以 AppWizard 設計電動自行車 GUI 專案

NuMicro® 32 位系列微控制器範例程式介紹

文件資訊

應用簡述	本範例程式使用 SEGGER emWin AppWizard 工具，結合 M467 系列開發平台與開發資源，完成電動自行車圖形化使用介面。
BSP 版本	M460_Series_BSP_CMSIS_V3.00.002 emWin Software Package for M460 V3.0
開發平台	NuMaker-HMI-M467

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

本範例程式將示範如何使用 SEGGER emWin AppWizard 工具，搭配 NuMaker-HMI-M467 開發平台，逐步完成一個電動自行車專案的畫面設計、畫面互動。

1.1 原理

1.1.1 系統方塊圖

此範例程式直接使用新唐 GUI 開發平台 - NuMaker-HMI-M467 開發板，開發時會使用到的元件包括解析度 480x272 且屏幕大小 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。

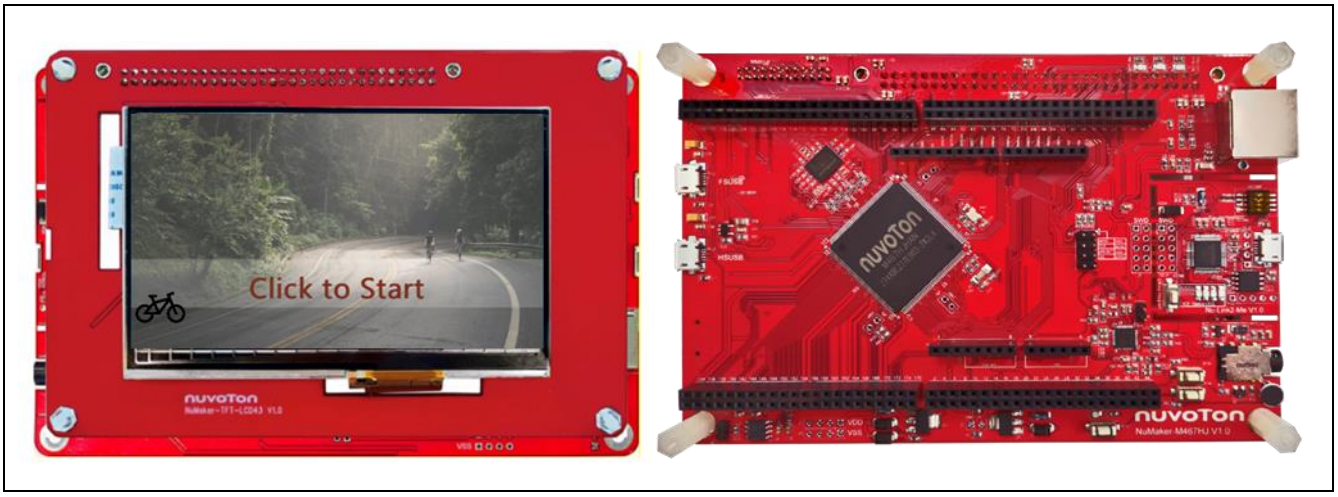


圖 1-1 NuMaker-HMI-M467 開發板

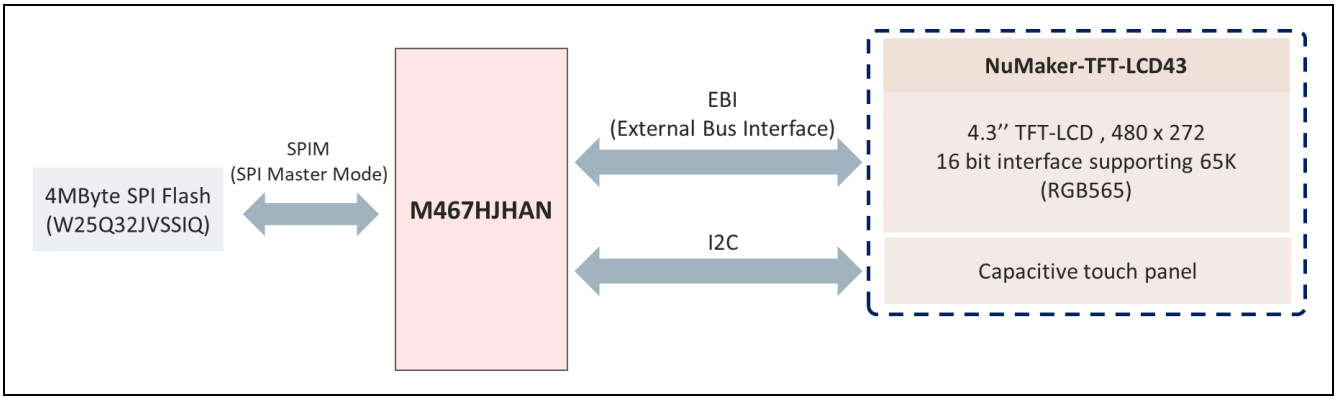


圖 1-2 系統方塊圖

1.1.2 電動自行車畫面

此範例欲設計畫面如下：

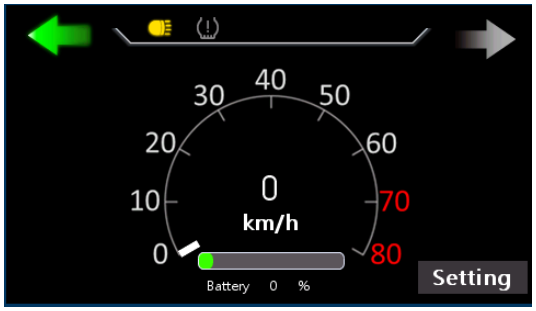
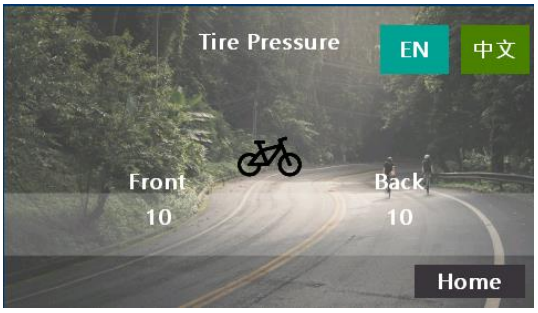
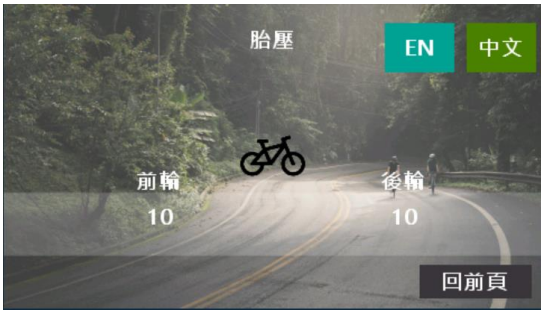
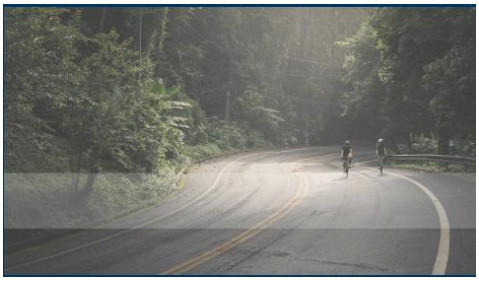
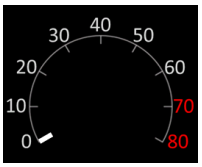
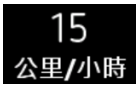
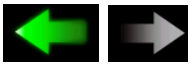
	語言切換至EN(英文)	語言切換至中文
頁面一		
頁面二		

表 1-1 電動自行車畫面

1.1.3 電動自行車畫面使用元素總覽

Ebike App Description	Ebike App GUI	Object
畫面分隔線		Image
480x272 背景圖片		

<table>
<tr><td>腳踏車圖片</td><td> </td><td></td></tr>
<tr><td>480x272 素色背景圖</td><td> </td><td>Box</td></tr>
<tr><td>速度儀表</td><td> 
</td><td>Rotary</td></tr>
<tr><td rowspan="2">速度數值、速度單位</td><td>英文介面
</td><td rowspan="6">TEXT</td></tr>
<tr><td>中文介面 
</td></tr>
<tr><td rowspan="2">電量、電量數值、電量單位</td><td>英文介面
</td></tr>
<tr><td>中文介面
</td></tr>
<tr><td rowspan="2">胎壓名稱、胎壓數值</td><td>英文介面
</td></tr>
<tr><td>中文介面
</td></tr>
<tr><td>電量顯示條</td><td> </td><td>Progbar</td></tr>
<tr><td>方向燈</td><td>  </td><td rowspan="2">Button</td></tr>
<tr><td>大燈</td><td> </td></tr>
</table>

胎壓警示燈		
頁面切換按鈕	英文介面	
	中文介面	
語言切換按鈕		

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四大階段，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、interaction 設定、動畫設定，第三階段使用 AppWizard 匯出程式碼並同時加入客製化程式，以完成更多應用功能，第四階段則是實際燒錄在板端。

以下於 1.2.1~1.2.4 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-HMI-M467，相關資源可參考 參考線路、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 項目即可開始下載，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) 依圖 1-3 AppWizard 工具安裝檔路徑 **Error! Reference source not found.**，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。

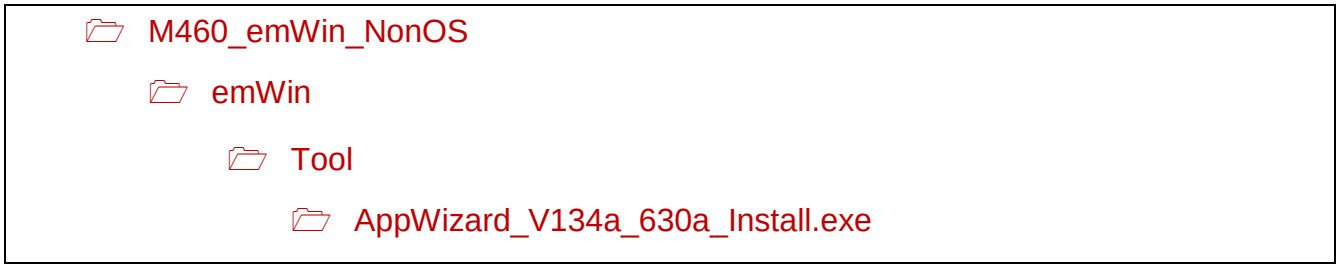


圖 1-3 AppWizard 工具安裝檔路徑

1.2.2 在 AppWizard 工具中編輯與設定 GUI

1. 於桌面捷徑開啟 AppWizard 工具。

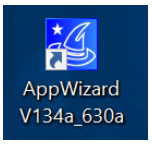


圖 1-4 AppWizard 桌面捷徑圖式

2. 點選 ”Create new project” 以創建新專案。

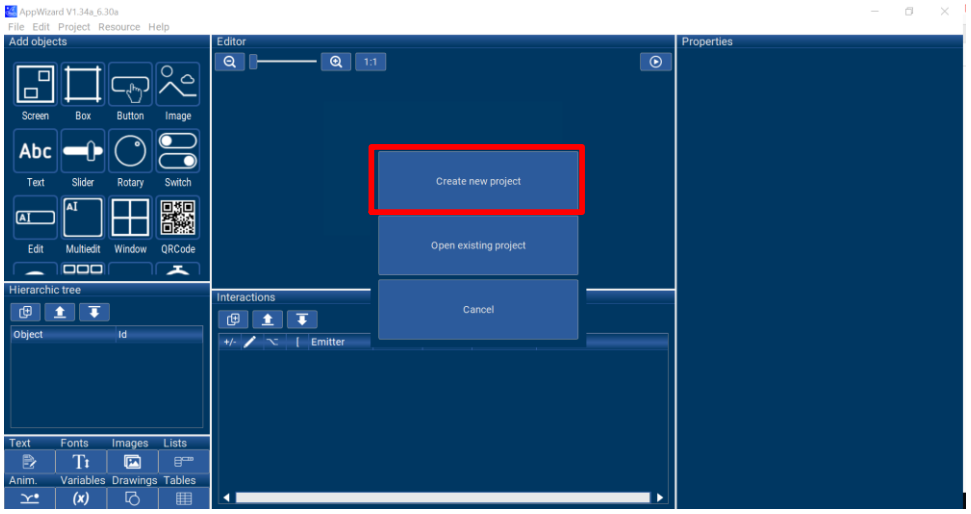


圖 1-5 創建專案頁面

3. 依圖 1-6 設定專案路徑、名稱，符合使用屏幕的顏色格式：16 Bit, GUICC_M565，勾選 ”Enable Multibuffering”，點選 ”ok” 。

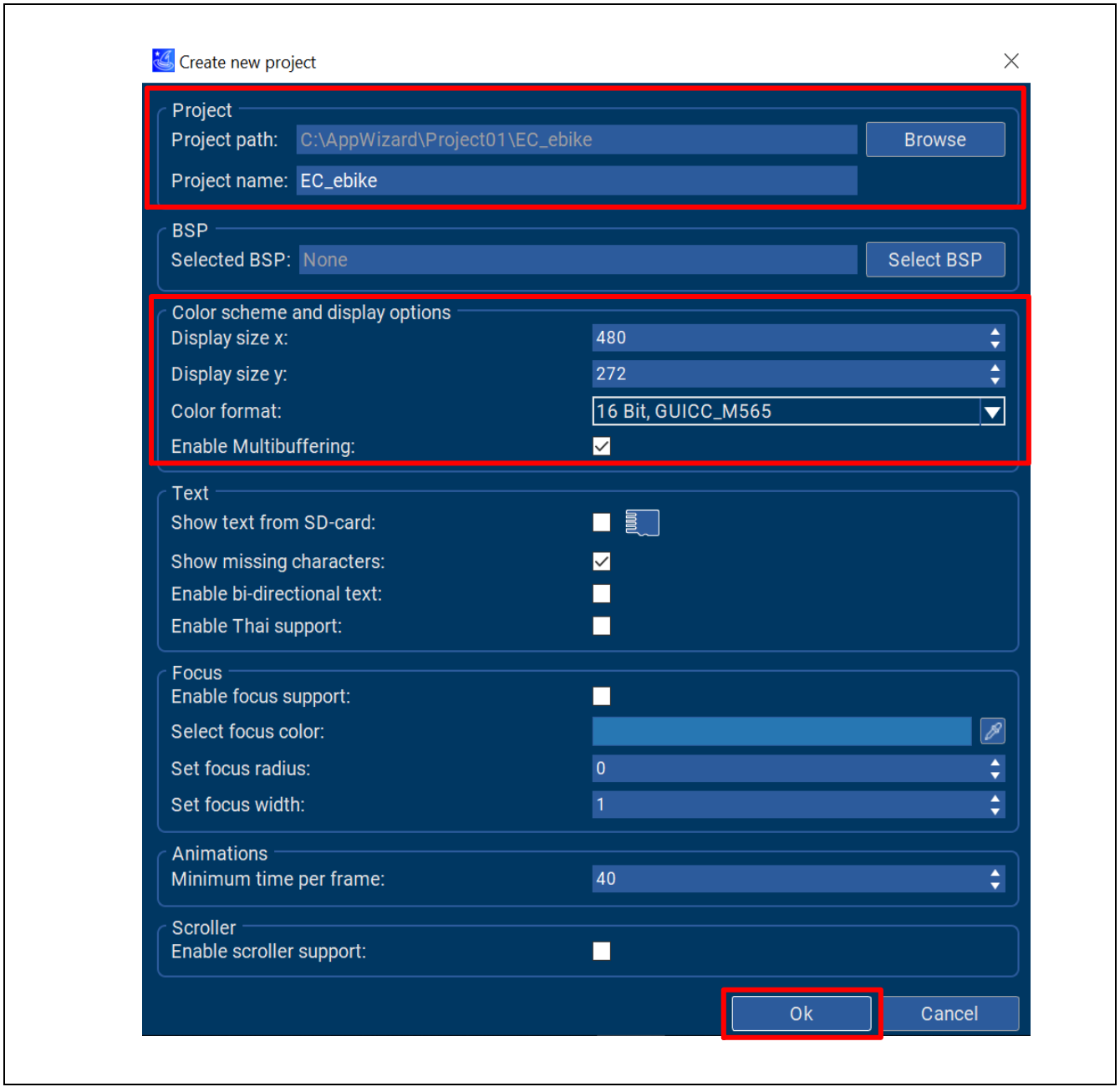


圖 1-6 新專案設定項目

4. 於 Add objects 點選兩次 Screen，以創建兩個畫面，並在 Hierarchic tree 中，為 Screen 分別命名為 ID_SCREEN_Speed 與 ID_SCREEN_Tier，以利辨認。

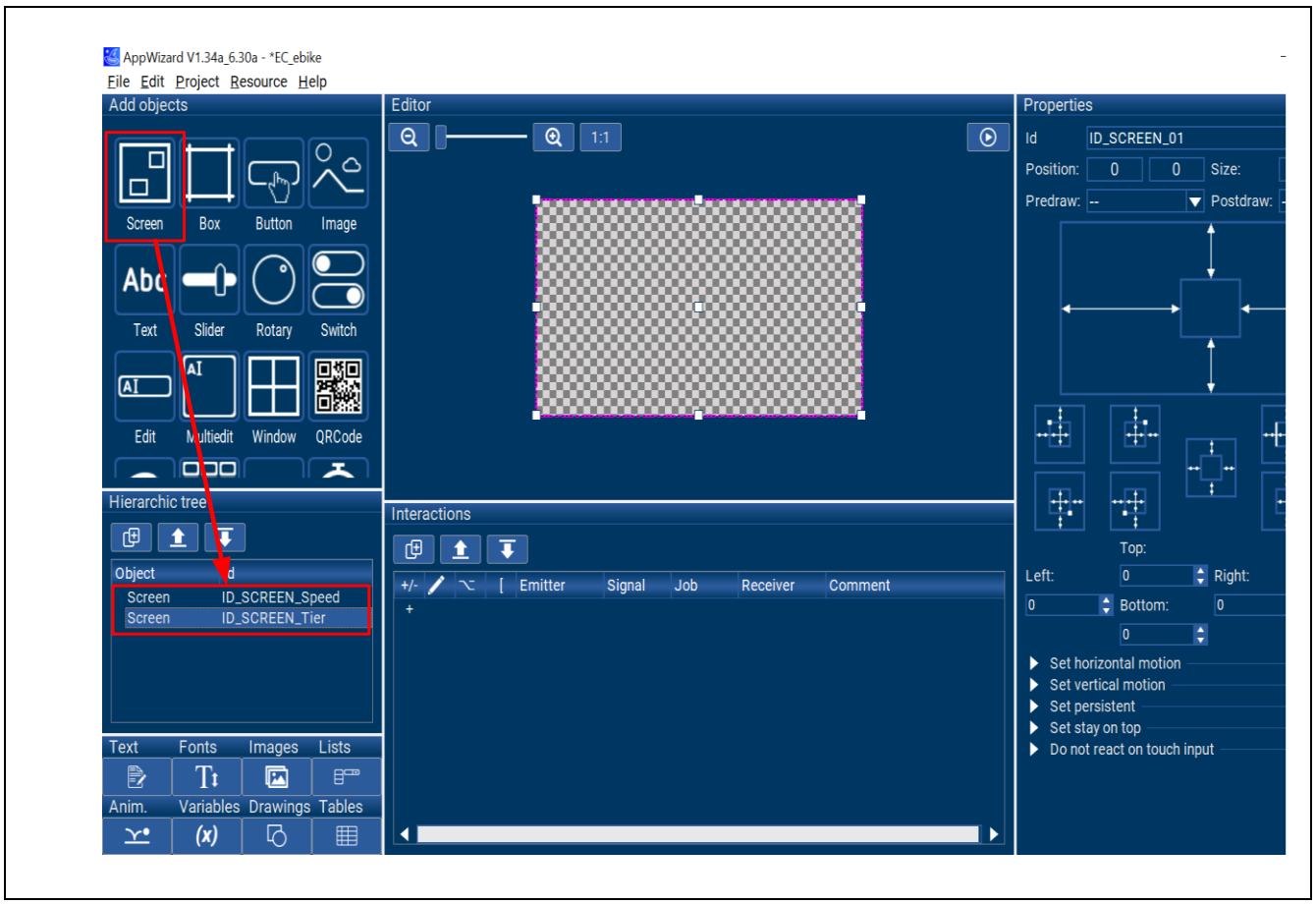


圖 1-7 新增 Screen

5. 點選 ID_SCREEN_Speed 以編輯第一個畫面，於 Add objects 中點選 Box，以創建一個純色的背景。

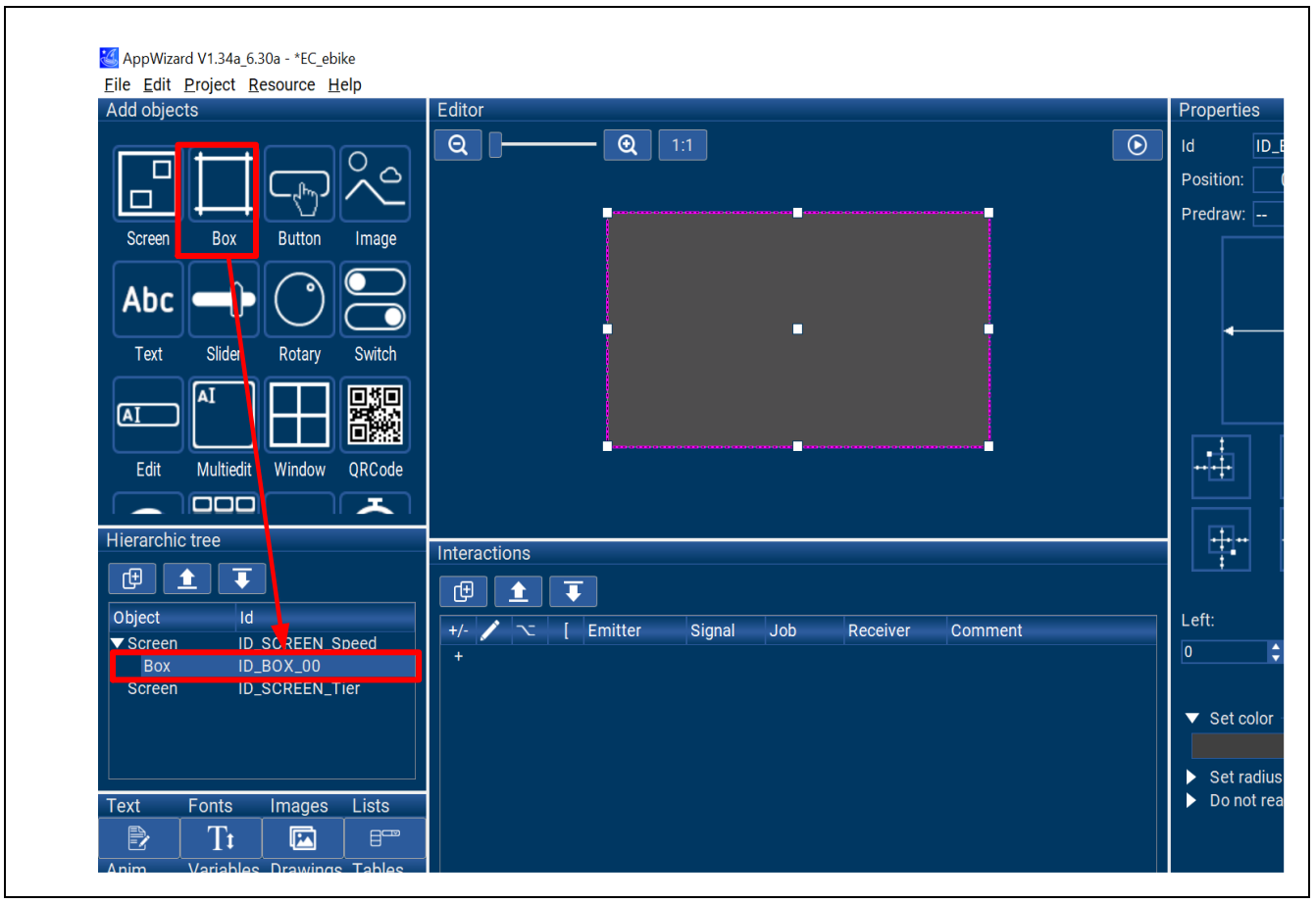


圖 1-8 以 Box 創建素色背景

6. 在 Properties 中，設定 Box 的相關屬性，點選 Set color 設定為黑色。

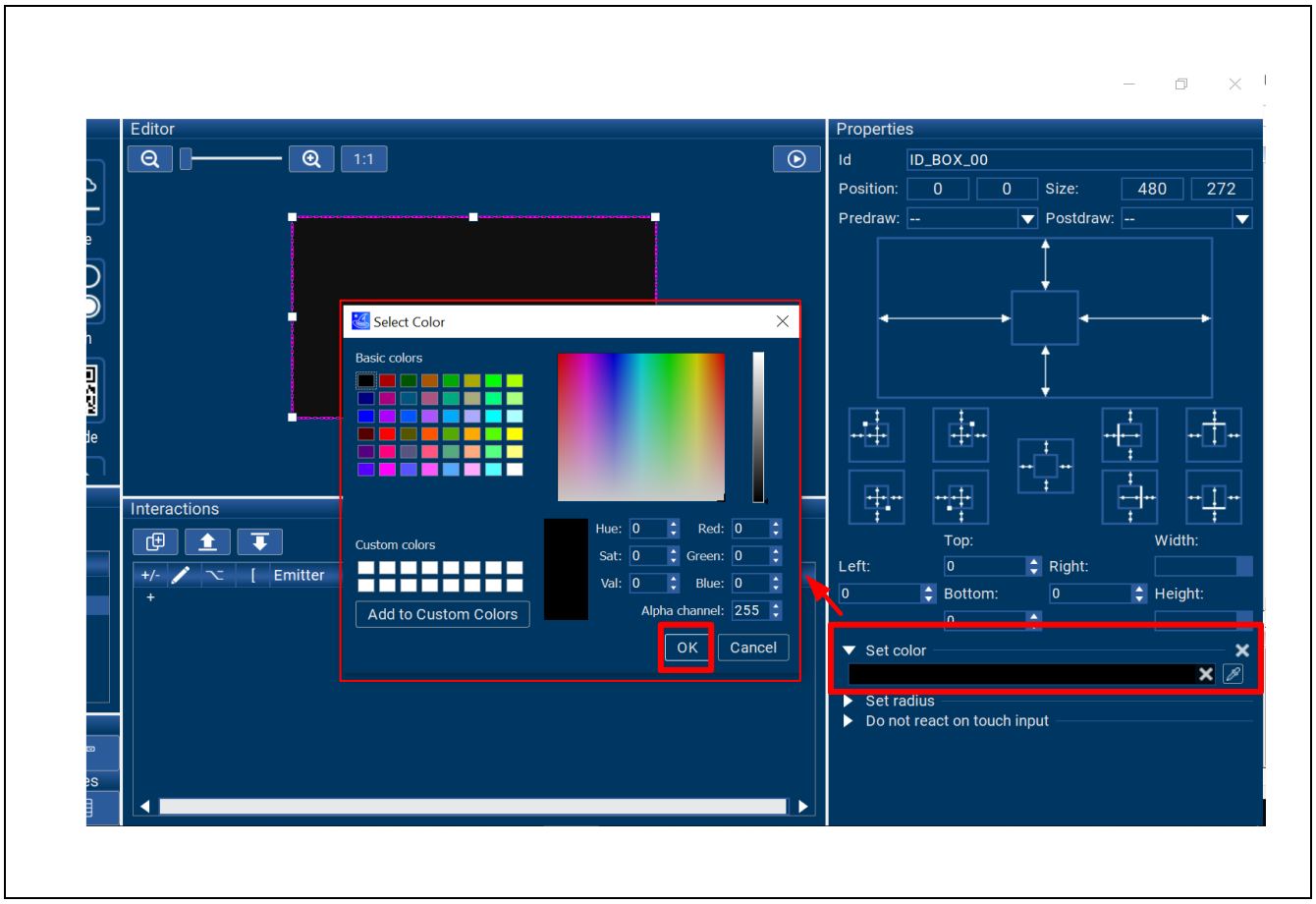


圖 1-9 Box 屬性設定

7. 為製作速度儀表，在 Add objects 中，點選 Rotary，並於 Hierarchic tree 中重新命名為 ID_ROTARY_Speed 以利辨認。

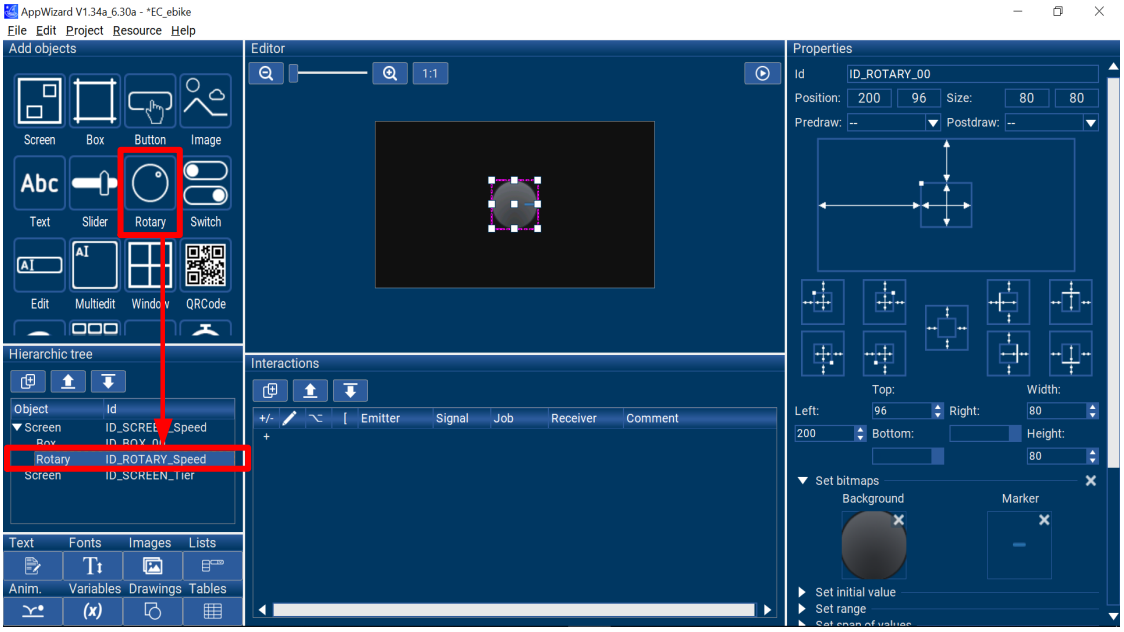


圖 1-10 以 Rotary 創建速度儀表

8. 在 Properties 中，設定 Rotary 的相關屬性：

1) 依圖 1-11，點選 Set bitmaps，為 Background 設定客製化圖片，選擇 Import to project，一次選擇專案會使用的圖片資源。此專案所使用的圖片可參考圖 4-1 的目錄資訊從以下路徑選擇 PNG 圖片檔：EC_M467_AppWizard_Ebike_V1.00\EC_ebike\Resource

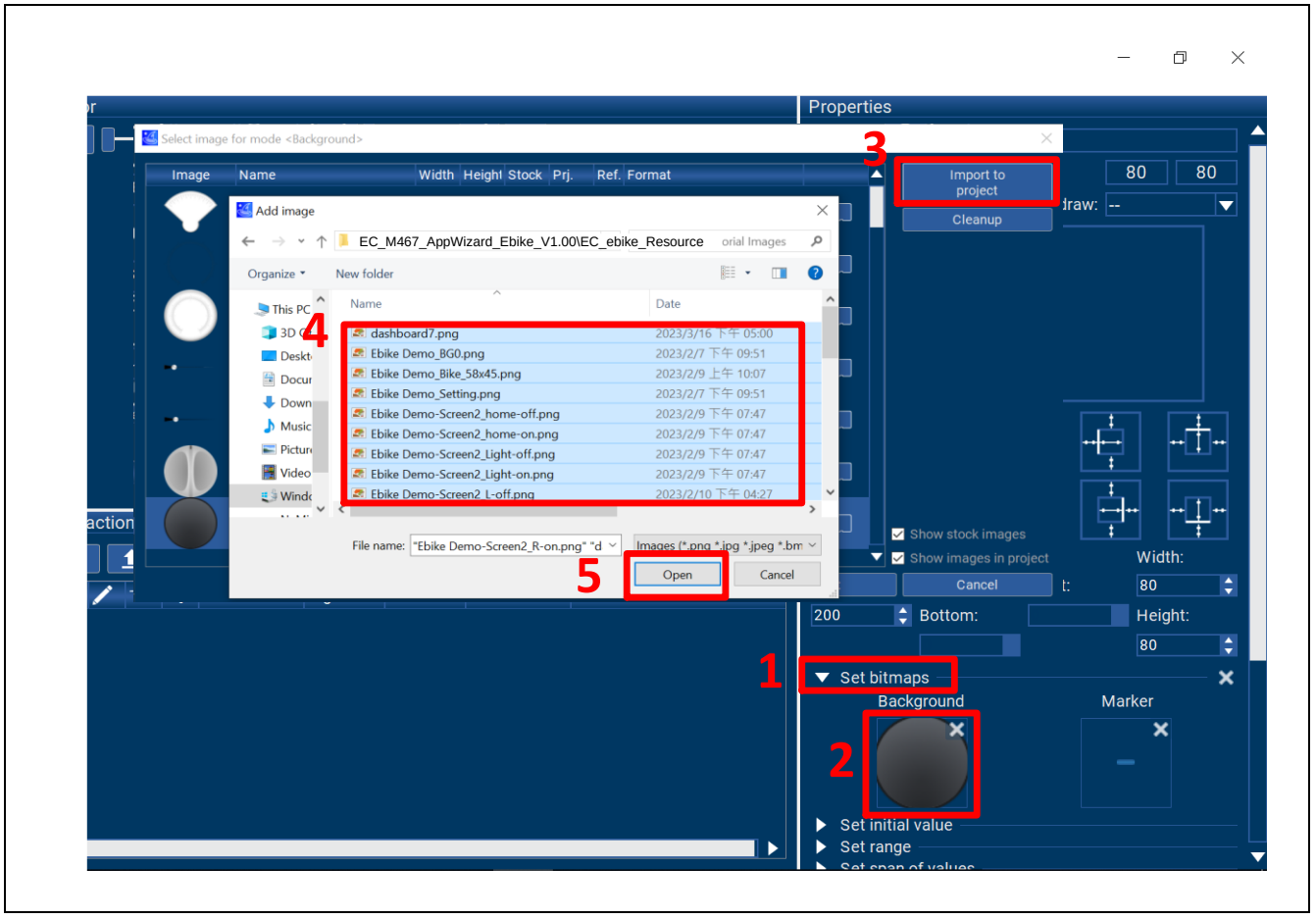


圖 1-11 Rotary 屬性設定-bitmap

2) 依圖 1-12，匯入圖片後，選擇圖片 ，並按下 Select。

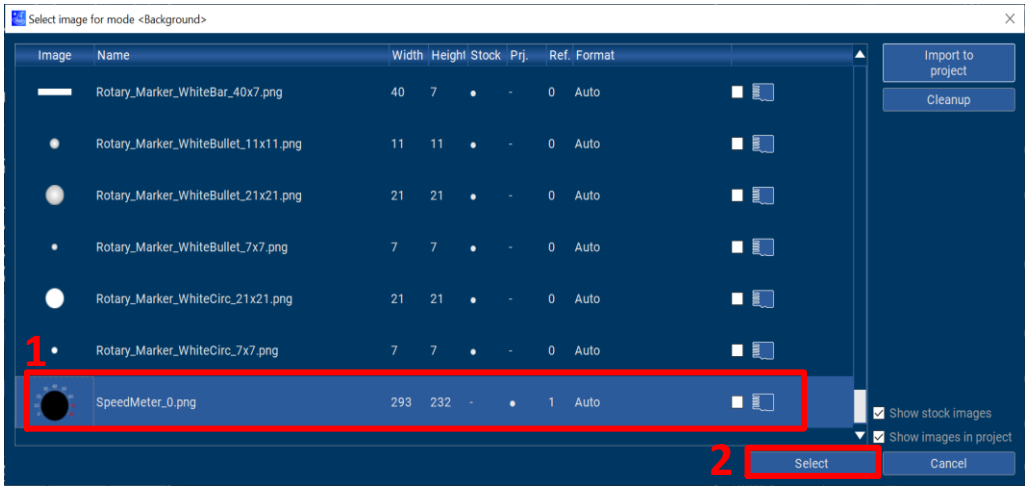


圖 1-12 Rotary 屬性設定-bitmap

3) 依圖 1-13 重覆相同動作，對 Marker 設定 bitmap，選擇圖片 (Rotary_Marker_WhiteBar_20x7.png)。

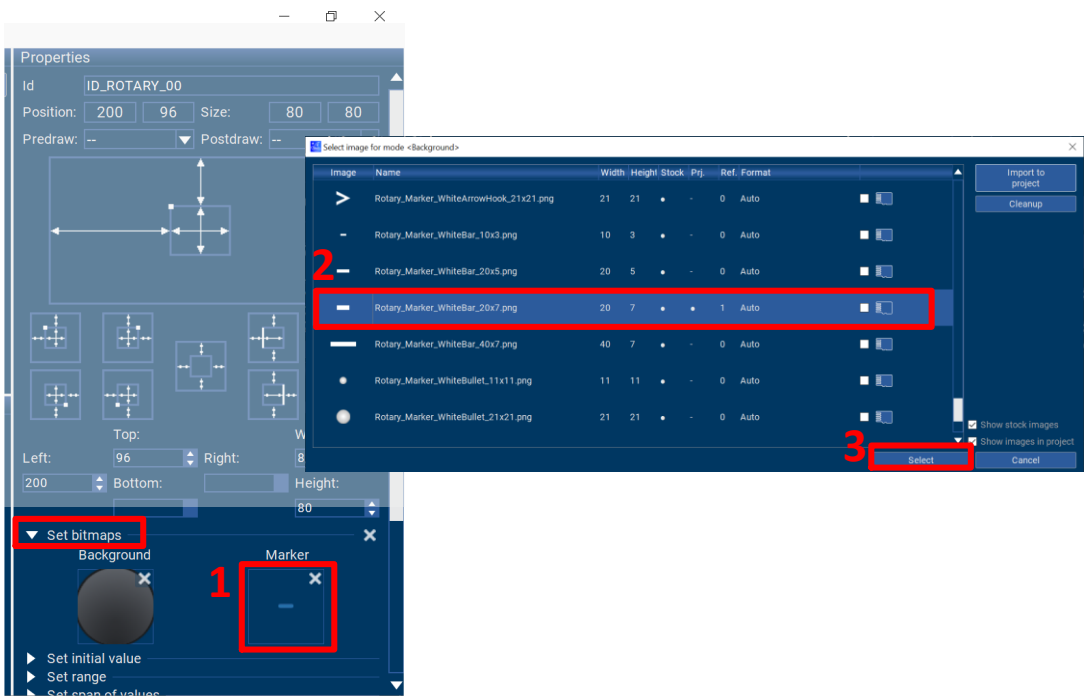


圖 1-13 Rotary 屬性設定-bitmap

4) 因系統設定初始角度 0°位置，如圖 1-14 所示，為符合圖片 的旋轉角度與旋轉值，依照表 1-3 說明，設定 initial value、range、span of values、offset，實際設定項目如圖 1-15。

圖 1-14 Rotary 系統角度設定

Properties	Description	Setting Value
Initial Value	Starting speed value	0
Span of values	Range of speed value	Min:0 Max:80
Range	Range of rotation angle and direction	Positive:0 Negative:2400 (measured in 10th of degrees)
Offset	Starting angle	2100 (measured in 10th of degrees)

表 1-3 Rotary 角度設定說明

5) 為對齊 marker 旋轉位置，依照表 1-4 說明，設定 marker alignment、radius，實際設定項目如圖 1-15。

Properties	Description	Setting Value
Marker alignment	Position of marker	Offset X:1 Offset Y:23
Radius	Radius of rotary	86

表 1-4 Rotary marker 設定說明

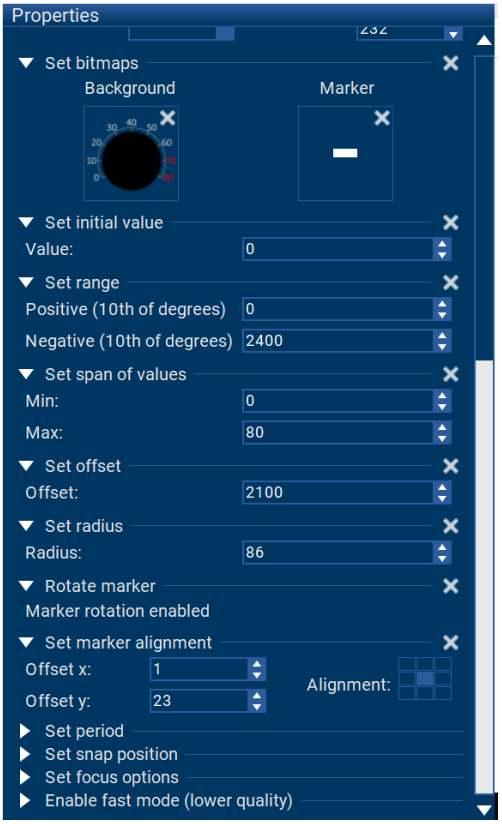


圖 1-15 Rotary 屬性設定

6) 以 Rotary 呈現速度儀表，設定完成後如圖 1-16 所示。

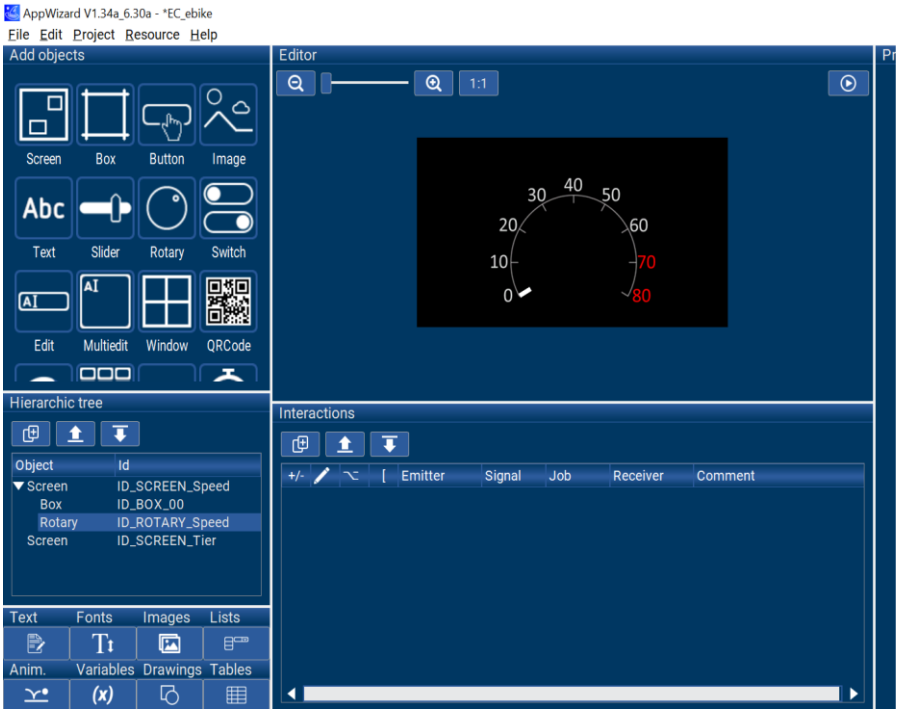


圖 1-16 以 Rotary 完成速度儀表

9. 增加數字以顯示速度數值，在 Add objects 中，點選 Text，並於 Hierarchic tree 中重新命名為 ID_TEXT_Speed 以利辨認。

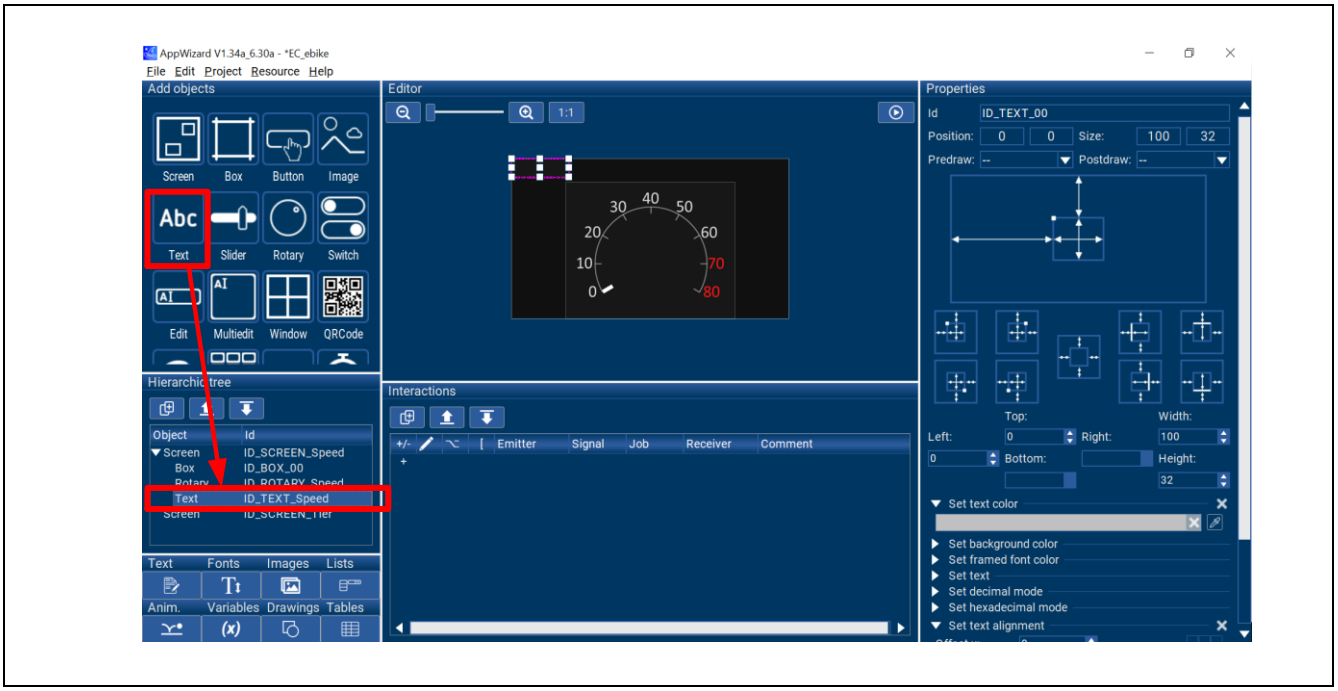


圖 1-17 以 TEXT 創建速度數值

10. 在 Properties 中，設定 TEXT 的相關屬性：

    1) 設定顏色、decimal mode、text alignment、數值顯示範圍(range)，實際設定項目如圖 1-19 所示。

    2) 依圖 1-18 步驟，設定字型(font) 。

圖 1-18 TEXT 屬性設定-字型

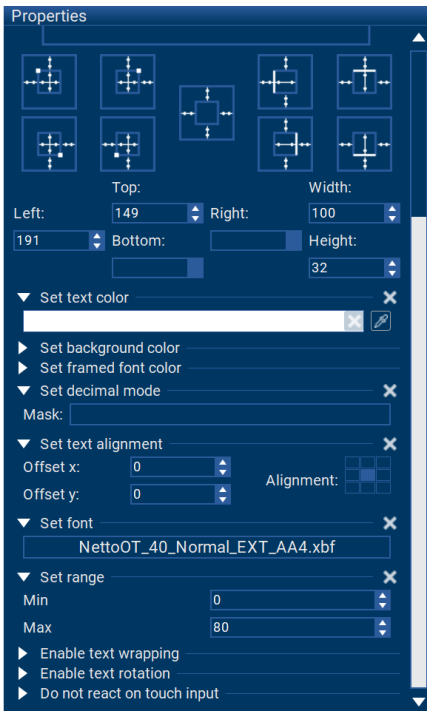


圖 1-19 TEXT 屬性設定

3) 以 TEXT 呈現速度數值，設定完成後如圖 1-20 所示。

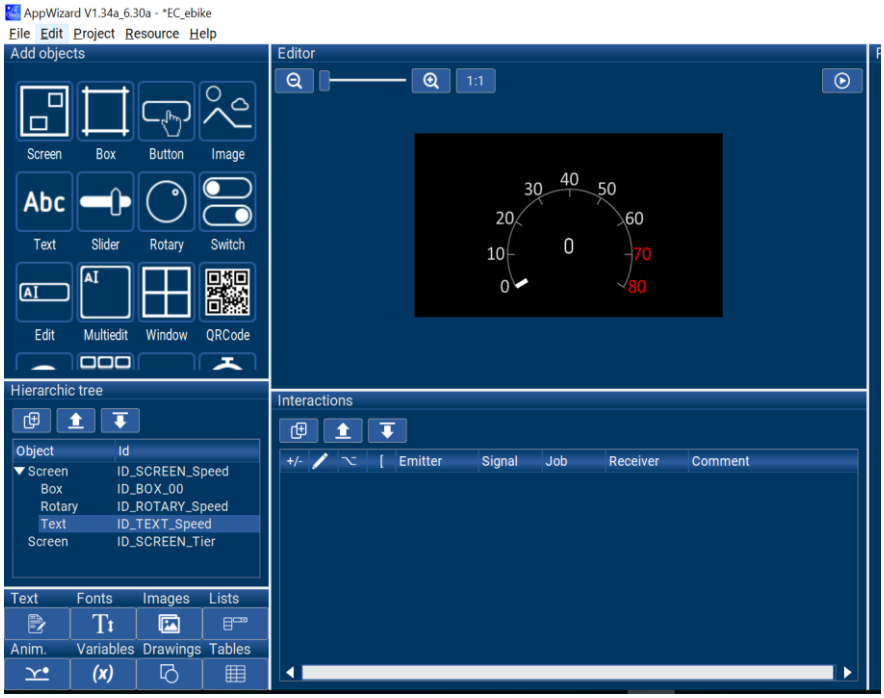


圖 1-20 以 TEXT 完成速度數值顯示

11. 增加文字以顯示數值名稱、數值單位、標題等，在 Add objects 中，點選 Text。

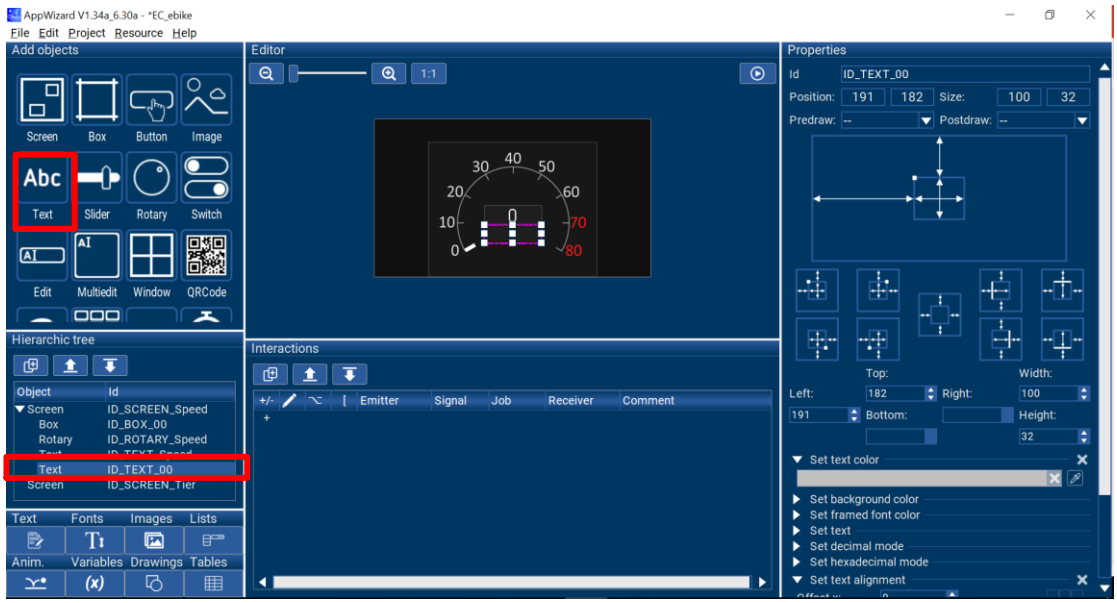


圖 1-21 以 TEXT 創建速度單位文字

1) 設定顏色、text alignment。

2) 依圖 1-22~圖 1-26 步驟，設定顯示的文字，並同時顯示中文與英文。

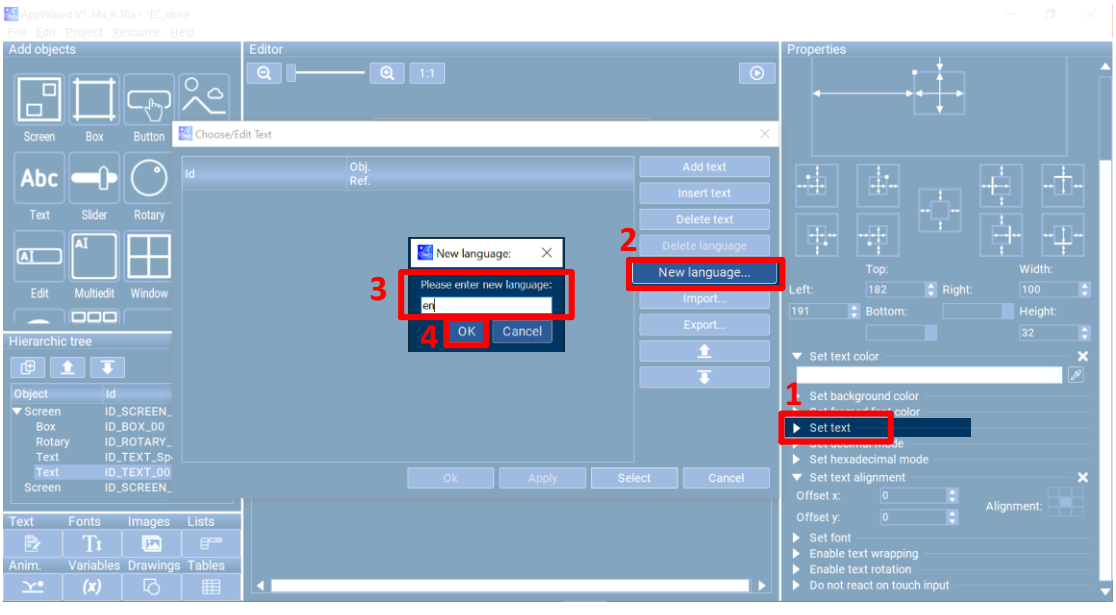


圖 1-22 TEXT-創建第一種語言-英文

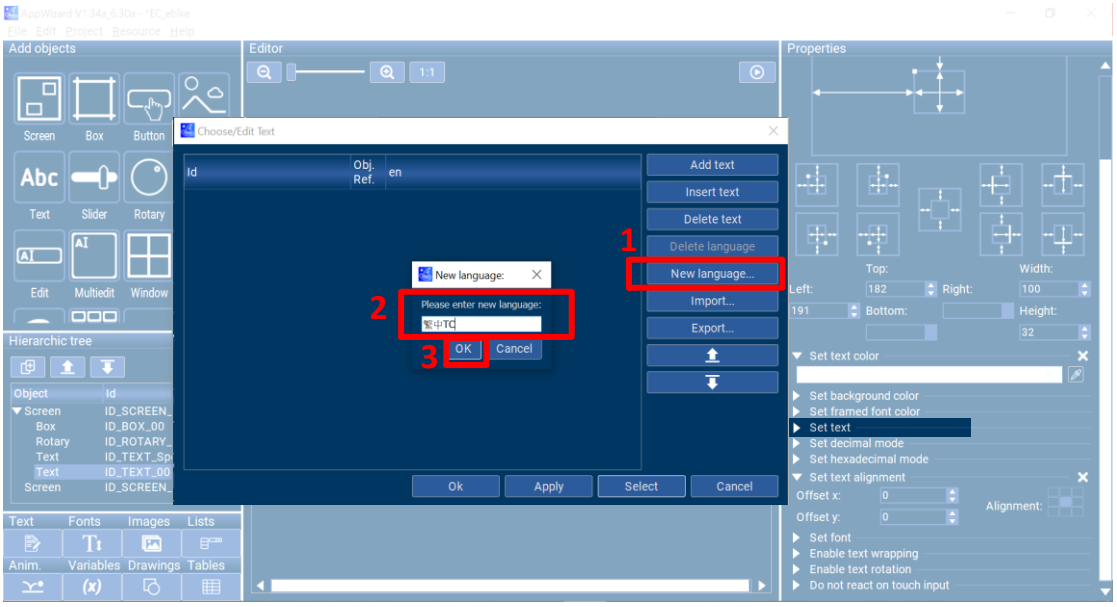


圖 1-23 TEXT-創建第二種語言-中文

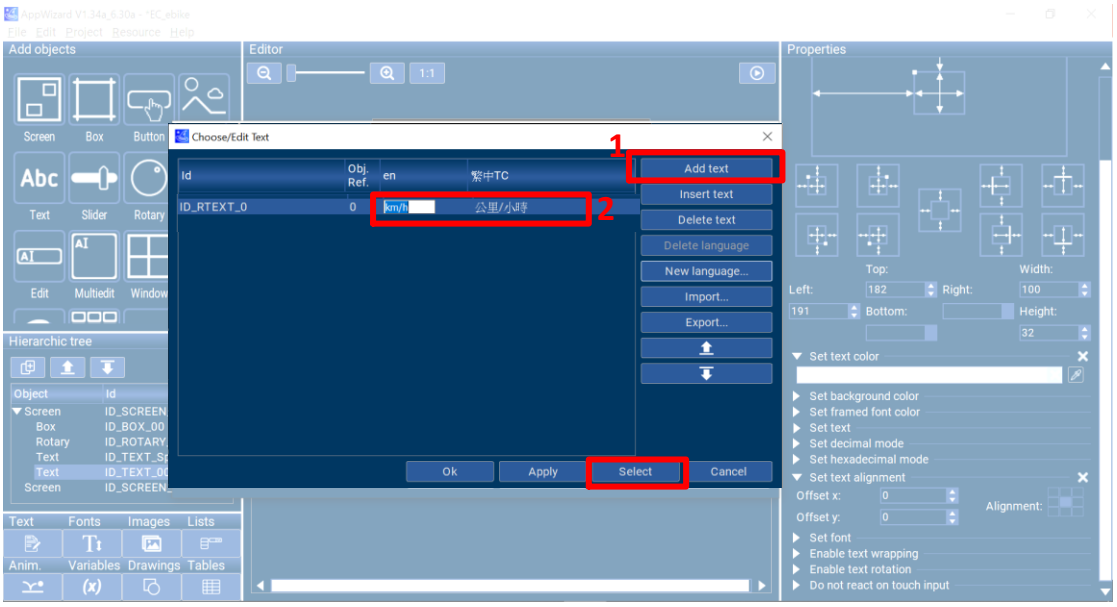


圖 1-24 TEXT- 分別編輯中文與英文的文字內容

3) 新增並選擇能同時顯示英文與中文的字型(font) 。

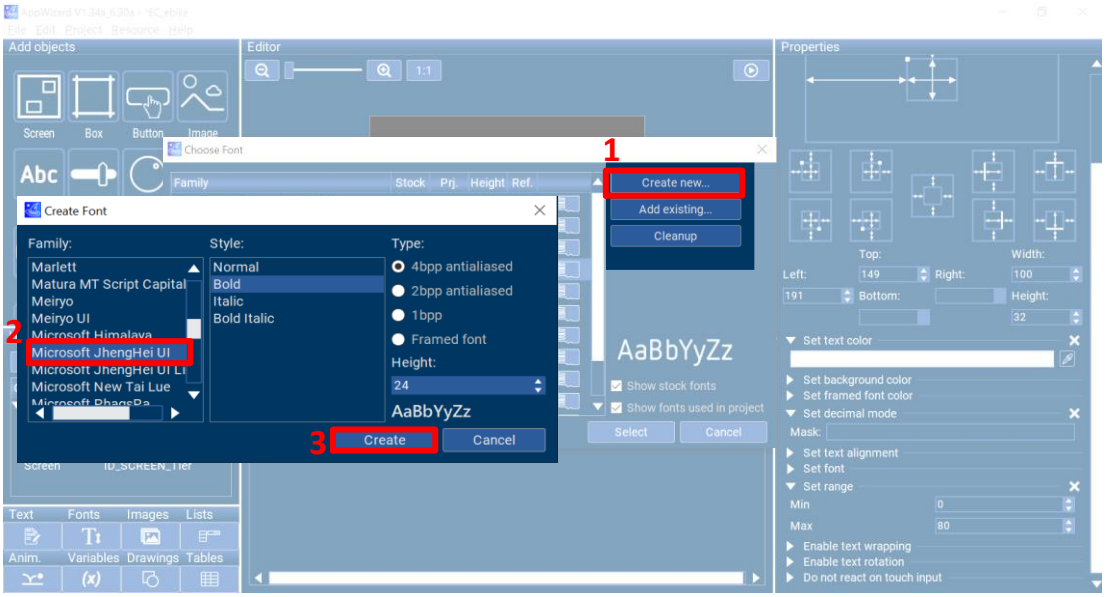


圖 1-25 TEXT 屬性設定-字型

4) 設定 codepoint range，跳出視窗”Define codepoints to be used”後，勾選 Project text only 即僅選擇專案會使用到的字符範圍。

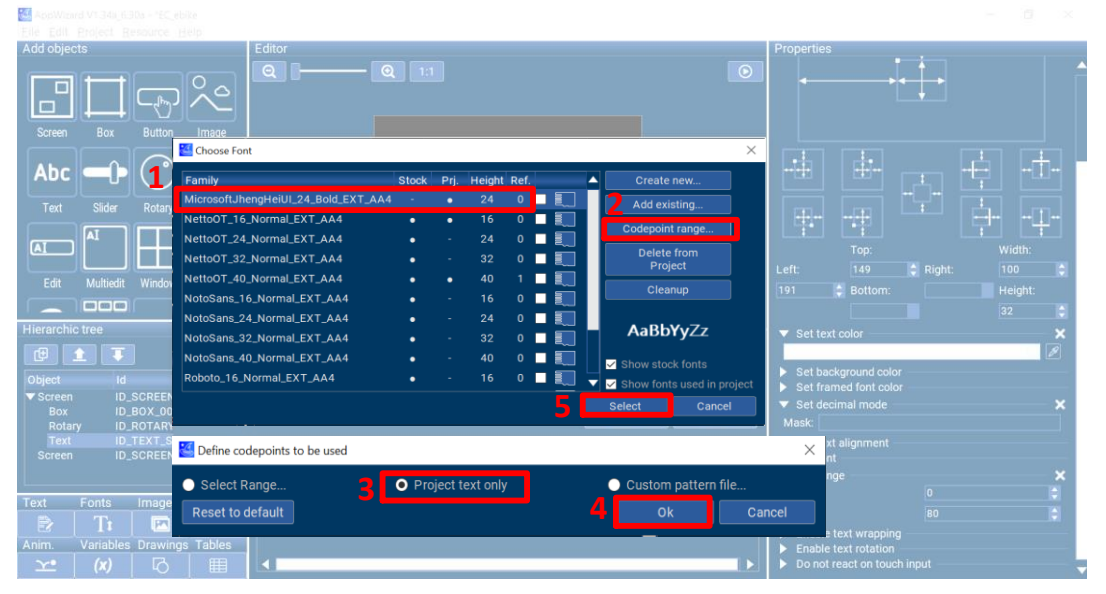


圖 1-26 TEXT 屬性設定-字型與字符範圍

5) 以 TEXT 呈現英文與中文文字，設定完成後如圖 1-27 所示。

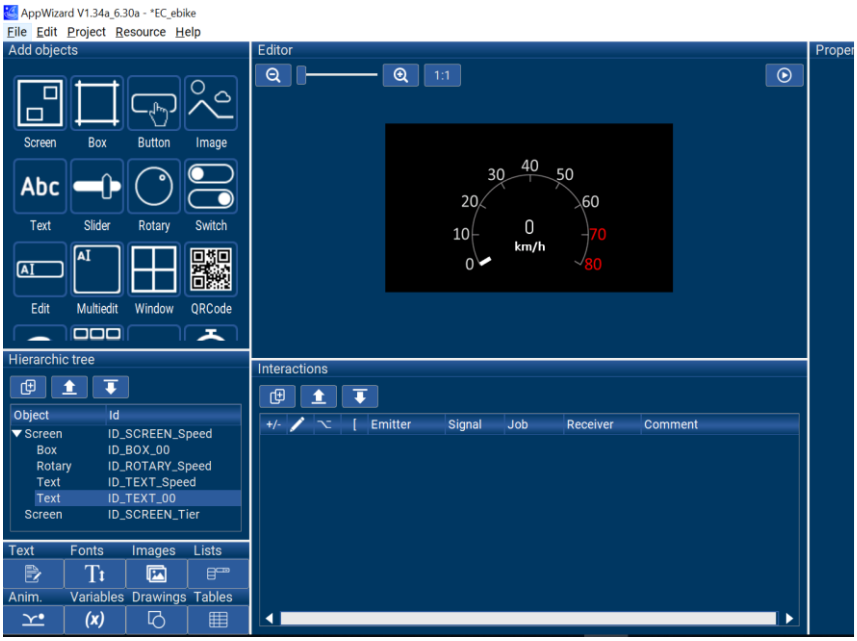


圖 1-27 以 TEXT 完成速度單位顯示

12. 使用 Image 物件，新增分隔線至畫面中，在 Add objects 中，點選 Image。

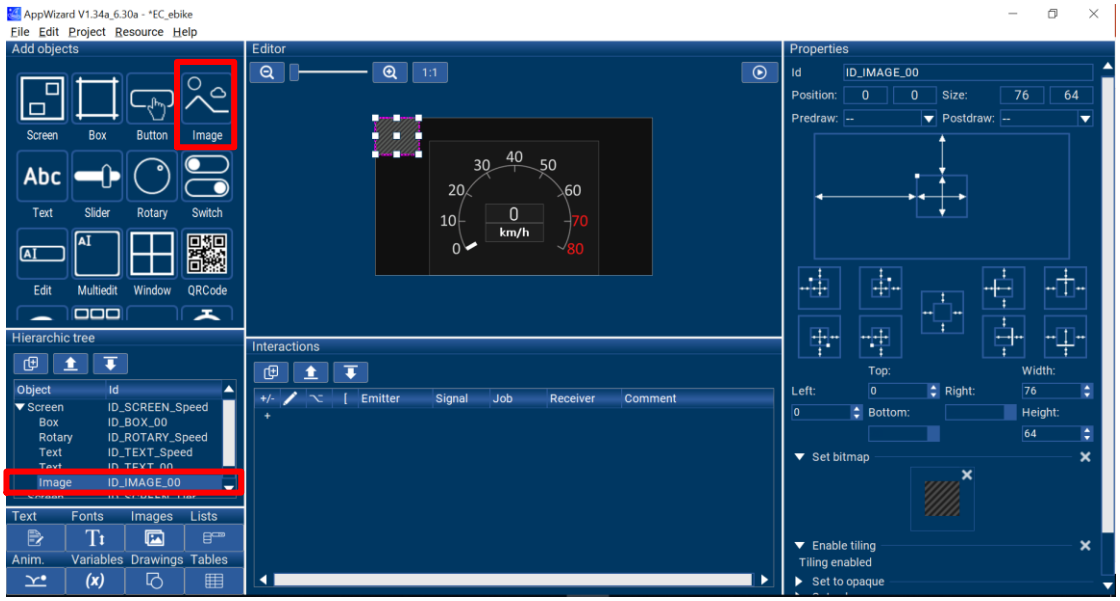


圖 1-28 以 Image 新增畫面分隔線

13. 在 Properties 中，設定 Image 的相關屬性：

1) 依圖 1-29，點選 Set bitmap，設定客製化圖片，並關閉 Enable tiling 功能，取消無限延伸。

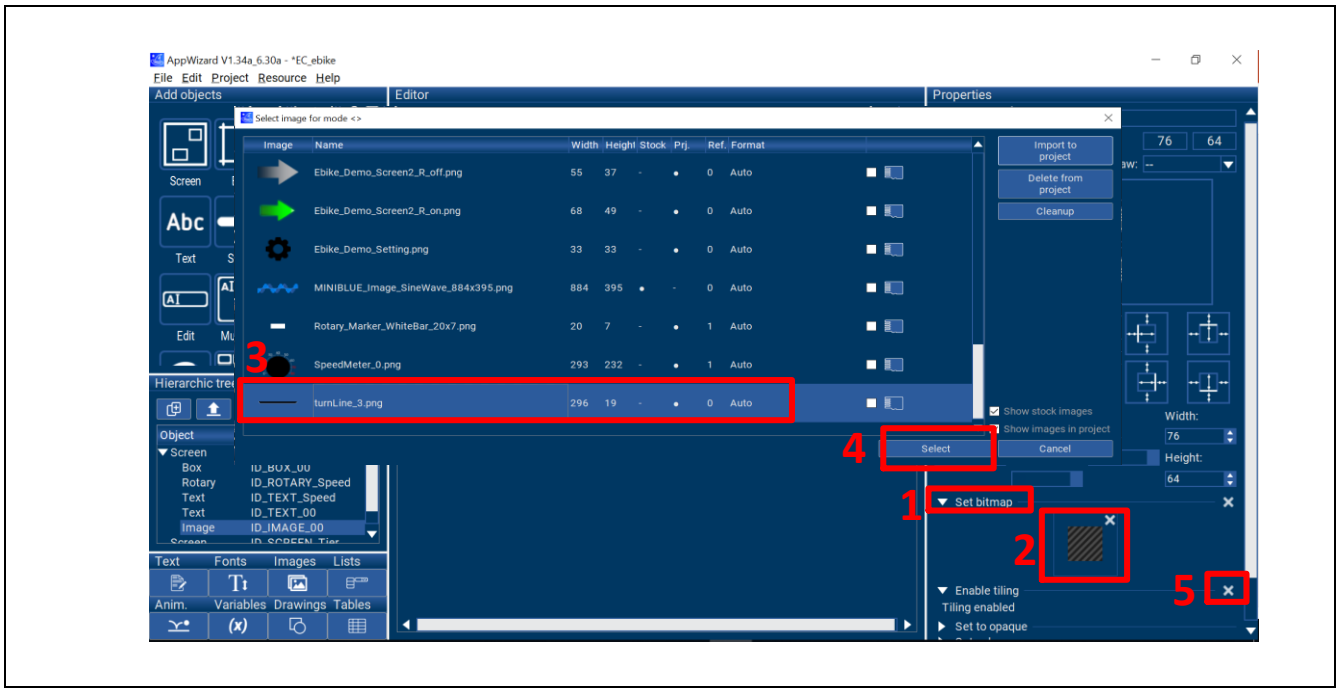


圖 1-29 Image 屬性設定 - bitmap

2) 設定完成如圖 1-30 所示。

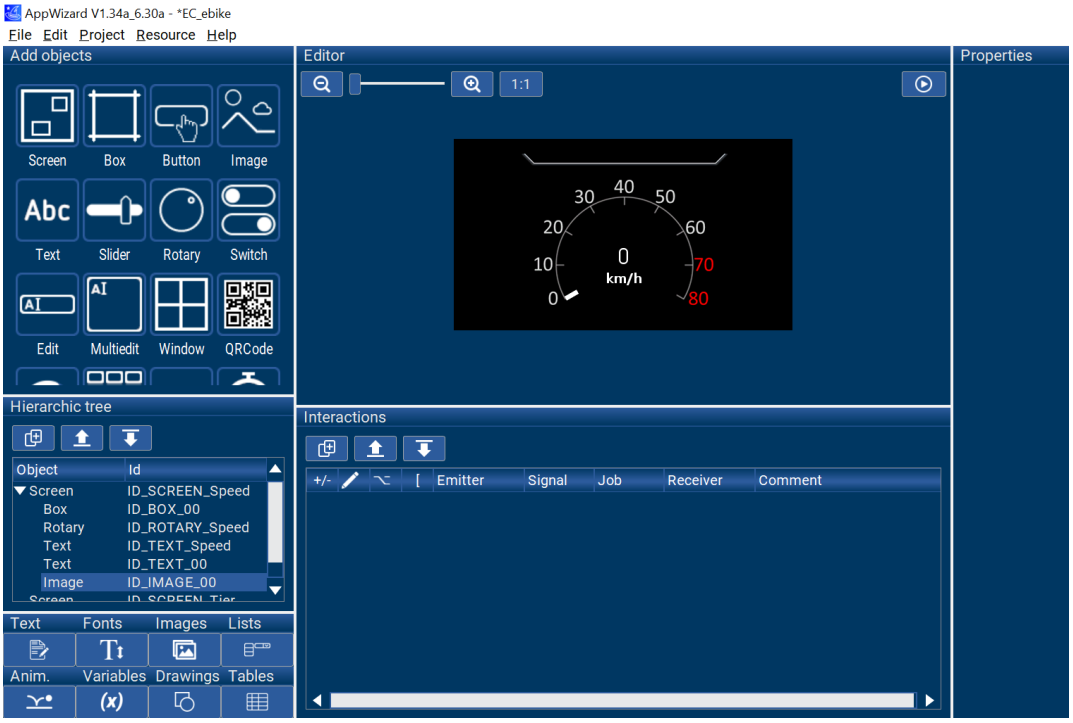


圖 1-30 以 Image 完成畫面分隔線

14. 使用 button 物件表示方向燈功能，在 Add objects 中，點選 Button ，並於 Hierarchic tree 中重新命名為 “ID_BUTTON_Left” 以利辨認。

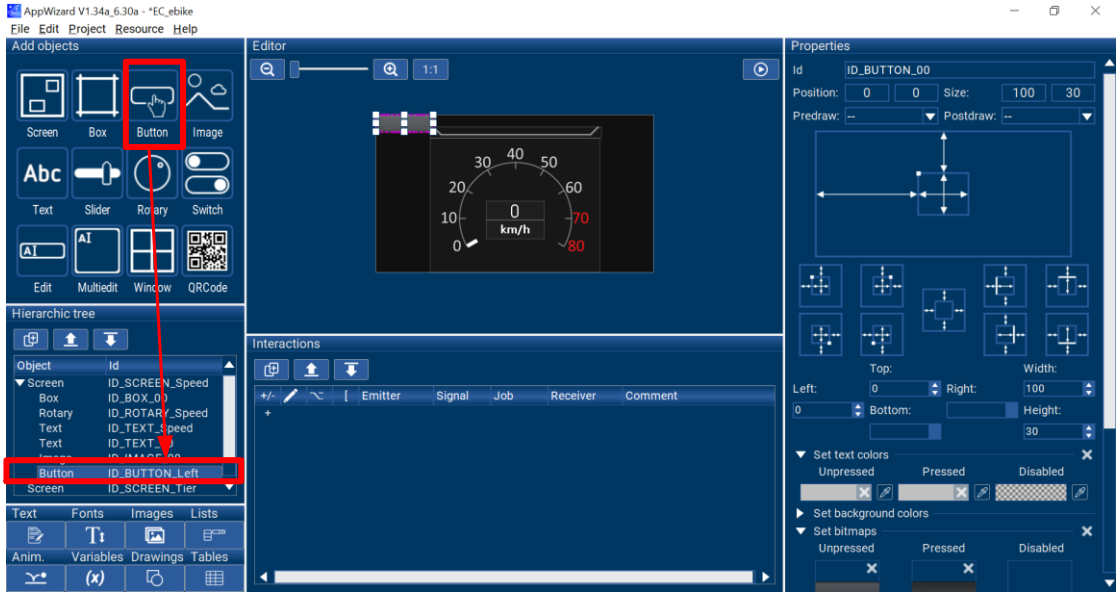
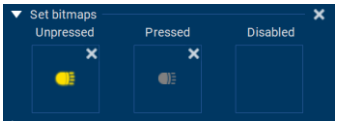


圖 1-31 以 Button 設計方向燈

15. 在 Properties 中，設定 Button 的相關屬性：

1) 此次範例會使用到的功能僅需設定按鈕未按下(Unpressed)、按下(Pressed)等不同狀態的 bitmap，bitmap 選擇方式可參考步驟 13，此範例分別對左右方向燈、警示燈、大燈警示燈設定對應圖片。

BUTTON ID	Bitmap
ID_BUTTON_Left	
ID_BUTTON_Right	
ID_BUTTON_Light	

ID_BUTTON_Warn	

表 1-5 點陣圖

2) 設置完成並調整位置，如圖 1-32 所示。

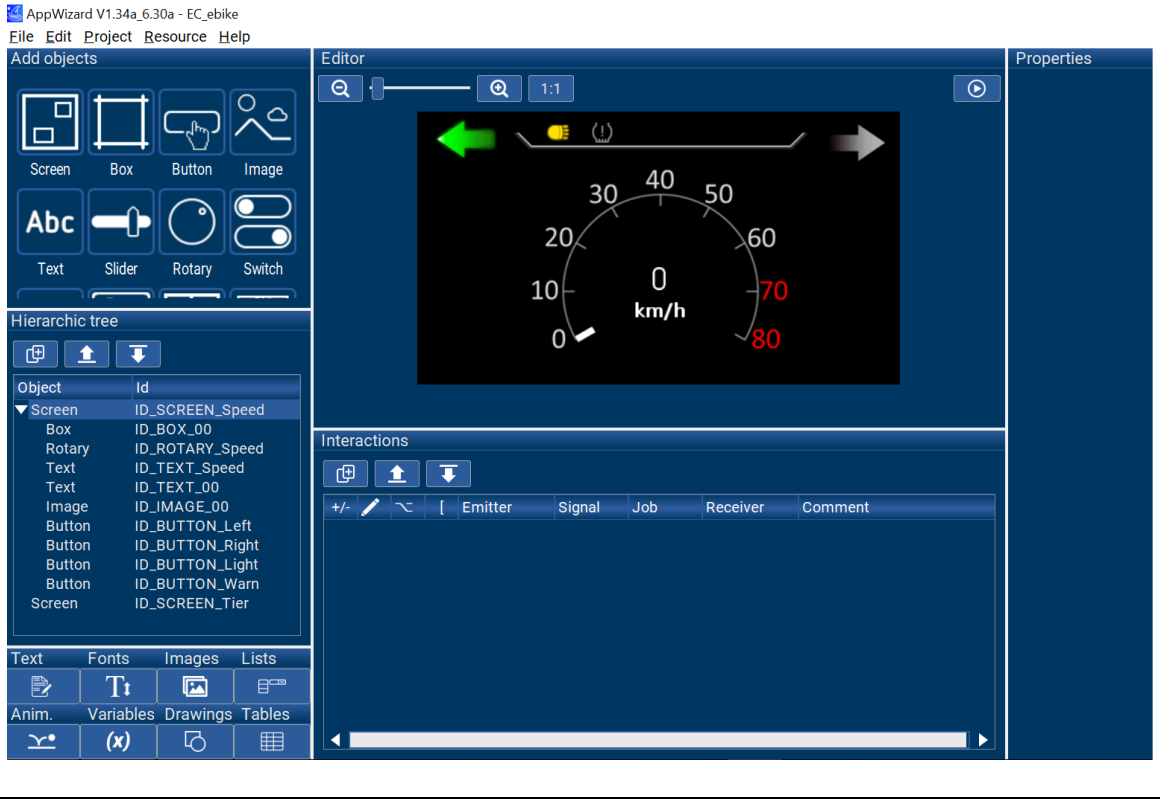


圖 1-32 以 Button 完成大燈、警示燈與方向燈

16. 新增”跳至設定頁”按鈕，不同於步驟 14 使用圖片(bitmap)作為按鈕的介面，此按鈕將不使用圖片資源，在 Add objects 中，點選 Button ，並於 Hierarchic tree 中重新命名為 ID_BUTTON_Setting 以利辨認。

    1) 設定文字顏色、背景顏色、取消 bitmap 設定、設定文字與字型，設定內容如圖 1-33。

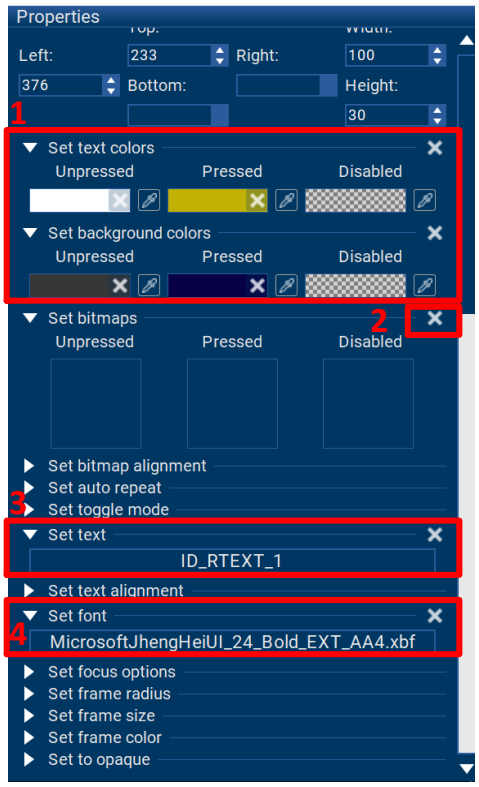


圖 1-33 Button 屬性設定

17. 設定並調整完位置，如圖 1-34。

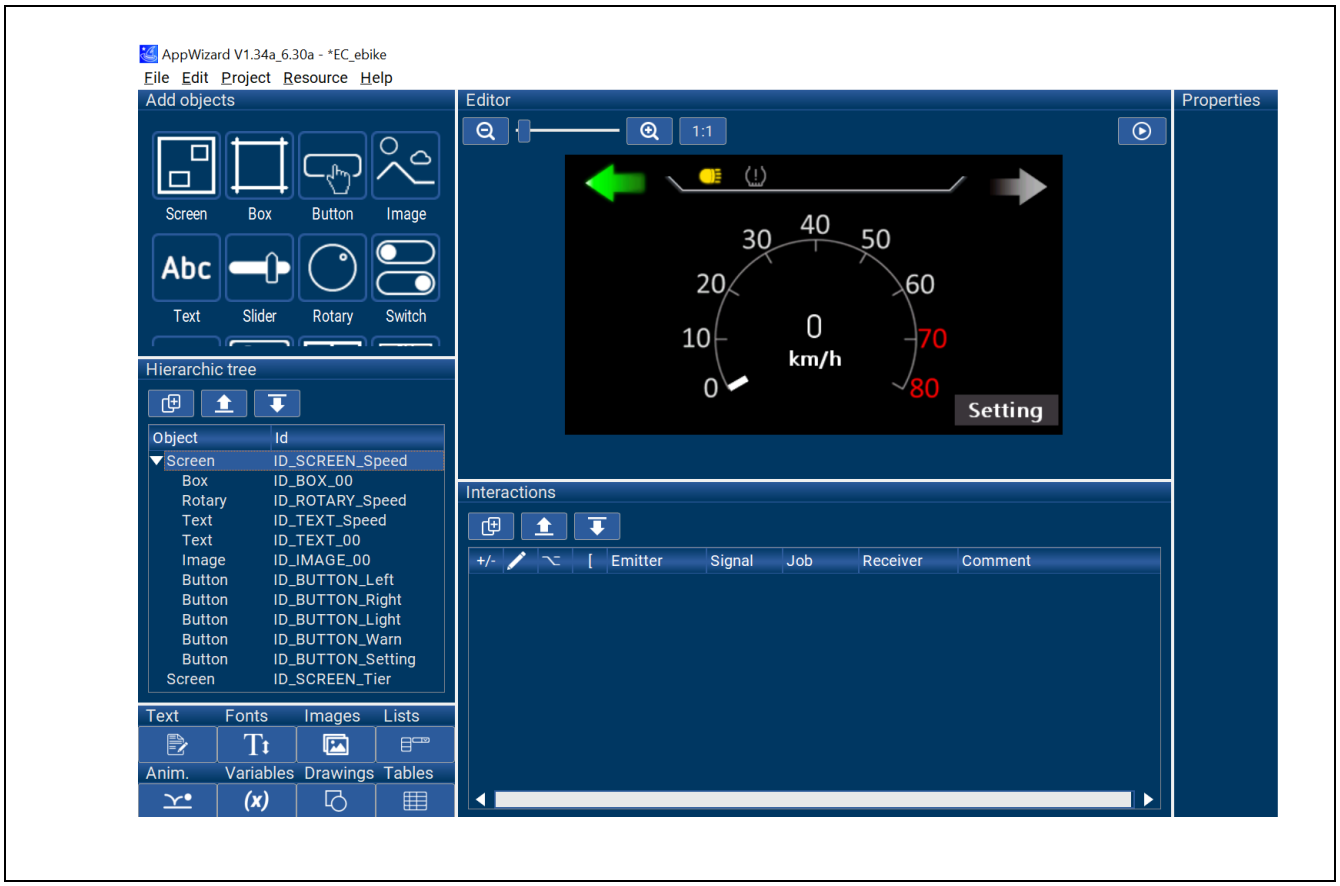


圖 1-34 以 Button 完成頁面跳轉按鈕

18. 使用 Progbar 物件，作為顯示電池電量功能，在 Add objects 中，點選 Progbar ，並於 Hierarchic tree 中重新命名為 ID_Progbar_Battery 以利辨認。

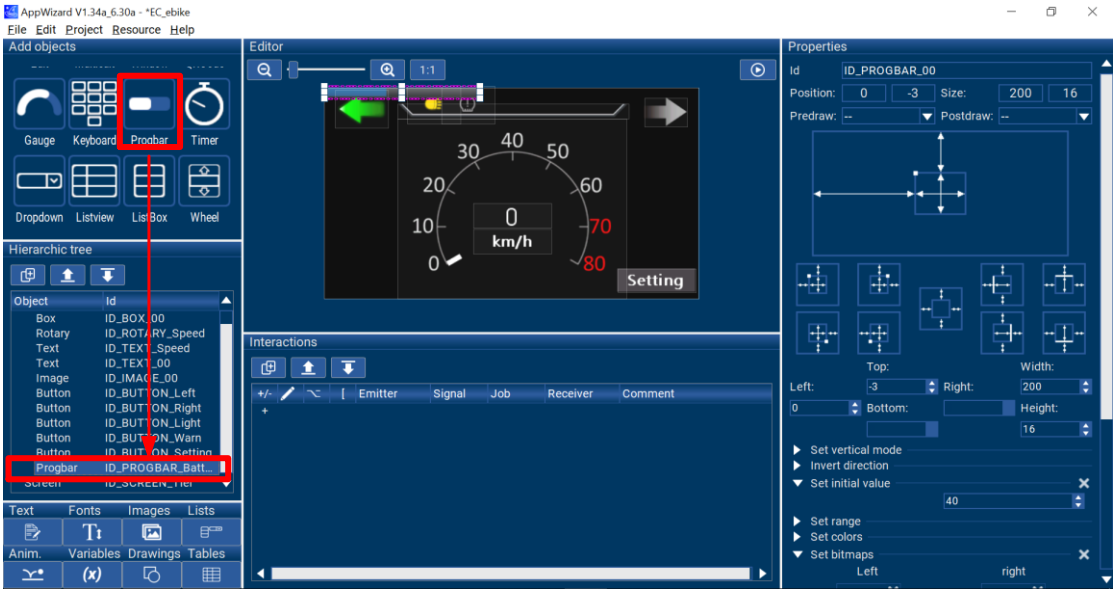


圖 1-35 以 Progbar 設計電池電量顯示

19. 在 Properties 中，設定 Progbar 的相關屬性

1) 依順序並如圖 1-36 所示，設定初始值(initial value)、數值範圍(range)、顏色(color)、取消 bitmap 功能、取消 tiling 功能、設定半徑(radius)、設定邊框大小(frame size)、設定邊框顏色(frame color) 。

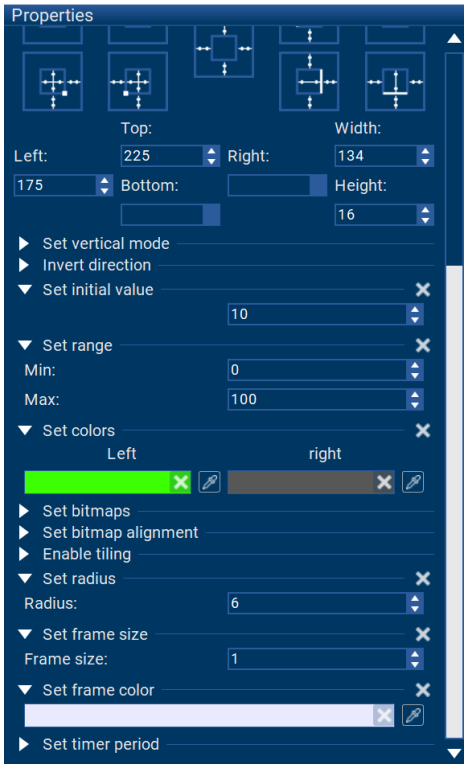


圖 1-36 Progbar 屬性設定

2) 設定並調整完位置，如圖 1-37。

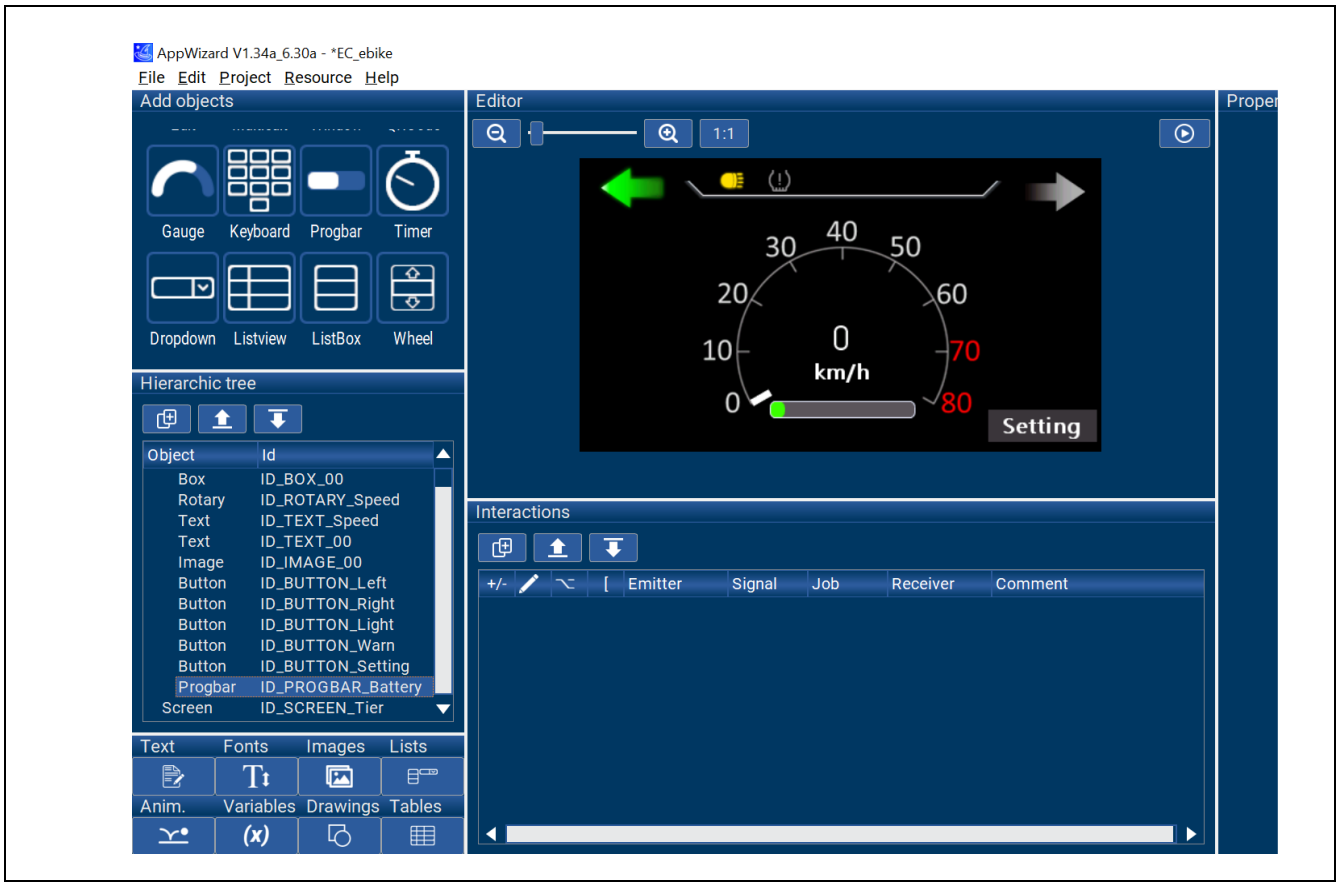


圖 1-37 以 Progbar 完成電量顯示條

20. 使用 TEXT，新增電池說明文字、單位，請依照步驟 11 操作，此範例相關設定可參考圖 1-38，完成後如圖 1-39 所示。

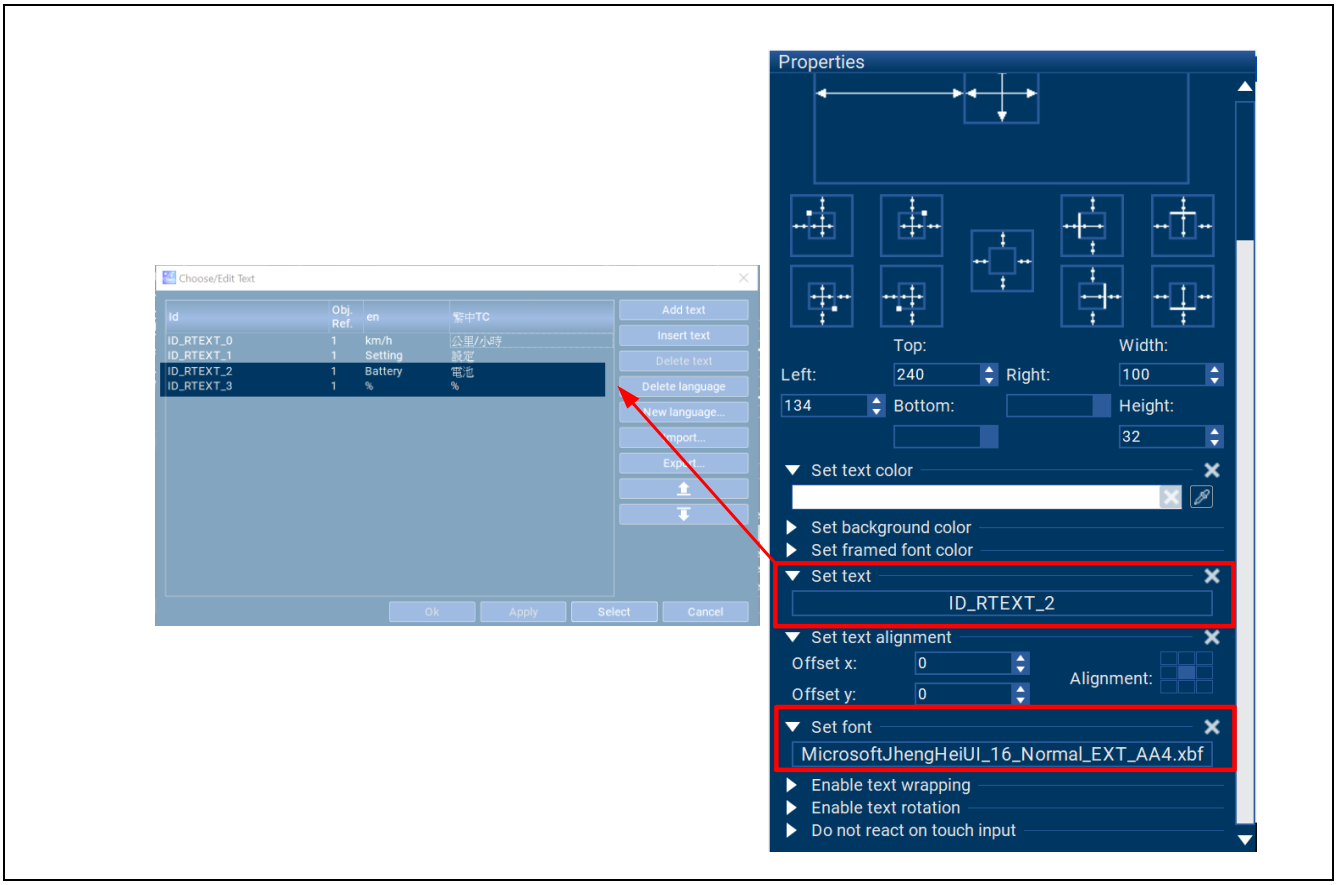


圖 1-38 TEXT 屬性設定

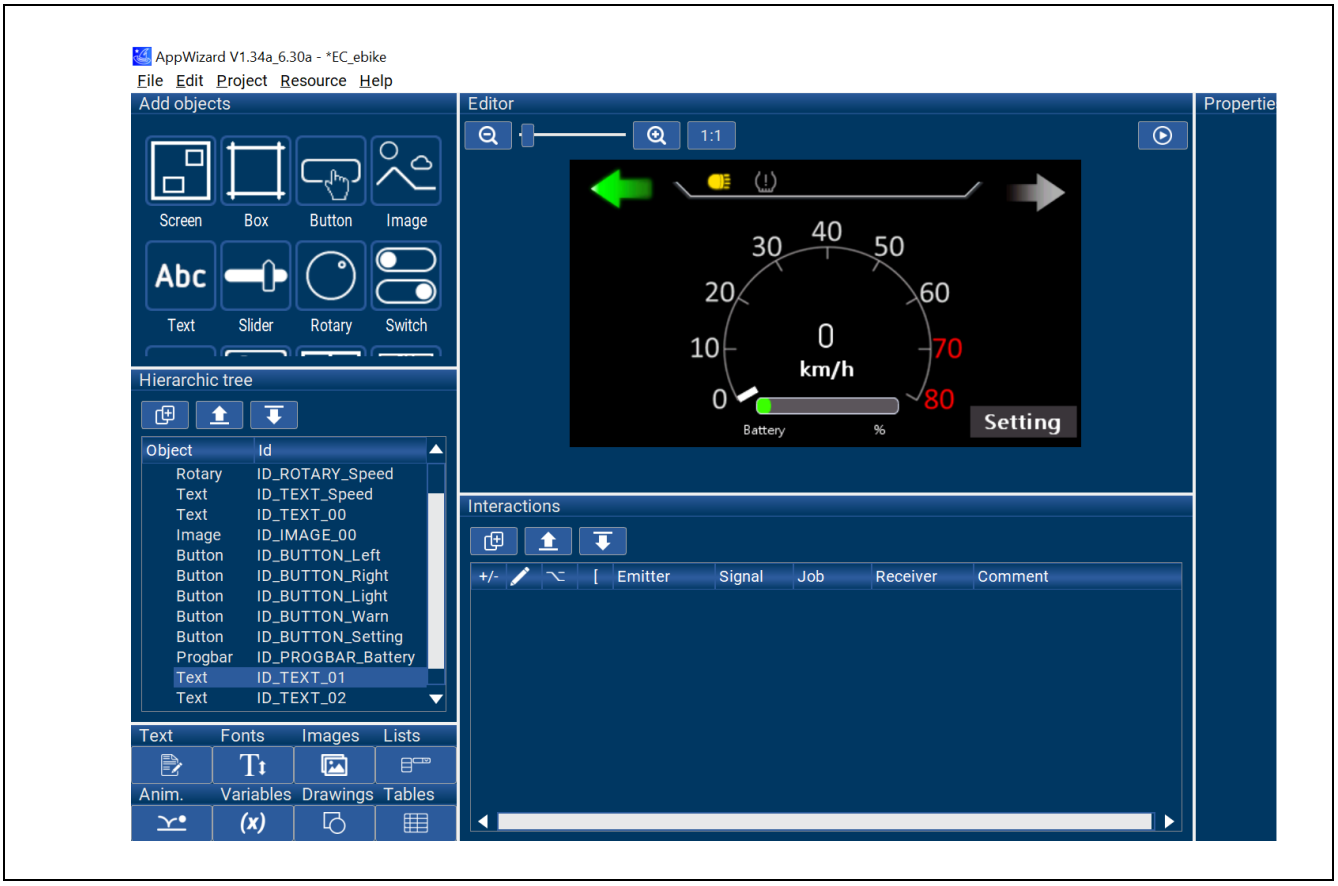


圖 1-39 以 TEXT 完成電池文字與單位顯示

21. 使用 TEXT，新增電池電量數值顯示，請依照步驟 9 ~ 步驟 10 操作，此範例相關設定可參考圖 1-40，完成後如圖 1-41 所示。

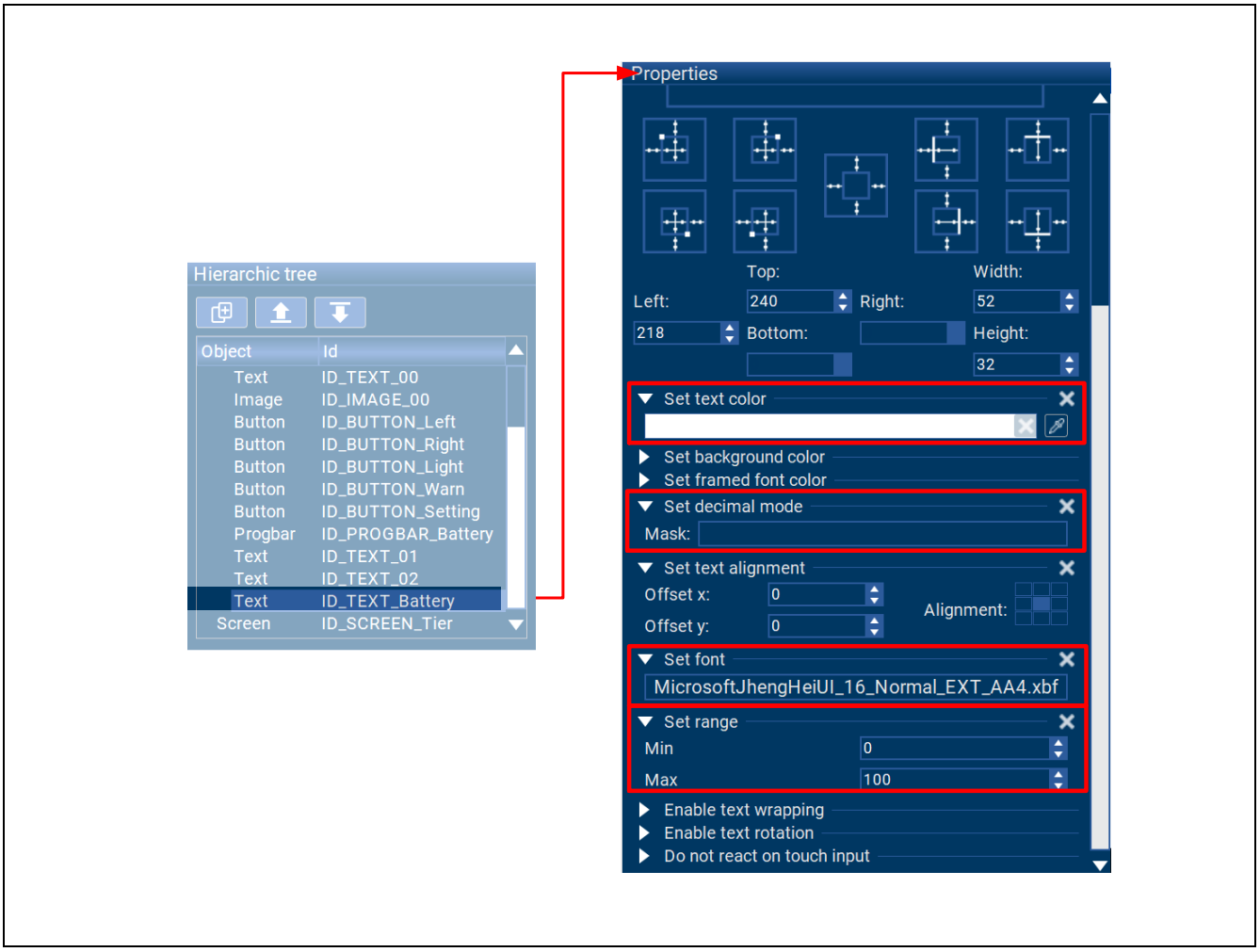


圖 1-40 TEXT 屬性設定

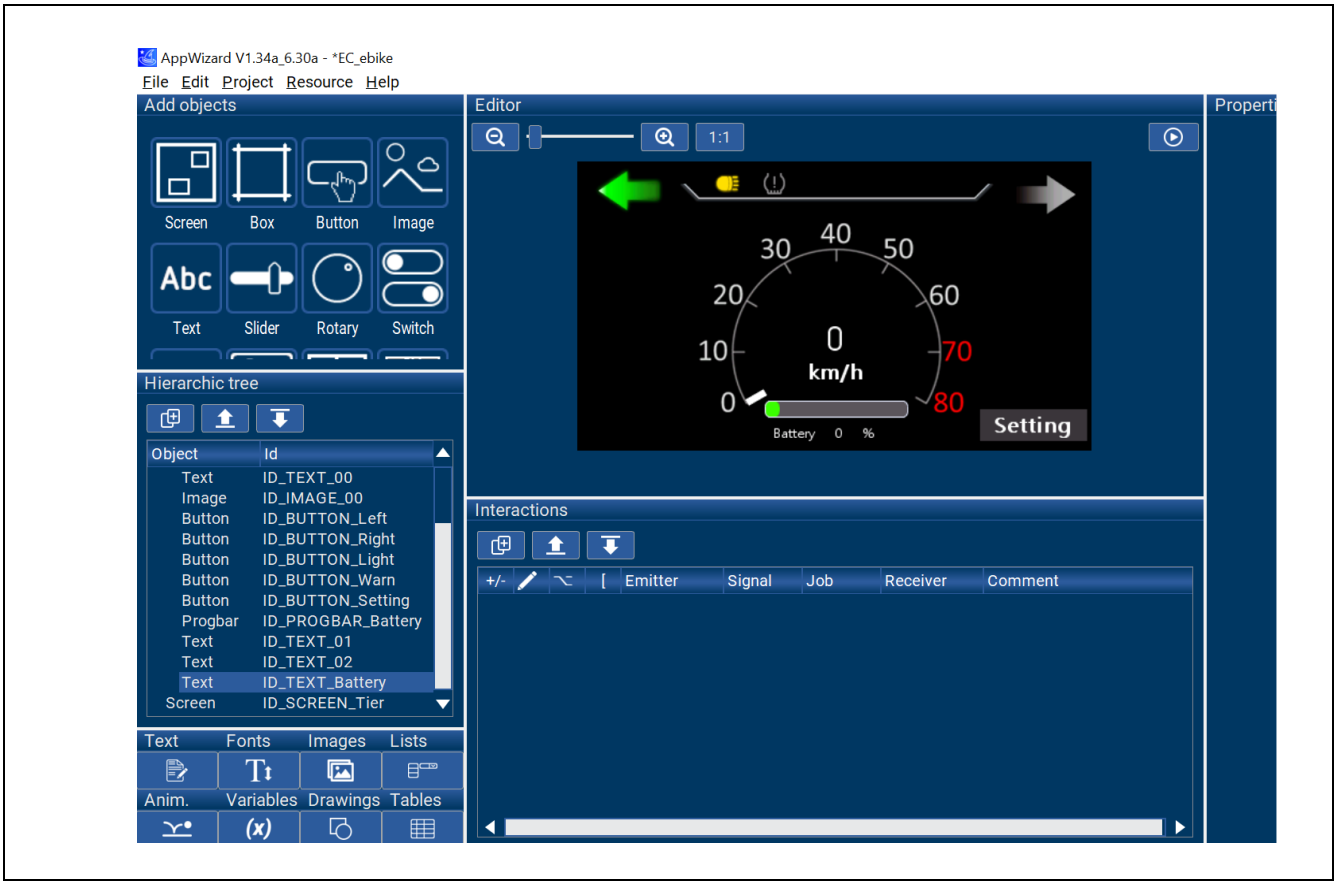


圖 1-41 以 TEXT 完成電池電量數值顯示

22. 開始編輯胎壓頁面，於 Hierarchic tree 中點選 ID_SCREEN_Tire。

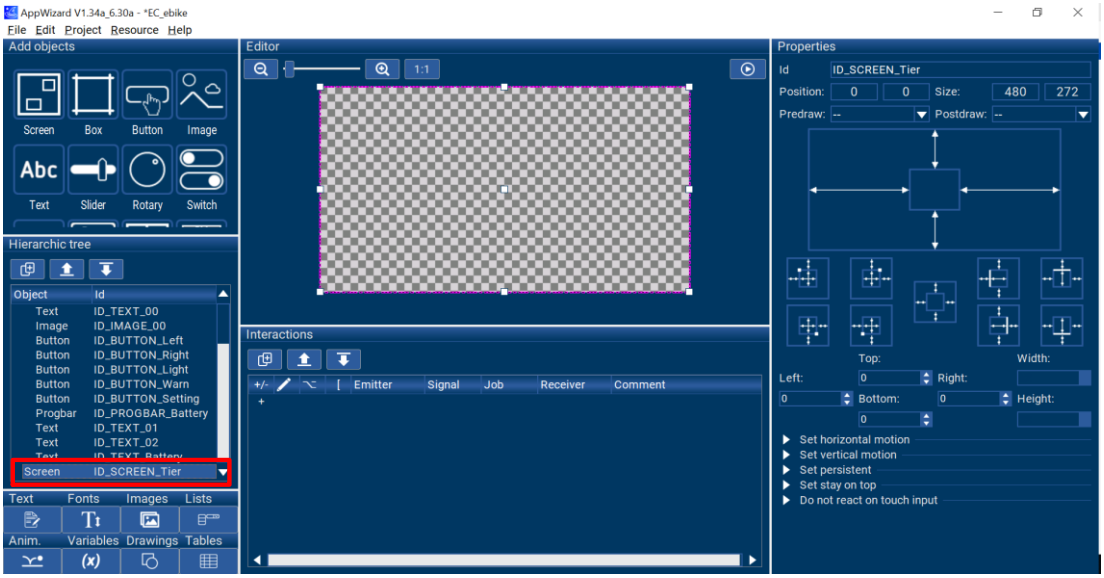


圖 1-42 點選 ID_SCREEN_TIRE 以編輯胎壓頁面

23. 使用 Image，匯入一張大小 480x272 的圖當背景，在 Add objects 中，點選 Image。

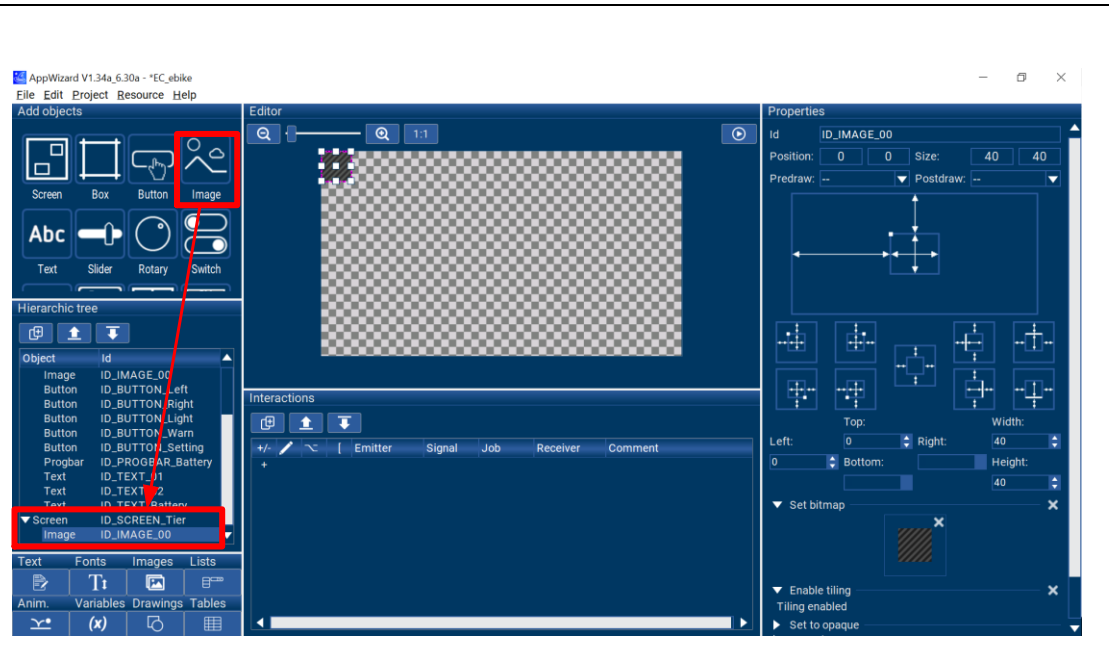


圖 1-43 使用 Image 匯入背景圖片

24. 依照步驟 12~13，選擇圖片，完成後如圖 1-44 所示。

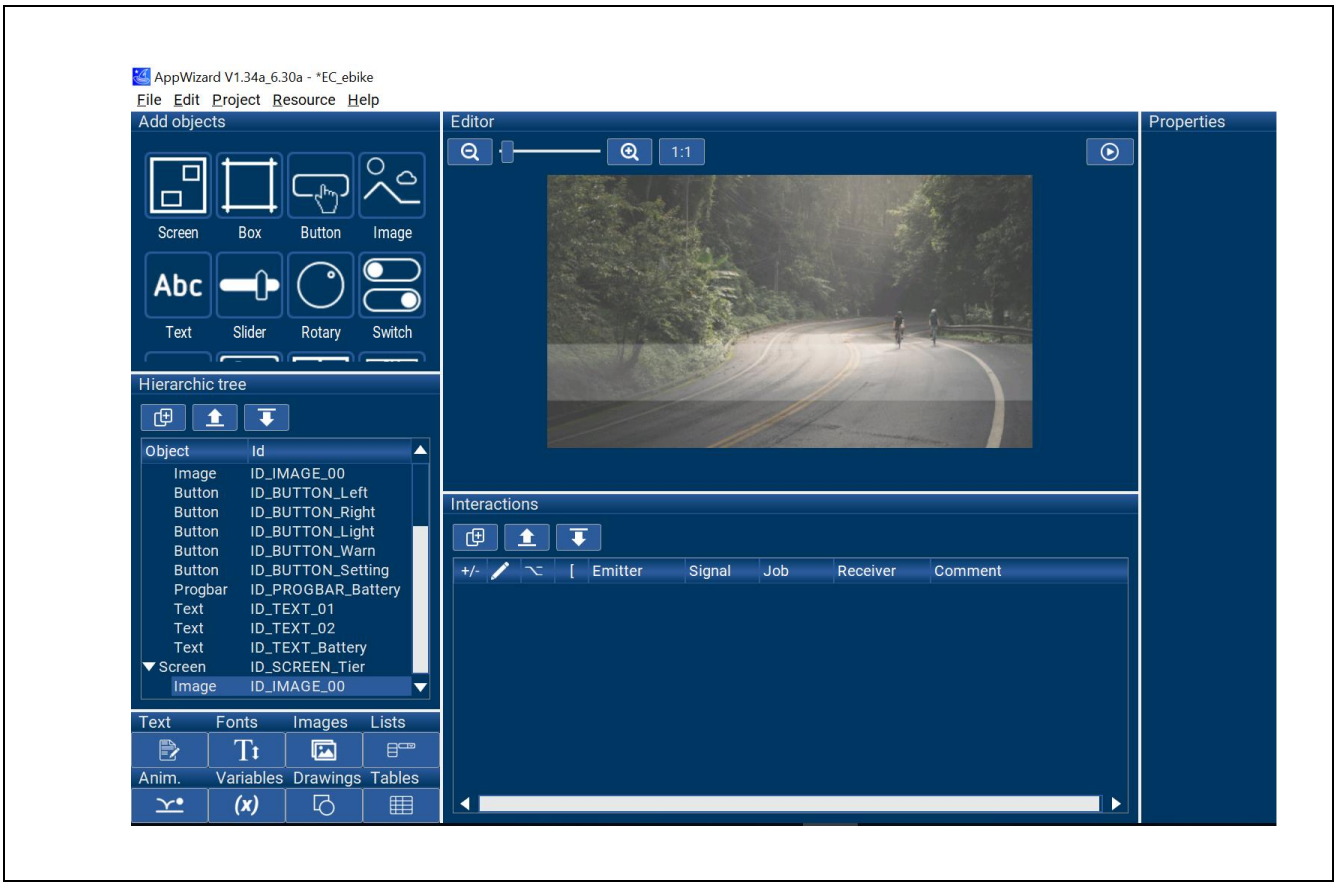


圖 1-44 以 Image 完成背景圖片

25. 選擇腳踏車圖片，並依照步驟 12~13，使用 Image 匯入，完成後如圖 1-45 所示。

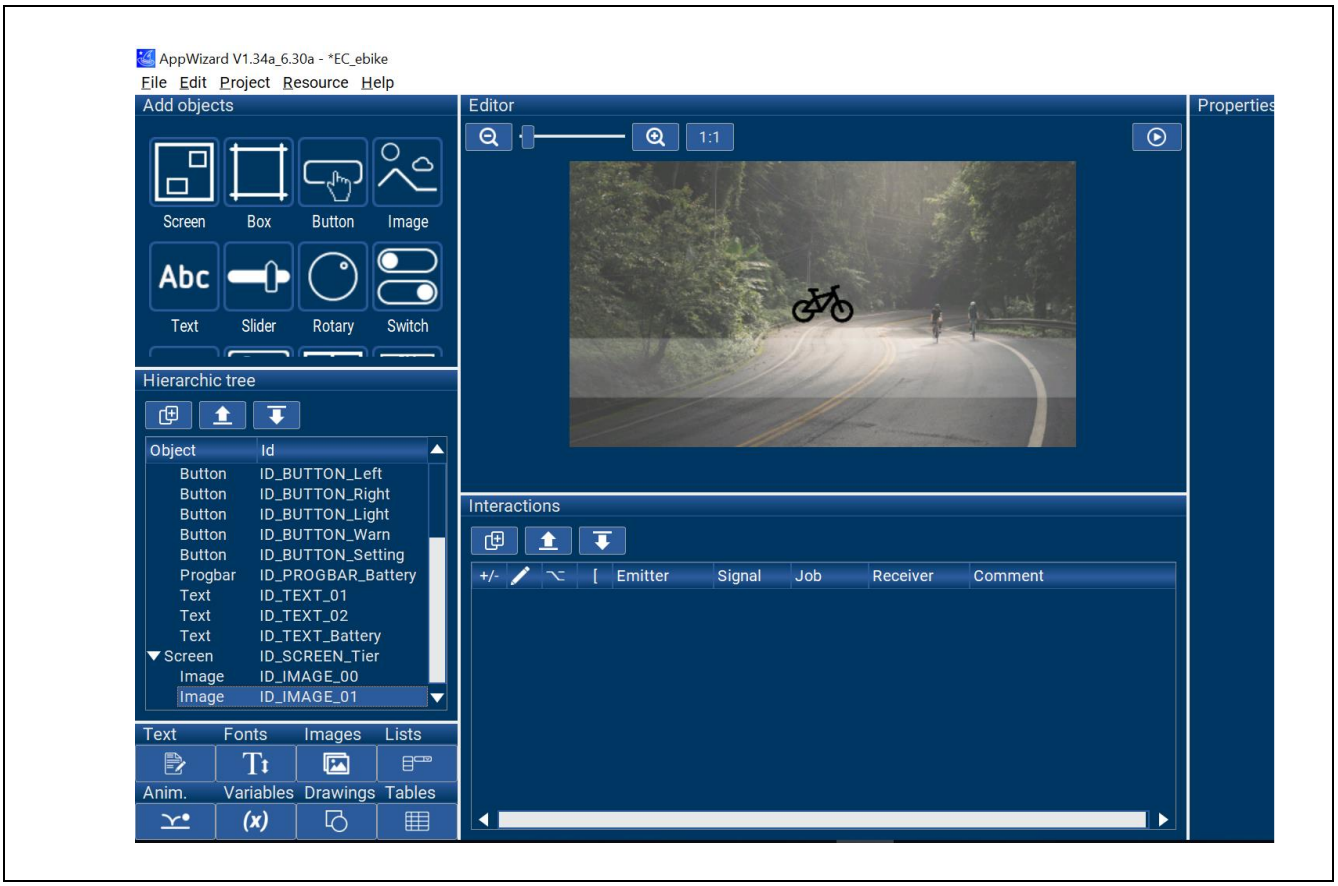


圖 1-45 以 Image 完成腳踏車圖片

26. 使用 TEXT，新增標題、胎壓文字，請依照步驟 11 操作，此範例相關設定可參考圖 1-46，完成後如圖 1-47 所示。而胎壓單位，如 psi、kPa、bar 等，亦可使用相同方式設計顯示於畫面中。

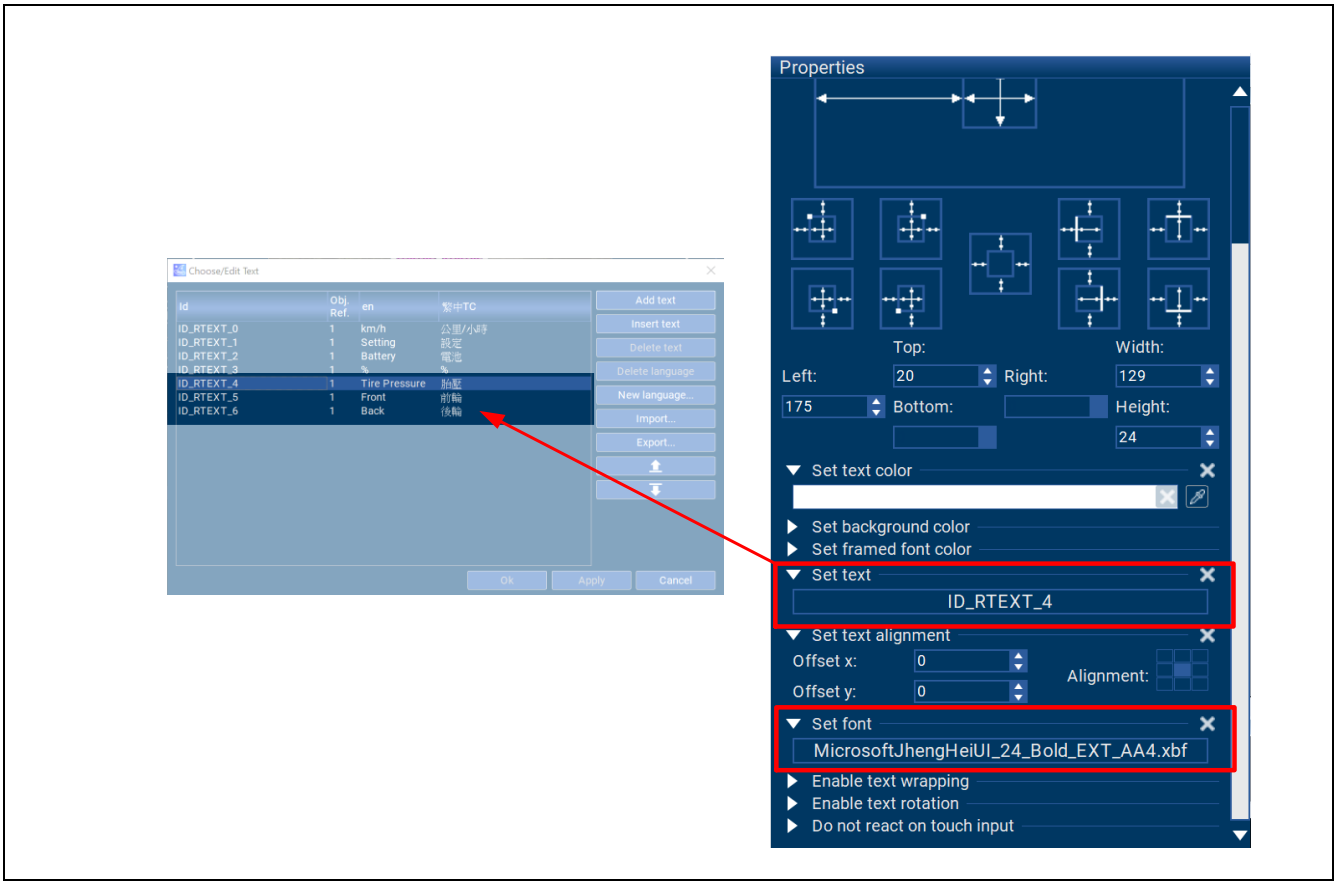


圖 1-46 TEXT 屬性

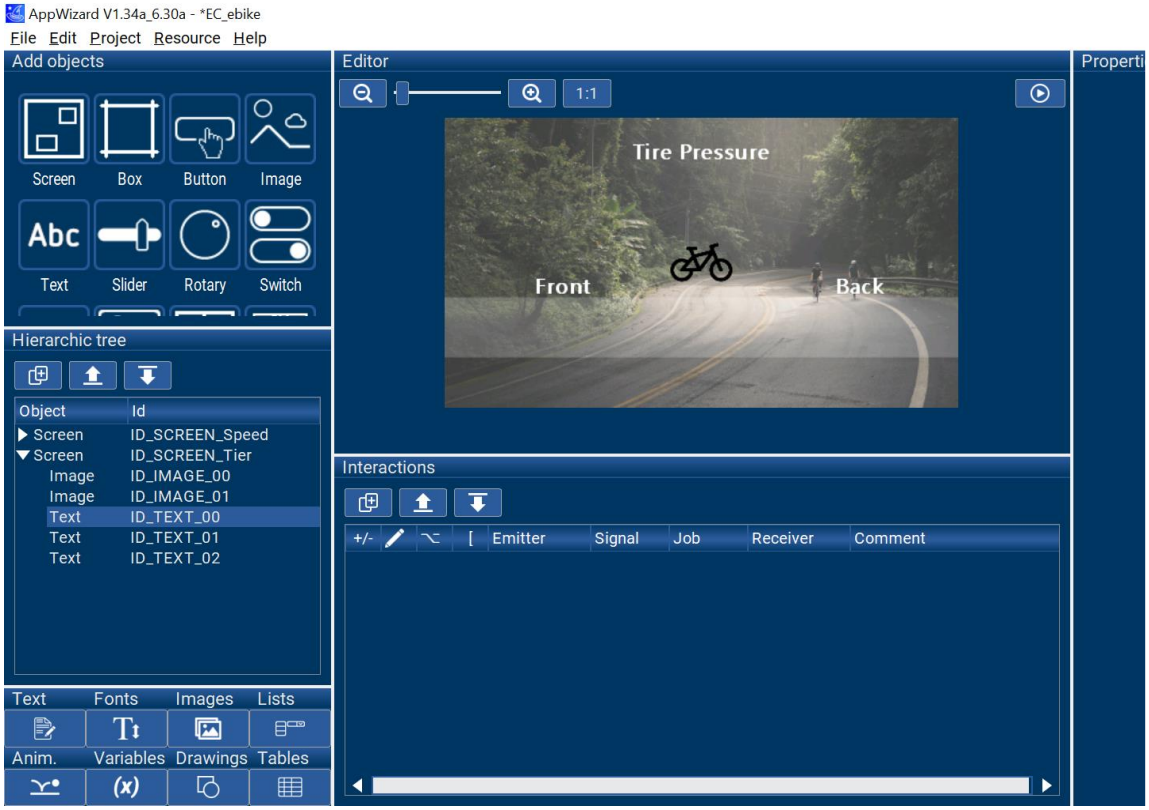


圖 1-47 以 TEXT 完成胎壓文字顯示

27. 使用 TEXT，新增胎壓數值顯示，請依照步驟 9 ~ 步驟 10 操作，此範例相關設定可參考圖 1-48，完成後如圖 1-49 所示。

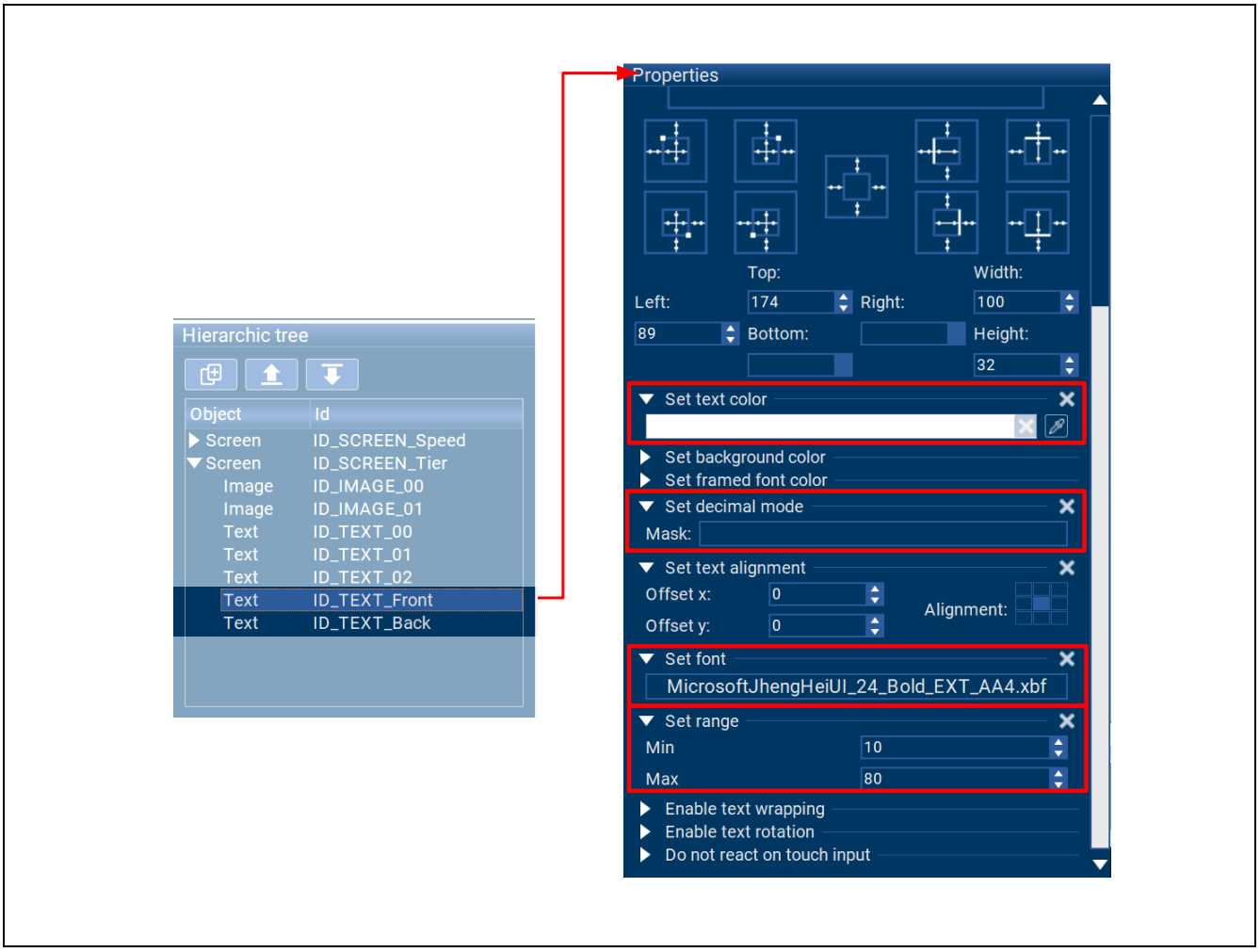


圖 1-48 TEXT 屬性設定

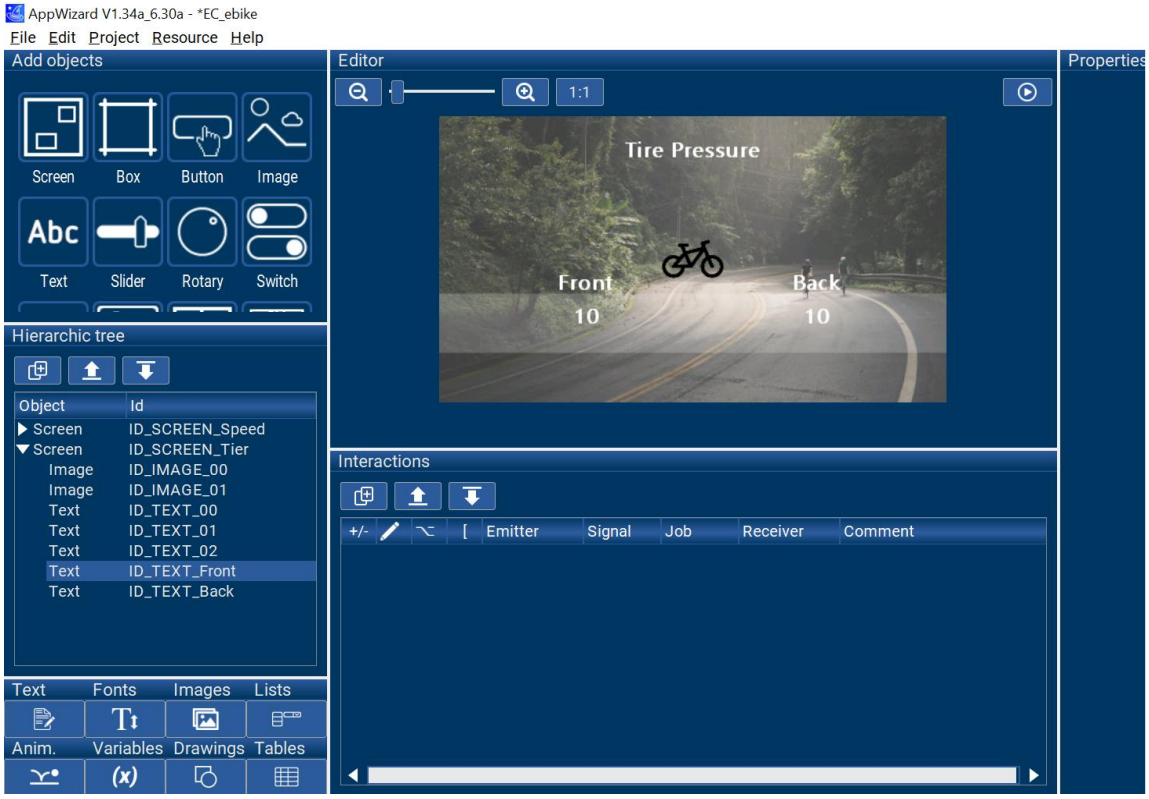


圖 1-49 以 TEXT 完成胎壓數值顯示

28. 依照步驟 16，新增”返回前頁”之按鈕，完成後如圖 1-50 所示。

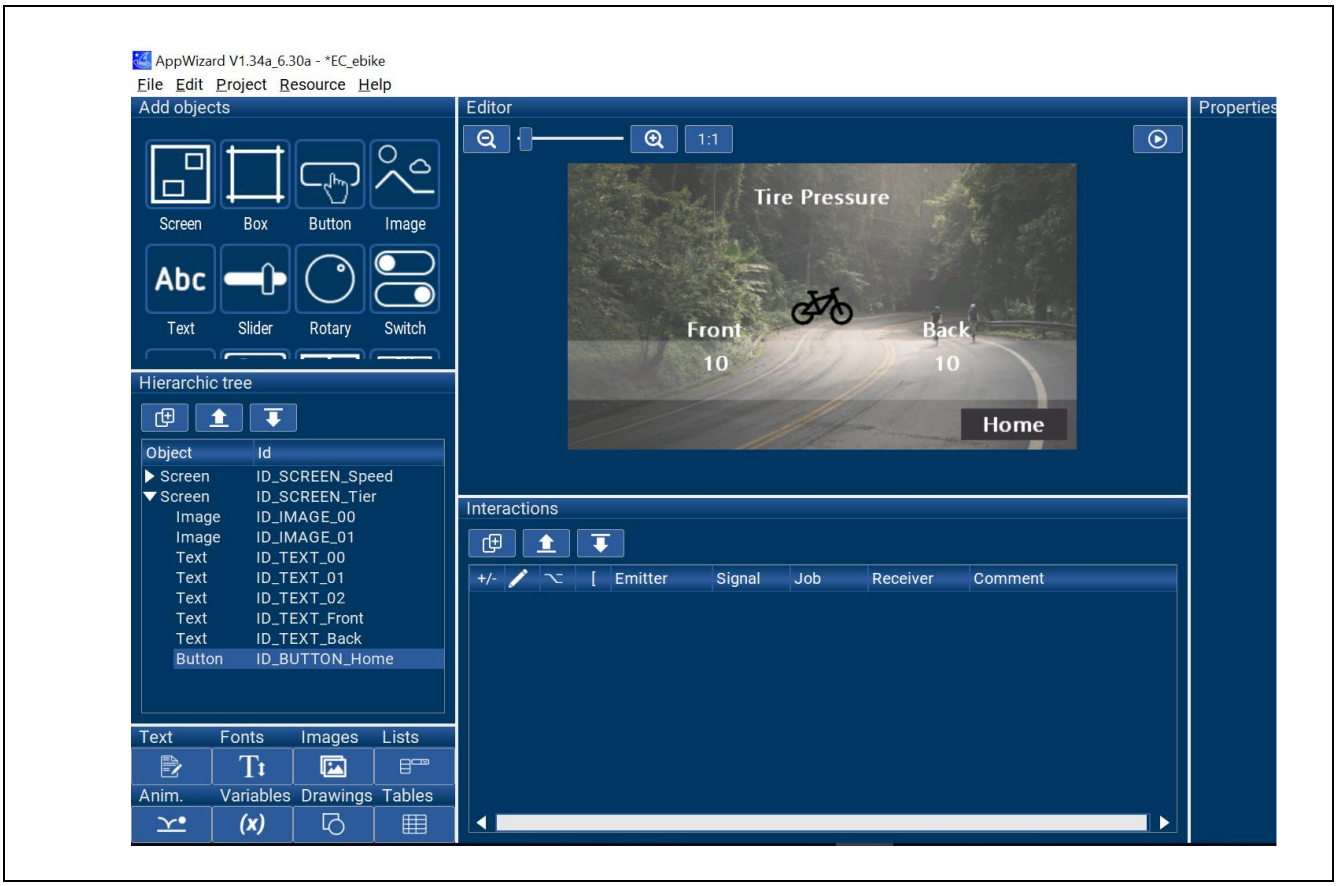


圖 1-50 以 Button 新增返回前頁按鈕

29. 依照步驟 16，新增語言切換”中文” “EN”之按鈕，完成後如圖 1-51 所示。

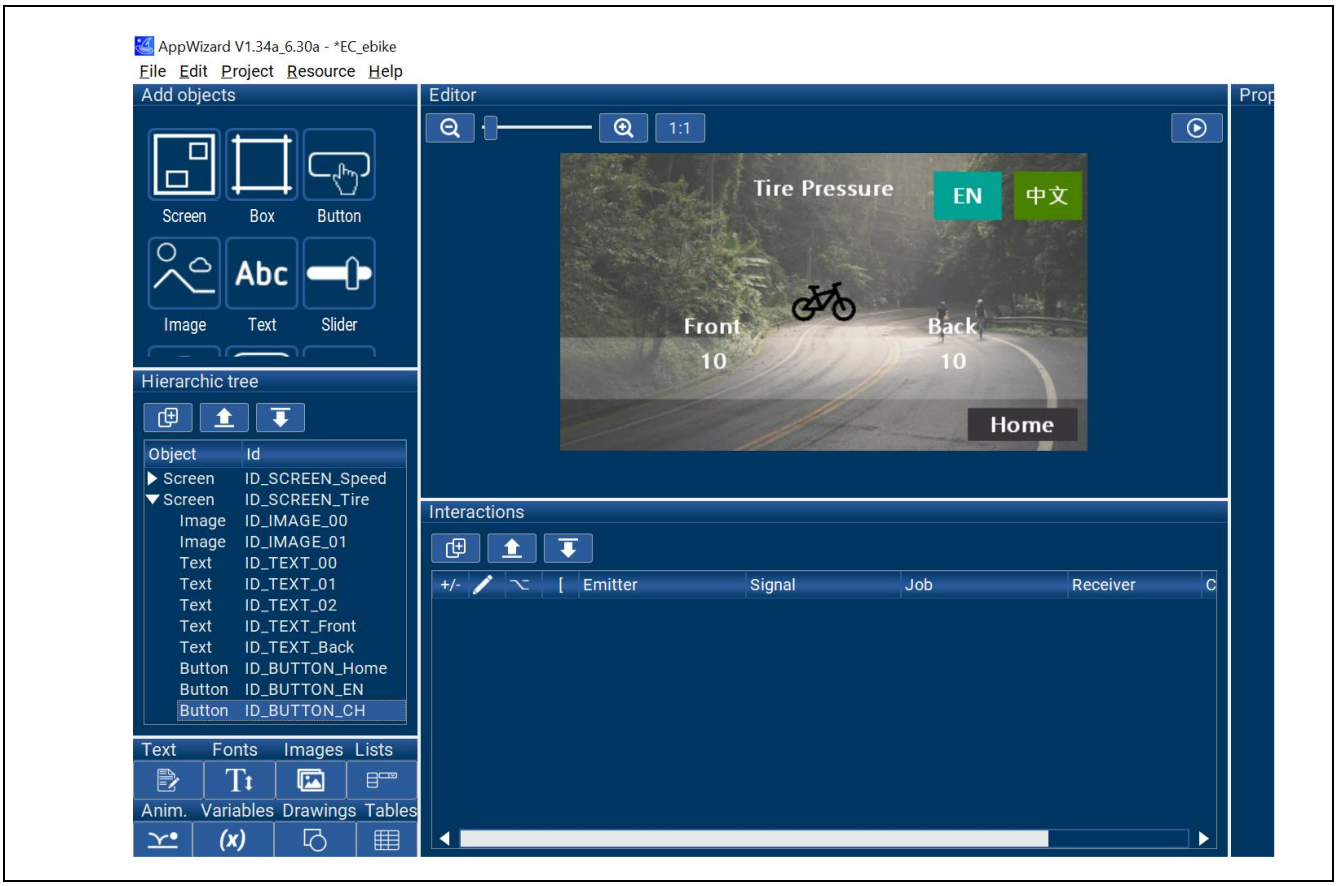


圖 1-51 以 Button 新增語言切換按鈕

30. 開始設定互動(Interactions)功能：

1) 使用按鈕做為頁面返回與切換，點選 ID_SCREEN_Tire 底下的 ID_BUTTON_Home，於 Interactions 欄位設定如圖 1-52 所示，即設定完成頁面 Tire 切換至頁面 Speed；由頁面 Speed 按鈕切換至頁面 Tire 亦然，如圖 1-53。

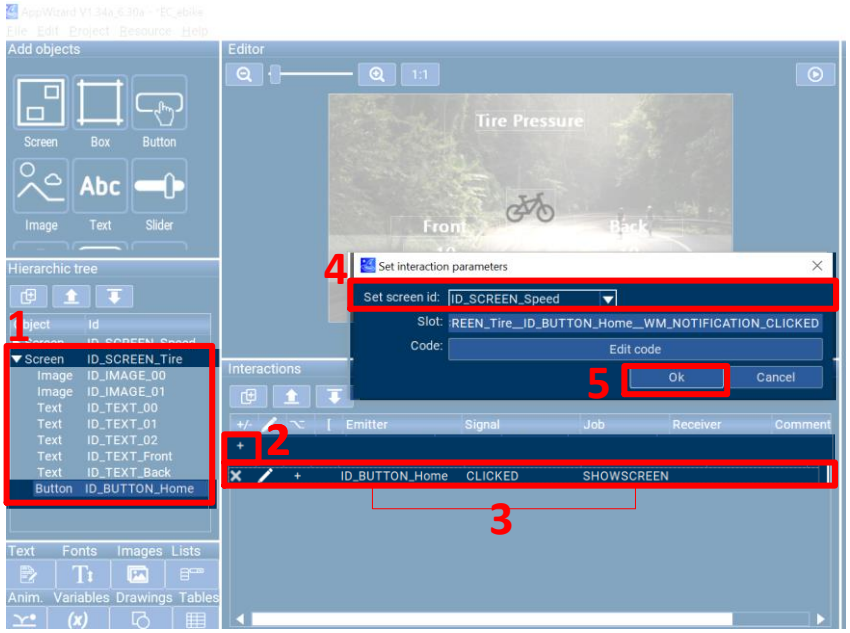


圖 1-52 Interaction 設定-頁面切換

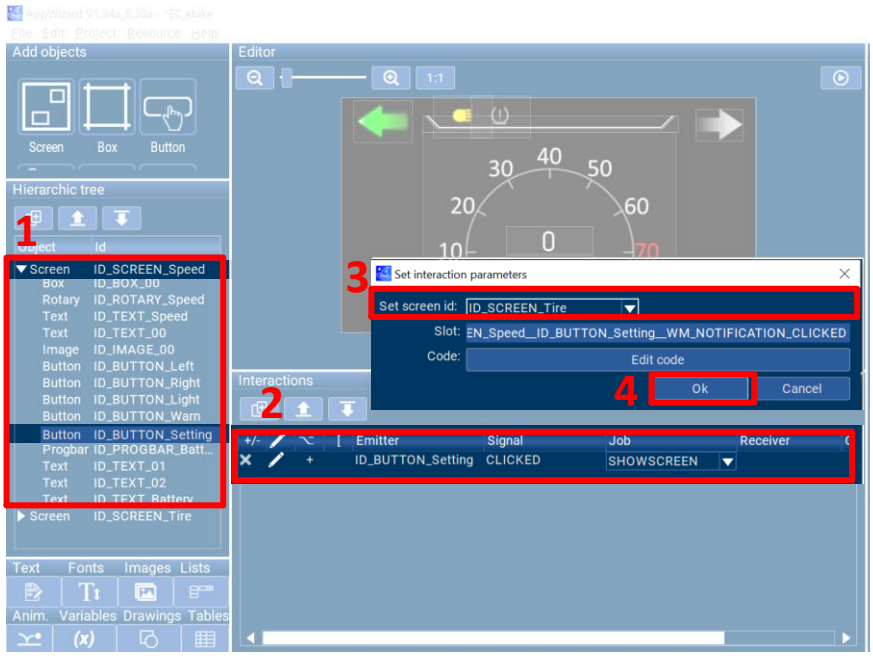


圖 1-53 Interaction 設定-頁面切換

2) 使用按鈕做為語言切換功能，點選 ID_SCREEN_Tire 底下的 ID_BUTTON_EN 設定步驟如圖 1-54 所示，即設定完成按下”EN”按鈕顯示英文功能；顯示中文功能亦然，設定步驟如圖 1-55 所示。

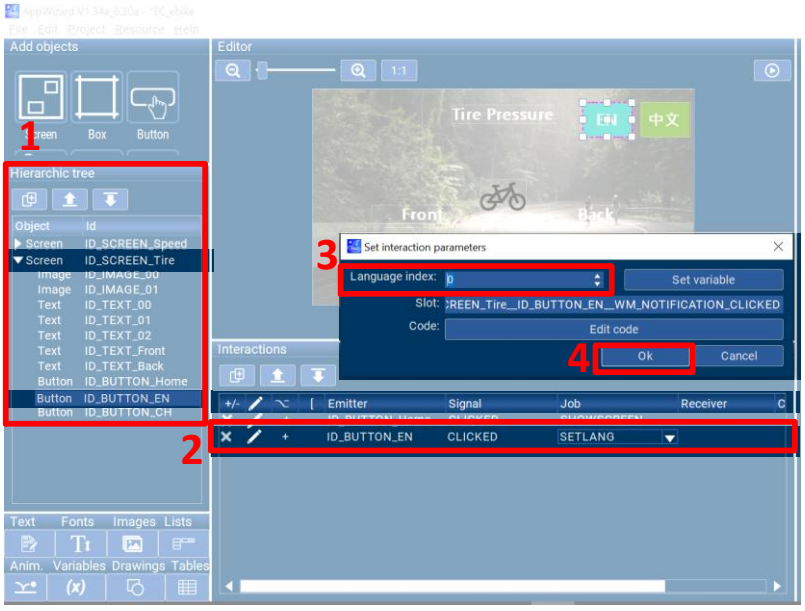


圖 1-54 Interaction 設定-語言切換

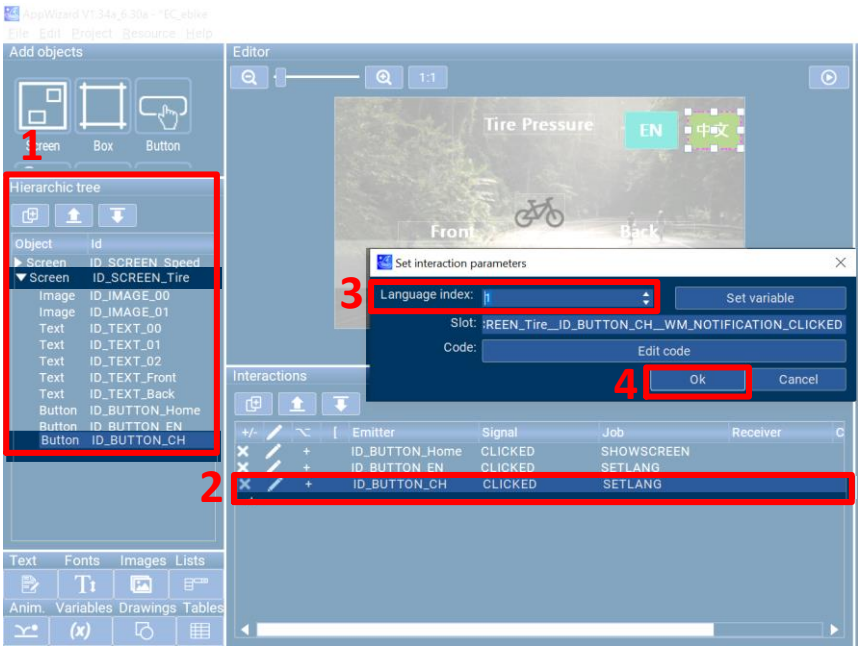


圖 1-55 Interaction 設定-語言切換

3) 為了從 MCU 計算完後設定數值給電量、前後輪的胎壓，需要使用 Variable 功能，並於程式碼中賦予 Variable 值，再使用 interaction 功能將 variable 值賦予給欲顯示設值的 TEXT，流程如圖 1-56 所示，實際步驟請依照圖 1-57~圖 1-59 操作，程式碼中步驟請參考 2 程式介紹。

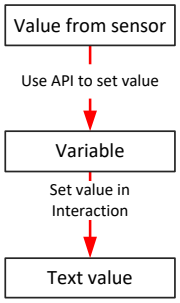


圖 1-56 顯示 MCU 計算的數值於畫面上的流程

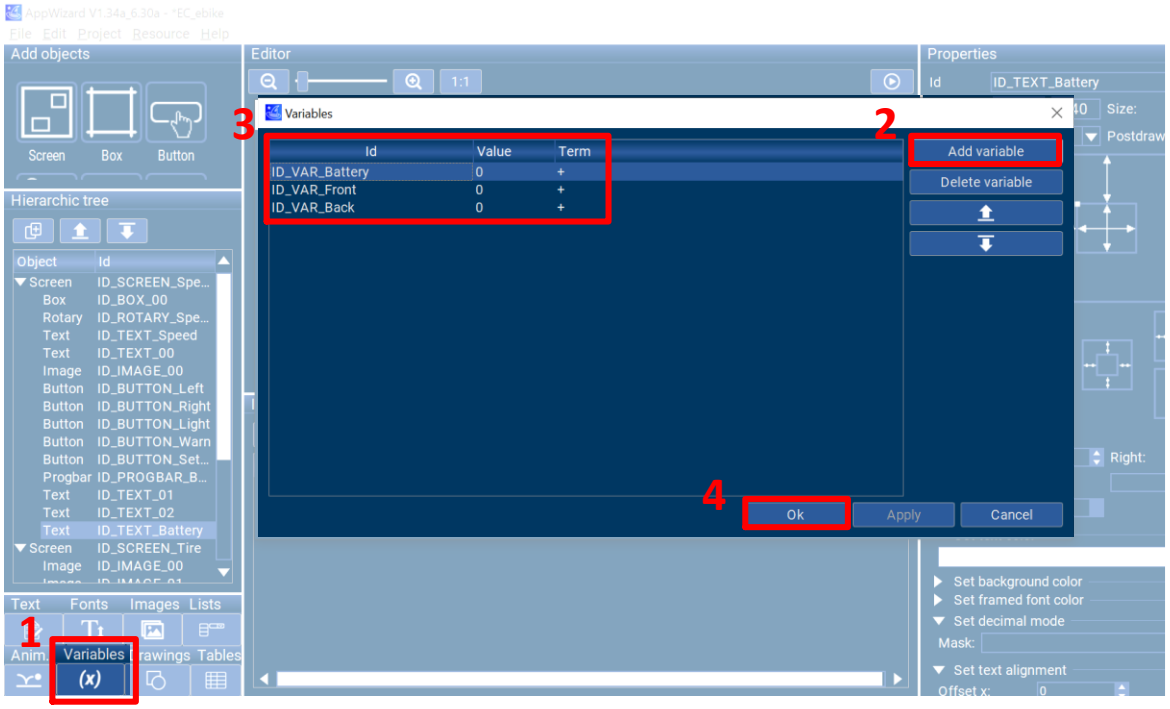


圖 1-57 Variable 設定

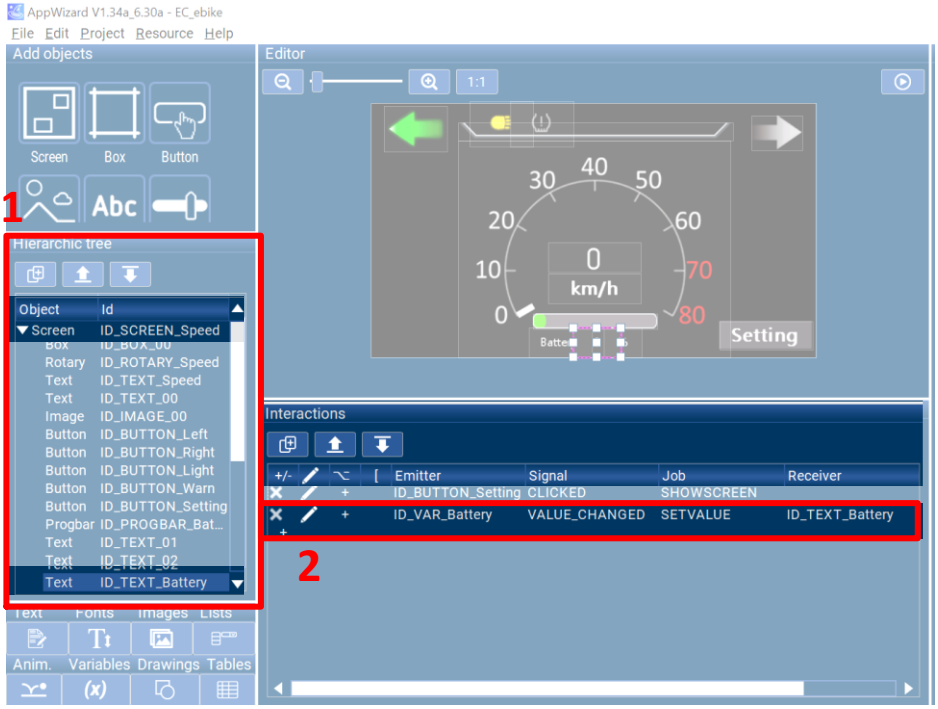


圖 1-58 Interaction 設定 - Variable

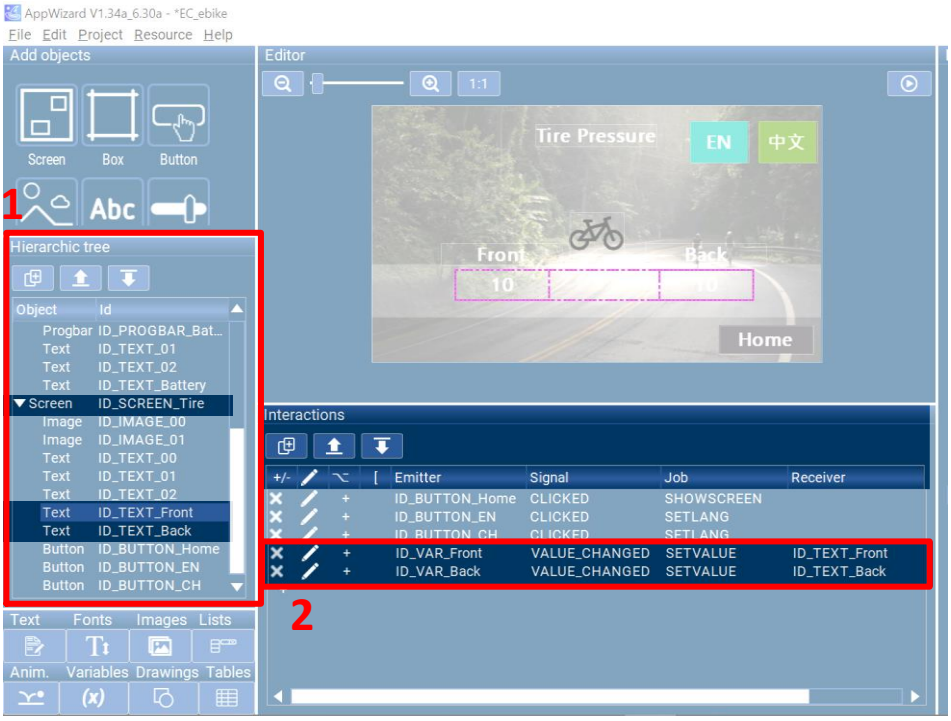


圖 1-59 Interaction 設定 – Variable

4) 使用動畫 animation 功能，模擬速度儀表變化，請依照圖 1-60~圖 1-63，設定速度數值自動變化，接著依照圖 1-64 設定速度儀表隨著速度數值變化而轉動。

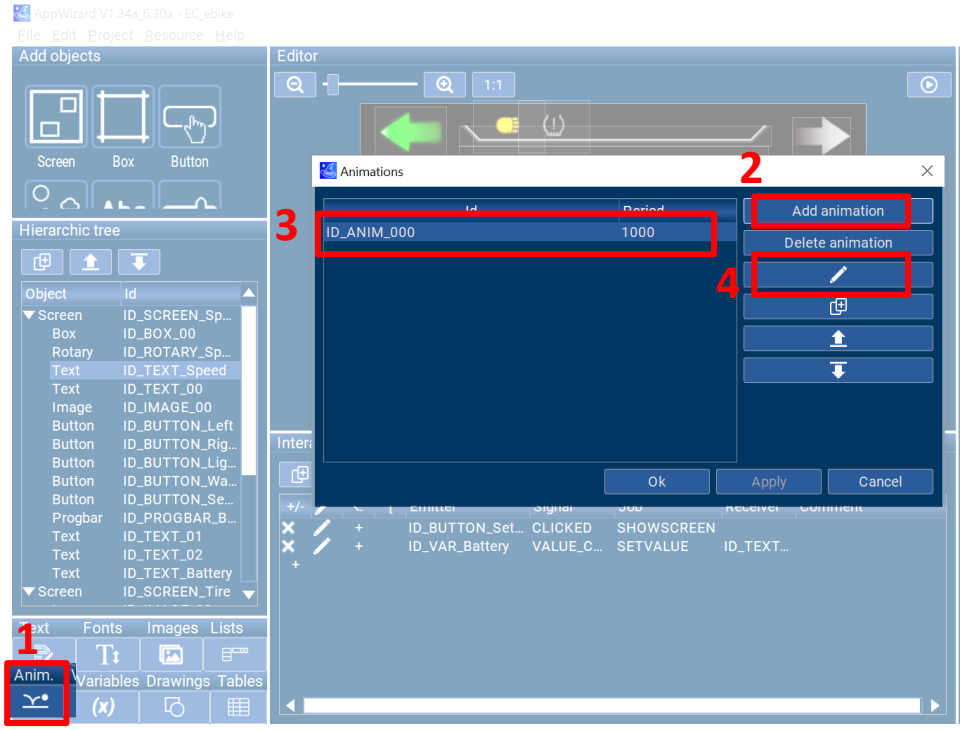


圖 1-60 Animation 設定

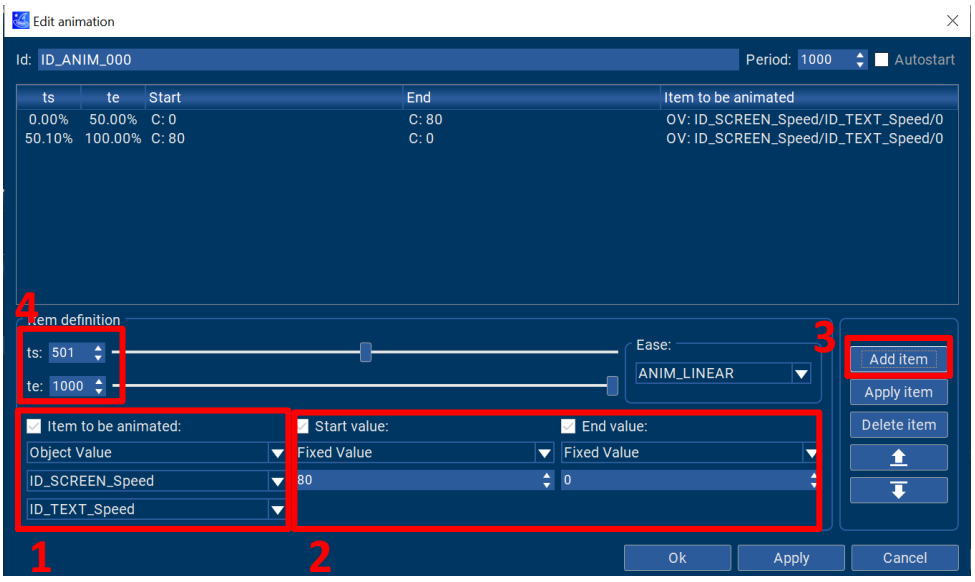


圖 1-61 Animation 設定

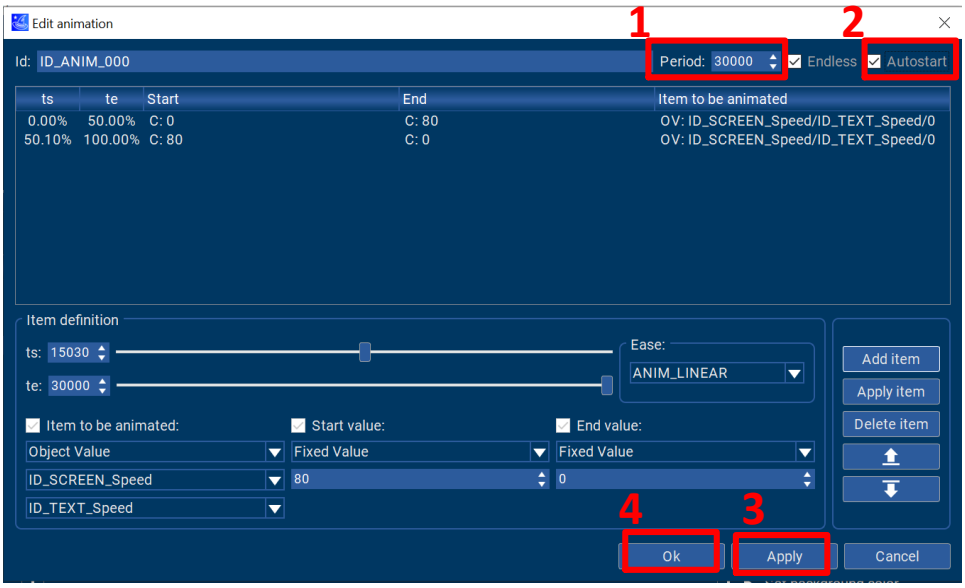


圖 1-62 Animation 設定

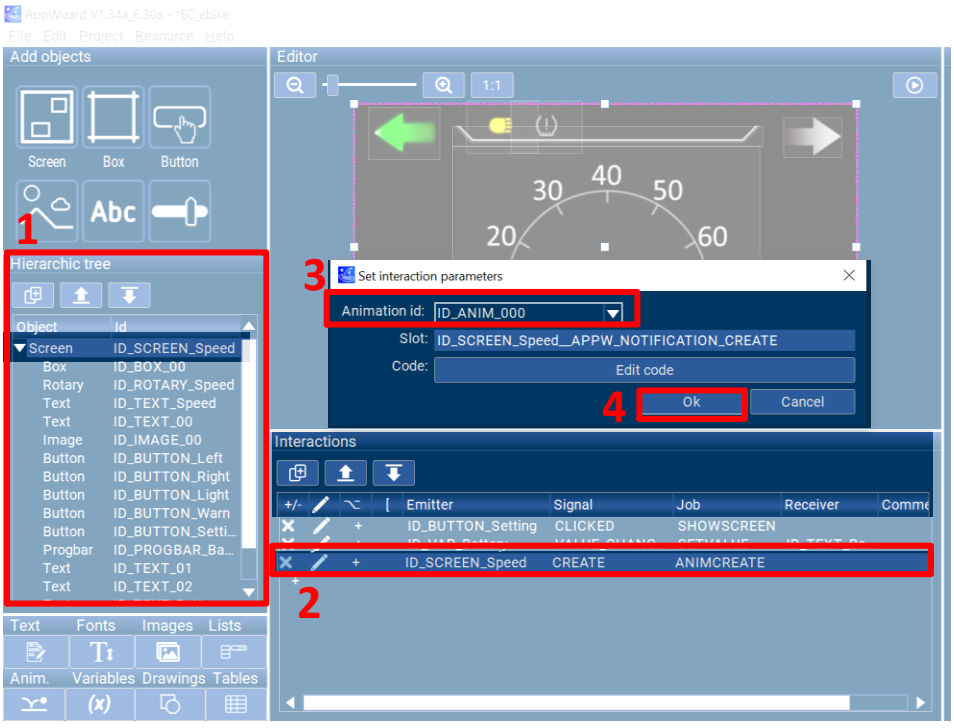


圖 1-63 Interaction 設定-開啟 Animation

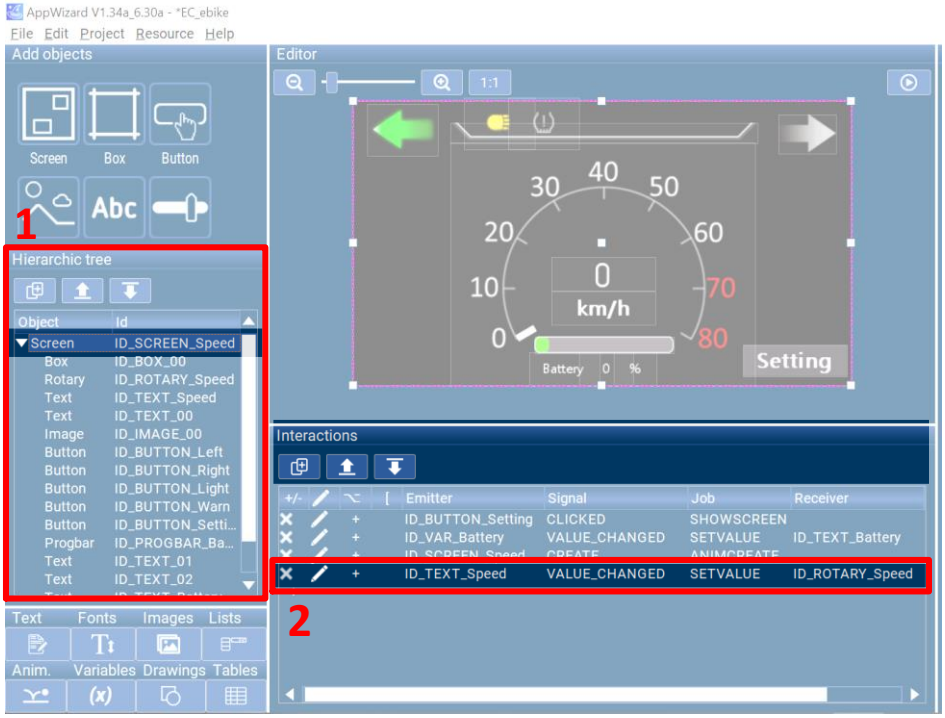


圖 1-64 Interaction 設定 – 動作連動

5) 設定電量數值與電量顯示圖連動，設定方式如圖 1-65。

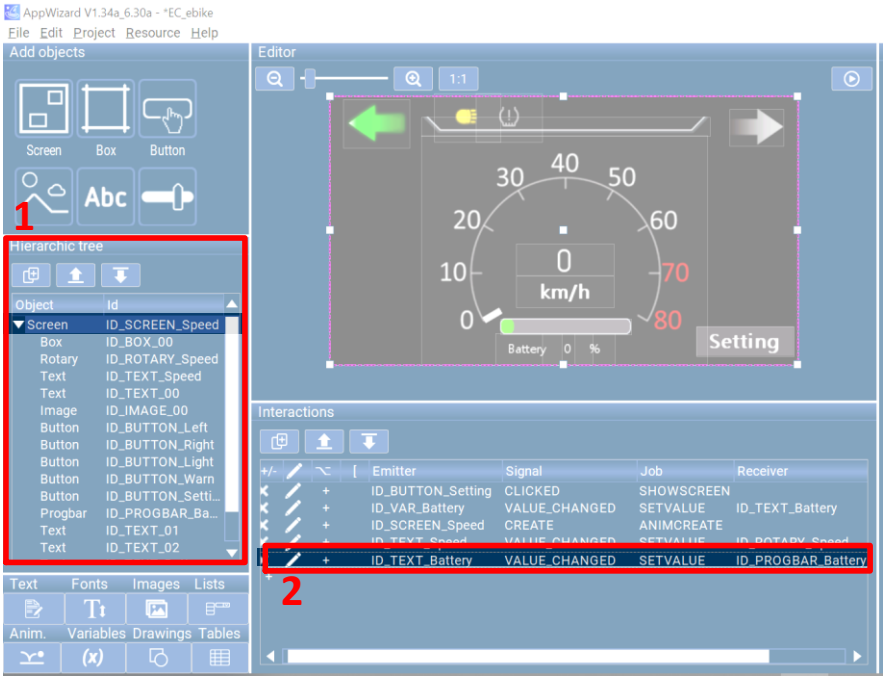


圖 1-65 Interaction 設定 – 動作連動

31. 設定以滑動頁面切換畫面，先於 Hierarchic tree 點選欲切換畫面的 Screen ID，再於 Properties 中設定水平或垂直動作，操作步驟如圖 1-66。

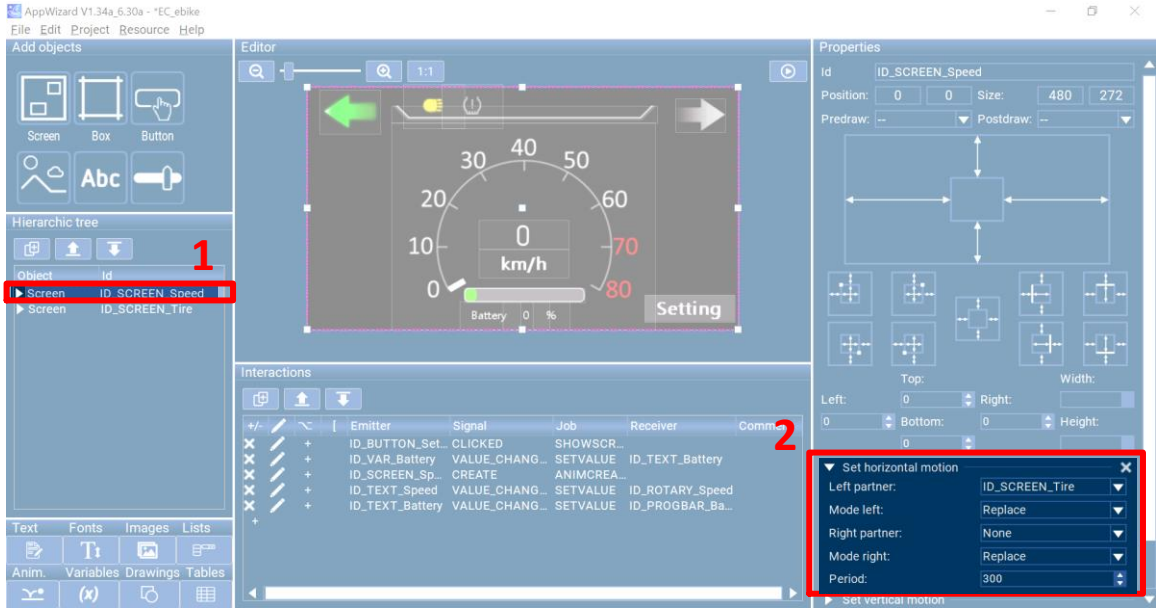


圖 1-66 Screen 屬性設定

1.2.3 程式碼編輯

1. 設計完成即可於 File-> Export & Save 輸出程式碼，如圖 1-67。此範例所使用之專案檔名稱為 EC_ebike.zip，檔案路徑可參考圖 4-1，在解壓縮後，可以使用 AppWizard 直接開啟該專案檔。

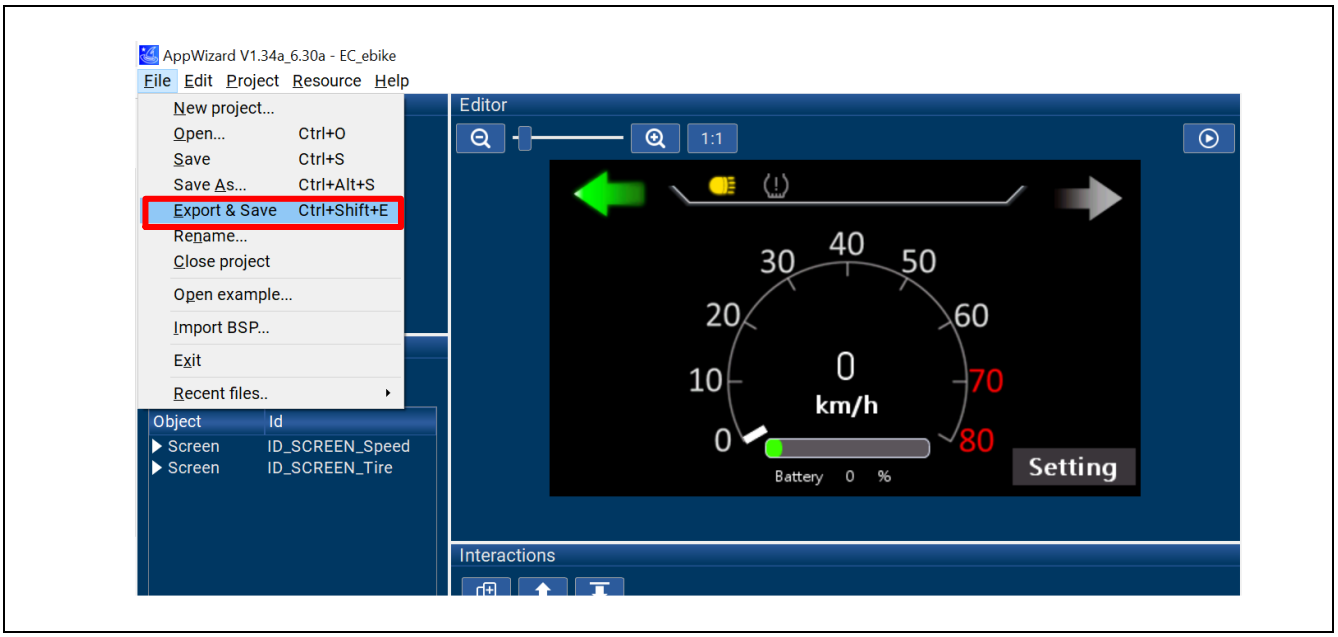


圖 1-67 輸出程式碼

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_Ebike_V1.00\SampleCode\ExampleCode\M467_AppWizard_Ebike\Application 中覆蓋原本檔案，步驟如圖 1-68~圖 1-69 所示。

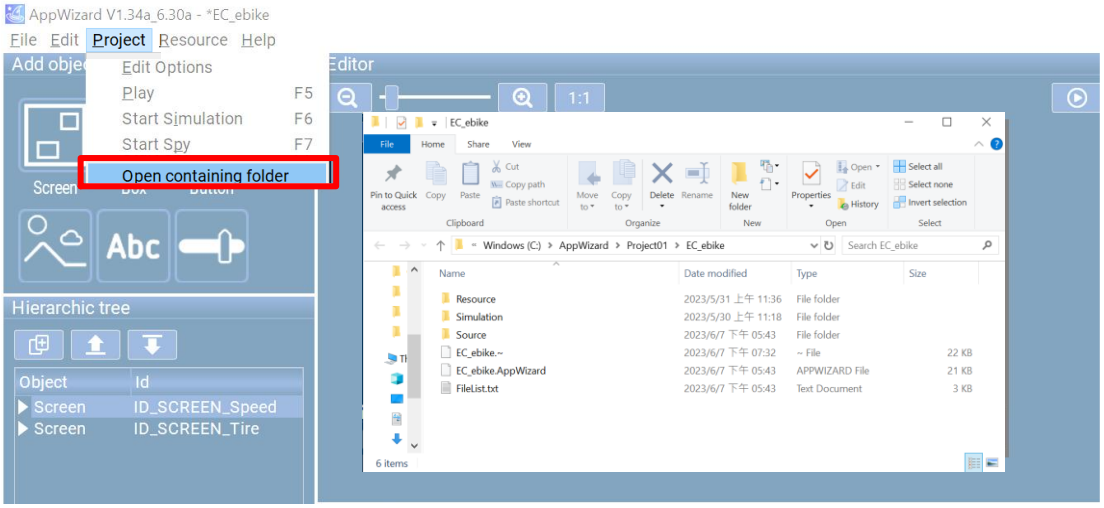


圖 1-68 開啟專案資料夾

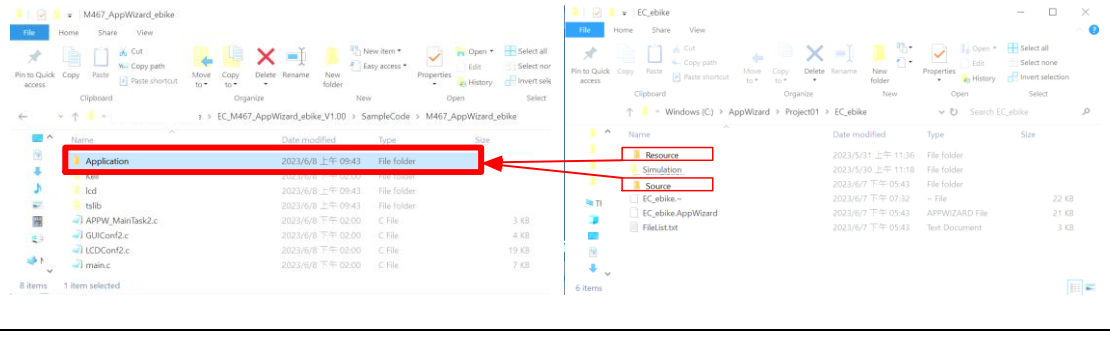


圖 1-69 移植至程式碼專案中

主程式碼請參考 2 程式介紹。

1.2.4 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

- 於 AppWizard 使用的圖片 format，需設定為符合屏幕格式，此範例設定為 “High color with alpha (565)”。

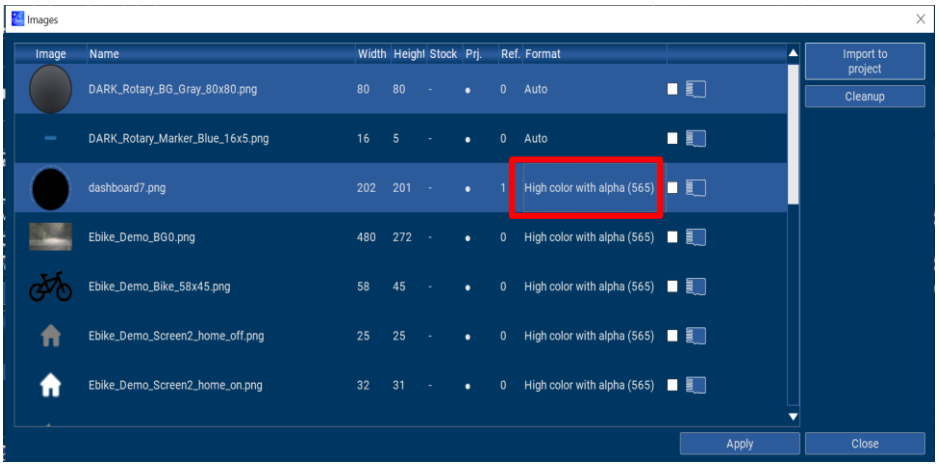


圖 1-70 Image 設定

1.3 執行結果

本範例程式執行編譯和燒錄的結果如圖 1-71 所示。

圖 1-71 執行結果

2. 程式介紹

主程式由 *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
- 應用主程式：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、文字檔案資源存至外部 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，直接到指定位址讀取 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_normal_ext_aa4.o (+RO)
     microsoftjhengheiui_24_bold_ext_aa4.o (+RO)
     nettoot_40_normal_ext_aa4.o (+RO)
     speedmeter_0.o (+RO)
     ebike_demo_screen2_l_off.o (+RO)
     ebike_demo_screen2_l_on.o (+RO)
     ebike_demo_screen2_light_off.o (+RO)
     ebike_demo_screen2_light_on.o (+RO)
     ebike_demo_screen2_pressure_on.o (+RO)
     ebike_demo_screen2_r_off.o (+RO)
     ebike_demo_screen2_r_on.o (+RO)
     ebike_demo_screen2_pressure_off.o (+RO)
     rotary_marker_whitebar_20x7.o (+RO)
     turnline_3.o (+RO)
     Ebike_Demo_BG0.o (+RO)
     ebike_demo_bike_58x45.o (+RO)
    }
}
```

程式進入應用主程式 MainTask2()後，將執行 AppWizard 相關函式，在 while 迴圈中會執行畫圖的動作，此範例程式同時新增 APPW_SetVarData()函式，將值從 MCU 傳出給 AppWizard 中的 Variable，再透過如步驟 3)所述之方法，數值便會呈現在畫面上。

```
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
    while (1)
    {
     // Drawing a frame
     GUI_Exec1();
     APPW_Exec();
     NVT_Draw1();
     APPW_SetVarData(ID_VAR_Battery, 88);
     APPW_SetVarData(ID_VAR_Back, 31);
     APPW_SetVarData(ID_VAR_Front, 40);
    }
}
```

AppWizard 執行的畫圖內容，會從 ID_Screen_xxx_Slots.c 中讀取，可以試著在步驟 16 按鈕按下時的程式碼中加入 printf("Button Click\n")，在實際 Speed 畫面，若按下 Setting 按鈕跳至 Tire 頁面，UART 會於 PC 印出”Button Click”內容。

```
/*
File        : ID_SCREEN_Speed_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_Speed.h"

/*** Begin of user code area ***/
/*** End of user code area ***/

/*********************************************************************
*
*       Public code
*
**********************************************************************
*/
/*********************************************************************
*
*       cbID_SCREEN_Speed
*/
void cbID_SCREEN_Speed(WM_MESSAGE * pMsg) {
  GUI_USE_PARA(pMsg);
}

/*********************************************************************
*
*       ID_SCREEN_Speed__ID_BUTTON_Setting__WM_NOTIFICATION_CLICKED
*/
void ID_SCREEN_Speed__ID_BUTTON_Setting__WM_NOTIFICATION_CLICKED(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-HMI-M467
- 線路示意圖
  - 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。

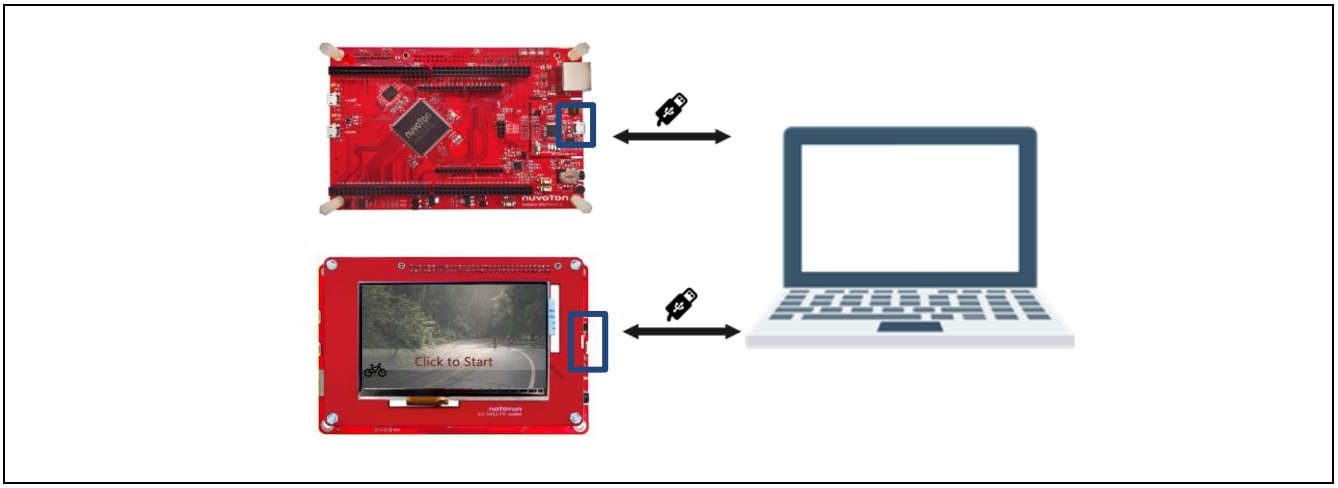


圖 3-1 以 USB 線將 NuMaker-HMI-M467 與 PC 相連

4. 目錄資訊

EC_M467_AppWizard_Ebike_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	Source file of example code
ThirdParty	Header files for emWin project
EC_ebike.zip	Ebike 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，雙擊 *M467_AppWizard_Ebike.uvprojx*。
2. 進入編譯模式介面
   - 編譯
   - 下載程式至記憶體
   - 進入 / 離開除錯模式
3. 進入除錯模式介面
   - 執行程式

6. 修訂紀錄

Date	Revision	Description
2023.05.31	1.00	初始發布。

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.