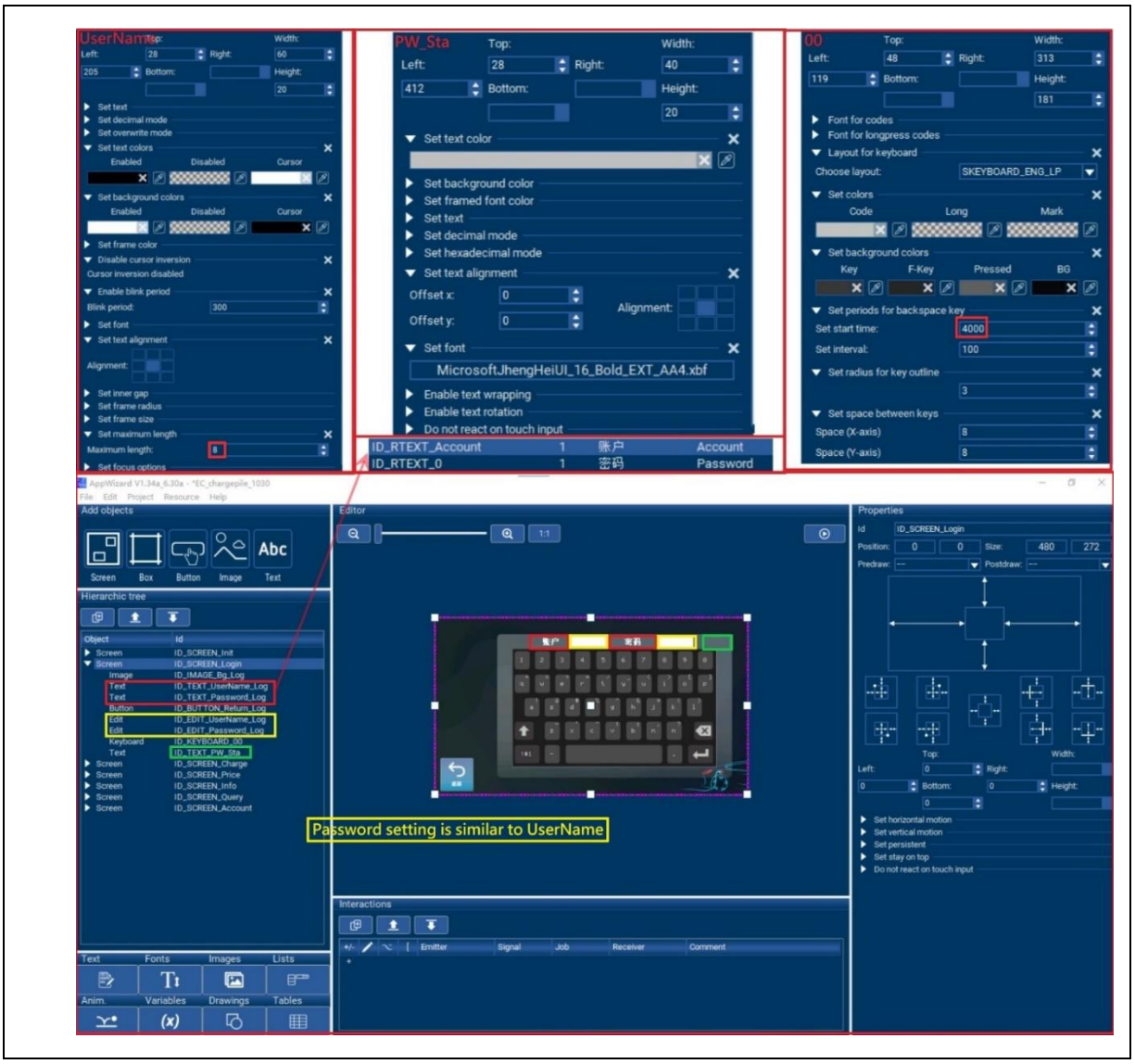


圖 1-36 添加所有物件並設置它們的相關屬性

1.2.3 在 AppWizard 工具中進行 Interactions 設定

1. Login 螢幕交互設計。包括按“Return”鍵返回 Init 螢幕；接收用戶名、密碼的輸入；結合代碼即時顯示密碼是否輸入正確。

   1) 按“Return”鍵返回 Init 螢幕。

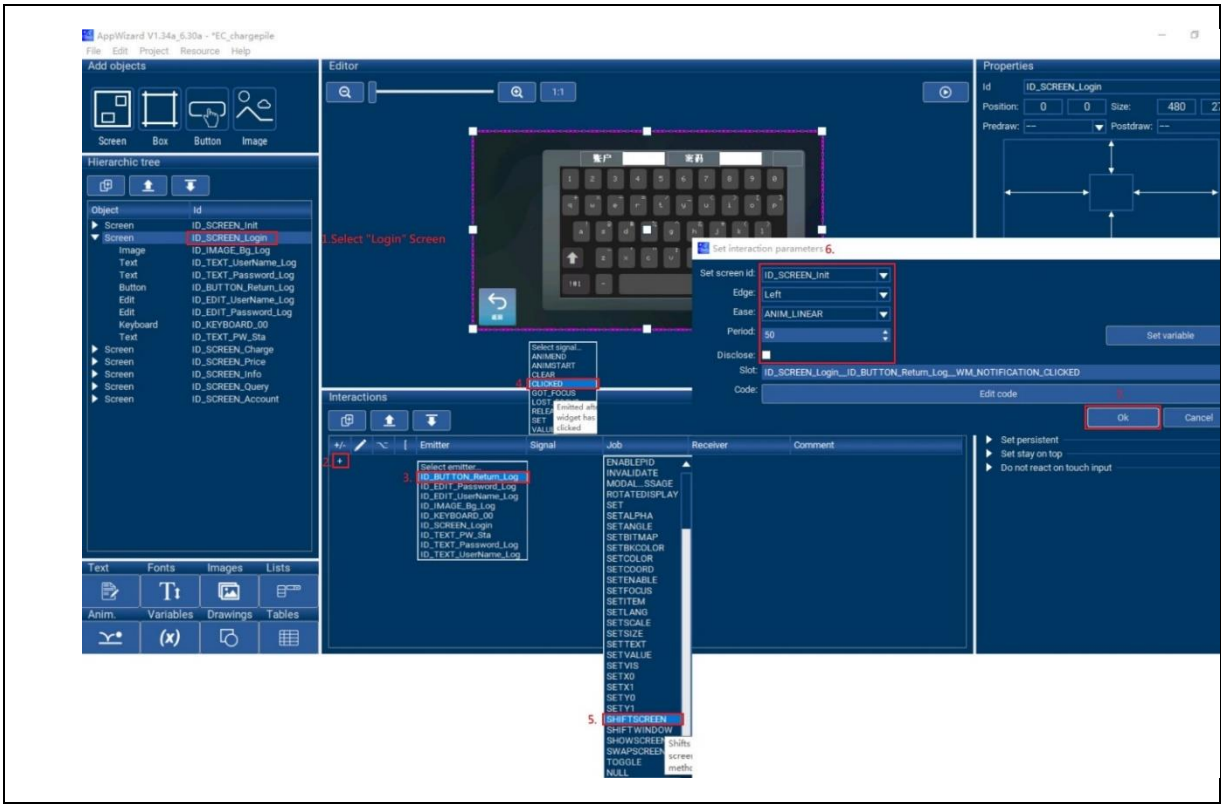


圖 1-37 按“Return”鍵返回 Init 螢幕

2) 載入 Login 螢幕，聚焦“UserName”物件。

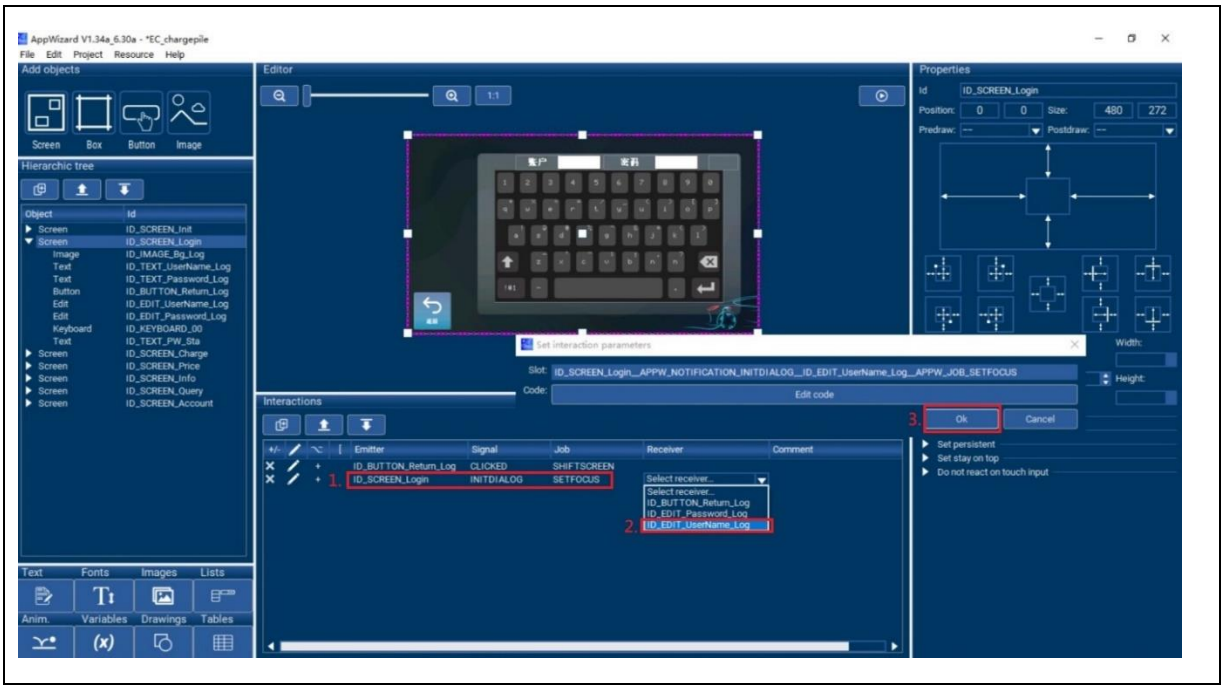


圖 1-38 載入 Login 螢幕聚焦“UserName”物件

3) 載入 Login 螢幕，根據“Lan”值的不同為 “Return”鍵設置不同的圖案。

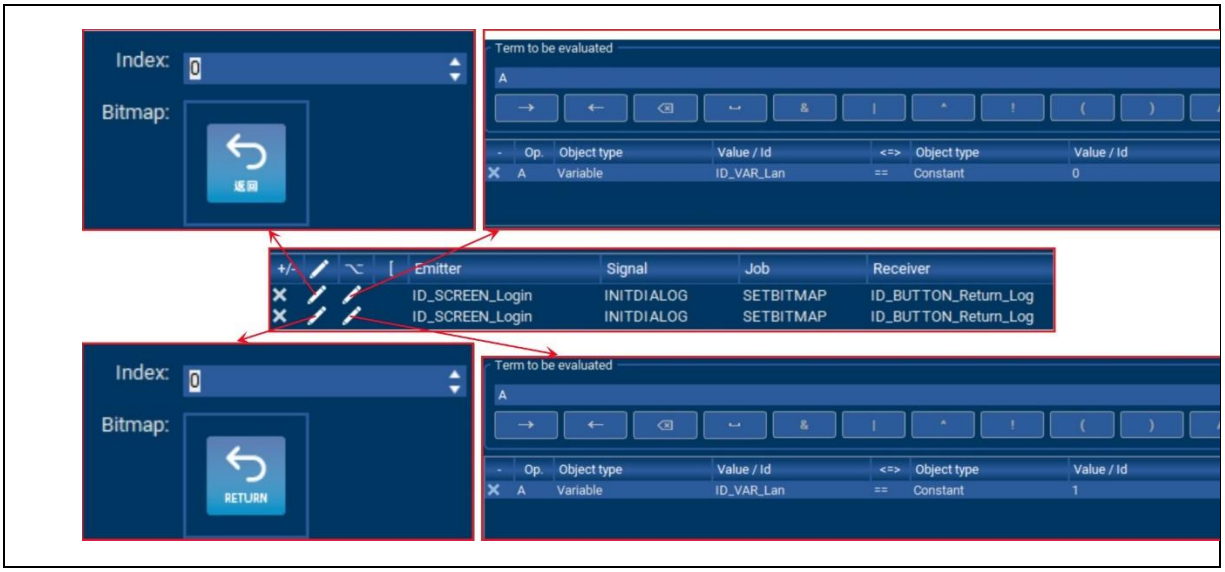


圖 1-39 載入 Login 螢幕為 “Return” 鍵設置不同的圖案

4) 定義密碼狀態變數“PW_1”。當輸入密碼正確的時候由代碼設置“PW_1”為 1，當輸入密碼錯誤的時候由代碼設置“PW_1”為 0。預設正確密碼為“ qqqqqqqq”。

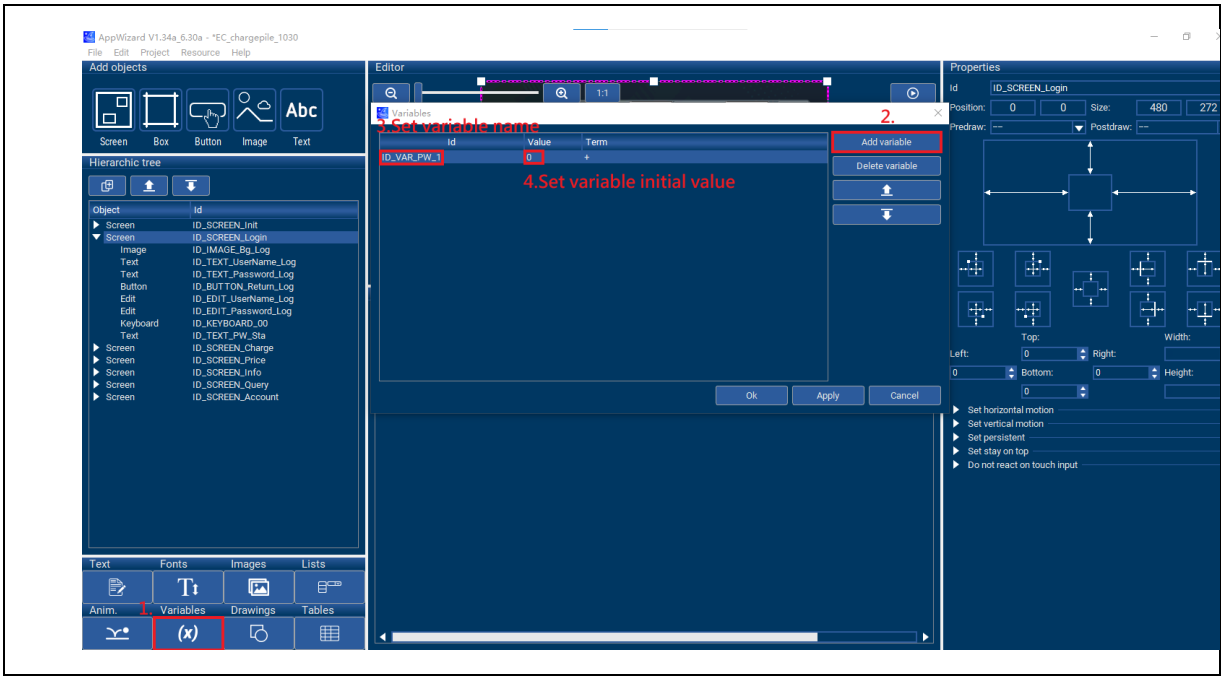


圖 1-40 “PW_1” 變數定義

5) 用戶名和密碼處理邏輯。

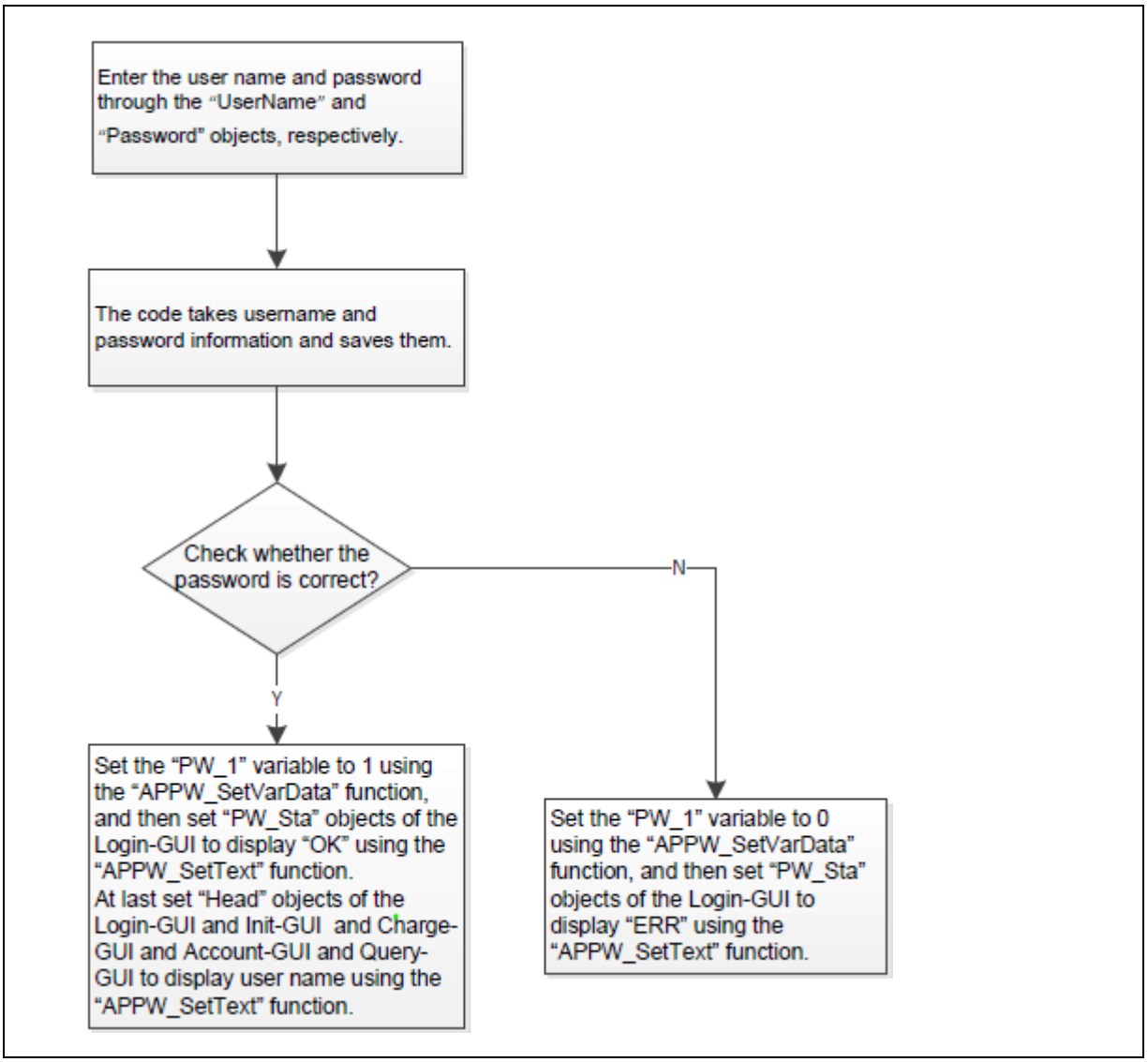


圖 1-41 用戶名和密碼處理邏輯

2. Init-GUI 交互設計。包括即時時鐘顯示；結合代碼進行用戶名顯示；按不同的鍵切換到不同的螢幕，按中英文切換鍵(“Lan”鍵)進行中英文切換；充電、帳戶、查詢鍵圖案根據“PW_1”變數的值進行不同的顯示。

1) 定義即時時鐘顯示所需變數(“Hour”、“Min”、“Sec”、“SecCnt”)。

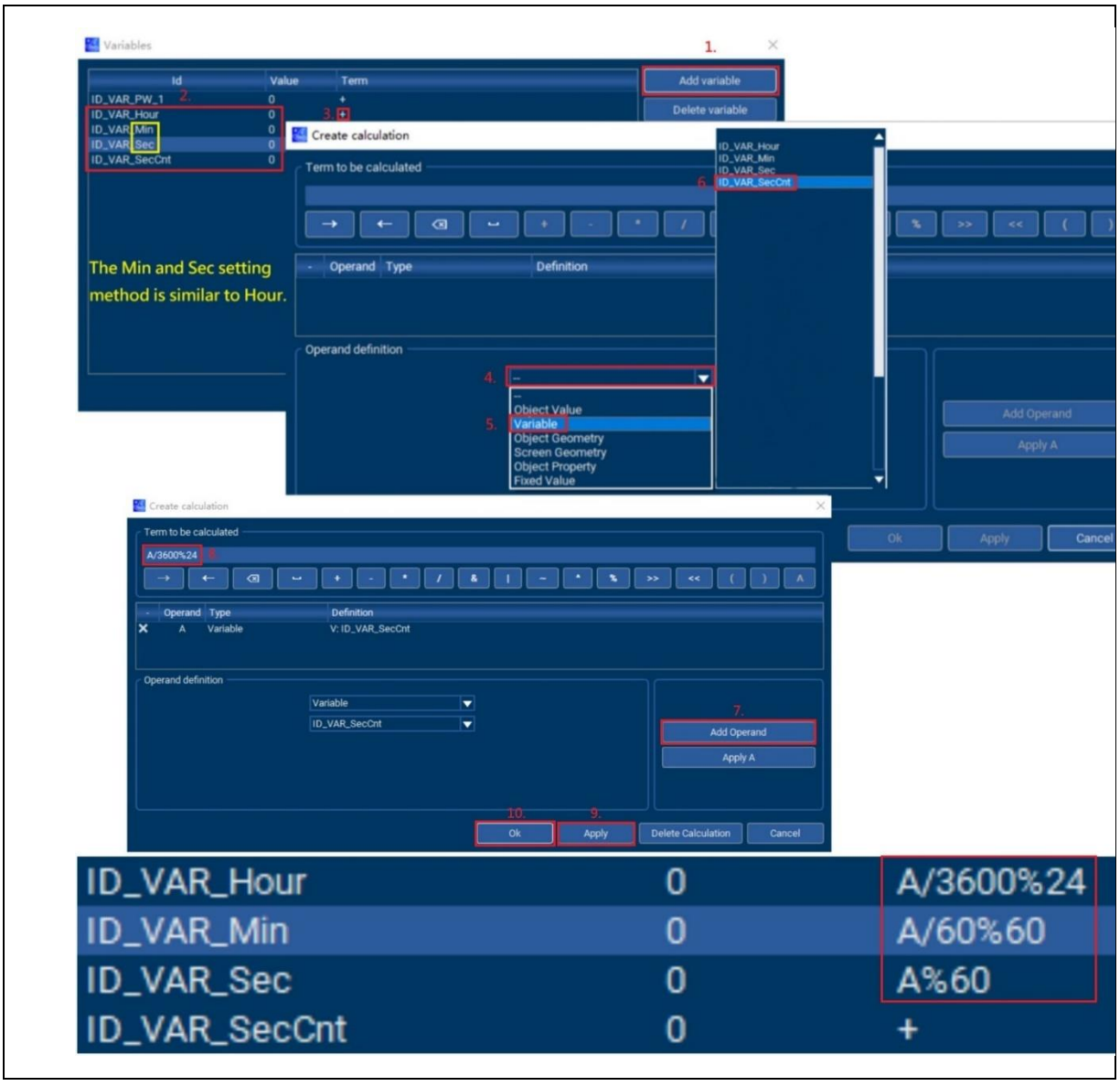


圖 1-42 即時時鐘相關變數定義

2) 即時時鐘顯示設計。上電啟動從 00 時 00 分 00 秒開始計時。

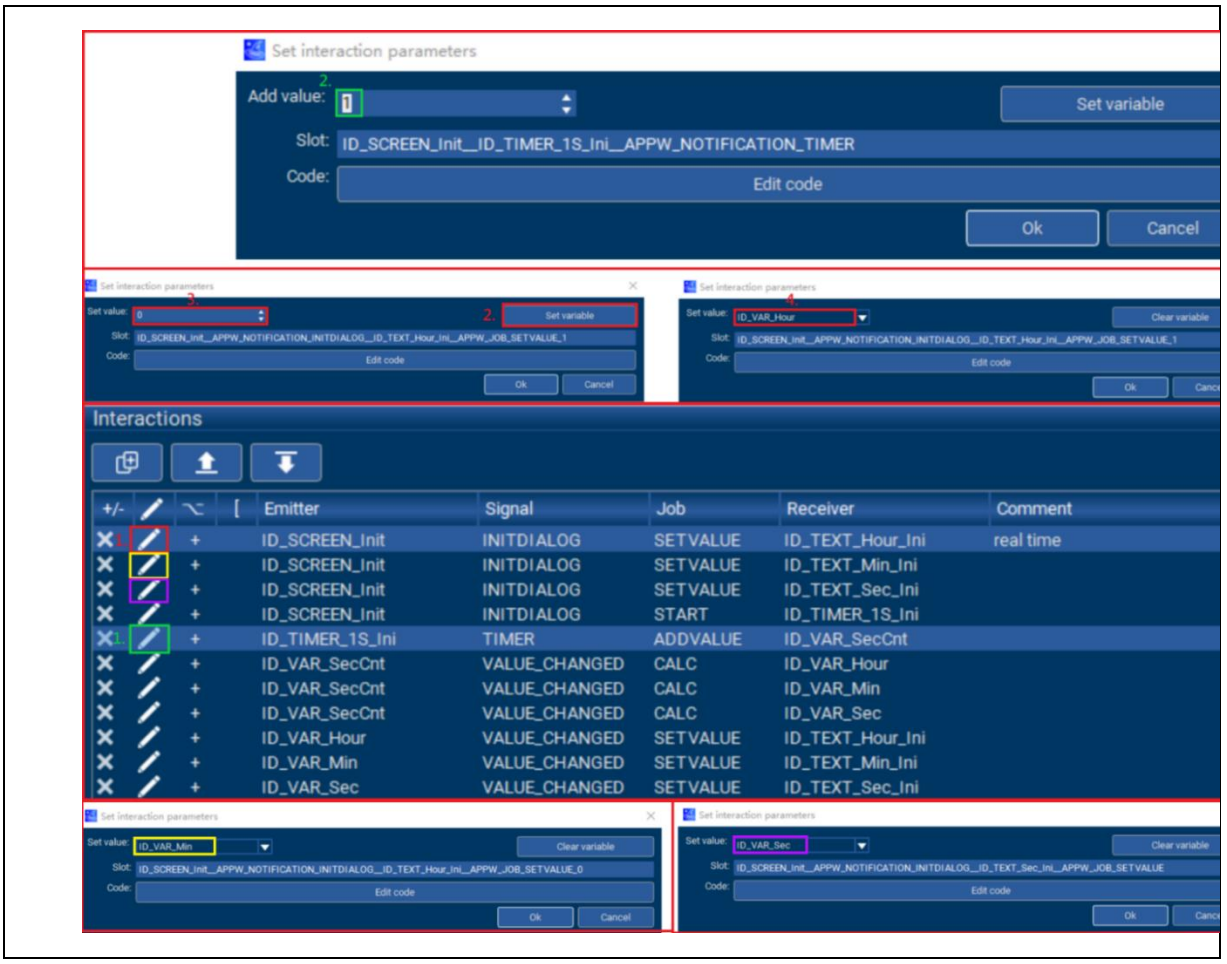


圖 1-43 即時時鐘顯示設計

3) 定義語言變數“Lan”。為 0 表示中文，為 1 表示英文。

圖 1-44 定義語言變數

4) 載入 Init 螢幕更新“Lan”鍵圖案。

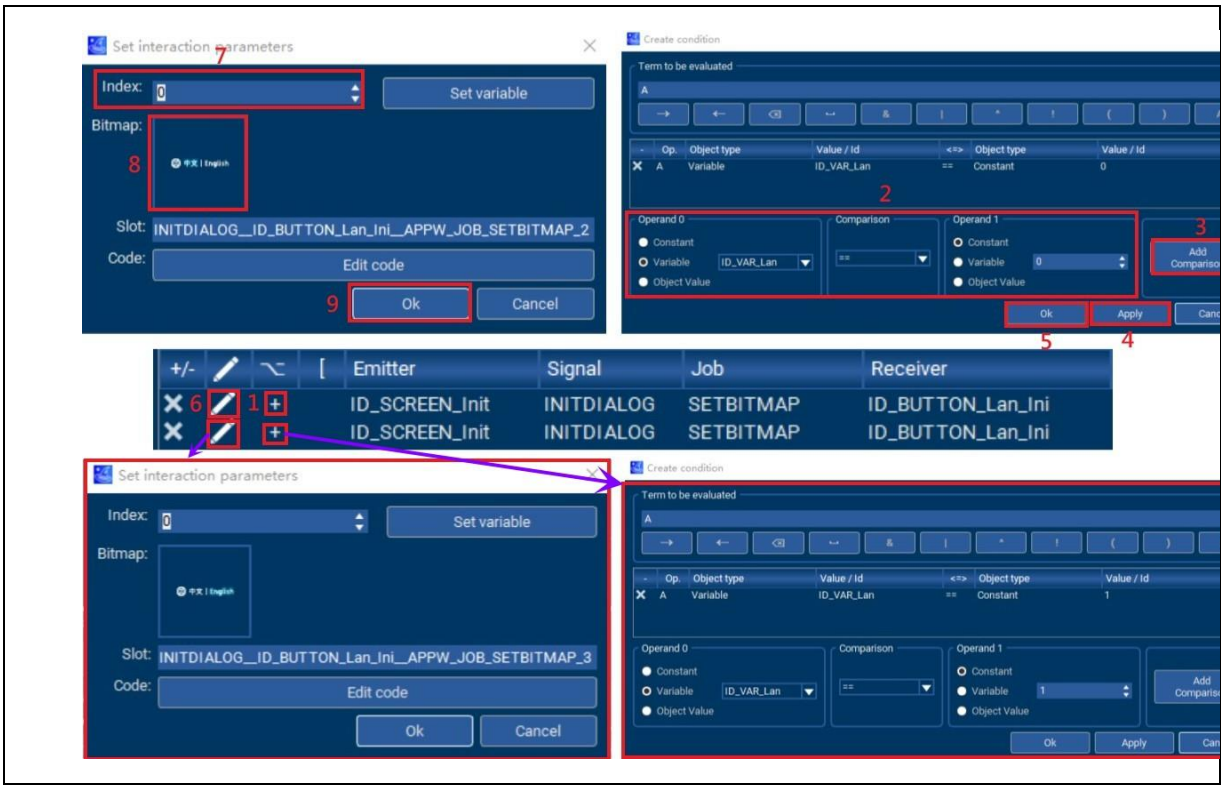


圖 1-45 載入 Init 螢幕更新“Lan”鍵圖案

5) 載入 Init 螢幕更新“Charge”鍵圖案。

圖 1-46 載入 Init 螢幕更新“Charge”鍵圖案

6) 載入 Init 螢幕更新背景“Bg”、 “Login” 鍵 、“Price” 鍵 、“Info” 鍵圖案，參考 1.2.3 章 2 節 4 步驟。載入 Init 螢幕更新“Acc”鍵 、“Query” 鍵 ，參考 1.2.3 章 2 節 5 步驟。

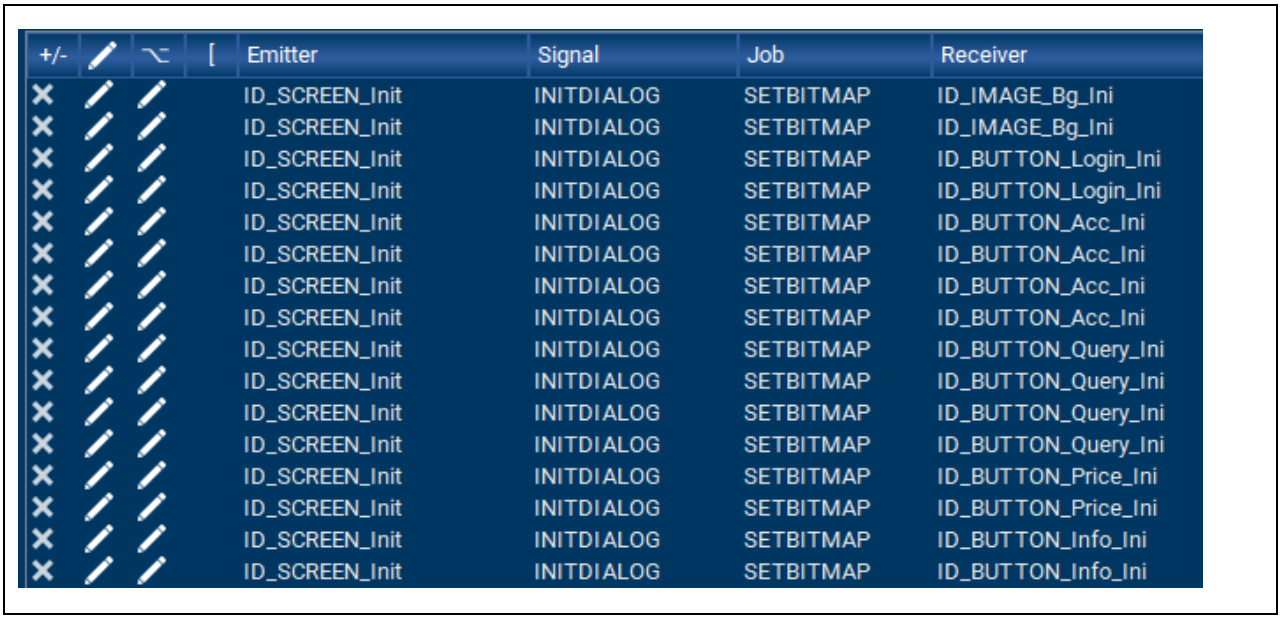


圖 1-47 載入 Init 螢幕更新“Bg ”、“Login” 、“Price” 、“Info” 、“Acc” 、“Query”圖案

7) 按“Lan”鍵改變“Lan”變數的值，值的範圍是 0 或 1。

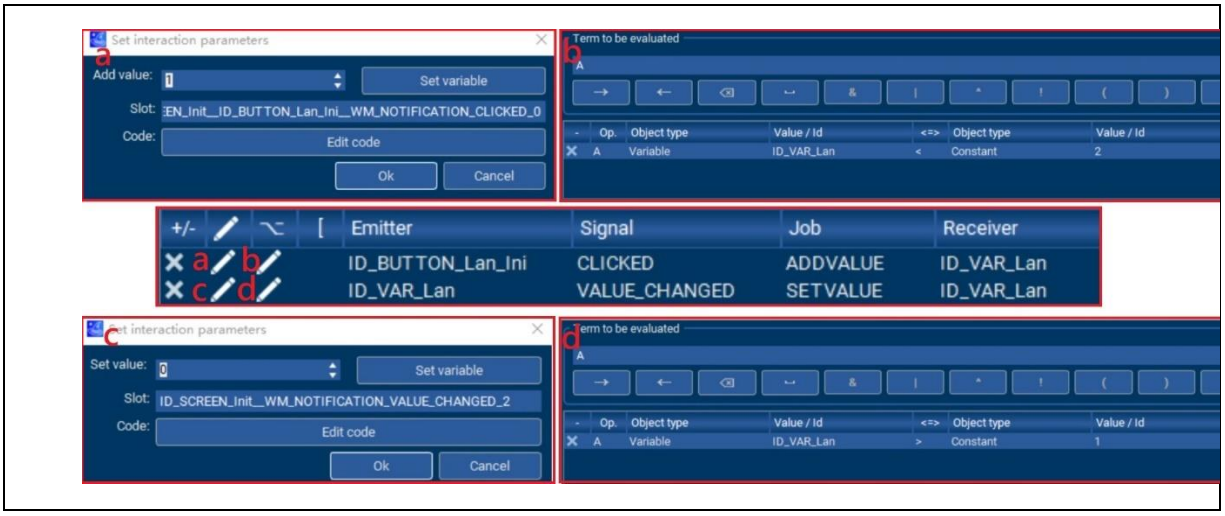


圖 1-48 按“Lan”鍵改變“Lan”變數的值

8) 按“Lan”鍵即時更新“Lan”鍵圖案。

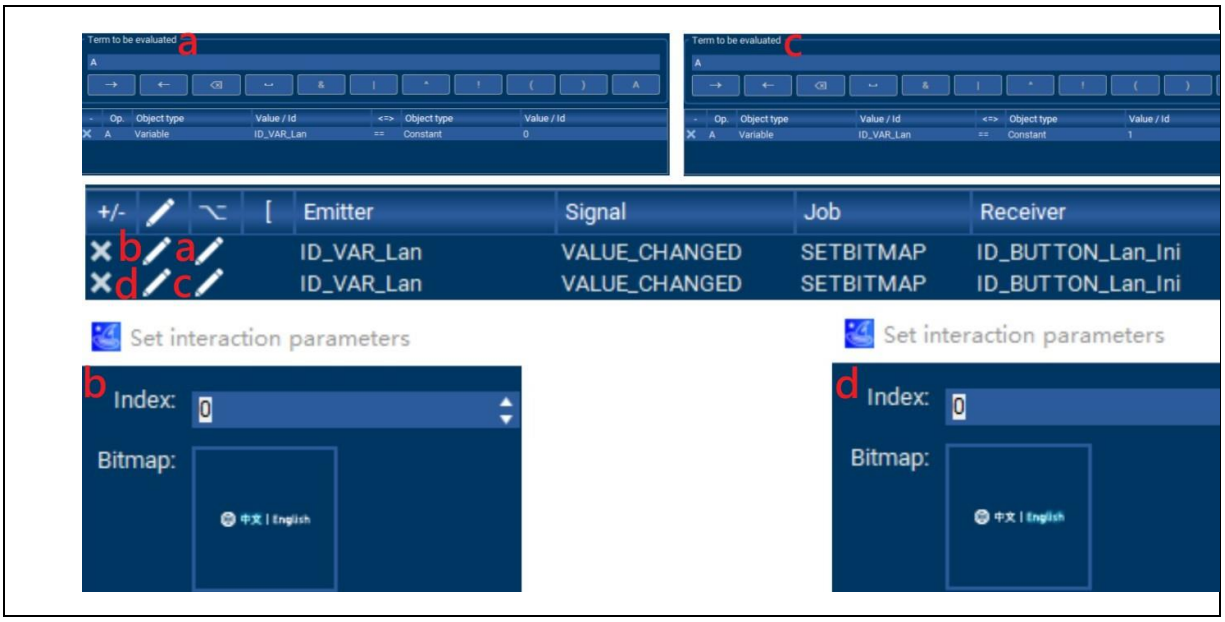


圖 1-49 按“Lan”鍵即時更新“Lan”鍵圖案

9) 設置“Lan”變數為 0 對應語言索引 0，“Lan”變數為 1 對應語言索引 1。

圖 1-50 設置“Lan”變數與語言索引的對應關係

10) 點擊“Login”、“Price”、“Info”、“Charge”、“Account”、“Query”鍵進行螢幕切換。

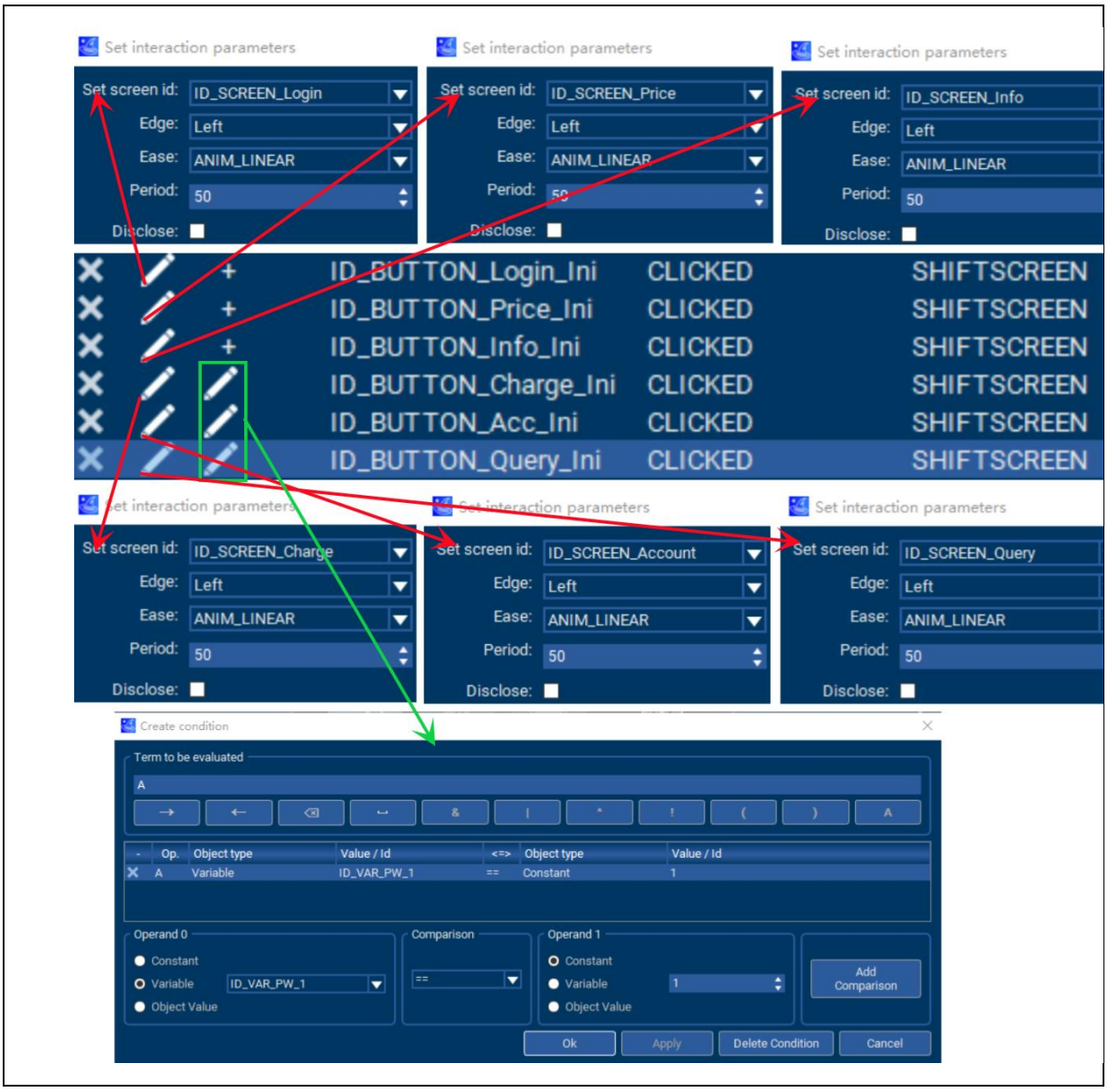


圖 1-51 點擊進行螢幕切換

11) “PW_1”變數變化結合 “Lan”變數的值，更新充電、帳戶、查詢鍵圖案。

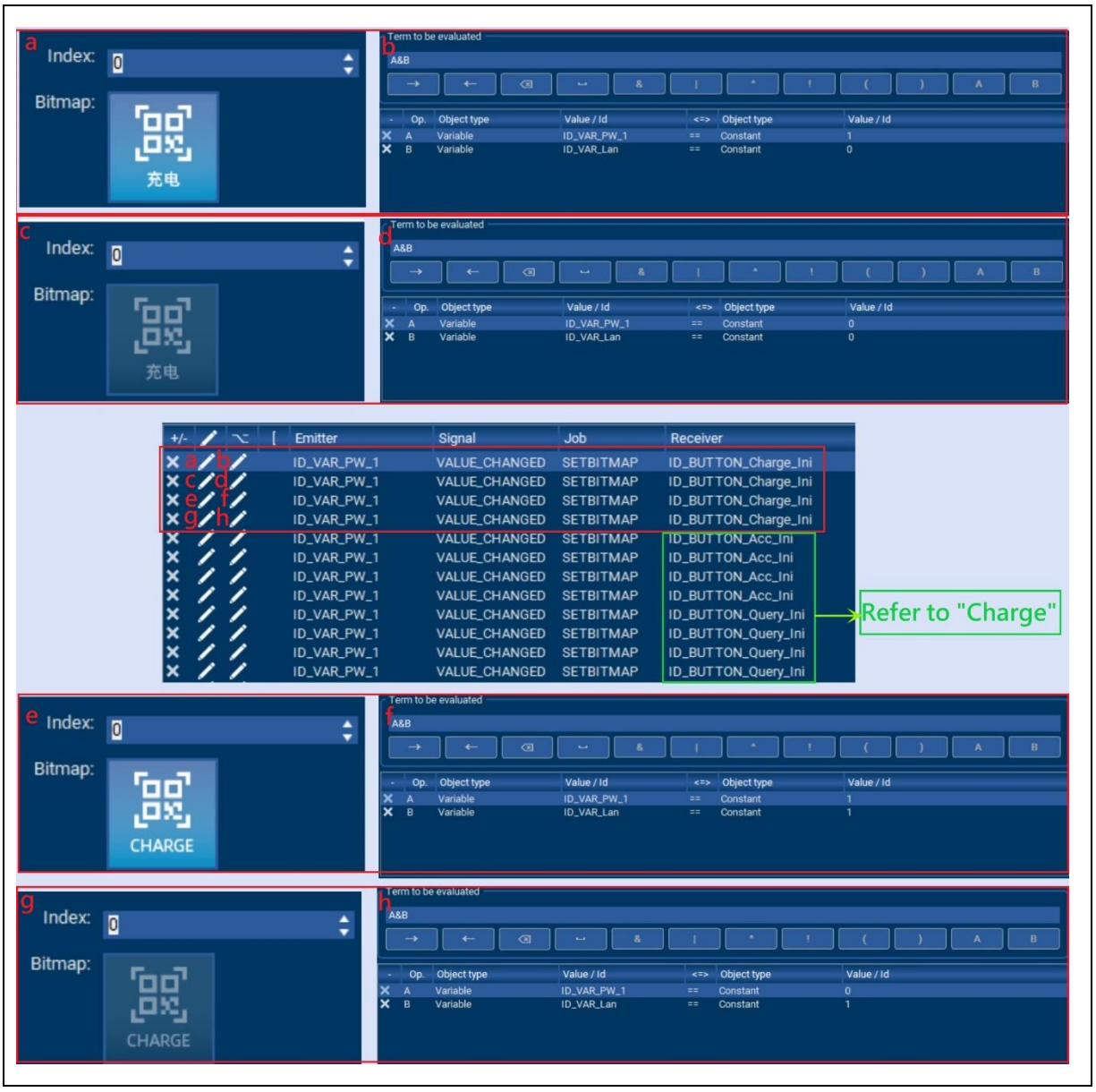


圖 1-52 充電、帳戶、查詢鍵圖案顯示

12) “Lan”變數變化結合 “PW_1”變數的值，更新登錄、充電、帳戶、查詢、價格、資訊鍵及背景圖案。

圖 1-53 各按鍵及背景圖案顯示

3. Charge 螢幕交互設計。主要包括按“Return”鍵返回 Init 螢幕；即時時鐘顯示；結合代碼實現用戶名顯示；實現顯示及控制充電狀態，包括充電時長顯示；電壓、電流、功率即時顯示；當前電價顯示；總充電電量及總充電費用以及尖、峰、平、谷各時間段充電電量及各時間段充電費用顯示。充電百分比顯示；通過按鍵可以啟動充電或停止充電。

1) 定義充電變數“Charging”，為 0 表示停止充電，為 1 表示充電中。

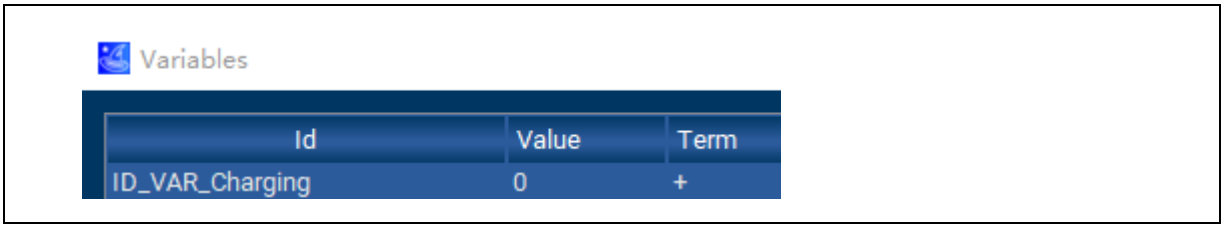


圖 1-54 定義充電變數“Charging”

2) 按“Return”鍵返回 Init 螢幕。

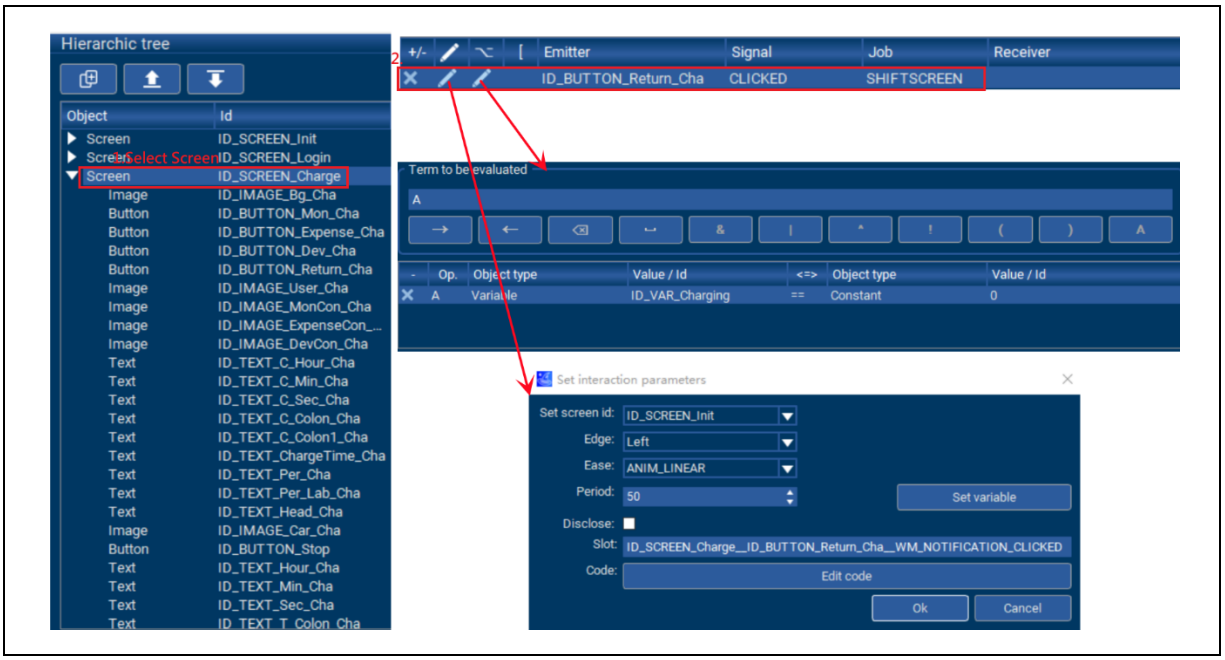


圖 1-55 按“Return”鍵返回 Init 螢幕

3) 即時時鐘顯示設計。參考 1.2.3 章 2 節 2 步驟。

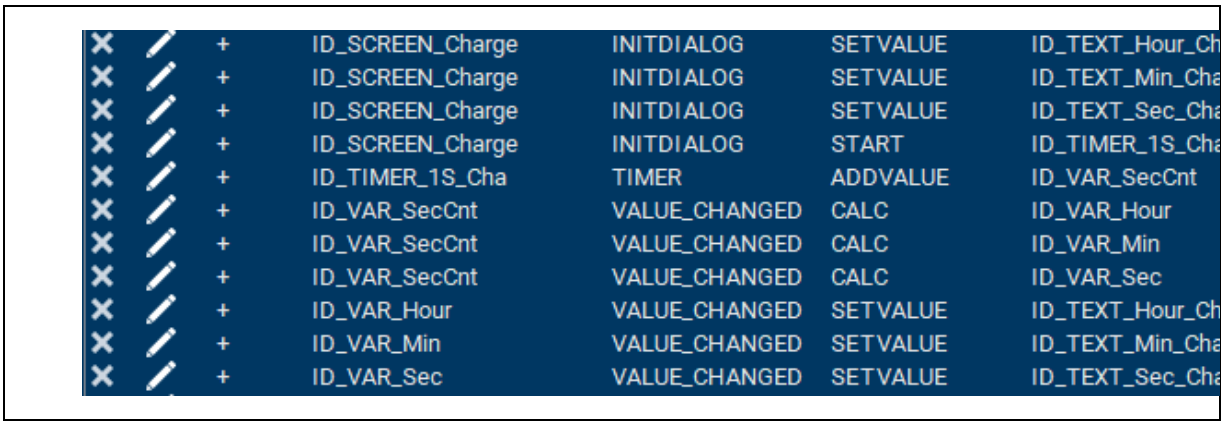


圖 1-56 即時時鐘顯示

4) 定義充電相關變數(“ConstantVolMode”、“Charge_Hour”、“Charge_Min”、“Charge_Sec”、“Charge_SecCnt”)。“ConstantVolMode”為 0 表示為恒流模式，“ConstantVolMode”為 1 表示為恒壓模式。

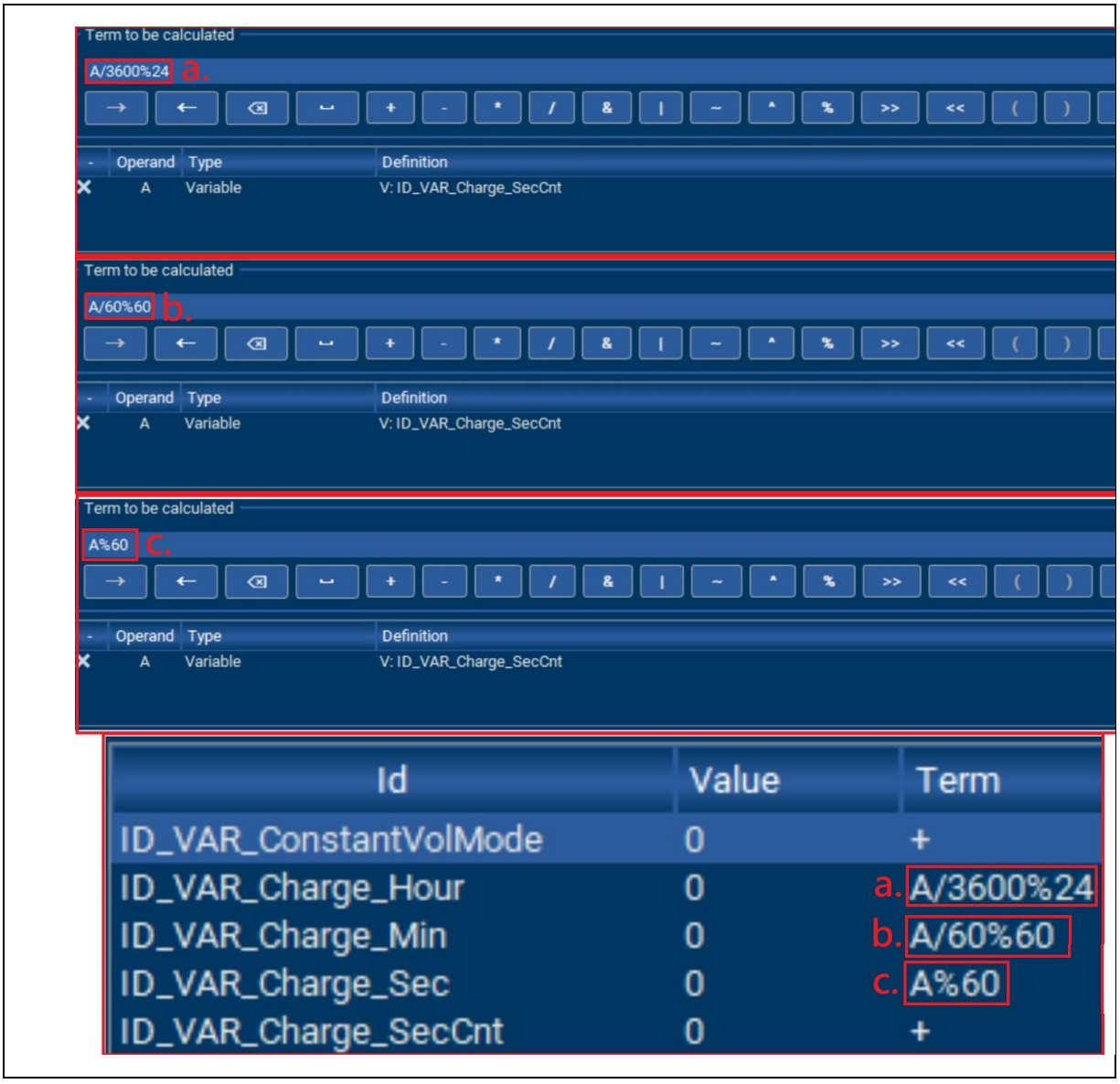


圖 1-57 定義充電相關變數

5) 充電時間顯示。參考 1.2.3 章 3 節 3 步驟。

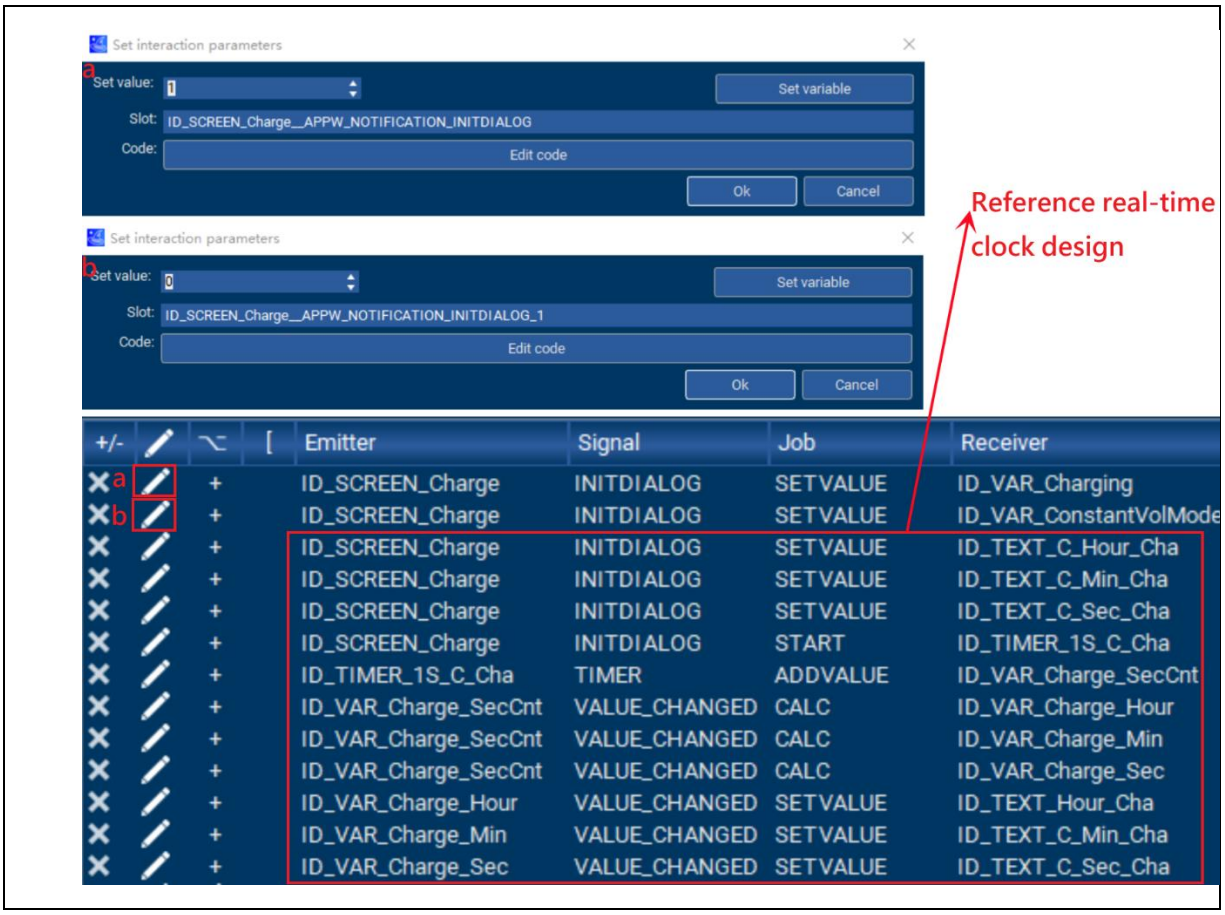


圖 1-58 充電時間顯示

6) 充電模式更新。“ChargeSecCnt”小於 60 為恒流模式，大於等於 60 為恒壓模式。

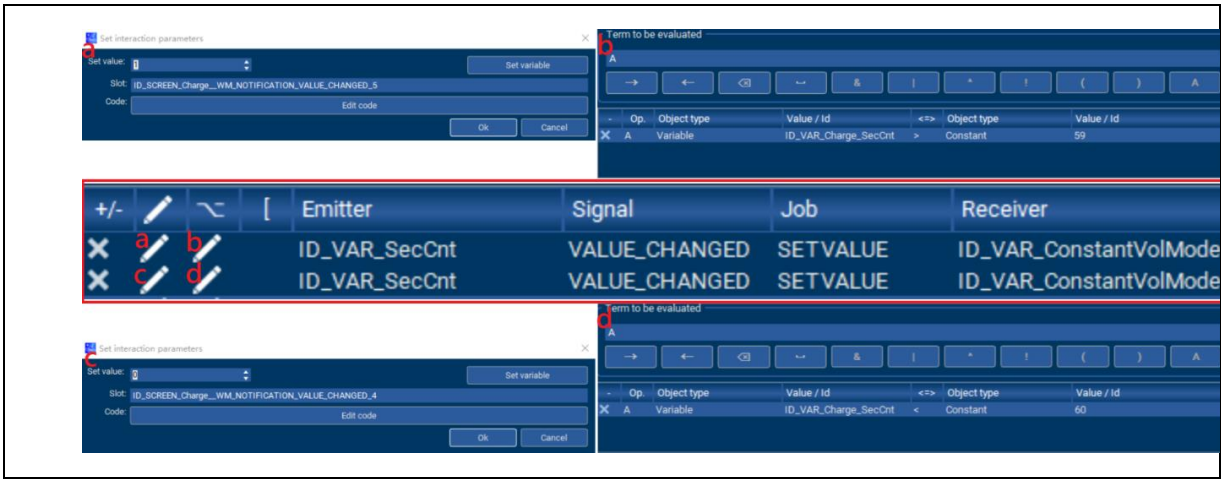


圖 1-59 充電模式更新

7) 定義當前價格變數 “CurPrice”。

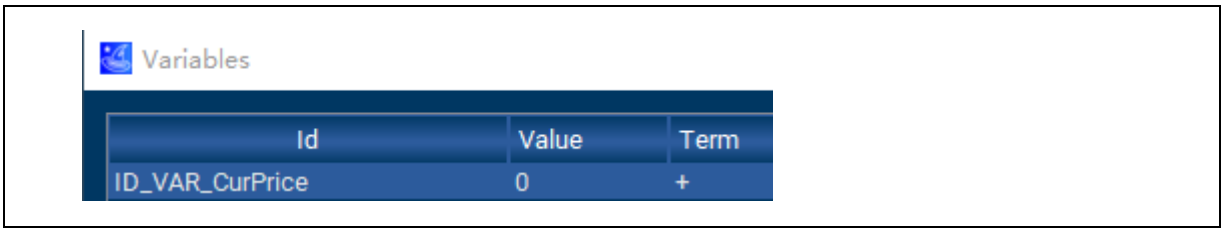


圖 1-60 當前價格變數定義

8) 當前價格更新。為了方便計算電量及費用，假定 1 秒相當於實際 4 分鐘。

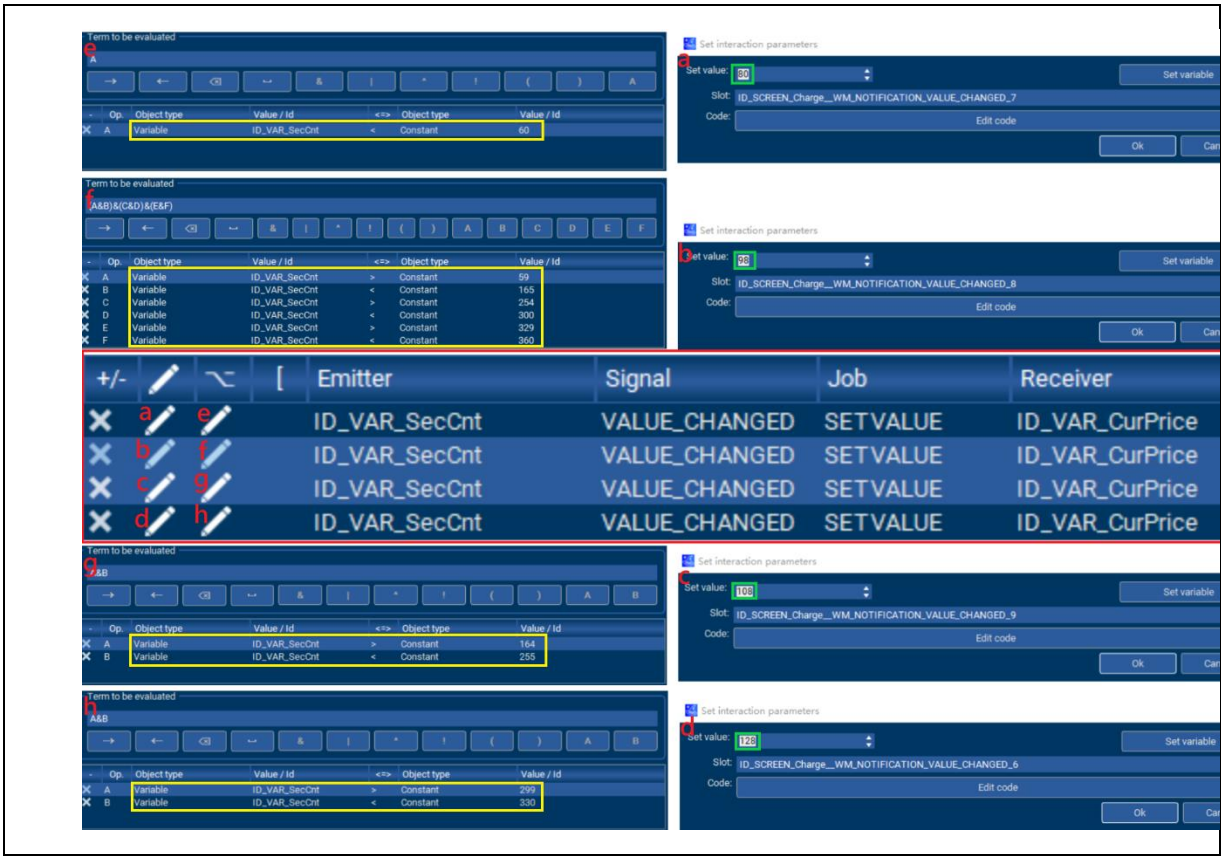


圖 1-61 當前價格更新設計

9) 定義電流、電壓變數(“Cur”、“Voltage”)。初值分別為 30A、281V。恒流模式每隔 4 分鐘電壓升高 1.16V；恒壓模式每隔 4 分鐘電流變小 1.8A。

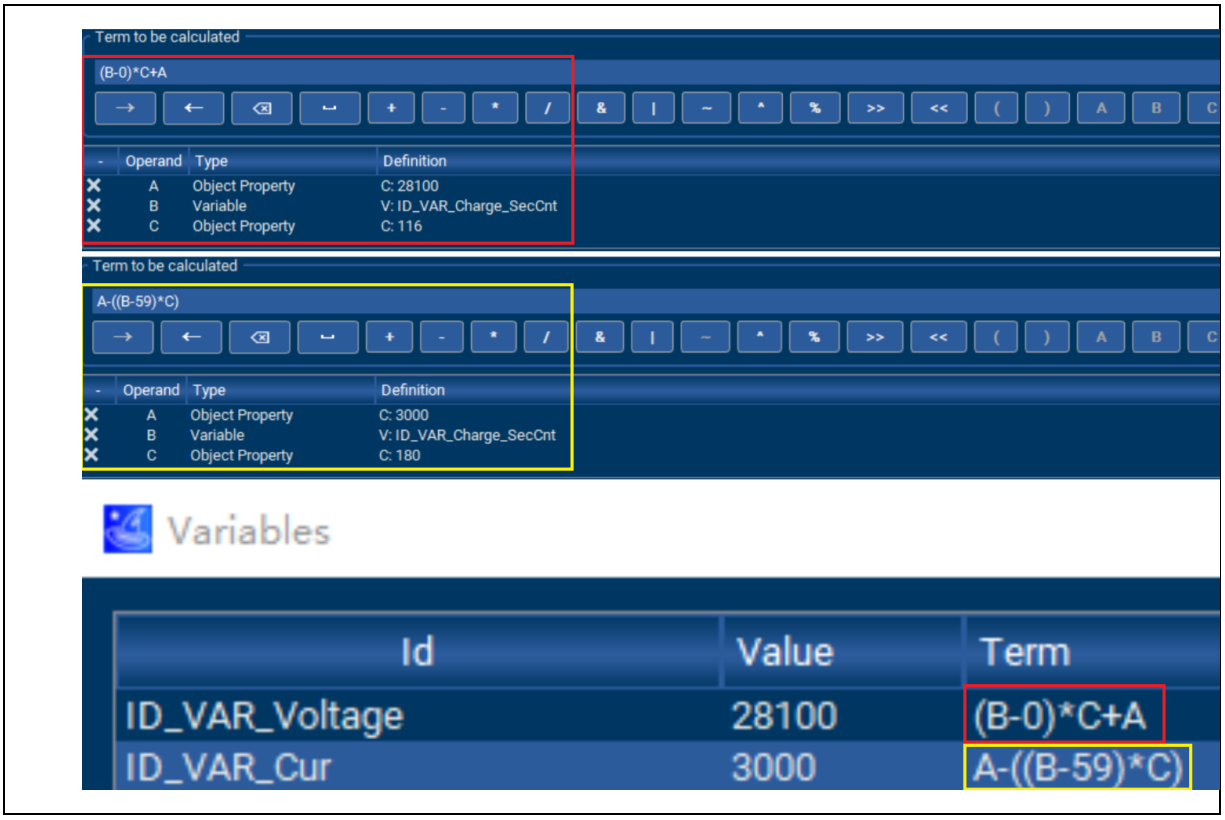


圖 1-62 電流、電壓變數定義

10) 電流、電壓顯示。

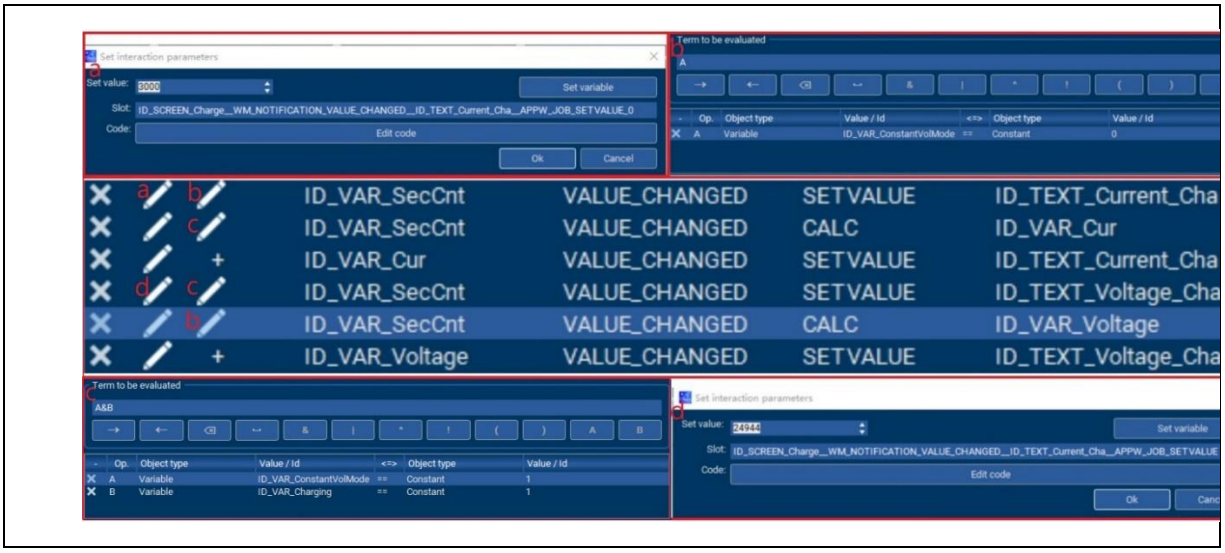


圖 1-63 電流、電壓顯示

11) 定義即時電量(“RealCapacity”)、即時消費(“RealCost”)及累計電量(“SumCapacity”)、累計消費(“SumCost”)變數。

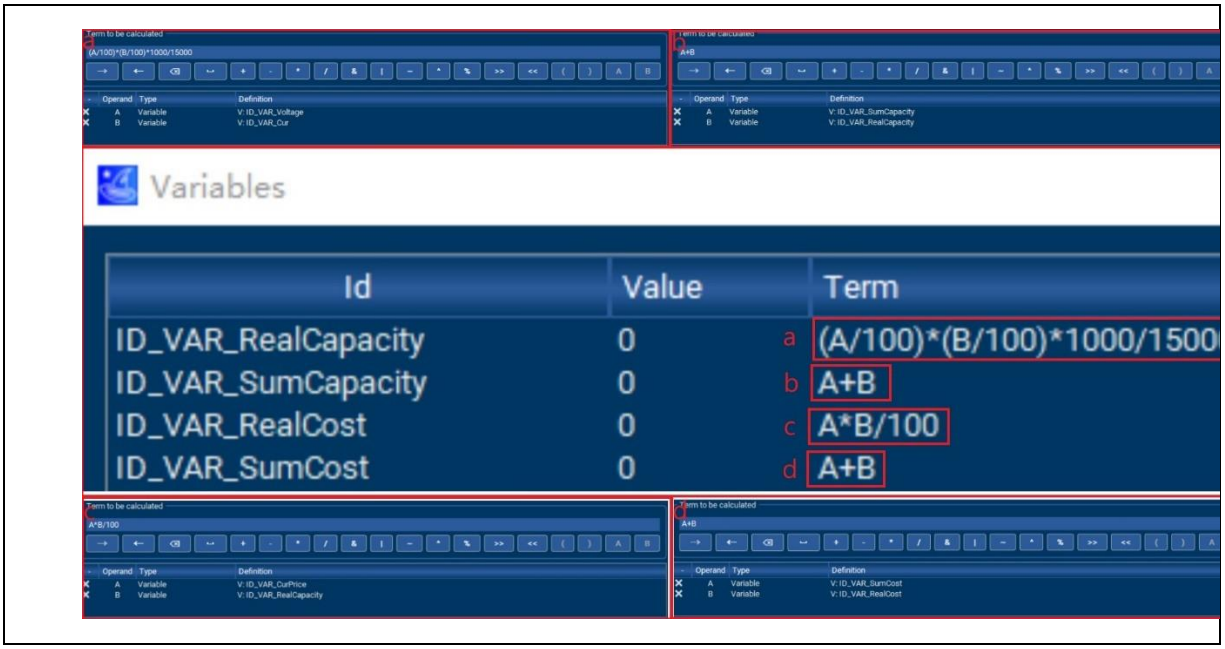


圖 1-64 定義即時電量、即時消費和累計電量、累計消費

12) 計算即時電量、即時消費及累計電量、累計消費，並顯示累計電量和累計消費。

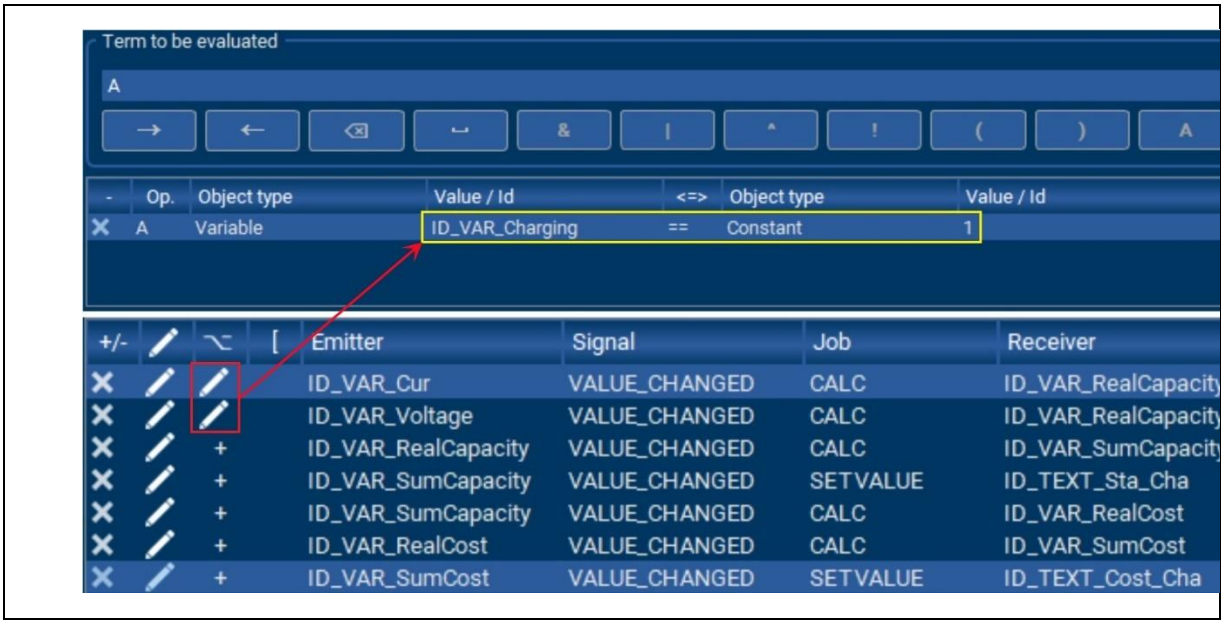


圖 1-65 計算即時電量、即時消費及累計電量、累計消費

13) 停止充電，包括充滿停止充電和按“Stop”鍵停止充電。

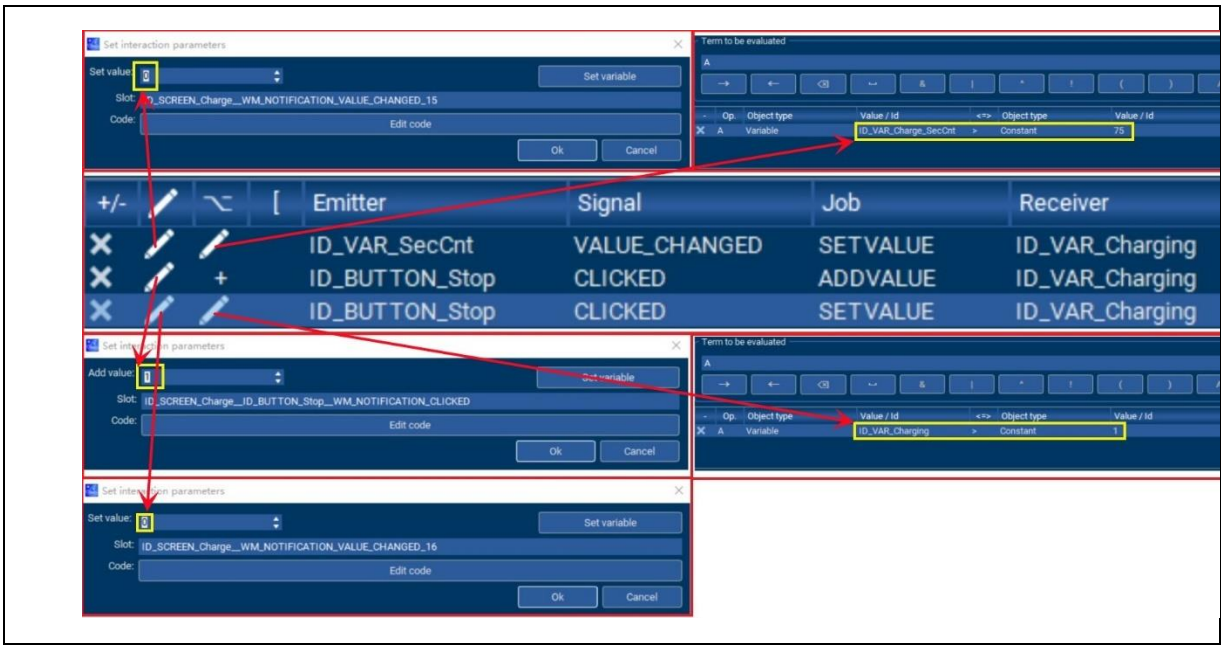


圖 1-66 停止充電

14) 啟動、停止充電的相關動作。

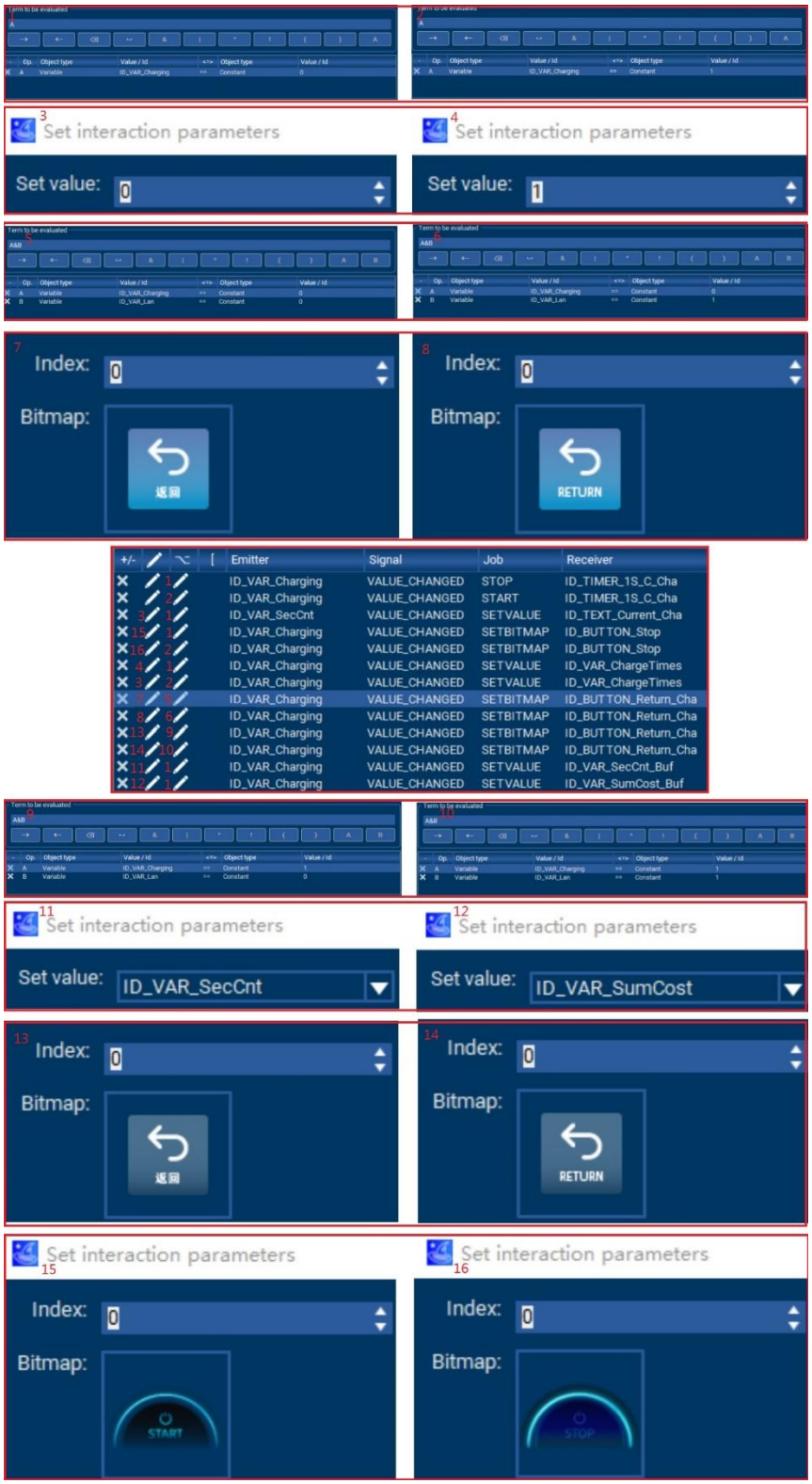


圖 1-67 啟動、停止充電的相關動作

15) 載入 Charge 螢幕，預設顯示監控視窗，並進入充電狀態。

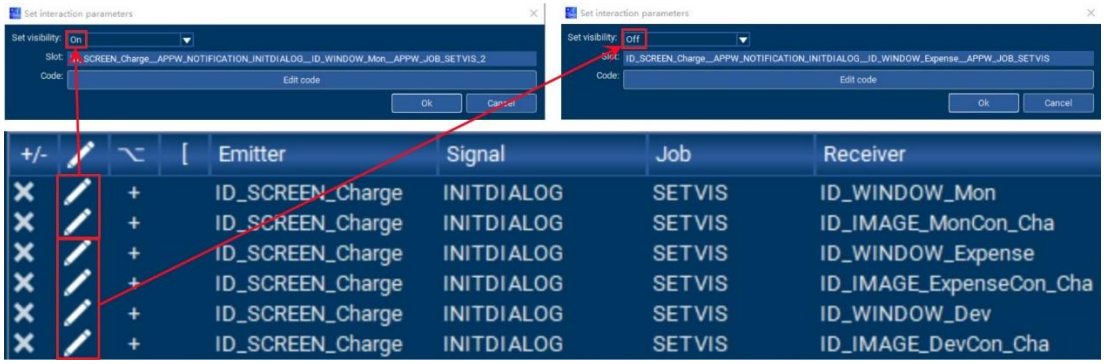


圖 1-68 載入 Charge 螢幕預設顯示監控視窗

16) 監控鍵按下顯示監控視窗資訊，消費、設備視窗資訊不可見。

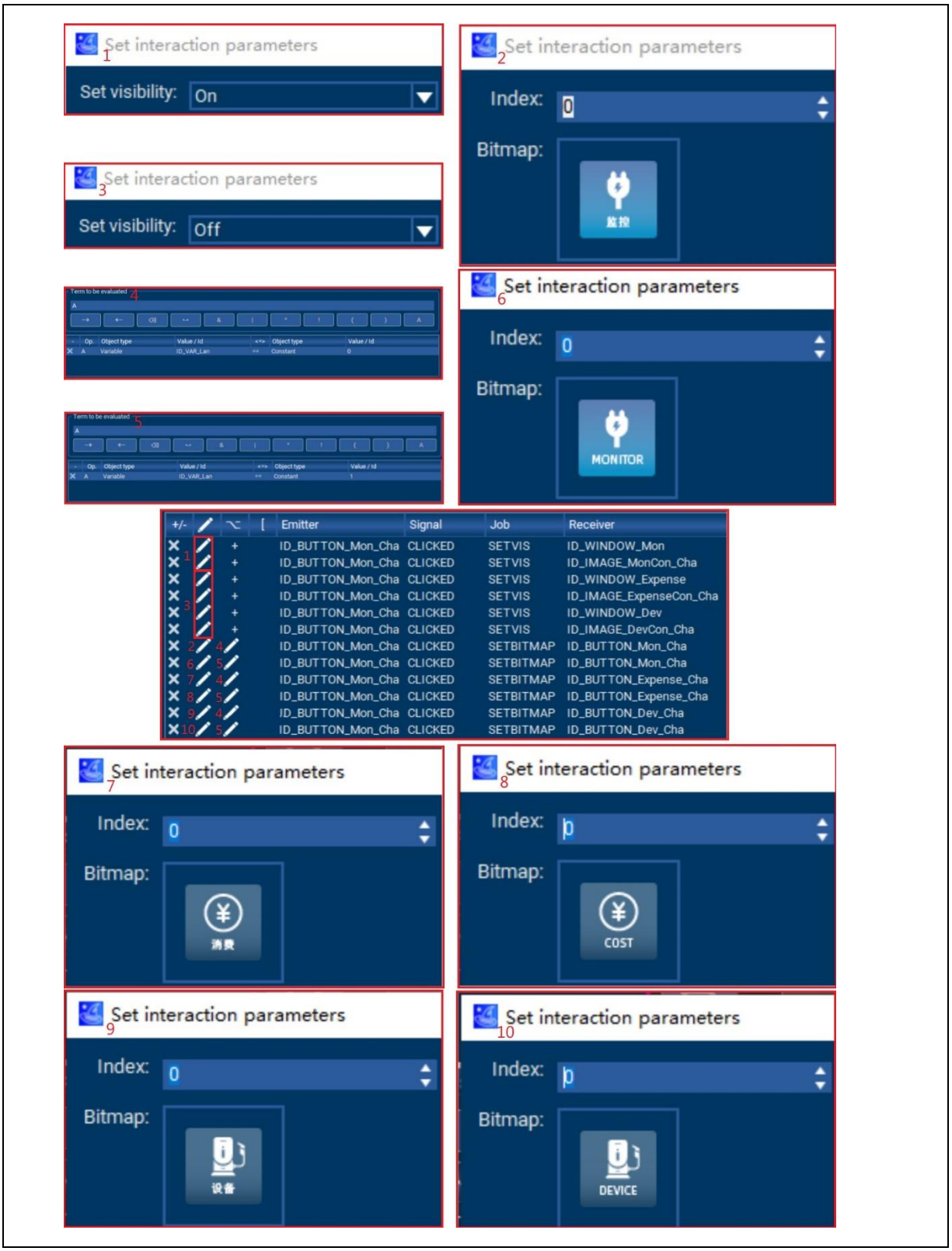
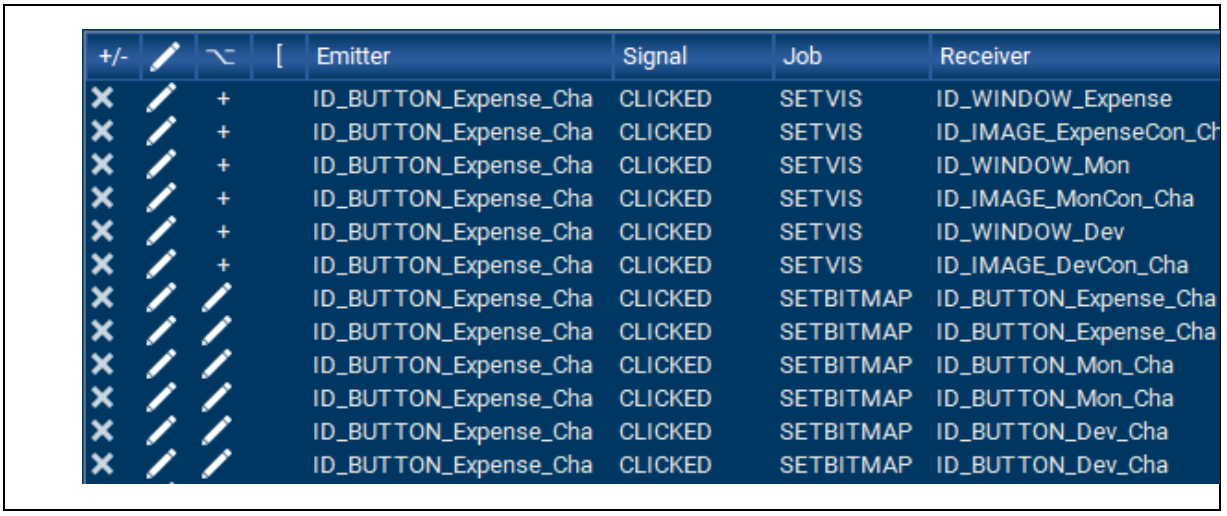


圖 1-69 監控鍵按下

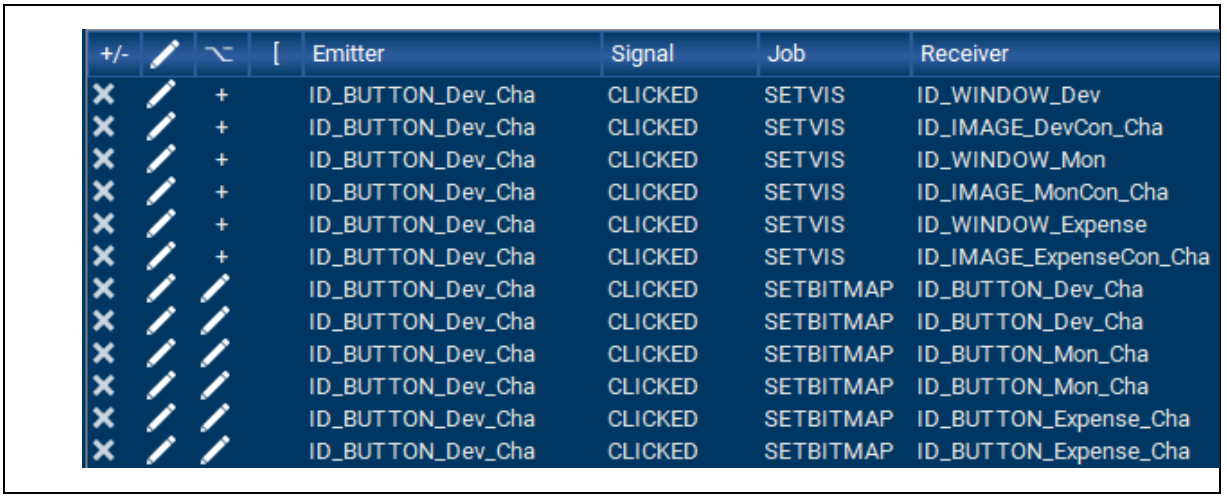
17) 消費鍵按下顯示消費視窗資訊，監控、設備視窗資訊不可見。參考 1.2.2 章 3 節 16 步驟。



+/-		⌥	[	Emitter	Signal	Job	Receiver
✕	✎	+		ID_BUTTON_Expense_Cha	CLICKED	SETVIS	ID_WINDOW_Expense
✕	✎	+		ID_BUTTON_Expense_Cha	CLICKED	SETVIS	ID_IMAGE_ExpenseCon_Ch
✕	✎	+		ID_BUTTON_Expense_Cha	CLICKED	SETVIS	ID_WINDOW_Mon
✕	✎	+		ID_BUTTON_Expense_Cha	CLICKED	SETVIS	ID_IMAGE_MonCon_Cha
✕	✎	+		ID_BUTTON_Expense_Cha	CLICKED	SETVIS	ID_WINDOW_Dev
✕	✎	+		ID_BUTTON_Expense_Cha	CLICKED	SETVIS	ID_IMAGE_DevCon_Cha
✕	✎	✎		ID_BUTTON_Expense_Cha	CLICKED	SETBITMAP	ID_BUTTON_Expense_Cha
✕	✎	✎		ID_BUTTON_Expense_Cha	CLICKED	SETBITMAP	ID_BUTTON_Expense_Cha
✕	✎	✎		ID_BUTTON_Expense_Cha	CLICKED	SETBITMAP	ID_BUTTON_Mon_Cha
✕	✎	✎		ID_BUTTON_Expense_Cha	CLICKED	SETBITMAP	ID_BUTTON_Mon_Cha
✕	✎	✎		ID_BUTTON_Expense_Cha	CLICKED	SETBITMAP	ID_BUTTON_Dev_Cha
✕	✎	✎		ID_BUTTON_Expense_Cha	CLICKED	SETBITMAP	ID_BUTTON_Dev_Cha

圖 1-70 消費鍵按下

18) 設備鍵按下顯示裝置視窗資訊，消費、監控視窗資訊不可見。參考 1.2.2 章 3 節 16 步驟。



+/-		⌥	[	Emitter	Signal	Job	Receiver
✕	✎	+		ID_BUTTON_Dev_Cha	CLICKED	SETVIS	ID_WINDOW_Dev
✕	✎	+		ID_BUTTON_Dev_Cha	CLICKED	SETVIS	ID_IMAGE_DevCon_Cha
✕	✎	+		ID_BUTTON_Dev_Cha	CLICKED	SETVIS	ID_WINDOW_Mon
✕	✎	+		ID_BUTTON_Dev_Cha	CLICKED	SETVIS	ID_IMAGE_MonCon_Cha
✕	✎	+		ID_BUTTON_Dev_Cha	CLICKED	SETVIS	ID_WINDOW_Expense
✕	✎	+		ID_BUTTON_Dev_Cha	CLICKED	SETVIS	ID_IMAGE_ExpenseCon_Cha
✕	✎	✎		ID_BUTTON_Dev_Cha	CLICKED	SETBITMAP	ID_BUTTON_Dev_Cha
✕	✎	✎		ID_BUTTON_Dev_Cha	CLICKED	SETBITMAP	ID_BUTTON_Dev_Cha
✕	✎	✎		ID_BUTTON_Dev_Cha	CLICKED	SETBITMAP	ID_BUTTON_Mon_Cha
✕	✎	✎		ID_BUTTON_Dev_Cha	CLICKED	SETBITMAP	ID_BUTTON_Mon_Cha
✕	✎	✎		ID_BUTTON_Dev_Cha	CLICKED	SETBITMAP	ID_BUTTON_Expense_Cha
✕	✎	✎		ID_BUTTON_Dev_Cha	CLICKED	SETBITMAP	ID_BUTTON_Expense_Cha

圖 1-71 設備鍵按下

19) 定義充電百分比變數“ChargePer”。

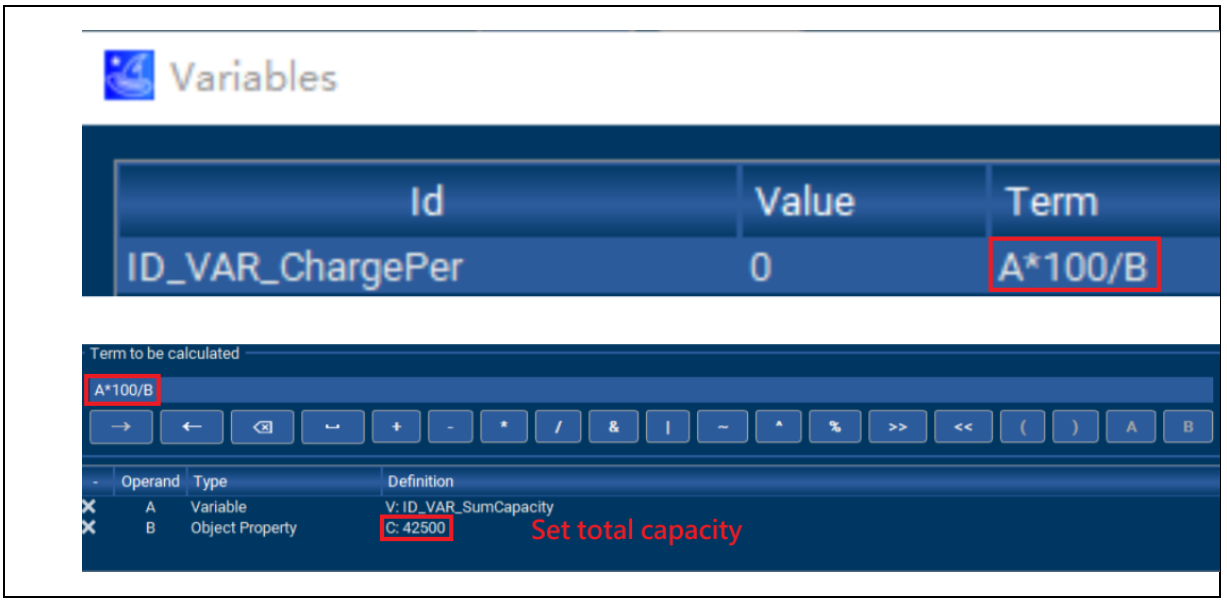


圖 1-72 充電百分比變數定義

20) 充電百分比顯示。0%~19%顯示 1 格電量；20%~39%顯示 2 格電量；40%~59%顯示 3 格電量；60%~79%顯示 4 格電量；80%~100%顯示 5 格電量。

圖 1-73 充電百分比顯示

21) 定義尖、峰、平、谷各時間段累計電量、累計費用變數。

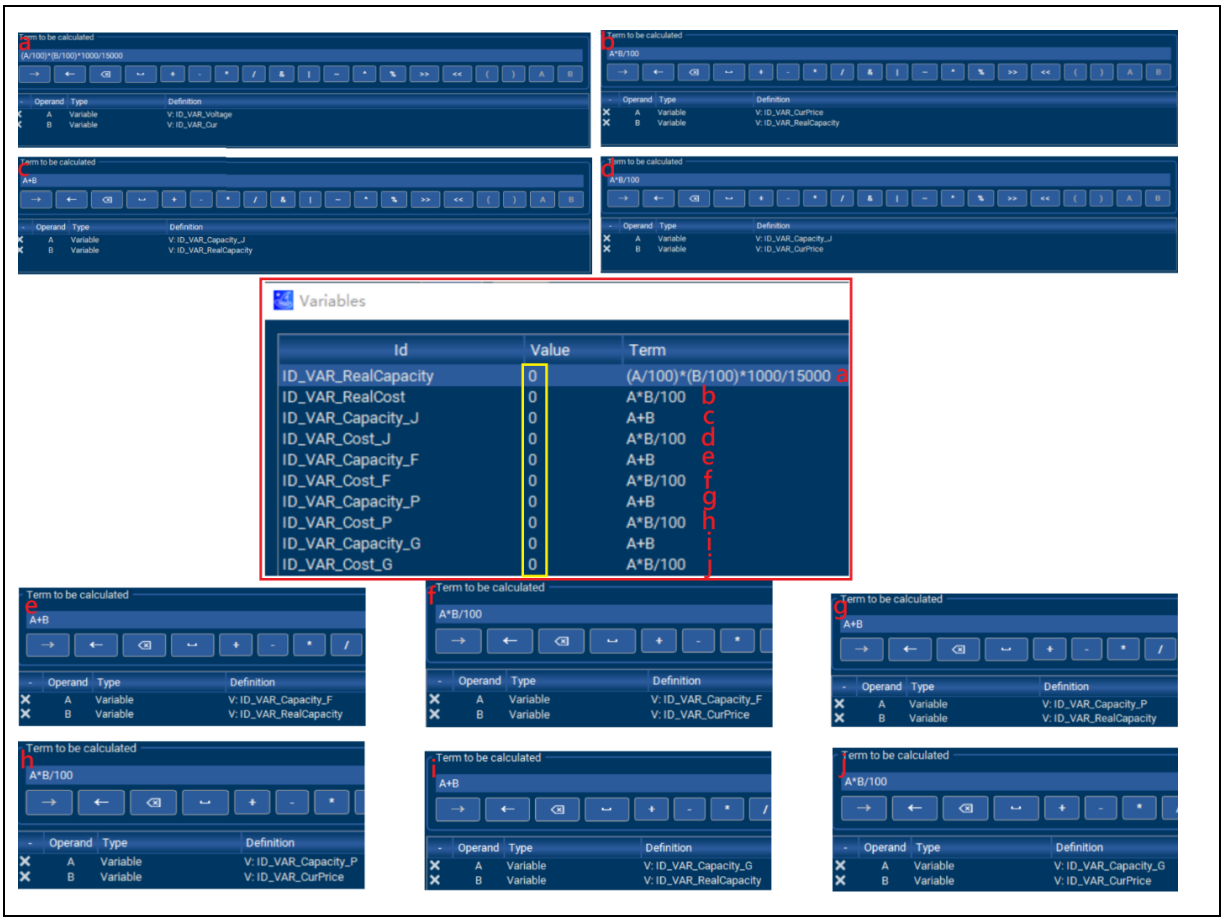


圖 1-74 電量、費用變數定義

22) 尖、峰、平、谷各時間段所消費的累計電量及累計費用顯示。

圖 1-75 尖、峰、平、谷各時間段累計電量、累計費用顯示

23) 定義功率變數。

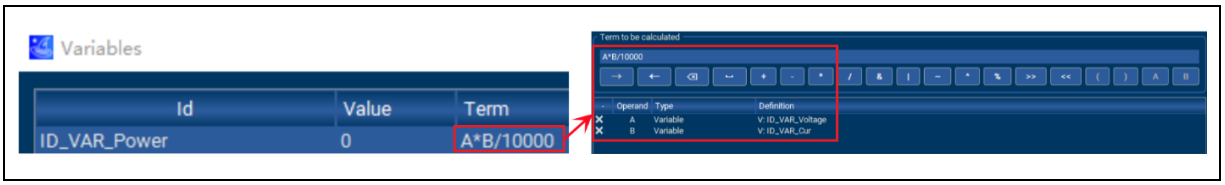


圖 1-76 功率變數定義

24) 功率顯示。

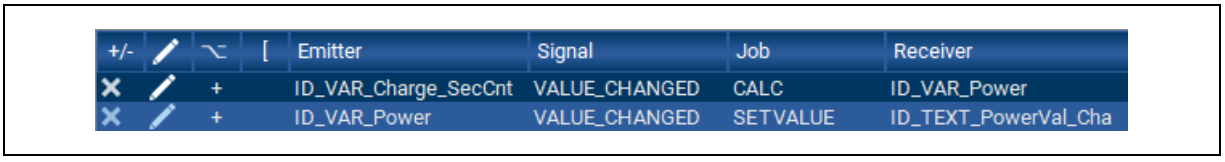


圖 1-77 功率顯示

25) 當前價格顯示。

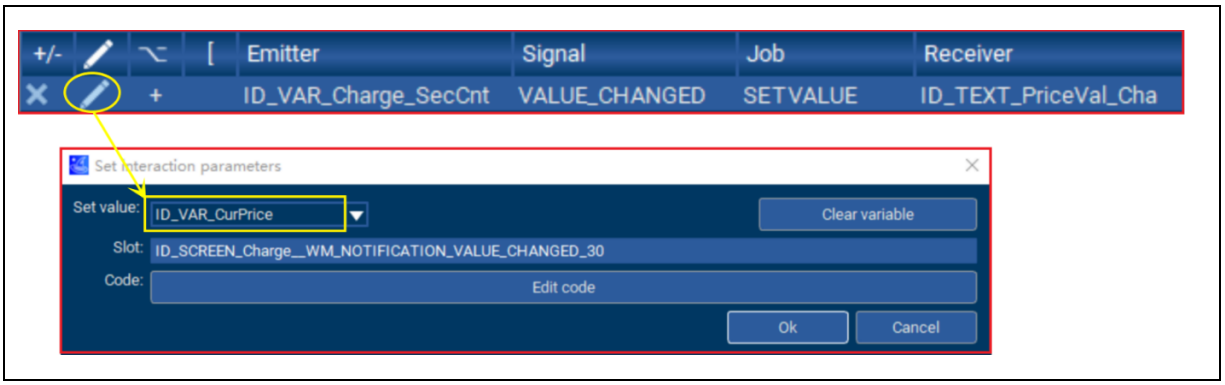


圖 1-78 當前價格顯示

26) 載入 Charge 螢幕，根據 “Lan”變數的值更新監控、消費、設備鍵圖案。

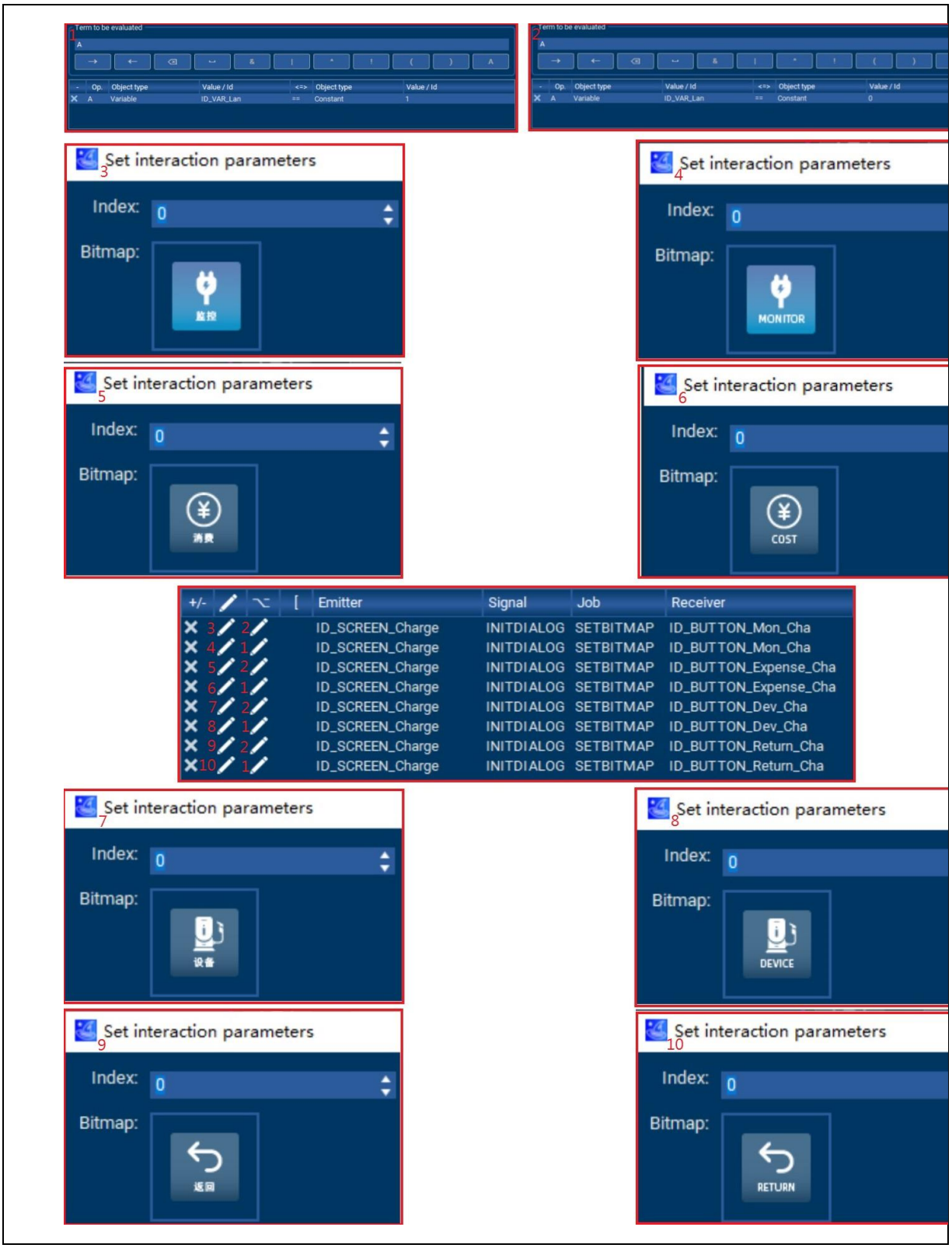


圖 1-79 更新新監控、消費、設備鍵圖案

4. Price 螢幕交互設計。包括按“Return”鍵返回 Init 螢幕，顯示即時時鐘、顯示各時間段的價格。

1) 按“Return”鍵返回 Init 螢幕。參考 1.2.3 章 1 節 1 步驟。

圖 1-80 按“Return”鍵返回 Init 螢幕

2) 即時時鐘顯示。參考 1.2.3 章 2 節 2 步驟。

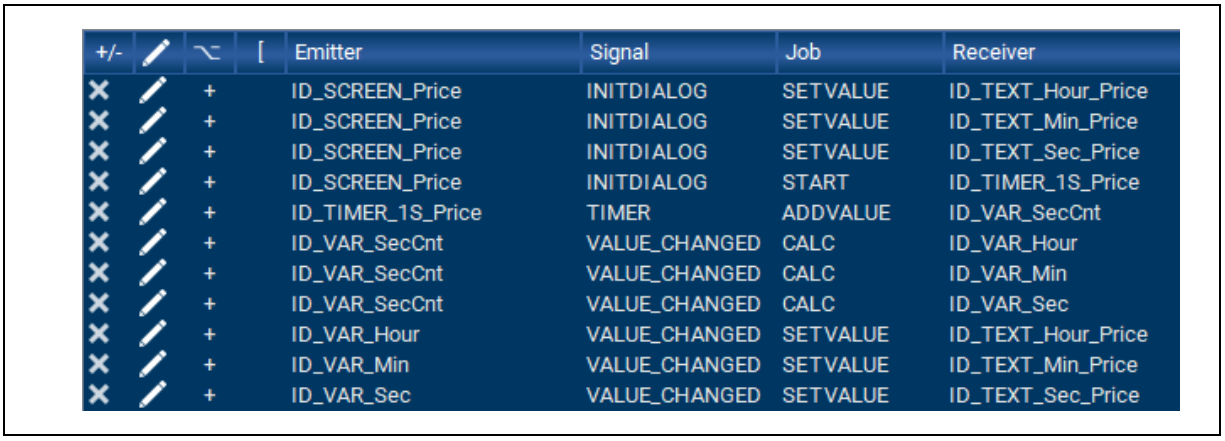


圖 1-81 即時時鐘顯示

3) 定義尖、峰、平、谷各時間段價格變數。

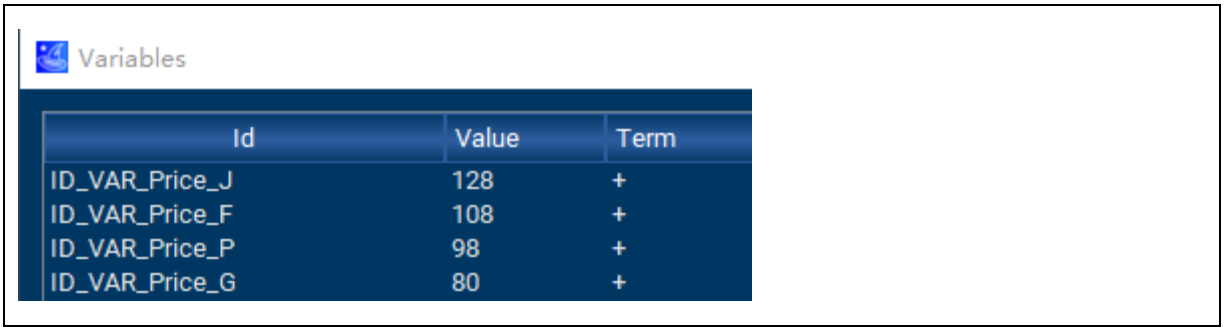


圖 1-82 尖、峰、平、谷各時間段價格變數定義

4) 顯示尖、峰、平、谷各時間段的價格。

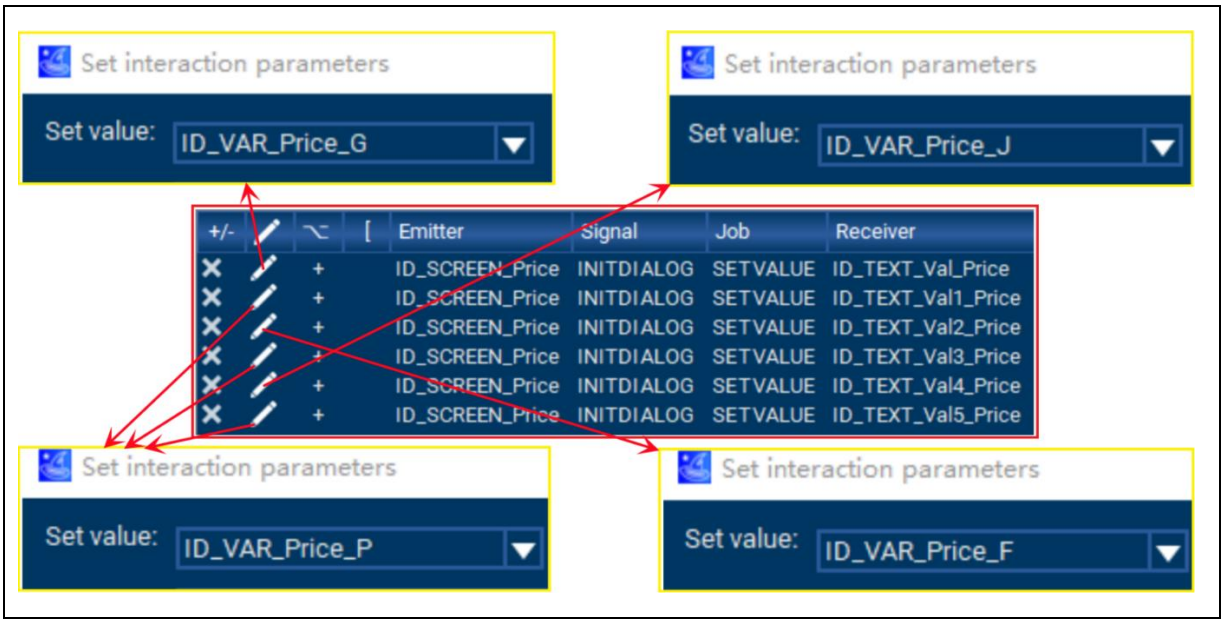


圖 1-83 尖、峰、平、谷各時間段價格顯示

5) 載入 Price 螢幕，根據 “Lan”變數的值更新 “Return”鍵圖案。參考 1.2.3 章 1 節 3 步驟。

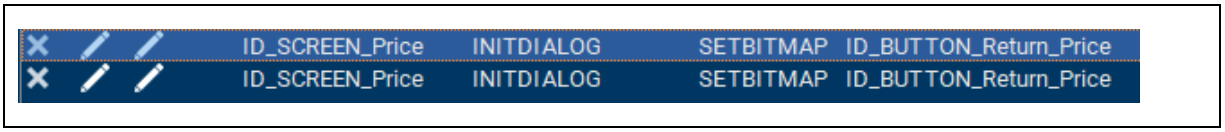


圖 1-84 更新“Return”鍵圖案

5. Info 螢幕設計。包括按“Return”鍵返回 Init 螢幕；顯示即時時鐘；顯示充電站名稱、支援的充電類型以及可用的充電槍號。

1) 按“Return”鍵返回 Init 螢幕。參考 1.2.3 章 1 節 1 步驟。

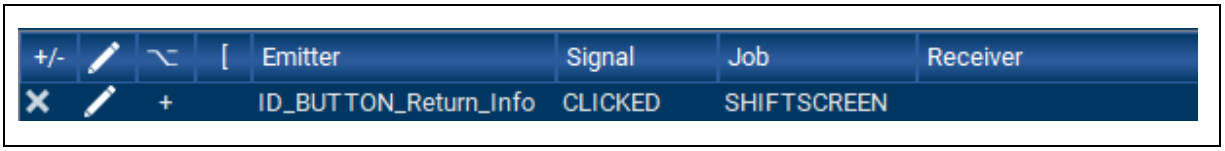
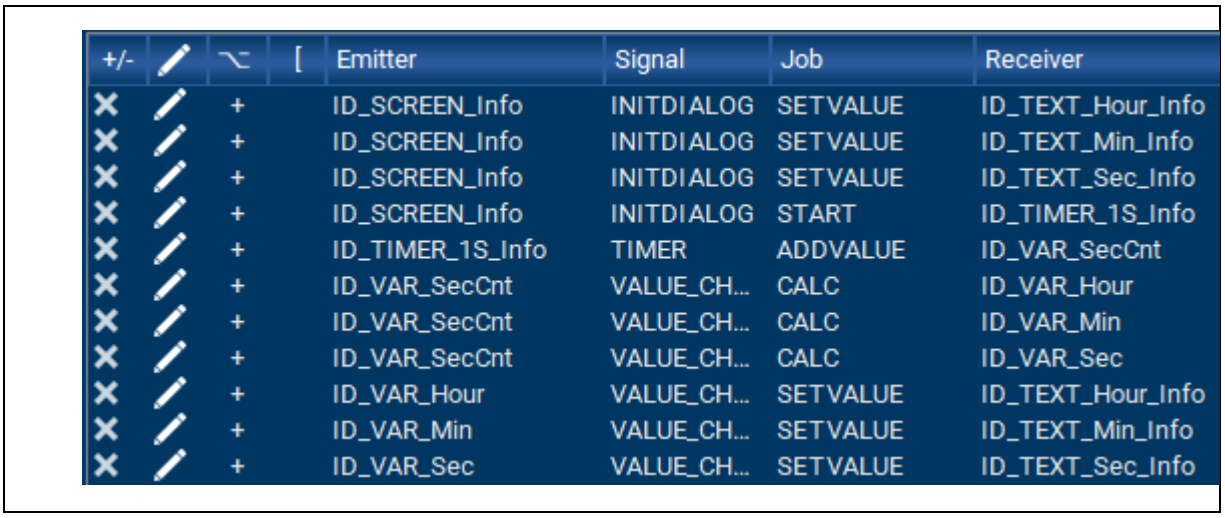


圖 1-85 按“Return”鍵返回 Init 螢幕

2) 即時時鐘顯示。參考 1.2.3 章 2 節 2 步驟。



+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_Info	INITDIALOG	SETVALUE	ID_TEXT_Hour_Info
+	ID_SCREEN_Info	INITDIALOG	SETVALUE	ID_TEXT_Min_Info
+	ID_SCREEN_Info	INITDIALOG	SETVALUE	ID_TEXT_Sec_Info
+	ID_SCREEN_Info	INITDIALOG	START	ID_TIMER_1S_Info
+	ID_TIMER_1S_Info	TIMER	ADDVALUE	ID_VAR_SecCnt
+	ID_VAR_SecCnt	VALUE_CH...	CALC	ID_VAR_Hour
+	ID_VAR_SecCnt	VALUE_CH...	CALC	ID_VAR_Min
+	ID_VAR_SecCnt	VALUE_CH...	CALC	ID_VAR_Sec
+	ID_VAR_Hour	VALUE_CH...	SETVALUE	ID_TEXT_Hour_Info
+	ID_VAR_Min	VALUE_CH...	SETVALUE	ID_TEXT_Min_Info
+	ID_VAR_Sec	VALUE_CH...	SETVALUE	ID_TEXT_Sec_Info

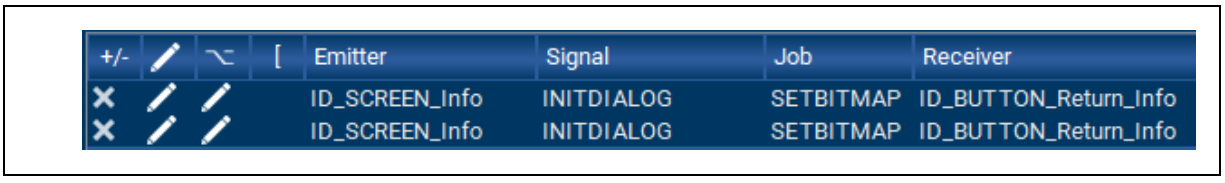
圖 1-86 即時時鐘顯示

3) 槍號初始化。

+/-	Emitter	Signal	Job	Receiver
+	ID_SCREEN_Info	INITDIALOG	SETVALUE	ID_TEXT_Gun1_Info
+	ID_SCREEN_Info	INITDIALOG	SETVALUE	ID_TEXT_Gun2_Info
+	ID_SCREEN_Info	INITDIALOG	SETVALUE	ID_TEXT_Gun3_Info

圖 1-87 槍號初始化

4) 載入 Info 螢幕，根據 “Lan”變數的值更新 “Return”鍵圖案。參考 1.2.3 章 1 節 3 步驟。

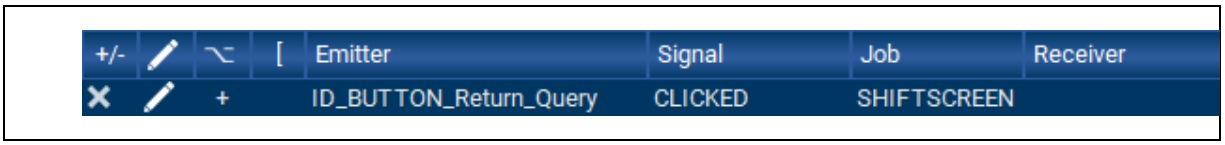


+/-	Emitter	Signal	Job	Receiver
	ID_SCREEN_Info	INITDIALOG	SETBITMAP	ID_BUTTON_Return_Info
	ID_SCREEN_Info	INITDIALOG	SETBITMAP	ID_BUTTON_Return_Info

圖 1-88 載入 Info 螢幕更新”Return”鍵圖案

6. Query 螢幕交互設計。包括按“Return”鍵返回 Init 螢幕；顯示即時時鐘，結合代碼顯示用戶名；上電後直到充電完成才會顯示最近一次的充電資訊。

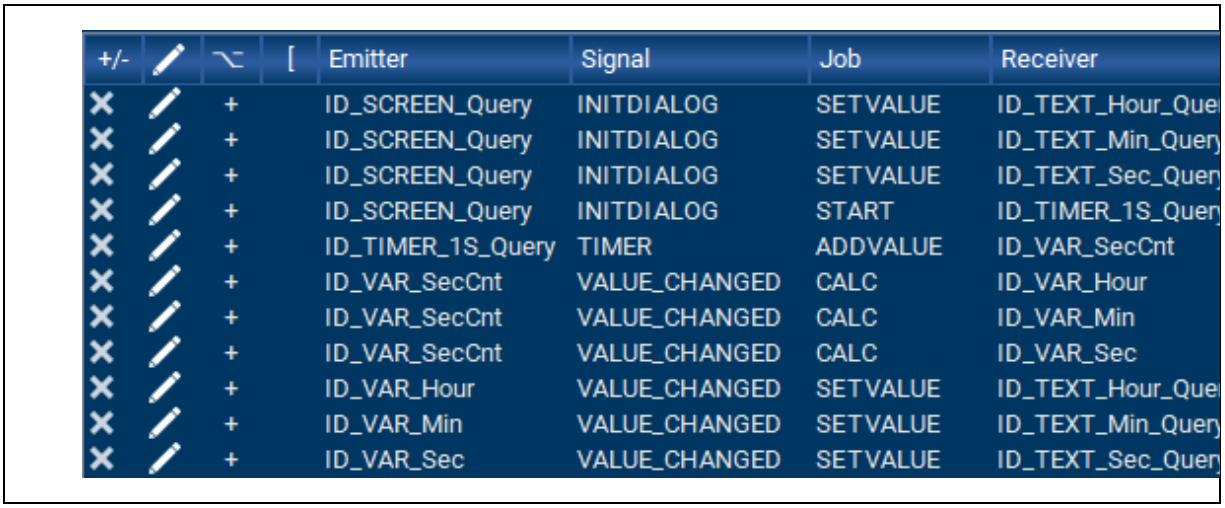
1) 按“Return”鍵返回 Init 螢幕。參考 1.2.3 章 1 節 1 步驟。



+/-		⌥	[	Emitter	Signal	Job	Receiver
✕	✎	+		ID_BUTTON_Return_Query	CLICKED	SHIFTSCREEN	

圖 1-89 按“Return”鍵返回 Init 螢幕

2) 即時時鐘顯示。參考 1.2.3 章 2 節 2 步驟。



+/-		⌥	[	Emitter	Signal	Job	Receiver
✕	✎	+		ID_SCREEN_Query	INITDIALOG	SETVALUE	ID_TEXT_Hour_Que
✕	✎	+		ID_SCREEN_Query	INITDIALOG	SETVALUE	ID_TEXT_Min_Quer
✕	✎	+		ID_SCREEN_Query	INITDIALOG	SETVALUE	ID_TEXT_Sec_Quer
✕	✎	+		ID_SCREEN_Query	INITDIALOG	START	ID_TIMER_1S_Quer
✕	✎	+		ID_TIMER_1S_Query	TIMER	ADDVALUE	ID_VAR_SecCnt
✕	✎	+		ID_VAR_SecCnt	VALUE_CHANGED	CALC	ID_VAR_Hour
✕	✎	+		ID_VAR_SecCnt	VALUE_CHANGED	CALC	ID_VAR_Min
✕	✎	+		ID_VAR_SecCnt	VALUE_CHANGED	CALC	ID_VAR_Sec
✕	✎	+		ID_VAR_Hour	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour_Que
✕	✎	+		ID_VAR_Min	VALUE_CHANGED	SETVALUE	ID_TEXT_Min_Quer
✕	✎	+		ID_VAR_Sec	VALUE_CHANGED	SETVALUE	ID_TEXT_Sec_Quer

圖 1-90 即時時鐘顯示

3) 螢幕載入視窗顯示。充電完成後才顯示視窗。

圖 1-91 螢幕載入視窗顯示

4) 定義時、分、秒變數及“SecCnt”備份變數、“SumCost”備份變數。

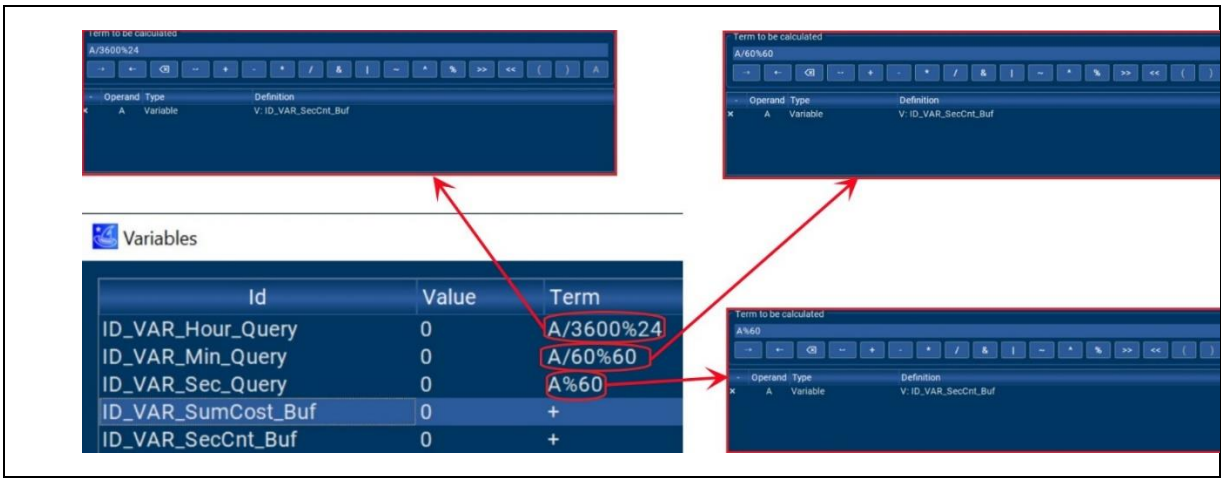


圖 1-92 時、分、秒變數及“SecCnt”備份變數、“SumCost”備份變數定義

5) 顯示最近一次充電資訊，包括時間、費用、充電站、槍號。

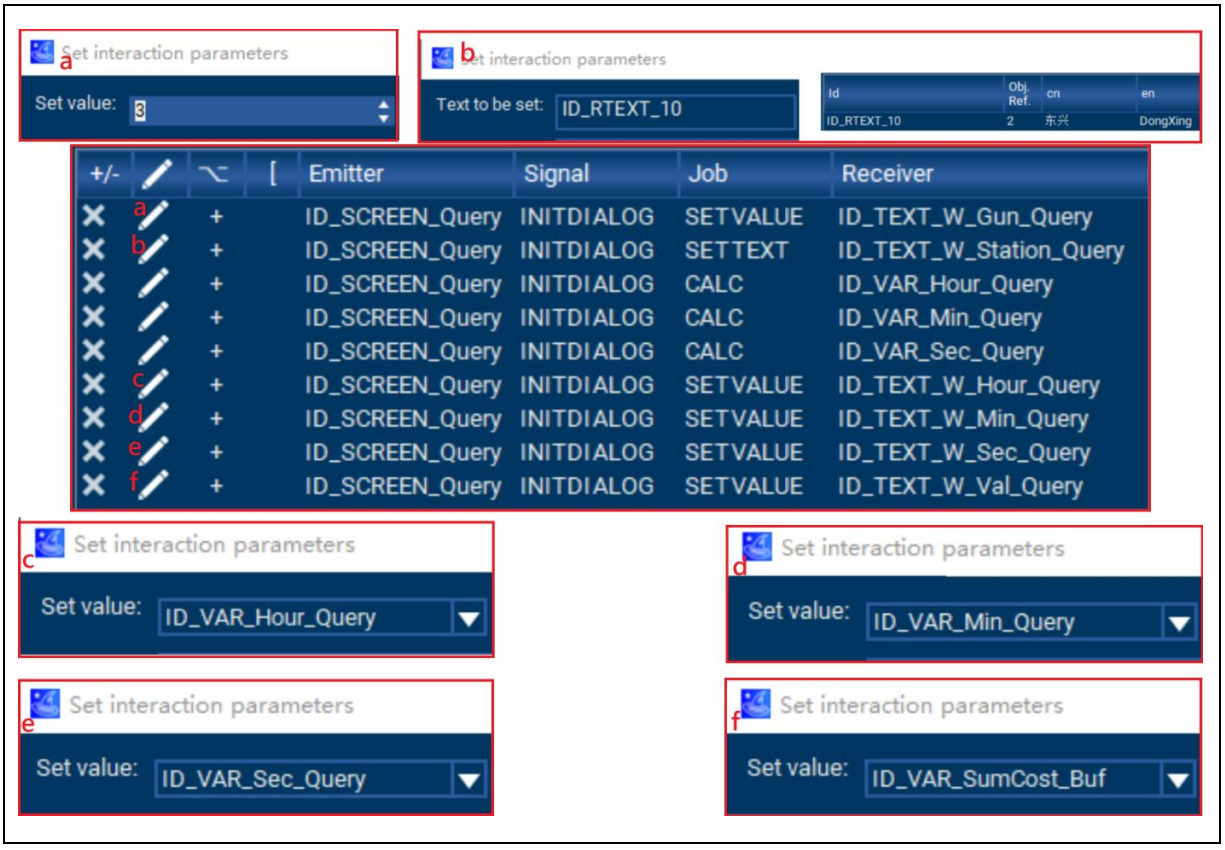
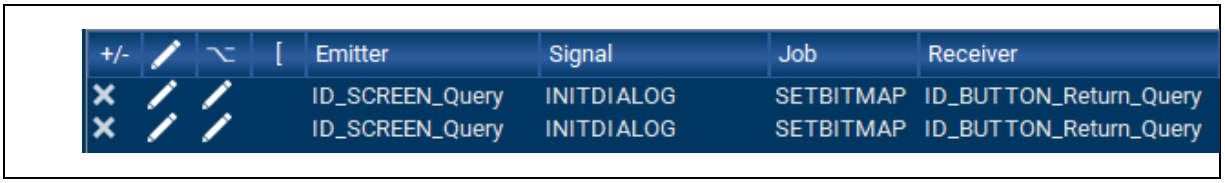


圖 1-93 最近一次充電資訊顯示

6) 載入 Query 螢幕，根據 “Lan”變數的值更新 “Return”鍵圖案。參考 1.2.3 章 1 節 3 步驟。

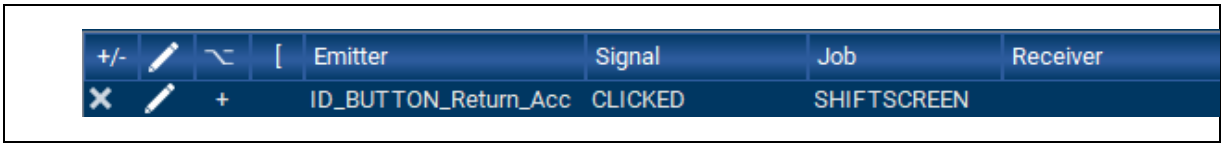


+/-	Emitter	Signal	Job	Receiver
	ID_SCREEN_Query	INITDIALOG	SETBITMAP	ID_BUTTON_Return_Query
	ID_SCREEN_Query	INITDIALOG	SETBITMAP	ID_BUTTON_Return_Query

圖 1-94 載入 Query 螢幕更新“Return”鍵圖案

7. Account 螢幕交互設計。包括按“Return”鍵返回 Init 螢幕；結合代碼顯示用戶名，顯示即時時鐘；顯示帳戶姓名、餘額、電話號碼。

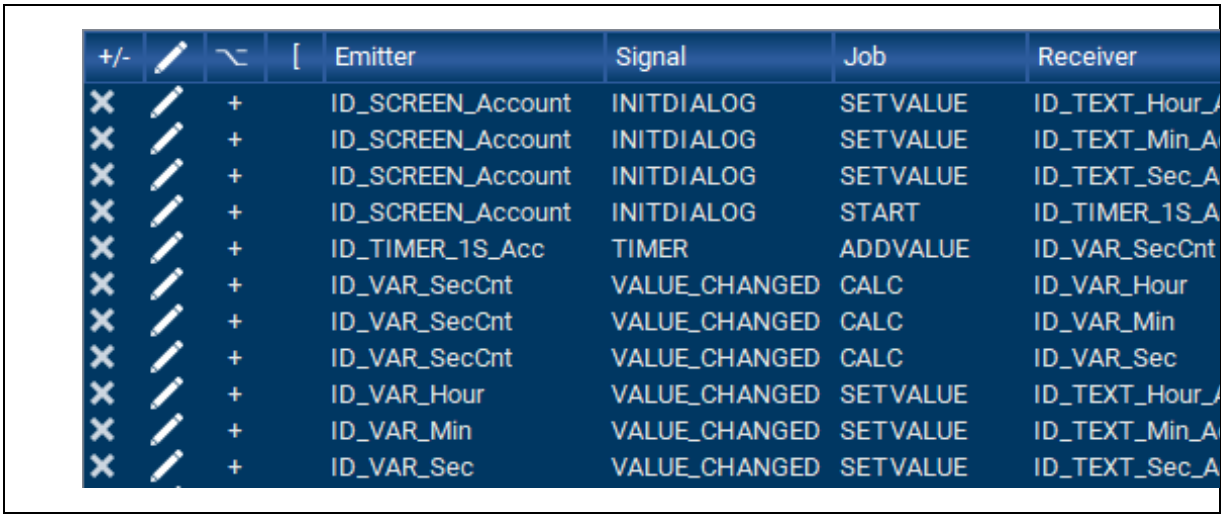
1) 按“Return”鍵返回 Init 螢幕。參考 1.2.3 章 1 節 1 步驟。



+/-	Emitter	Signal	Job	Receiver
	+ ID_BUTTON_Return_Acc	CLICKED	SHIFTSCREEN	

圖 1-95 按“Return”鍵返回 Init 螢幕

2) 即時時鐘顯示。參考 1.2.3 章 2 節 2 步驟。



+/-	Emitter	Signal	Job	Receiver
	+ ID_SCREEN_Account	INITDIALOG	SETVALUE	ID_TEXT_Hour_
	+ ID_SCREEN_Account	INITDIALOG	SETVALUE	ID_TEXT_Min_A
	+ ID_SCREEN_Account	INITDIALOG	SETVALUE	ID_TEXT_Sec_A
	+ ID_SCREEN_Account	INITDIALOG	START	ID_TIMER_1S_A
	+ ID_TIMER_1S_Acc	TIMER	ADDVALUE	ID_VAR_SecCnt
	+ ID_VAR_SecCnt	VALUE_CHANGED	CALC	ID_VAR_Hour
	+ ID_VAR_SecCnt	VALUE_CHANGED	CALC	ID_VAR_Min
	+ ID_VAR_SecCnt	VALUE_CHANGED	CALC	ID_VAR_Sec
	+ ID_VAR_Hour	VALUE_CHANGED	SETVALUE	ID_TEXT_Hour_
	+ ID_VAR_Min	VALUE_CHANGED	SETVALUE	ID_TEXT_Min_A
	+ ID_VAR_Sec	VALUE_CHANGED	SETVALUE	ID_TEXT_Sec_A

圖 1-96 即時時鐘顯示

3) 定義餘額變數。初始值為 1000。

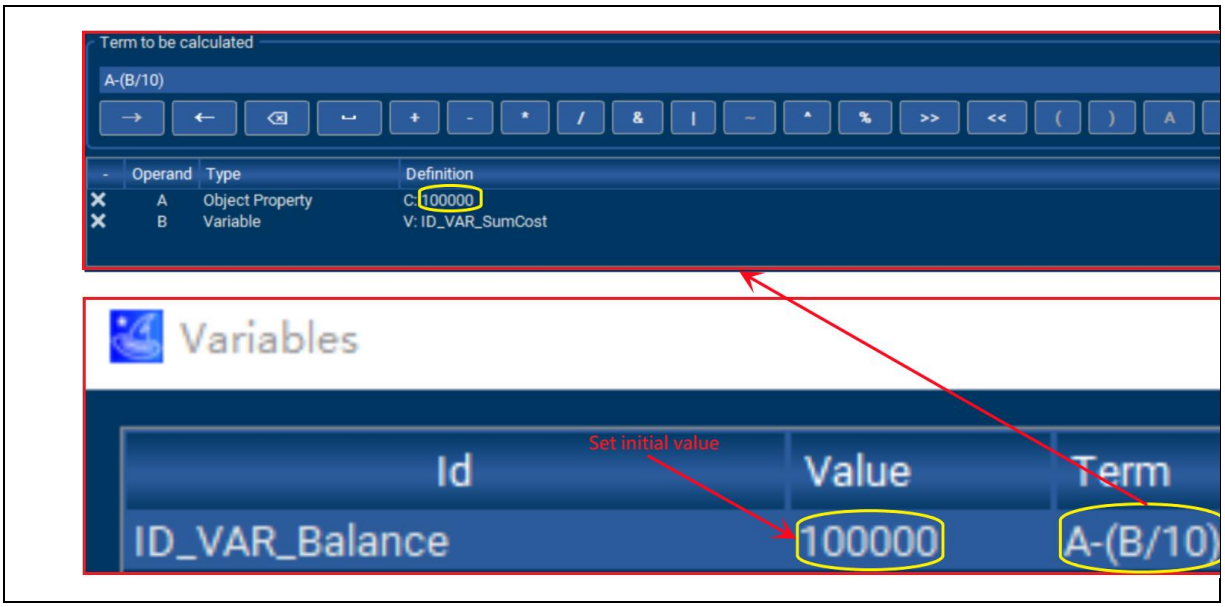


圖 1-97 餘額變數定義

4) 餘額顯示。

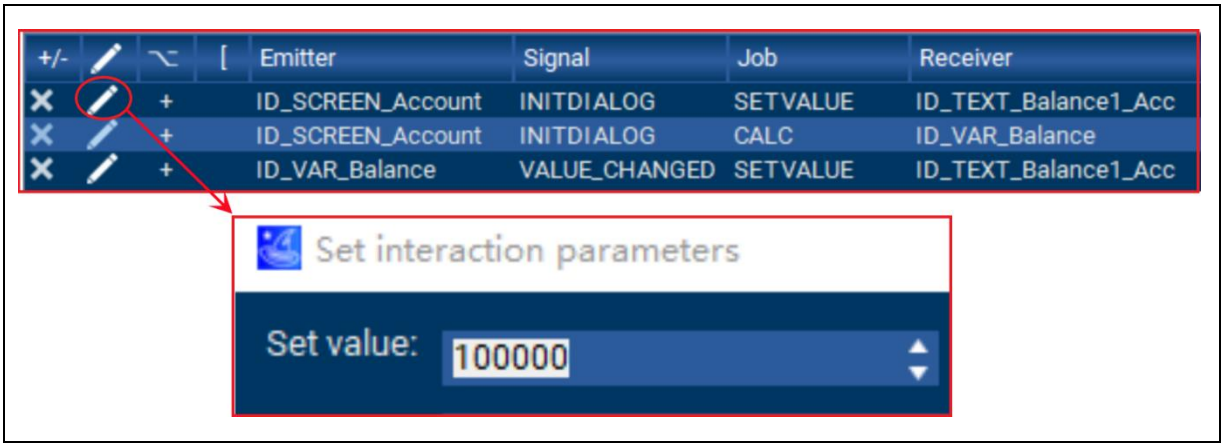


圖 1-98 餘額顯示

5) 載入 Account 螢幕，根據 “Lan”變數的值更新 “Return”鍵圖案。參考 1.2.3 章 1 節 3 步驟。

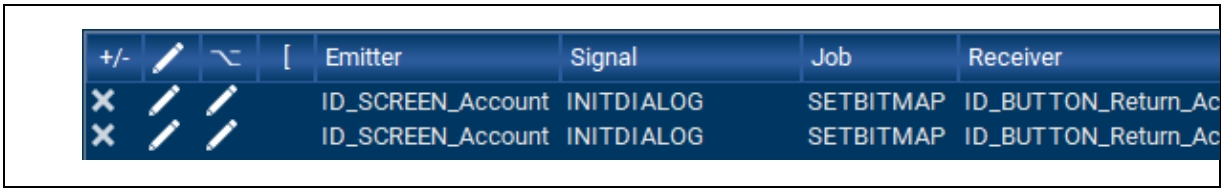


圖 1-99 載入 Account 螢幕更新“Return”鍵圖案

1.2.4 程式碼編輯

1. 設計完成即可於 File-> Export & Save 輸出程式碼，如下圖。此範例所使用之專案檔案名稱為 EC_ChargePile.zip，檔案路徑可參考圖 4-1，在解壓縮後，可以使用 AppWizard 直接開啟該項目檔。

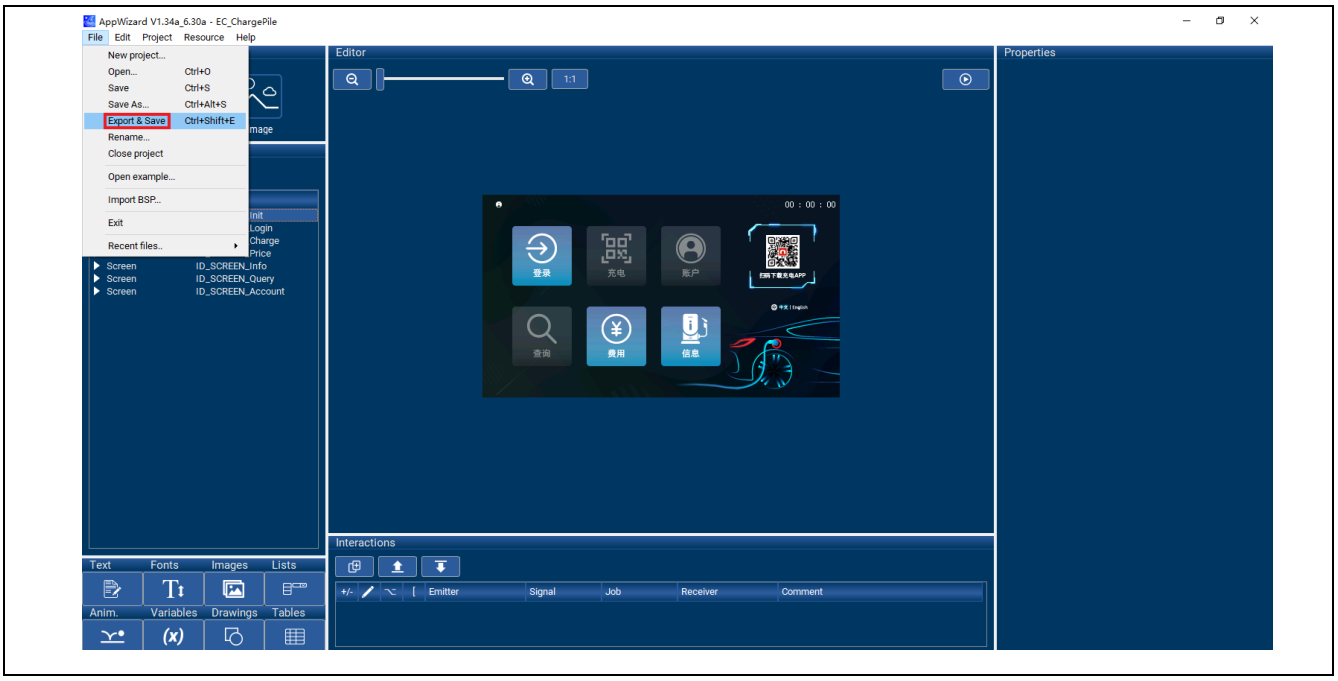


圖 1-100 輸出程式碼

2. 於 Project -> Open containing folder 開啟 AppWizard 專案資料夾，複製 Resource 與 Source 資料夾，於路徑：EC_M467_AppWizard_ChargePile_V1.00\SampleCode\Example\M467_AppWizard_ChargePile\Applicationn 中覆蓋原本檔案，步驟如下 2 圖所示。

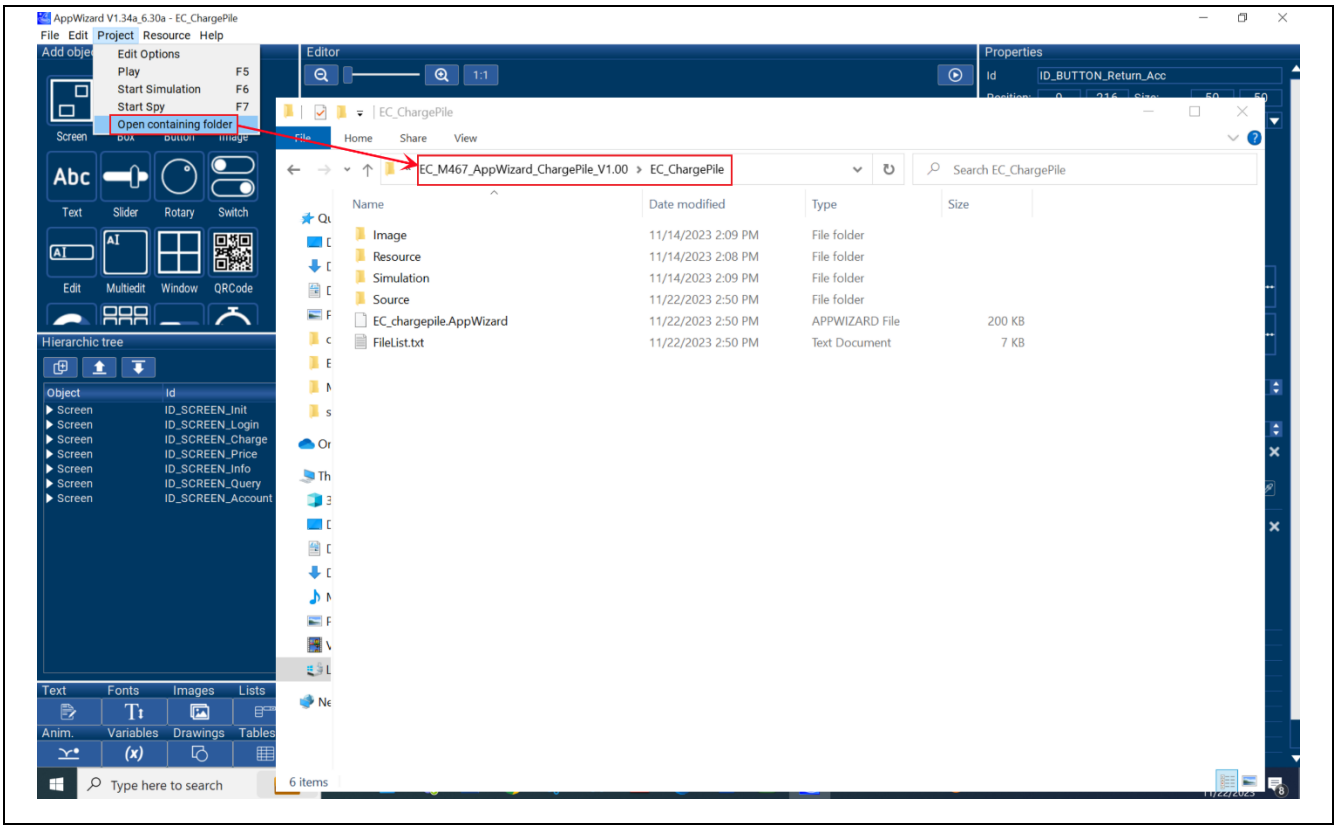


圖 1-101 開啟專案資料夾

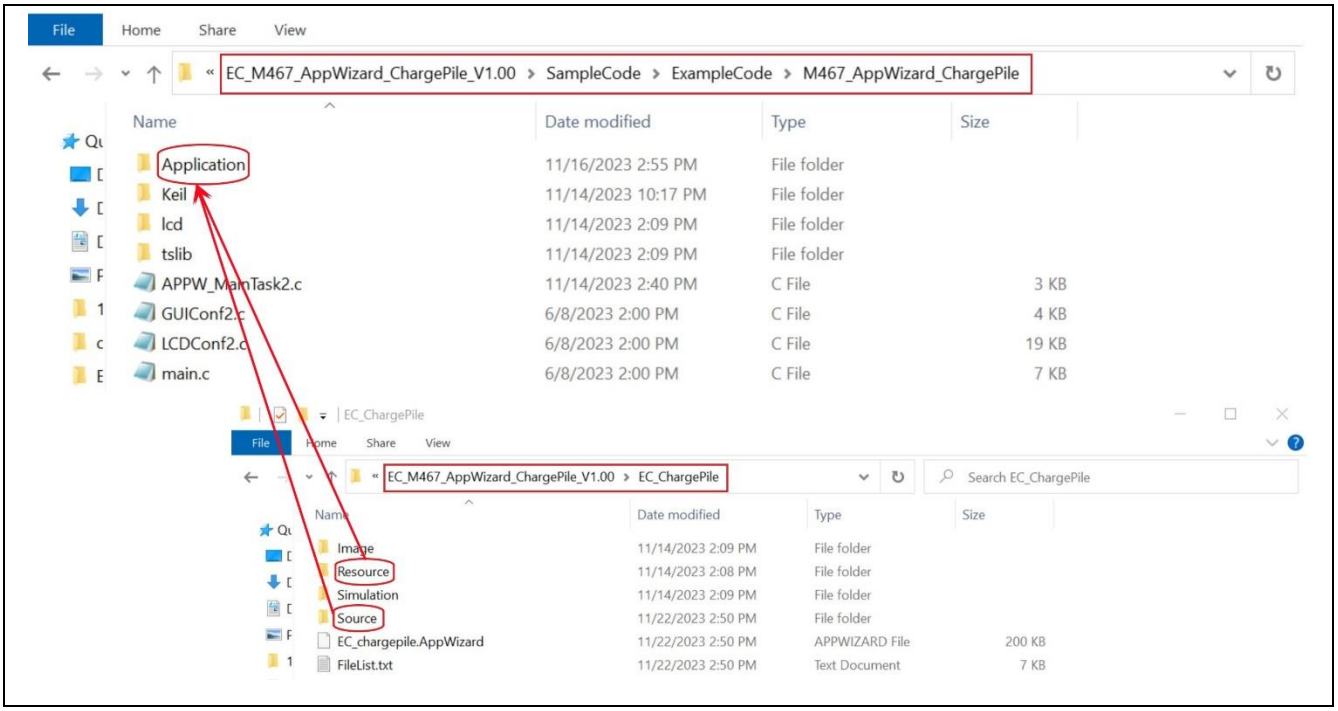


圖 1-102 移植至程式碼專案中

3. 主程序代碼請參考 2 代碼介紹。

1.2.5 燒錄運行在裝置中

實際運行在裝置後，仍可對整體執行效能做調整，相關優化設定如下所述

1. 於 AppWizard 使用的圖片 format，需設定為符合螢幕格式，此範例設定為 “High color with alpha (565)”或 “Auto”。

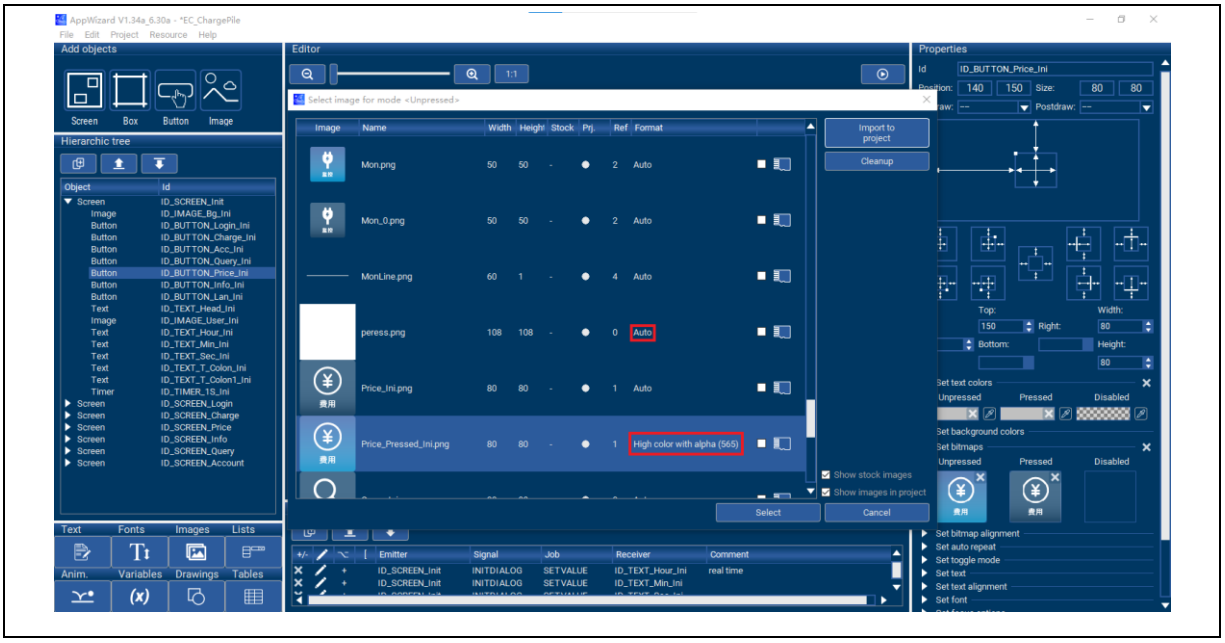


圖 1-103 Image 設定

1.3 執行結果

本範例代碼執行編譯和燒錄的結果如下圖所示。

圖 1-104 執行結果

2. 代碼介紹

主程序由 *main.c* 進入，包括 3 個部分如下：

- 系統時鐘與引腳初始化：sys_init
- 介面初始化與開啟：UART、SPIM、Timer、Touch 等
- 應用主程序：MainTask2()

```
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();
    // Startup CTP
    st1633i_low_level_init();
    st1633i_startup_sequence();

    printf("\r\n");
    printf("CPU @ %d Hz\r\n", SystemCoreClock);
    printf("+------------------------------------------------------------+\r\n");
    printf("|               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 介面，是為了將占較大空間的圖片、文字檔資源存至外部 SPI Flash，利用 SPI Synchronous Serial Interface Controller (SPI Master Mode)，在 Direct Memory Mapping Mode(DMM)的模式下，SPI Flash 可被視為 ROM Module，直接到指定地址讀取 SPI Flash，並在內建的 cache 幫忙下，以及使用四線模式(quad read)，提升整體讀取效率。

初始化 SPIM 的程式碼如下，需包括 SPIM 速度、SPI flash detection、開啟 SPIM cache mode、開啟 SPIM DMM mode、開啟 fast quad read mode。相關參數設定可參考 W25Q32JVTBIQ 之說明。

```
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, 200MHz/2 = 100*/

    SPIM_SET_RXCLKDLY_RDDLYSEL(0);     /* Insert 0 delay cycle. Adjust the sampling clock
of received data 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，亦須編輯 Scatter file，以讓編譯器知道 SPI Flash 的位置與內容。於 KEIL 環境中，依照路徑 Options for Target-> Linker->Scatter File-> Edit，開啟.sct 檔案並編輯內容如下。

```
;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
    {
     MicrosoftJhengHeiUI_16_Bold_EXT_AA4.o (+RO)
     MicrosoftJhengHeiUI_24_Bold_EXT_AA4.o (+RO)
     Account_Lock_Ini.o (+RO)
     Account_Pressed_Ini.o (+RO)
     Account_Lock_Ini_En.o (+RO)
     Account_Pressed_Ini_En.o (+RO)
     Bg_Cha.o (+RO)
     Bg2.o (+RO)
     Bg2_En.o (+RO)
     Bg3.o (+RO)
     Charge_Lock_Ini.o (+RO)
     Charge_Pressed_Ini.o (+RO)
     Charge_Lock_Ini_En.o (+RO)
     Charge_Pressed_Ini_En.o (+RO)
     che0.o (+RO)
     che1.o (+RO)
     che2.o (+RO)
     che3.o (+RO)
     che4.o (+RO)
     che5.o (+RO)
     Connect.o (+RO)
     Cost.o (+RO)
     Cost_0.o (+RO)
     Cost_En.o (+RO)
     Cost_0_En.o (+RO)
     Device.o (+RO)
     Device_0.o (+RO)
     Device_En.o (+RO)
     Device_0_En.o (+RO)
     ExpenseLine.o (+RO)
     Head_Ini.o (+RO)
     Info_Pressed_Ini.o (+RO)
     Info_Pressed_Ini_En.o (+RO)
     Lan_Cn.o (+RO)
     Lan_En.o (+RO)
     Login_Pressed_Ini.o (+RO)
     Login_Pressed_Ini_En.o (+RO)
     Mon.o (+RO)
     Mon_0.o (+RO)
     Mon_En.o (+RO)
     Mon_0_En.o (+RO)
     MonLine.o (+RO)
     Price_Pressed_Ini.o (+RO)
     Price_Pressed_Ini_En.o (+RO)
     Query_Lock_Ini.o (+RO)
```

```
    Query_Lock_Ini_En.o (+RO)
    Query_Pressed_Ini.o (+RO)
    Query_Pressed_Ini_En.o (+RO)
    Return0.o (+RO)
    Return1.o (+RO)
    Return0_En.o (+RO)
    Return1_En.o (+RO)
    Start.o (+RO)
    Stop_Pressed.o (+RO)
   }
}
```

程式進入應用主程序 MainTask2()後，將執行 AppWizard 相關函數，在 while 迴圈中會執行畫圖的動作，在此範例程式使用 APPW_SetVarData()函數，將值從 MCU 傳出給 AppWizard 中的 Variable；使用 APPW_GetText()函數，獲取 AppWizard 中的 Text 值；使用 APPW_SetText()函數，將字元從 MCU 傳給 AppWizard 中的 Text 變數。通過 1.2.3 章 1 節 5 步驟所述之方法，實現從 AppWizard 中 Login 螢幕輸入用戶名、密碼，代碼對輸入密碼進行檢查。如果密碼正確，首先設置“PW_1”變數為 1，然後設置字元“OK”顯示在 Login 螢幕“PW_Sta”物件上，最後設置用戶名顯示在 Login、Init、Charge、Account、Query 螢幕的“Head”物件上；如果密碼錯誤，首先設置“PW_1”變數設置為 0，然後設置字元“ERR”顯示在 Login 螢幕的“PW_Sta”物件。

```
void MainTask2(void)
{
    volatile int i=0;
    volatile int j=0;
    char UserName[20];
    char PW_Val[8] = {'q','q','q','q','q','q','q','q'};
    char PW_OK[3] = {'O','K',' '};
    char PW_ERR[3] = {'E','R','R'};
    char PW_Rev[9];
    // 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
    while (1)
    {
      // Drawing a frame
      GUI_Exec1();
      APPW_Exec();
      NVT_Draw1();
        i = APPW_GetText(ID_SCREEN_Login,ID_EDIT_UserName_Log,UserName,8);
        i = APPW_GetText(ID_SCREEN_Login,ID_EDIT_Password_Log,PW_Rev,9);

        for(i = 0;i < 8;i++)
        {
            if(PW_Rev[i] != PW_Val[i])
```

```
            {
                APPW_SetVarData(ID_VAR_PW_1, 0);
                APPW_SetText(ID_SCREEN_Login,ID_TEXT_PW_Sta,PW_ERR);
                break;
            }
        }
        if(i == 8)
        {
            APPW_SetVarData(ID_VAR_PW_1, 1);
            APPW_SetText(ID_SCREEN_Login,ID_TEXT_PW_Sta,PW_OK);
            APPW_SetText(ID_SCREEN_Init,ID_TEXT_Head_Ini,UserName);
            APPW_SetText(ID_SCREEN_Charge,ID_TEXT_Head_Cha,UserName);
            APPW_SetText(ID_SCREEN_Account,ID_TEXT_Head_Acc,UserName);
            APPW_SetText(ID_SCREEN_Query,ID_TEXT_Head_Query,UserName);
        }
    }
}
```

AppWizard 執行的畫圖內容，會從 ID_Screen_Init_Slots.c 中讀取。在 1.2.3 章 2 節 7 步驟中，“Lan”鍵按下的程式碼中加入 printf("Button Click\n")，那麼當 Lan 鍵按下，UART 會于 PC 印出”Button Click”內容。

```
/*
File        : ID_SCREEN_Init_Slots.c
Purpose     : AppWizard managed file, function content could be changed------------------
--------END-OF-HEADER------------------------------
*/
#include "Application.h"
#include "../Generated/Resource.h"
#include "../Generated/ID_SCREEN_Init.h"
/*** Begin of user code area ***/
/*** End of user code area ***/

/*********************************************************************
*
*       Public code
*
**********************************************************************
*/
/*********************************************************************
*
*       cbID_SCREEN_Init
*/
void cbID_SCREEN_Init(WM_MESSAGE * pMsg) {
  GUI_USE_PARA(pMsg);
}
/*********************************************************************
*
*       ID_SCREEN_Init__ID_BUTTON_Lan_Ini__WM_NOTIFICATION_CLICKED_0
*/
void ID_SCREEN_Init__ID_BUTTON_Lan_Ini__WM_NOTIFICATION_CLICKED_0(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);
  printf("Button Click\n");
}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP版本
  - M460_Series_BSP_CMSIS_V3.00.002
- emWin Software Package for M460版本
  - emWin Software Package for M460 V3.0
- IDE版本
  - Keil uVersion 5.38.0.0
- AppWizard版本
  - AppWizard V1.34a_6.30a

3.2 硬體需求

- 電路元件
  - NuMaker-M467HJ V1.0
- 示意圖
  - 如圖 3-1，將 Nu-Link2-Me 與子板之螢幕 USB connector 以 2 條 USB 線與 PC 相連

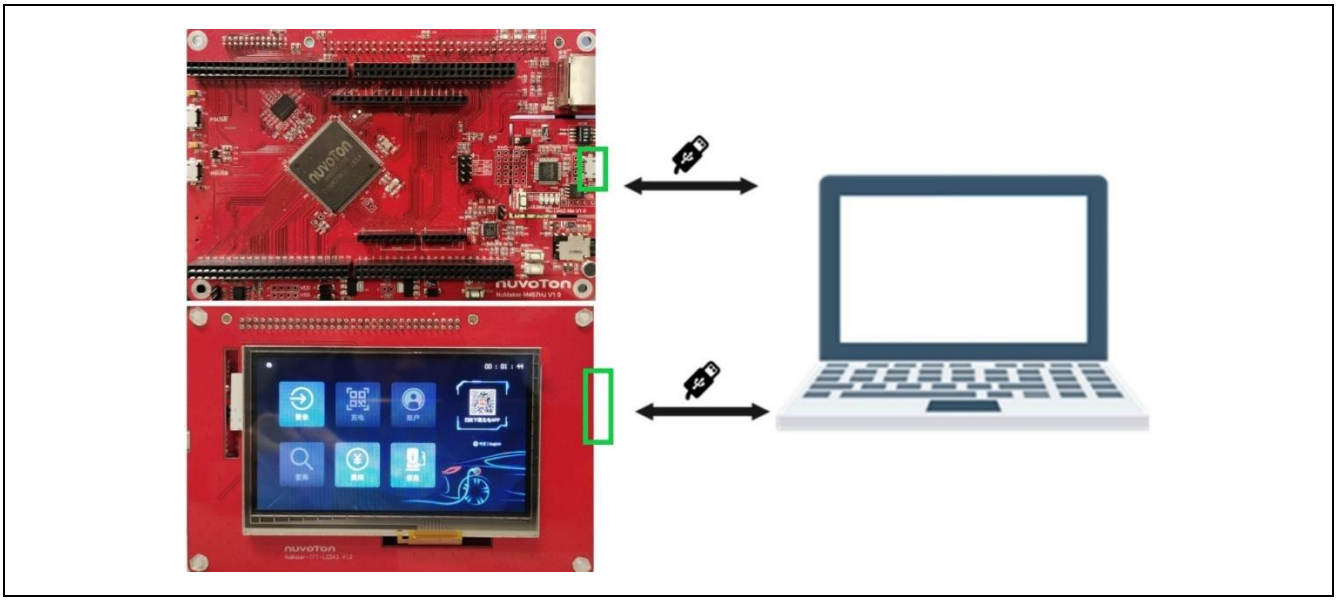


圖 3-1 以 USB 線將 NuMaker-M467HJ V1.0 與 PC 相連

4. 目錄資訊

EC_M467_AppWizard_ChargePile_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	Header files for emWin project
EC_ChargePile.zip	ChargePile sample project of AppWizard. Please open by AppWizard.exe.

圖 4-1 目錄資訊

5. 範例程式執行

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，按兩下 *M467_AppWizard_ChargePile.uvprojx*。

2. 進入編譯模式介面

   - 編譯
   - 下載代碼至記憶體
   - 進入 / 離開除錯模式

3. 進入除錯模式介面

   - 執行代碼

6. 修訂紀錄

Date	Revision	Description
2023.12.07	1.00	初始發佈。

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.