以 AppWizard 设计电动自行车 GUI 项目

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 SEGGER emWin AppWizard 工具，结合 M467 系列开发平台与开发资源，完成电动自行车图形化界面设计。
BSP 版本	M460_Series_BSP_CMSIS_V3.00.002 emWin Software Package for M460 V3.0
开发平台	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

本范例代码将示范如何使用 SEGGER emWin AppWizard 工具，搭配 NuMaker-HMI-M467 开发平台，逐步完成一个电动自行车项目的画面设计、画面互动。

1.1 原理

1.1.1 系统方块图

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

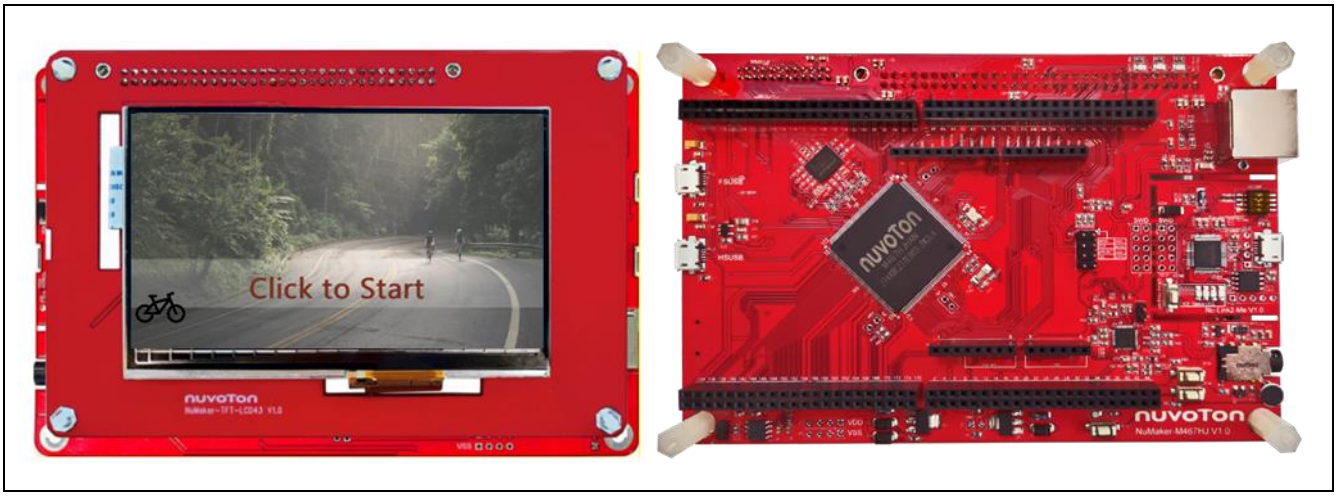


图 1-1 NuMaker-HMI-M467 开发板

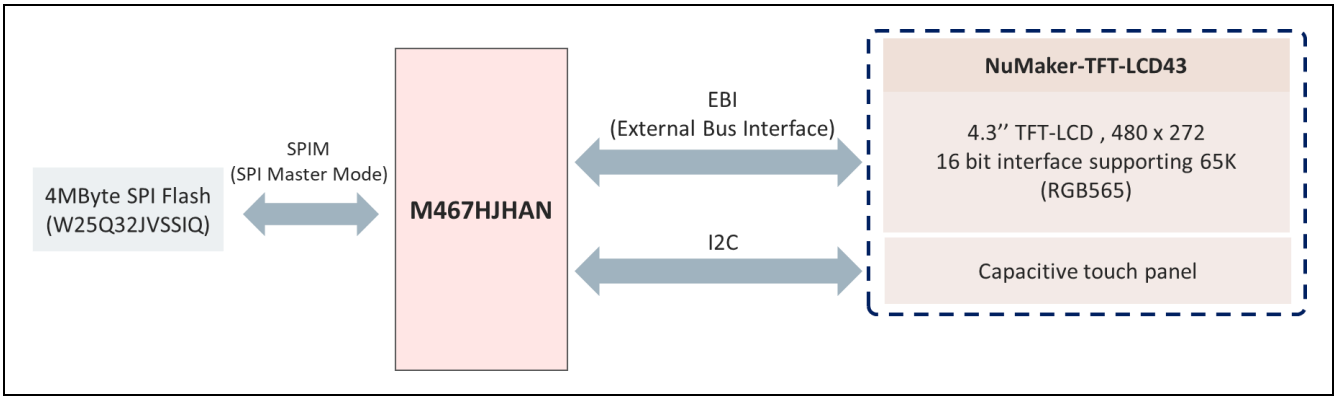


图 1-2 系统方块图

1.1.2 电动自行车画面

此范例欲设计画面如下：

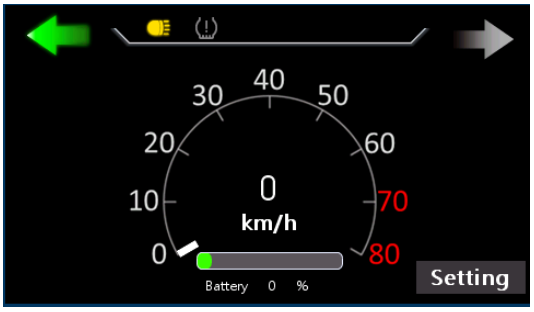
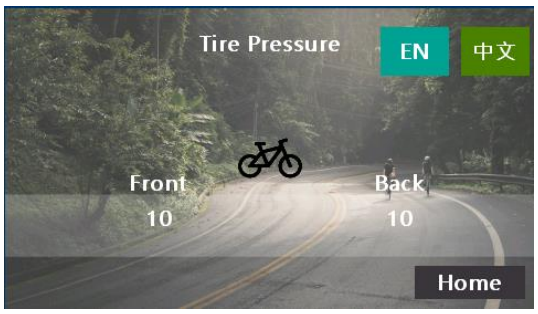
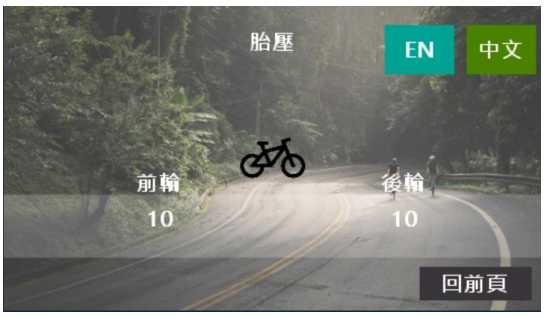
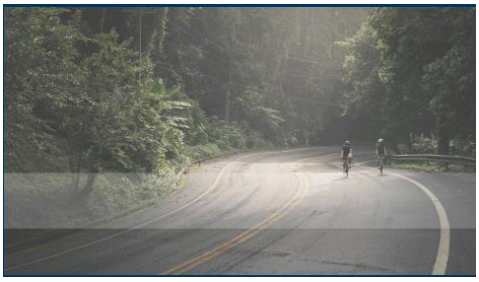
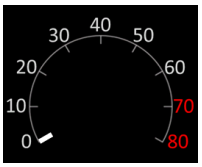
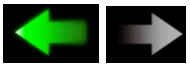
	语言切换至EN(英文)	语言切换至中文
页面一		
页面二		

表 1-1 电动自行车画面

1.1.3 电动自行车画面使用元素总览

Ebike App Description	Ebike App GUI	Object
画面分隔线		Image
480x272 背景图片		

脚踏车图片		
480x272 素色背景图		Box
速度仪表		Rotary
速度数值、速度单位	英文界面	TEXT
	中文界面	
电量、电量数值、电量单位	英文界面	
	中文界面	
胎压名称、胎压数值	英文界面	
	中文界面	
电量显示条		Progbar
方向灯		Button
大灯		

胎压警示灯		
页面切换按钮	英文界面	
	中文界面	
语言切换按钮		

表 1-2 画面元素

1.1.4 相关资源

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

1.2 执行步骤

使用 AppWizard 工具完成一项 GUI 项目的执行步骤可分为四大阶段，第一阶段需准备硬件与软件下载，第二阶段则于 AppWizard 中完成对应的画面设计、object 设定、Interactions 设定、动画设定，第三阶段使用 AppWizard 导出代码并同时加入客制化代码，以完成更多应用功能，第四阶段则是实际烧录在板端。

以下于 1.2.1-1.2.4 就各项阶段详细说明执行步骤。

1.2.1 准备相关硬件与软件资源

1. 硬件使用 NuMaker-HMI-M467，相关资源可参考 参考线路、开发板使用手册。
2. 软件资源需下载 AppWizard 软件工具：
   1) 请点击以下链接 emWin Software Package for M460 进入下载页面，点选 emWin Software Package for M460 项目即可开始下载，下载前需注册与登入新唐官网会员。
   2) 下载完成后，解压缩，开启 M460_emWin_NonOS 文件夹。
   3) 依图 1-3 路径，点击 AppWizard_V134a_630a_Install.exe 安装 AppWizard 工具，需注意若有旧的 AppWizard 版本，需将旧版本完全移除后，才能安装新版本。

图 1-3 AppWizard 工具安装文件路径

1.2.2 在 AppWizard 工具中编辑与设定 GUI

1. 于桌面快捷方式开启 AppWizard 工具。

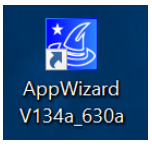


图 1-4 AppWizard 桌面快捷方式

2. 点选 ”Create new project” 以创建新专案。

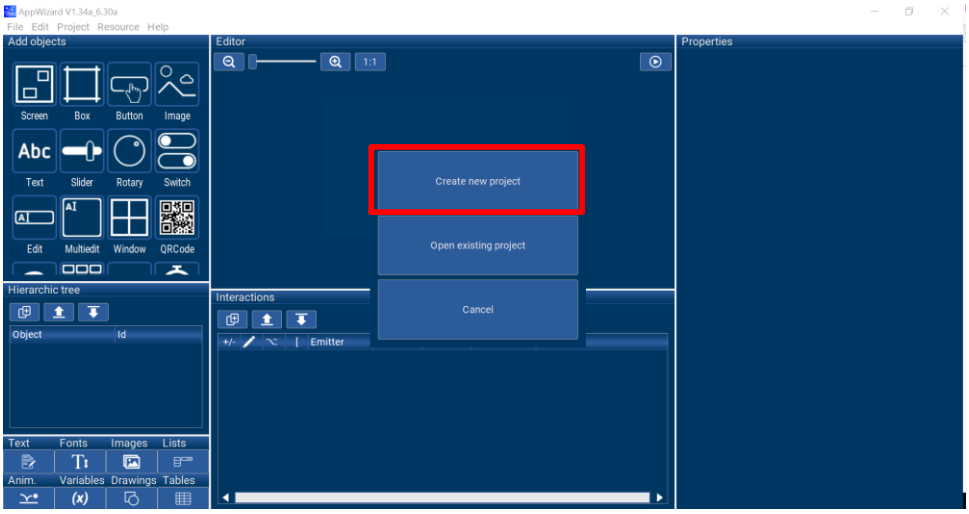


图 1-5 创建专案页面

3. 依图 1-6 新专案设定项目路径、名称，符合使用屏幕的颜色格式：16 Bit, GUICC_M565，勾选 ”Enable Multibuffering”，点选 ”ok”。

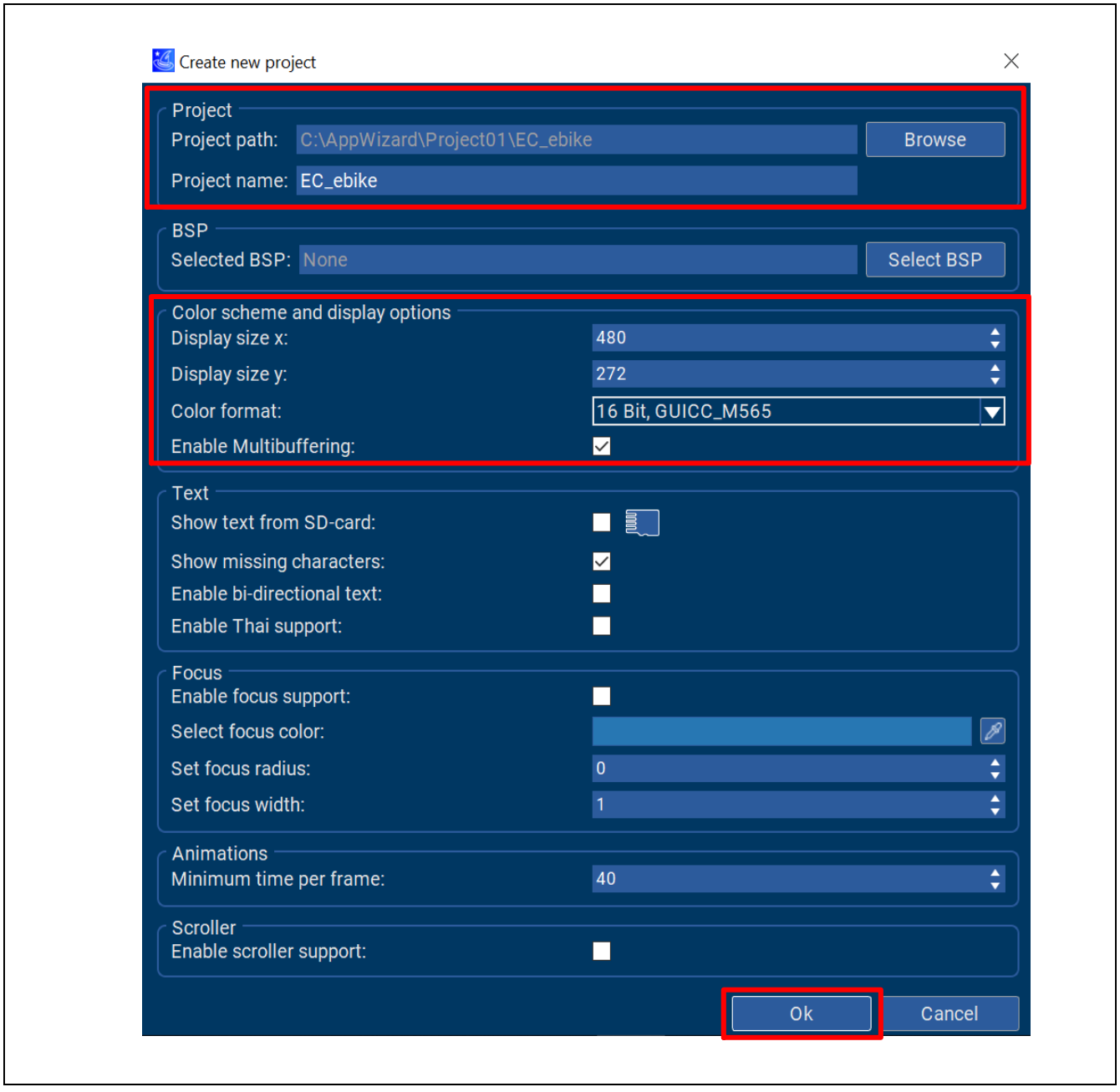


图 1-6 新专案设定项目

4. 于 Add objects 点选两次 Screen，以创建两个画面，并在 Hierarchic tree 中，为 Screen 分别命名为 ID_SCREEN_Speed 与 ID_SCREEN_Tier，以利辨认。

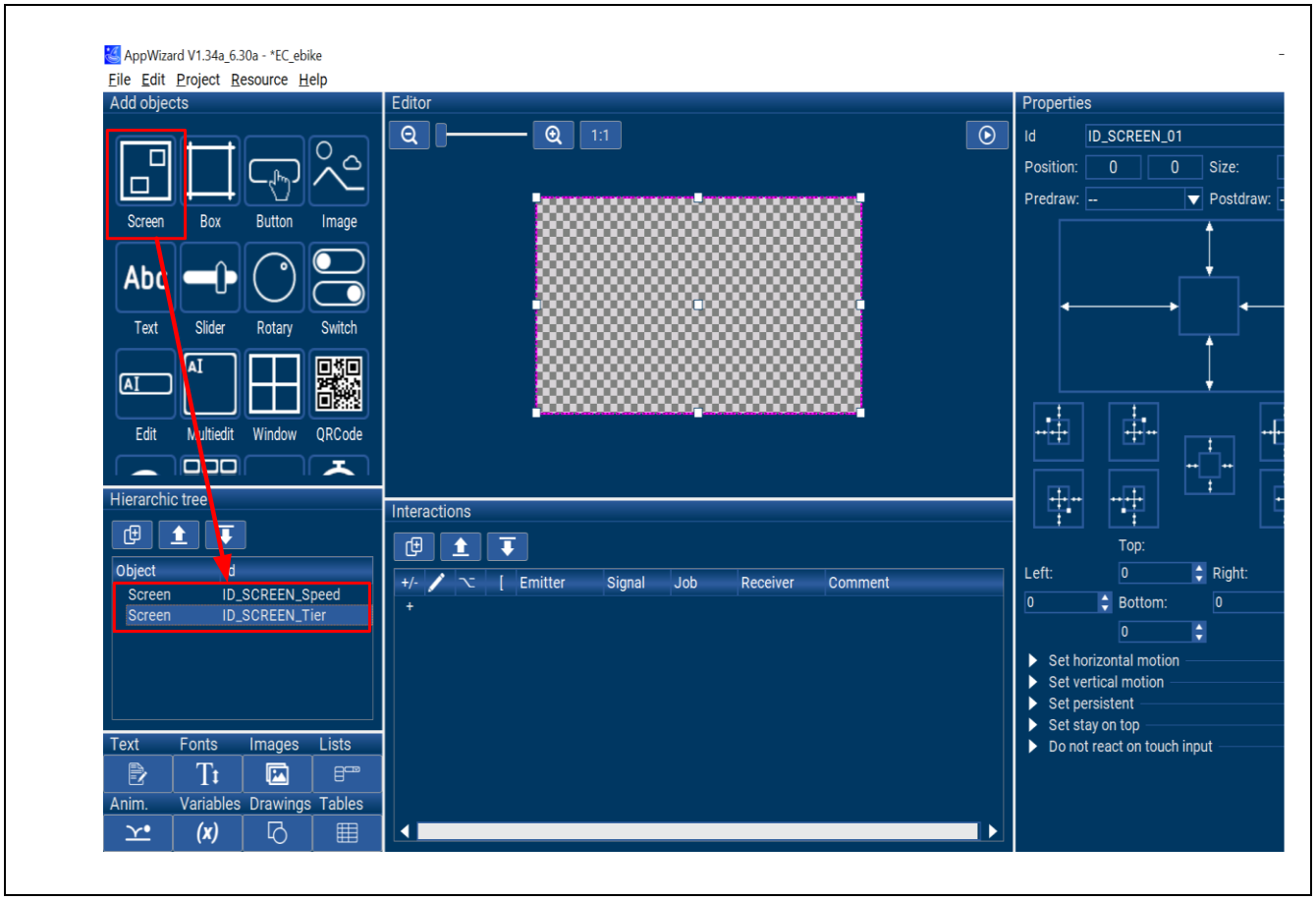


图 1-7 新增 Screen

5. 点选 ID_SCREEN_Speed 以编辑第一个画面，于 Add objects 中点选 Box，以创建一个纯色的背景。

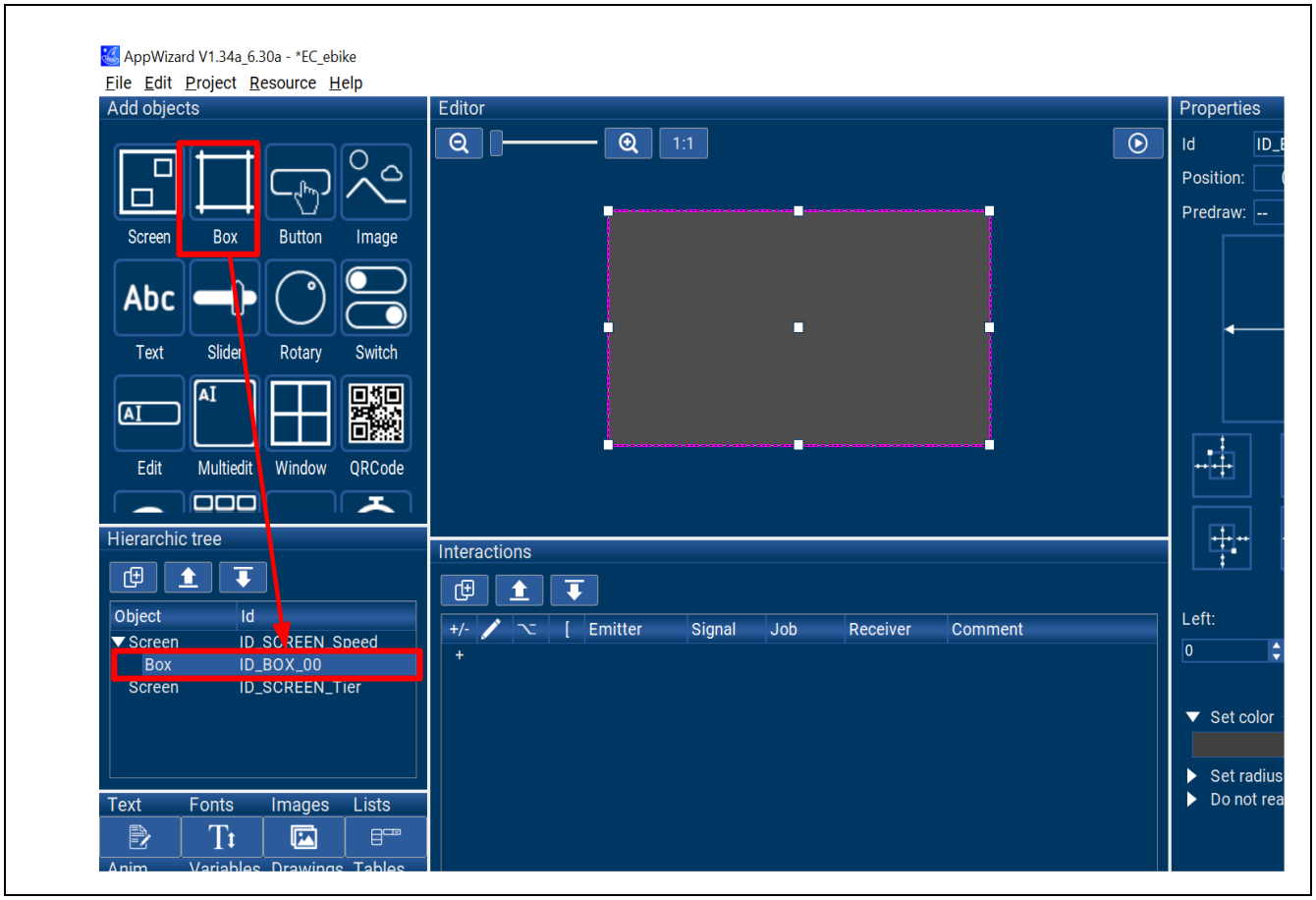


图 1-8 以 Box 创建素色背景

6. 在 Properties 中，设定 Box 的相关属性，点选 Set color 设定为黑色。

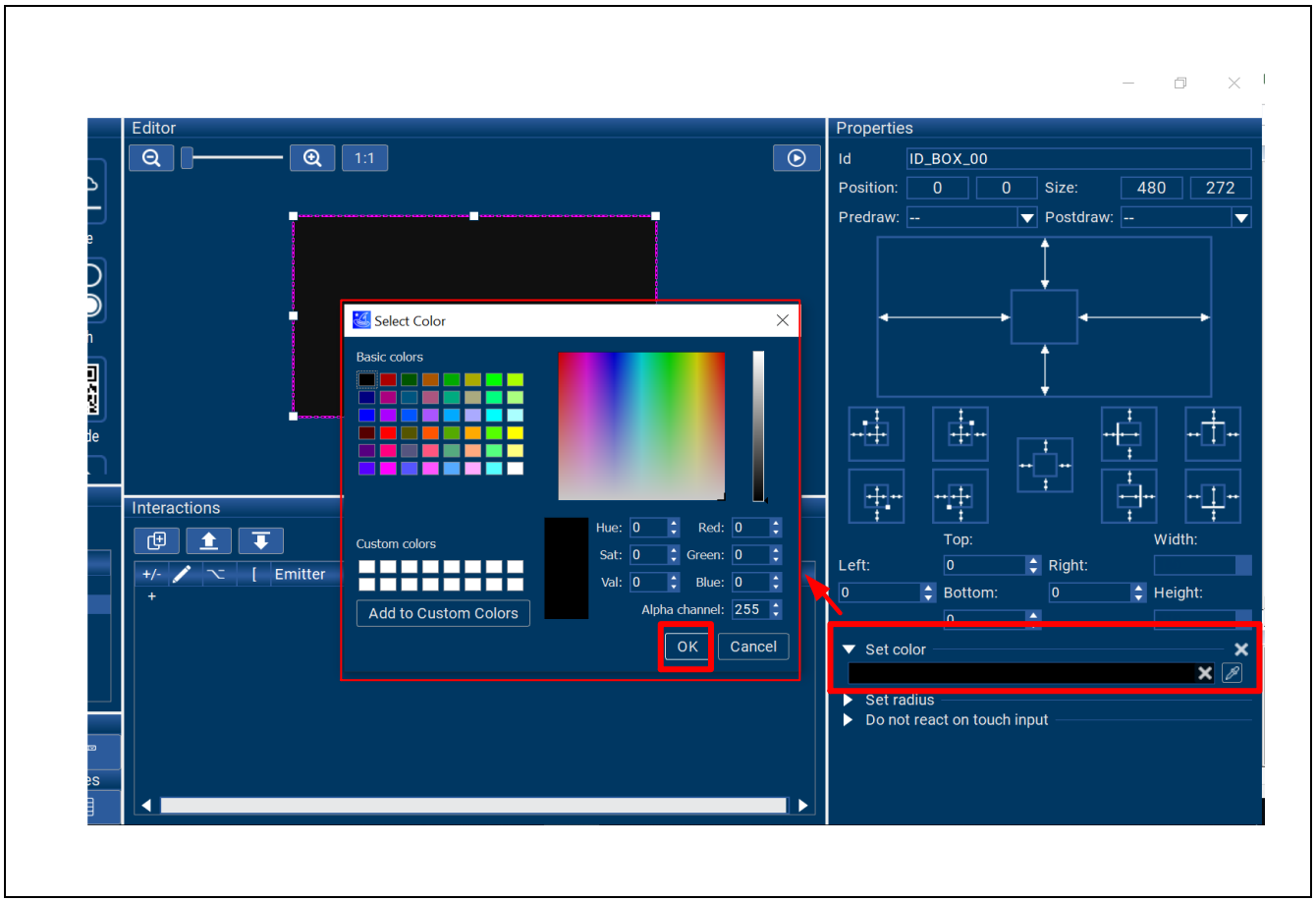


图 1-9 Box 属性设定

7. 为制作速度仪表，在 Add objects 中，点选 Rotary，并于 Hierarchic tree 中重新命名为 ID_ROTARY_Speed 以利辨认。

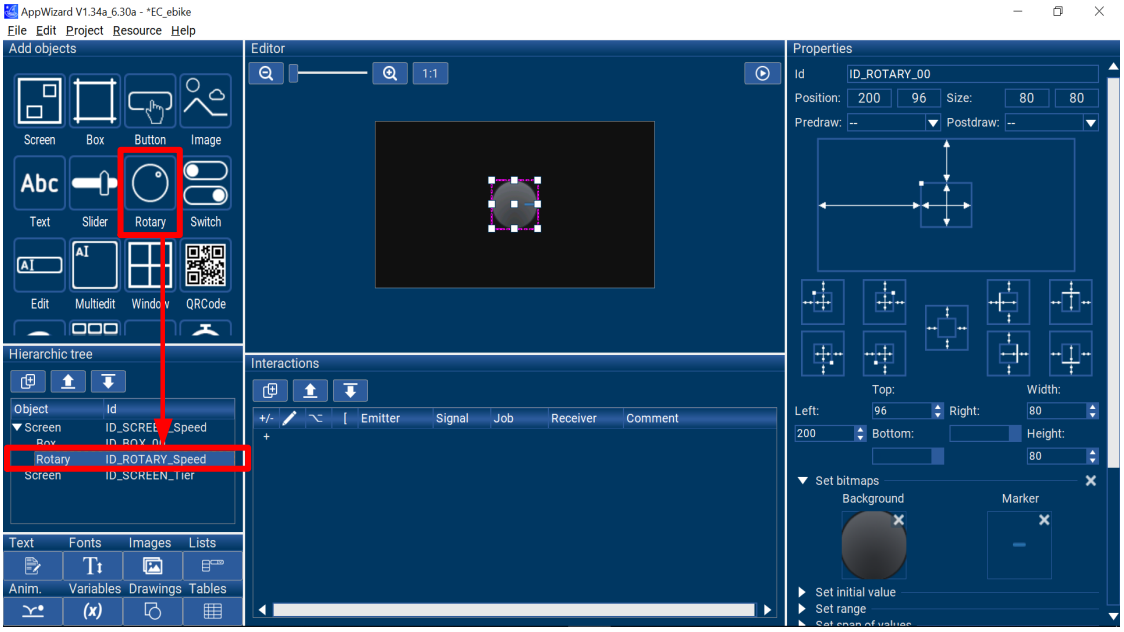


图 1-10 以 Rotary 创建速度仪表

8. 在 Properties 中，设定 Rotary 的相关属性：

1) 依图 1-11，点选 Set bitmaps，为 Background 设定客制化图片，选择 Import to project，一次选择项目会使用的图片资源。此项目所使用的图片可参考图 4-1 的目录信息从以下路径选择 PNG 图片文件：EC_M467_AppWizard_Ebike_V1.00\EC_ebike\Resource

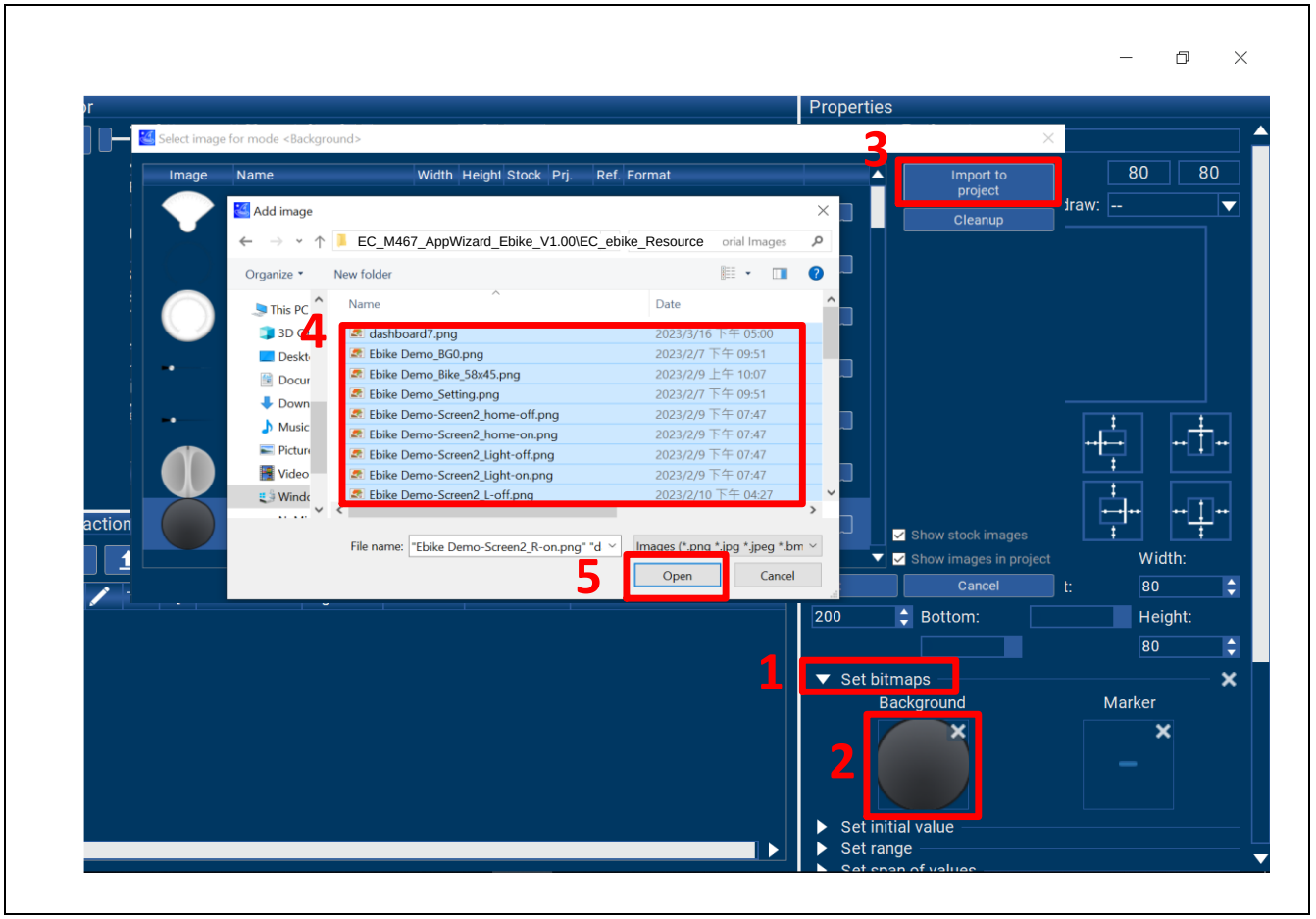


图 1-11 Rotary 属性设定-bitmap

2) 导入图片后，依图 1-12 选择图片 ，并按下 Select。

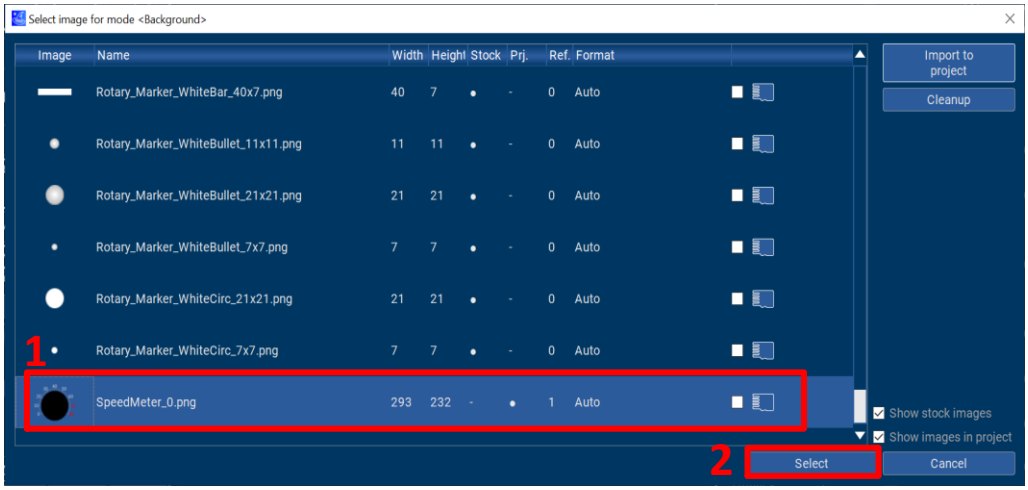


图 1-12 Rotary 属性设定-bitmap

3) 重复相同动作，依图 1-13 对 Marker 设定 bitmap，选择图片 (Rotary_Marker_WhiteBar_20x7.png)。

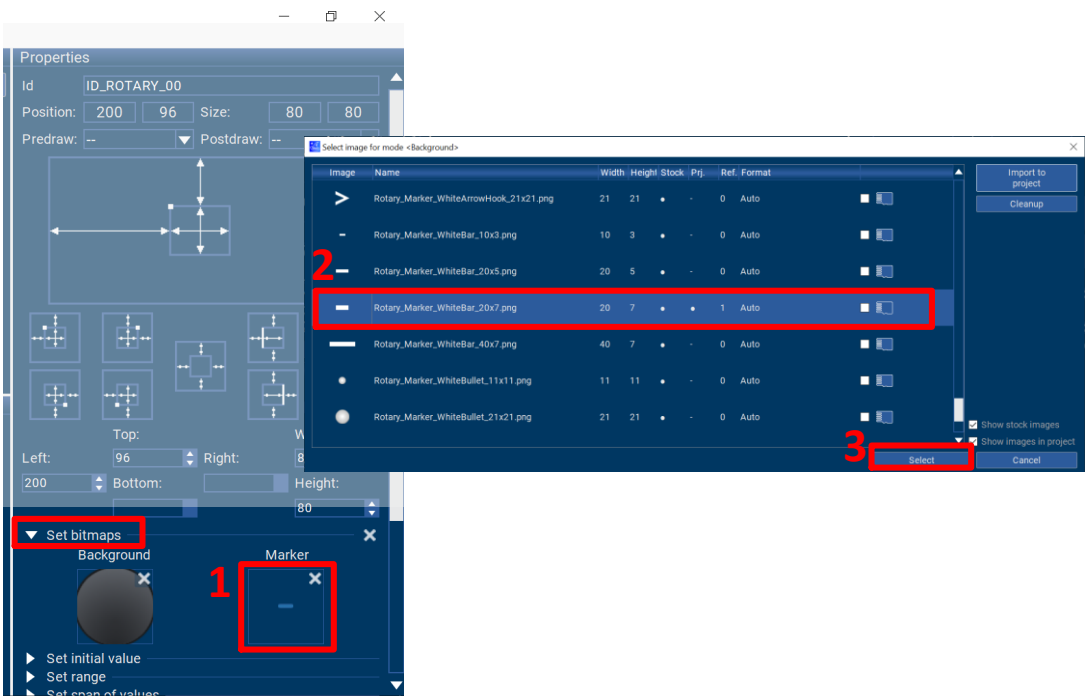


图 1-13 Rotary 属性设定-bitmap

4) 因系统设定初始角度 0°位置，如图 1-14 所示，为符合图片 的旋转角度与旋转值，依照表 1-3 说明，设定 initial value、range、span of values、offset，实际设定项目如图 1-15。

图 1-14 Rotary 系统角度设定

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 角度设定说明

5) 为对齐 marker 旋转位置，依照表 1-4 说明，设定 marker alignment、radius，实际设定项目如图 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 设定说明

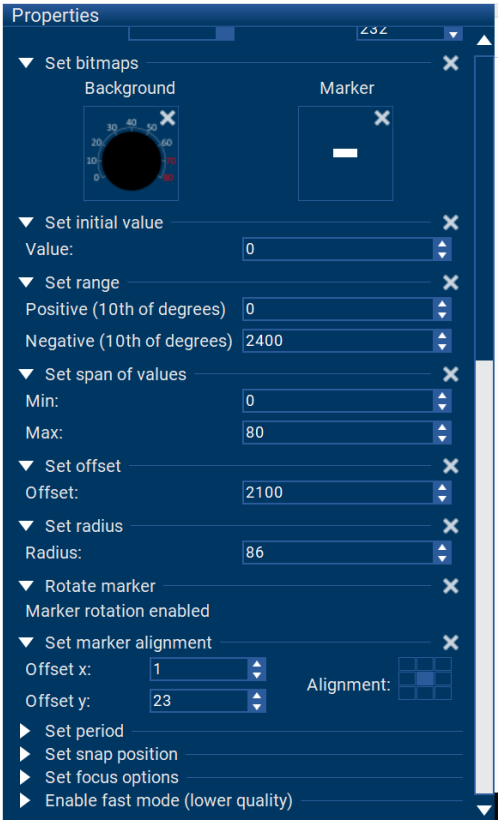


图 1-15 Rotary 属性设定

6) 以 Rotary 呈现速度仪表，设定完成后如图 1-16 所示。

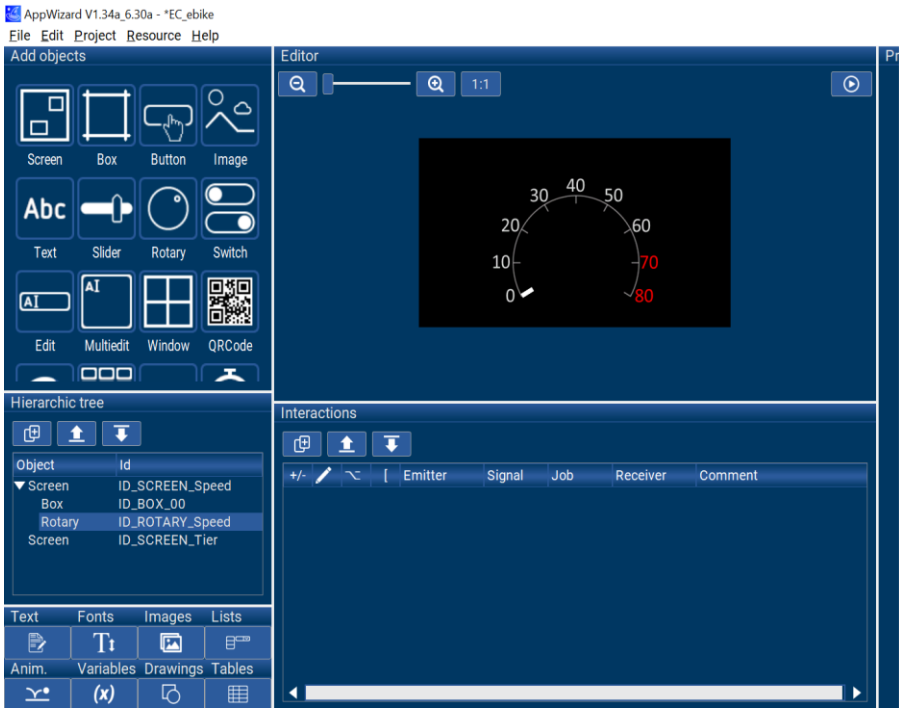


图 1-16 以 Rotary 完成速度仪表

9. 增加数字以显示速度数值，在 Add objects 中，点选 Text，并于 Hierarchic tree 中重新命名为 ID_TEXT_Speed 以利辨认。

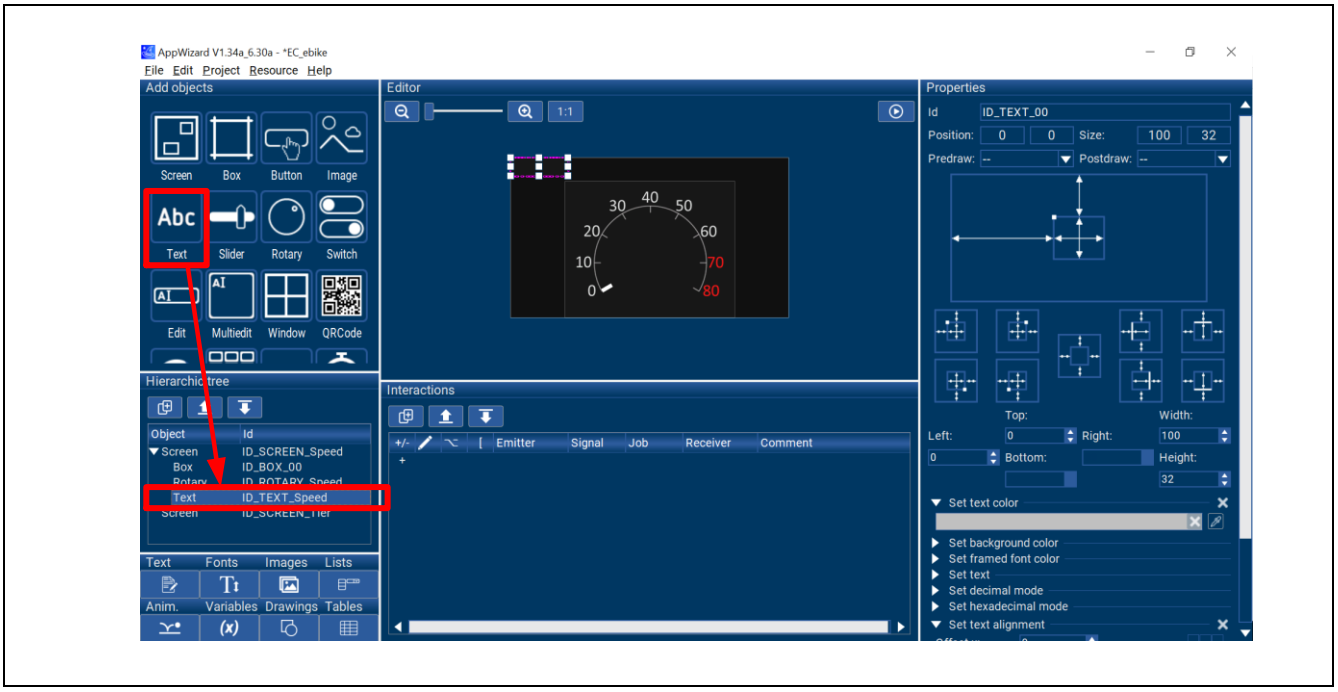


图 1-17 以 TEXT 创建速度数值

10. 在 Properties 中，设定 TEXT 的相关属性：

1) 设定颜色、decimal mode、text alignment、数值显示范围(range)，实际设定项目如图 1-19 所示。

2) 依图 1-18 步骤，设定字体(font) 。

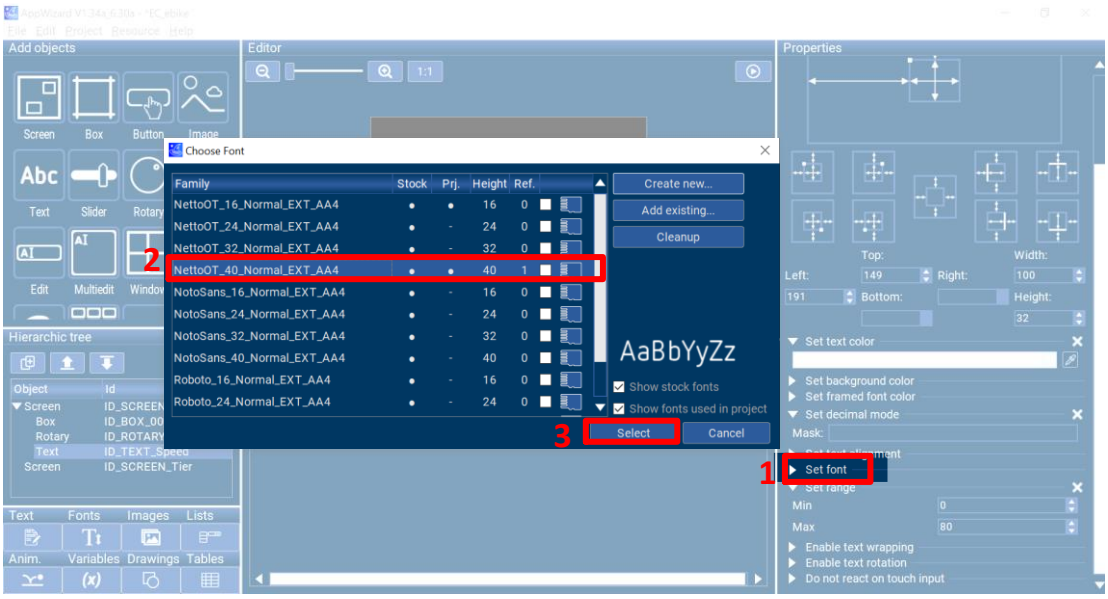


图 1-18 TEXT 属性设定-字体

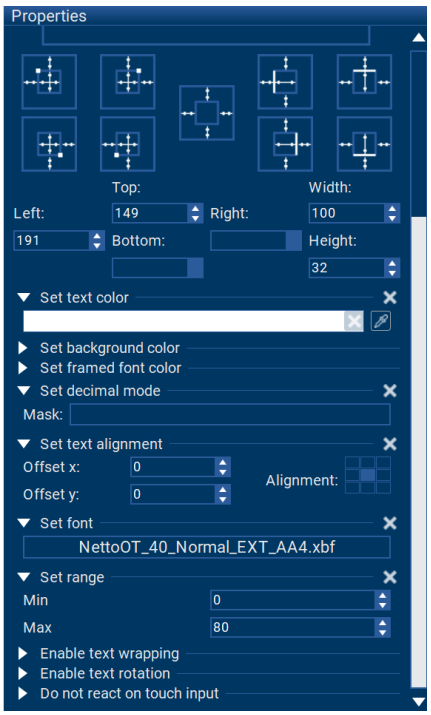


图 1-19 TEXT 属性设定

3) 以 TEXT 呈现速度数值，设定完成后如图 1-20 所示。

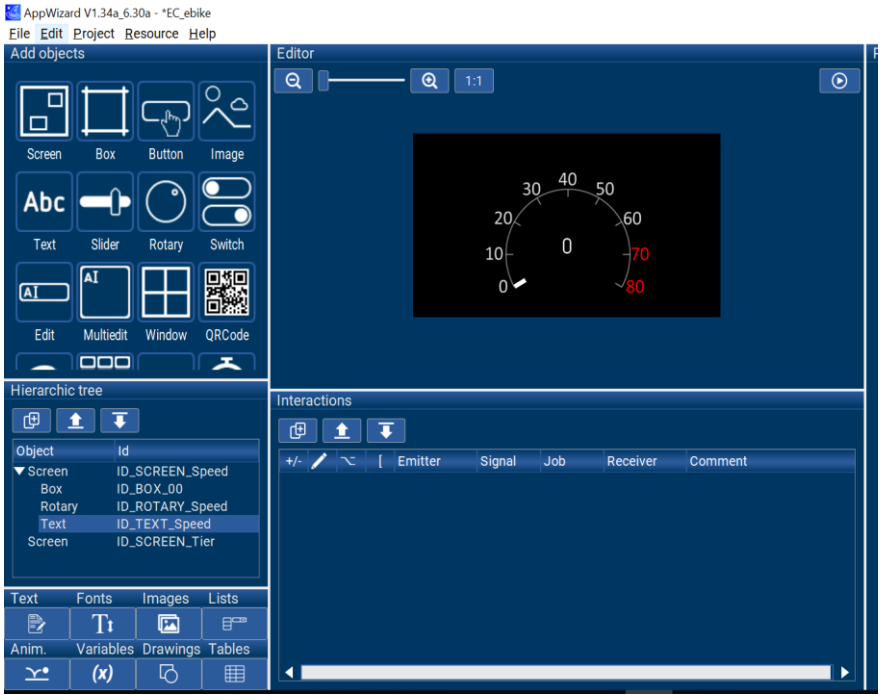


图 1-20 以 TEXT 完成速度数值显示

11. 增加文字以显示数值名称、数值单位、标题等，在 Add objects 中，点选 Text。

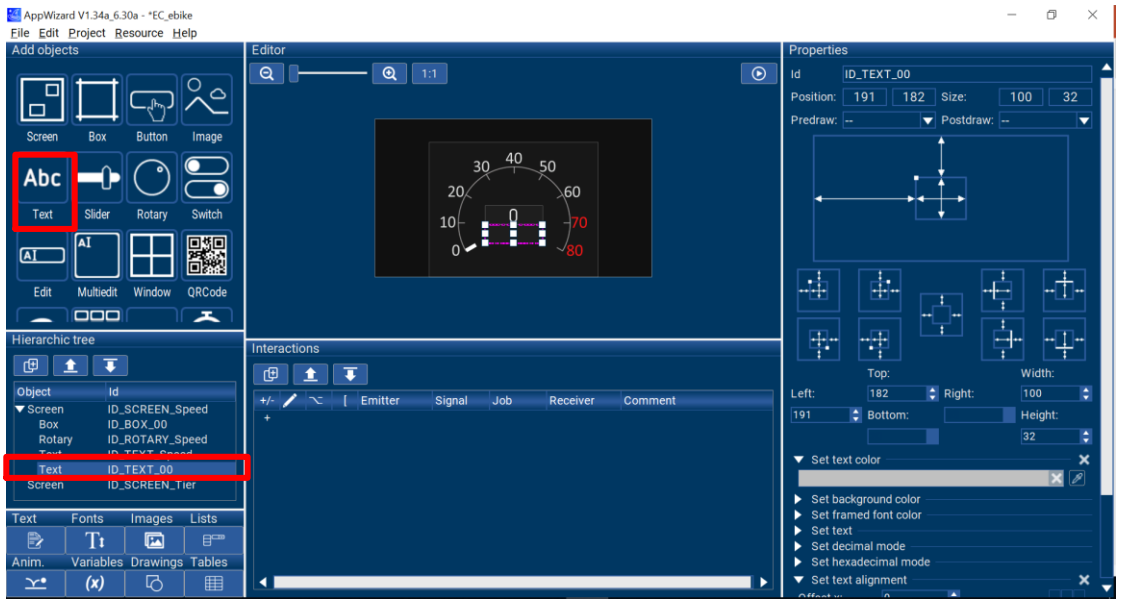


图 1-21 以 TEXT 创建速度单位文字

1) 设定颜色、text alignment。

2) 依图 1-22~图 1-26 步骤，设定显示的文字，并同时显示中文与英文。

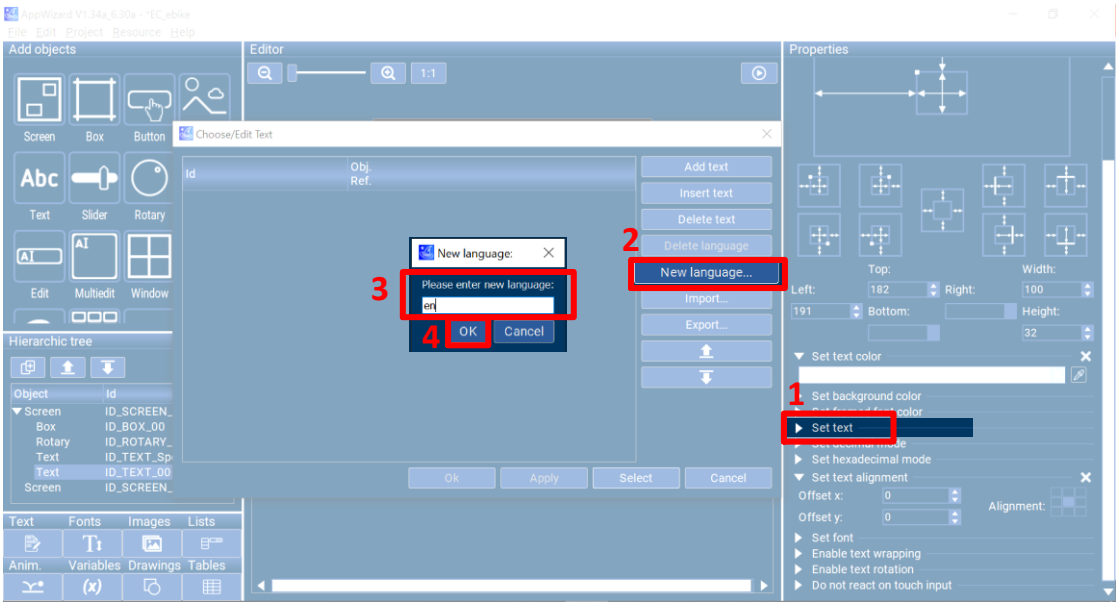


图 1-22 TEXT-创建第一种语言-英文

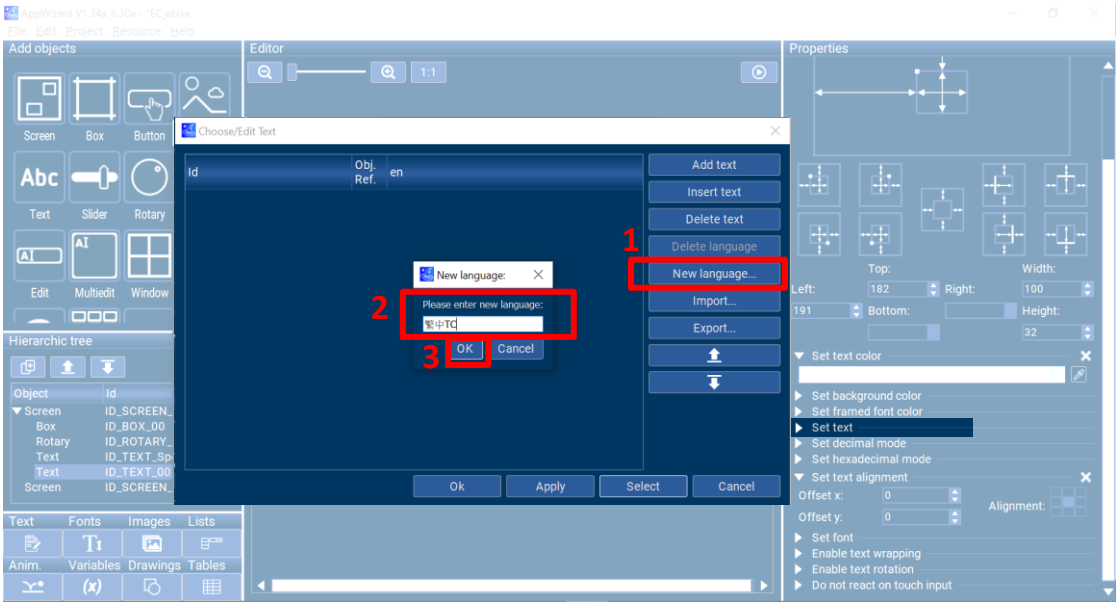


图 1-23 TEXT-创建第二种语言-中文

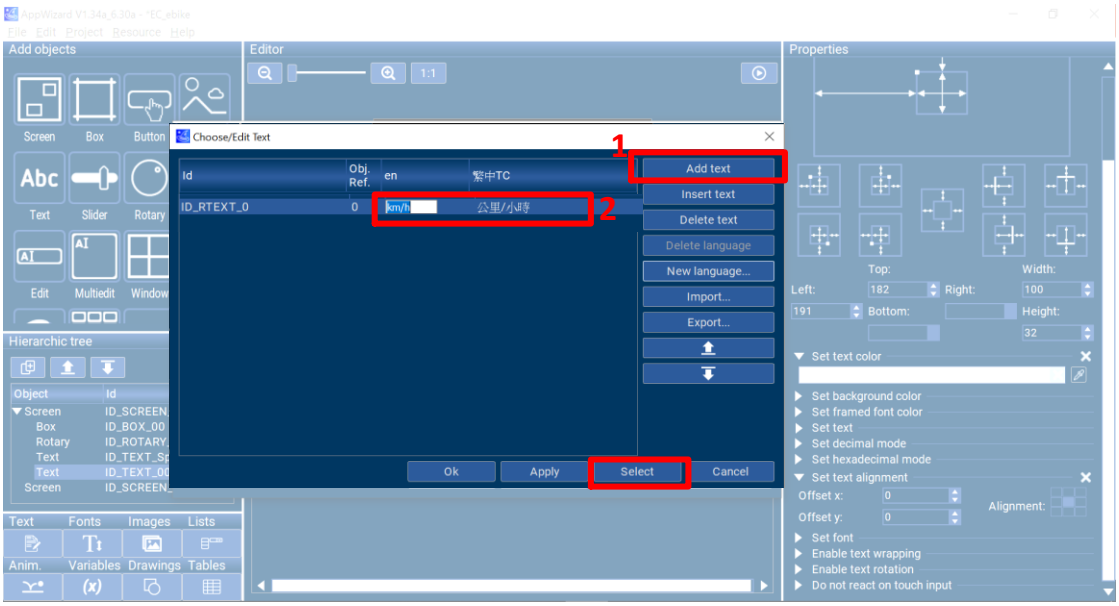


图 1-24 TEXT- 分别编辑中文与英文的文字内容

3) 新增并选择能同时显示英文与中文的字体(font) 。

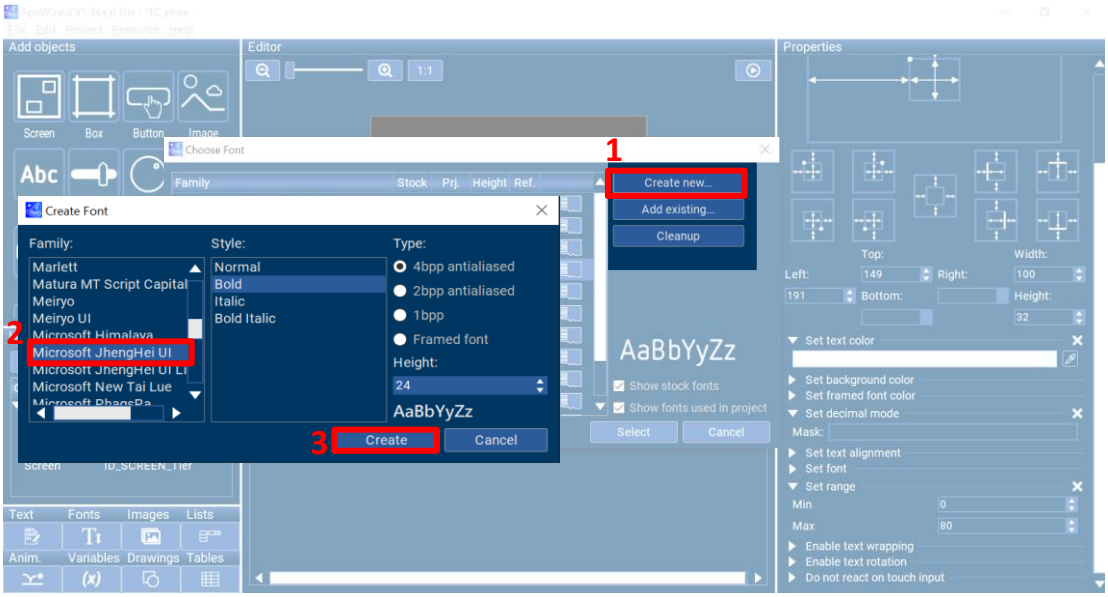


图 1-25 TEXT 属性设定-字体

4) 设定 codepoint range，跳出视窗”Define codepoints to be used”后，勾选 Project text only 即仅选择项目会使用到的字符范围。

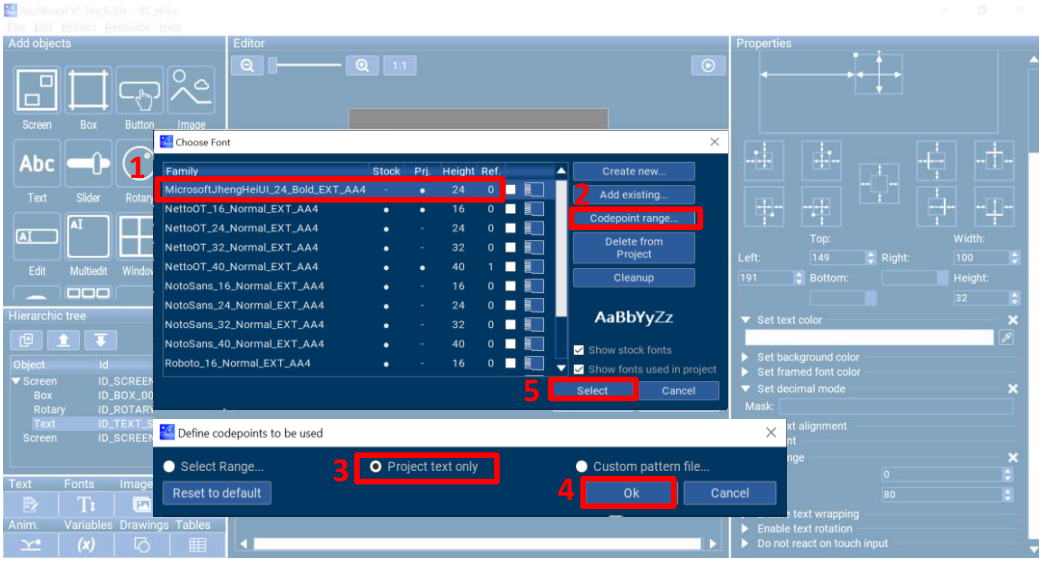


图 1-26 TEXT 属性设定-字体与字符范围

5) 以 TEXT 呈现英文与中文文字，设定完成后如图 1-27 所示。

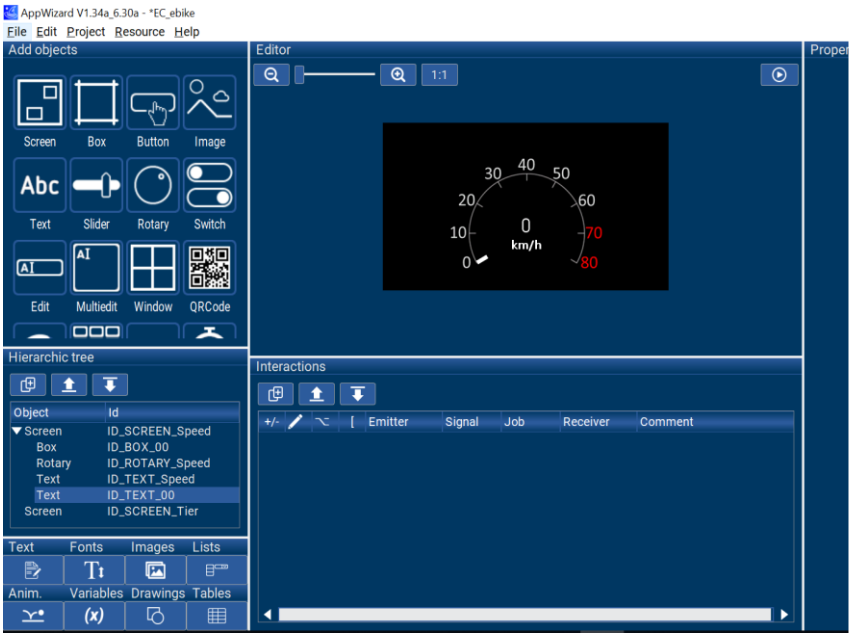


图 1-27 以 TEXT 完成速度单位显示

12. 使用 Image 物件，新增分隔线至画面中，在 Add objects 中，点选 Image。

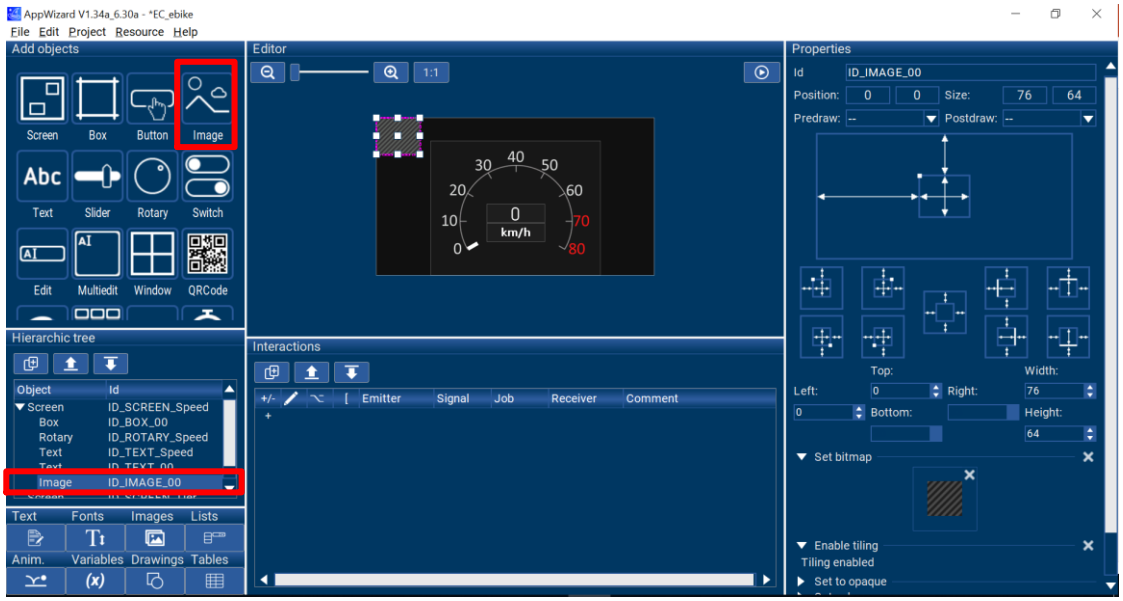


图 1-28 以 Image 新增画面分隔线

13. 在 Properties 中，设定 Image 的相关属性：

1) 依图 1-29，点选 Set bitmap，设定客制化图片，并关闭 Enable tiling 功能，取消无限延伸。

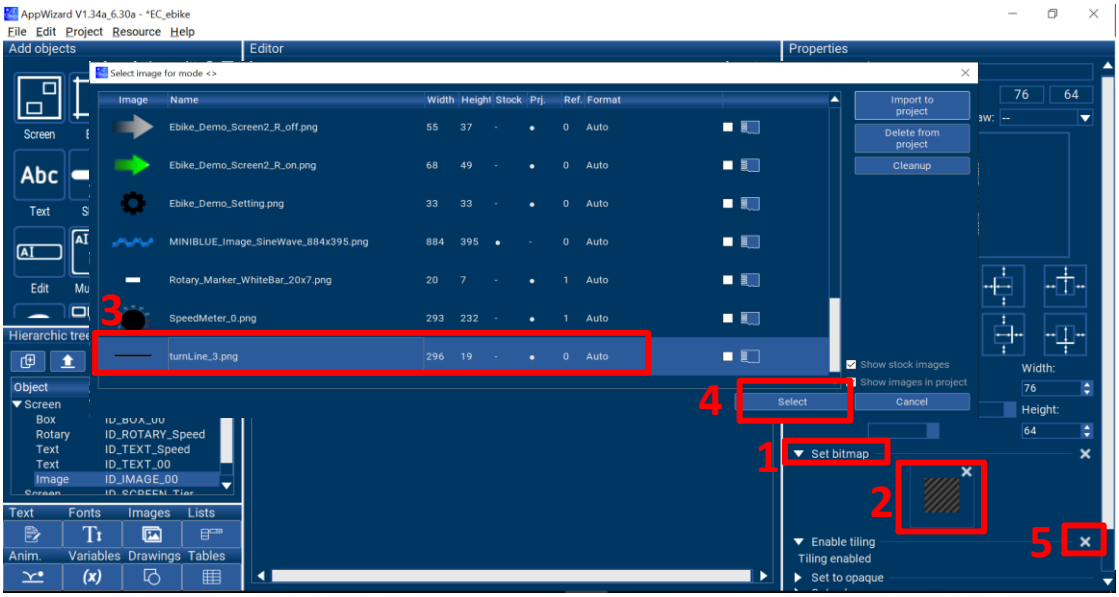


图 1-29 Image 属性设定 - bitmap

2) 设定完成如图 1-30 所示。

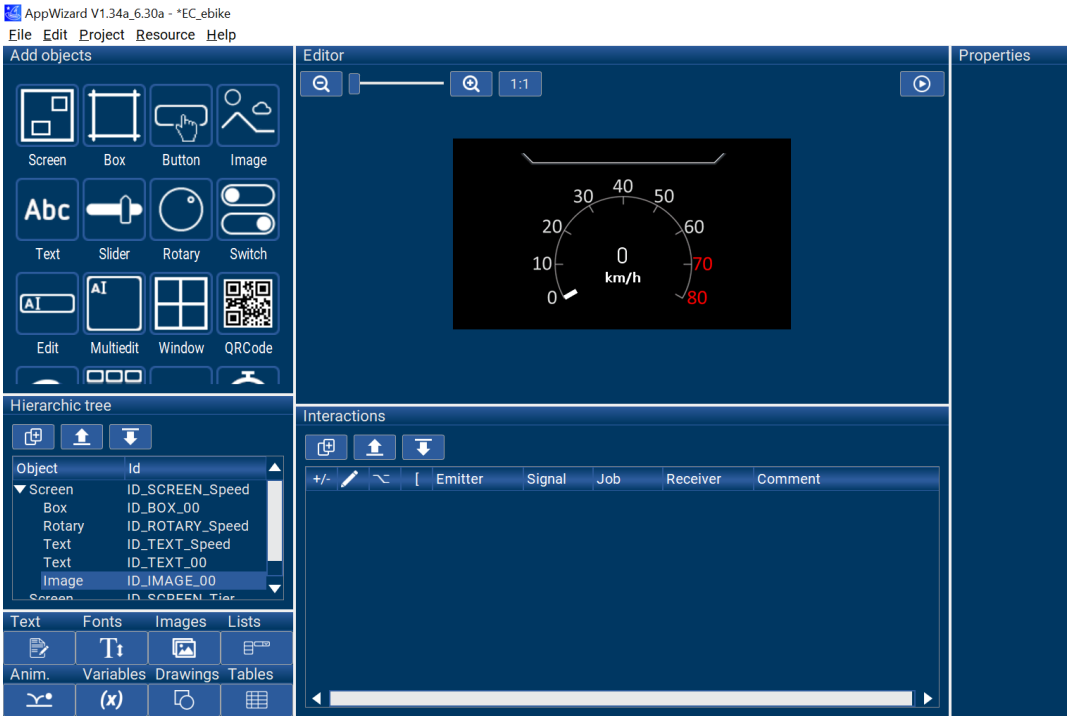


图 1-30 以 Image 完成画面分隔线

14. 使用 button 物件表示方向灯功能，在 Add objects 中，点选 Button ，并于 Hierarchic tree 中重新命名为 “ID_BUTTON_Left” 以利辨认。

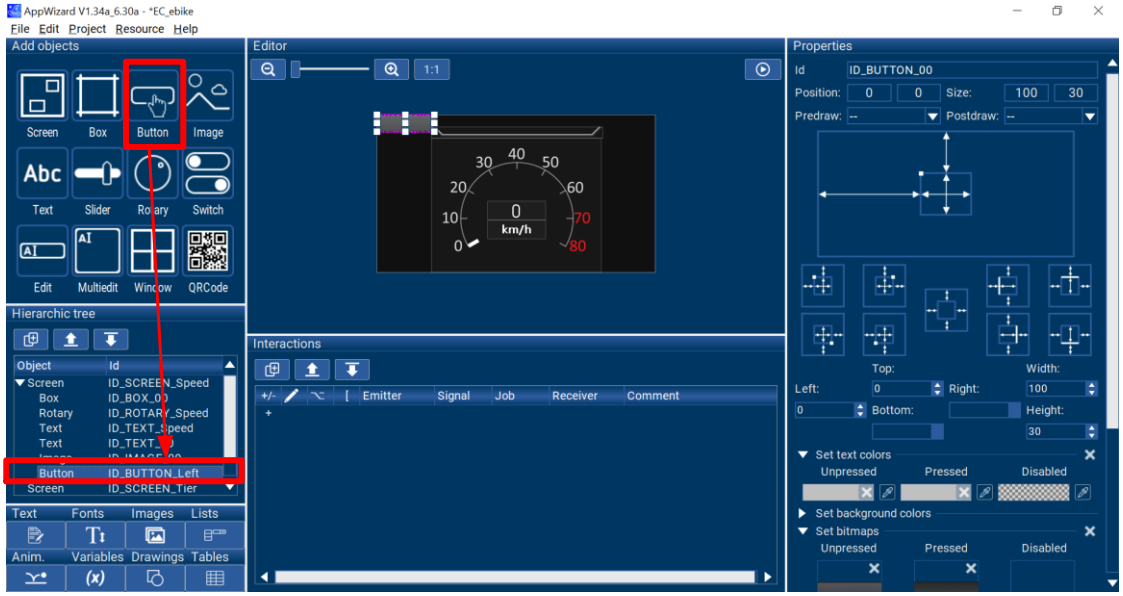
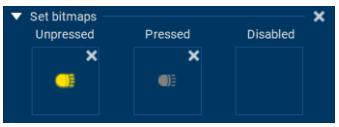


图 1-31 以 Button 设计方向灯

15. 在 Properties 中，设定 Button 的相关属性：

1) 此次范例会使用到的功能仅需设定按钮未按下(Unpressed)、按下(Pressed)等不同状态的 bitmap，bitmap 选择方式可参考步骤 13，此范例分别对左右方向灯、警示灯、大灯警示灯设定对应图片。

BUTTON ID	Bitmap
ID_BUTTON_Left	
ID_BUTTON_Right	
ID_BUTTON_Light	

ID_BUTTON_Warn	

表 1-5 位图

2) 设置完成并调整位置，如图 1-32 所示。

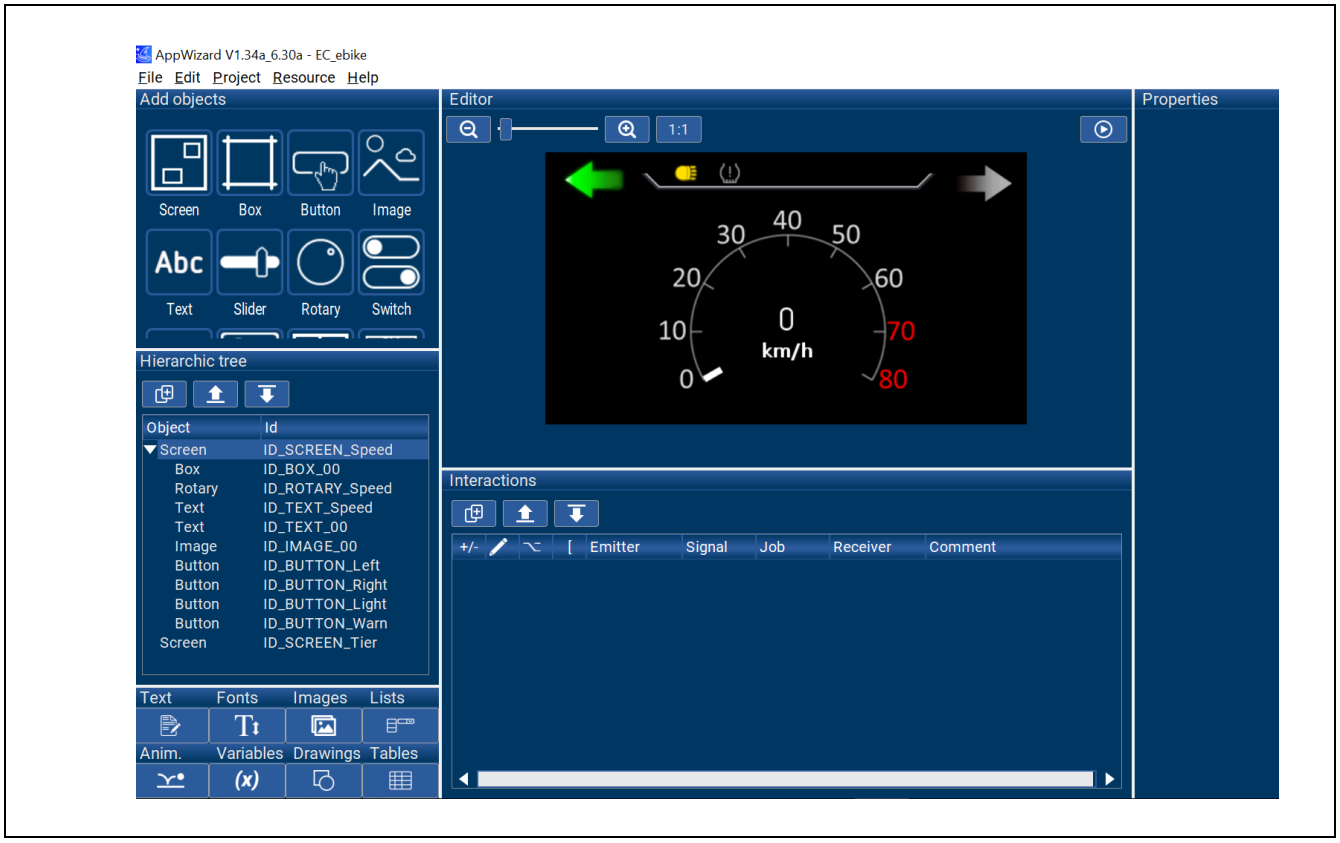


图 1-32 以 Button 完成大灯、警示灯与方向灯

16. 新增”跳至设定页”按钮，不同于步骤 14 使用图片(bitmap)作为按钮的接口，此按钮将不使用图片资源，在 Add objects 中，点选 Button ，并于 Hierarchic tree 中重新命名为 ID_BUTTON_Setting 以利辨认。

    1) 设定文字颜色、背景颜色、取消 bitmap 设定、设定文字与字体，设定内容如图 1-33。

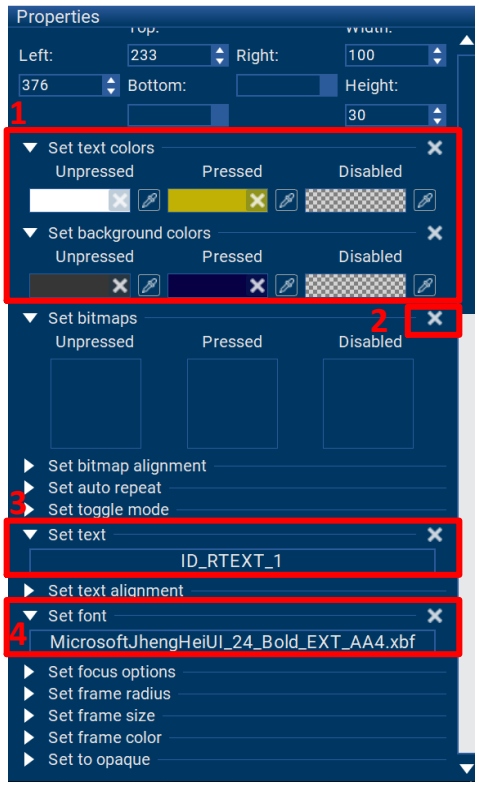


图 1-33 Button 属性设定

17. 设定并调整完位置，如图 1-34。

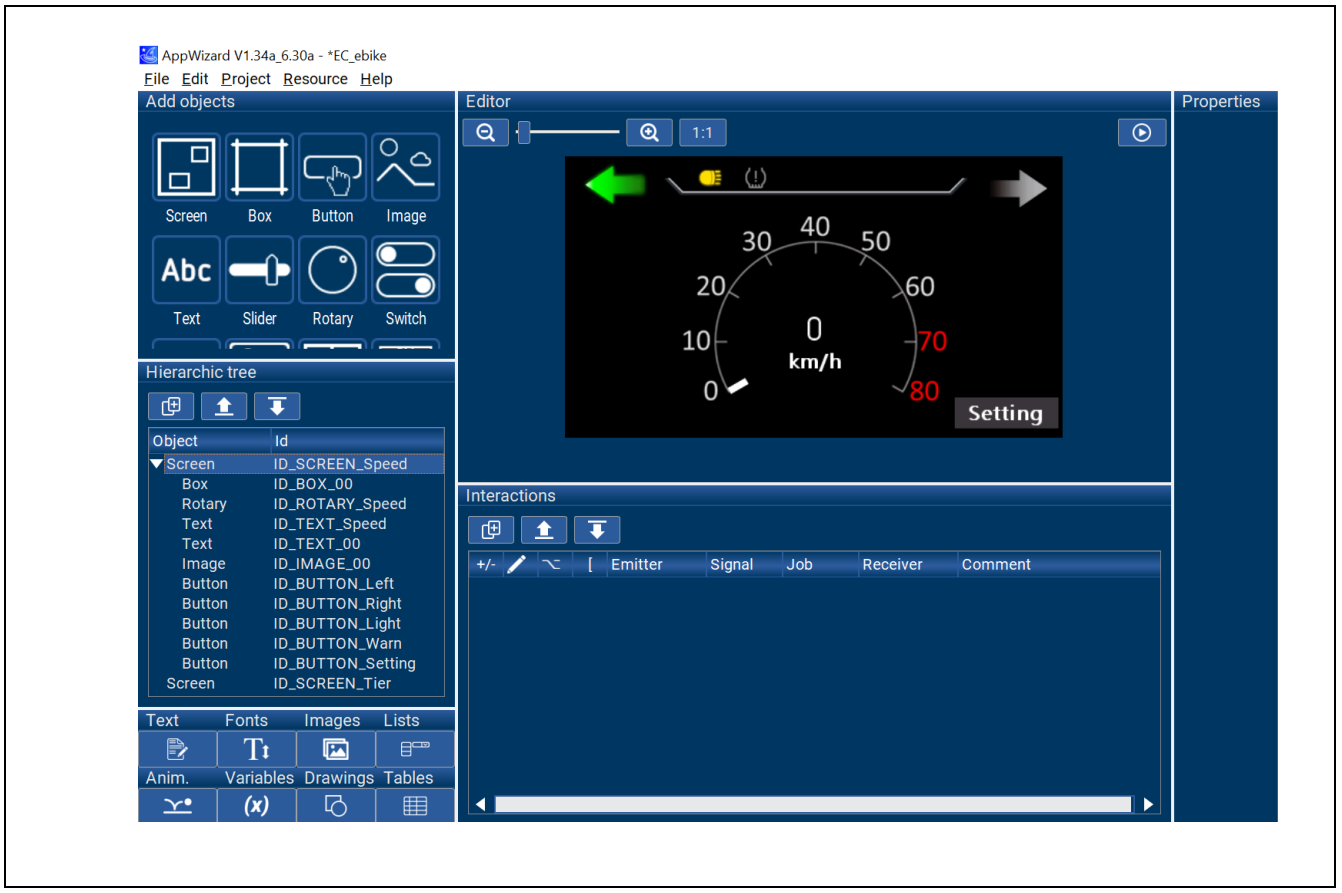


图 1-34 以 Button 完成页面跳转按钮

18. 使用 Progbar 物件，作为显示电池电量功能，在 Add objects 中，点选 Progbar ，并于 Hierarchic tree 中重新命名为 ID_Progbar_Battery 以利辨认。

图 1-35 以 Progbar 设计电池电量显示

19. 在 Properties 中，设定 Progbar 的相关属性

1) 依顺序并如图 1-36 所示，设定初始值(initial value)、数值范围(range)、颜色(color)、取消 bitmap 功能、取消 tiling 功能、设定半径(radius)、设定边框大小(frame size)、设定边框颜色(frame color) 。

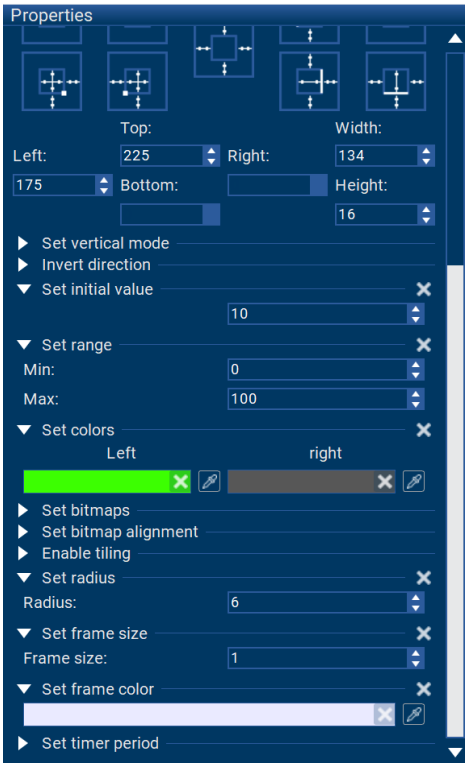


图 1-36 Progbar 属性设定

2) 设定并调整完位置，如图 1-37。

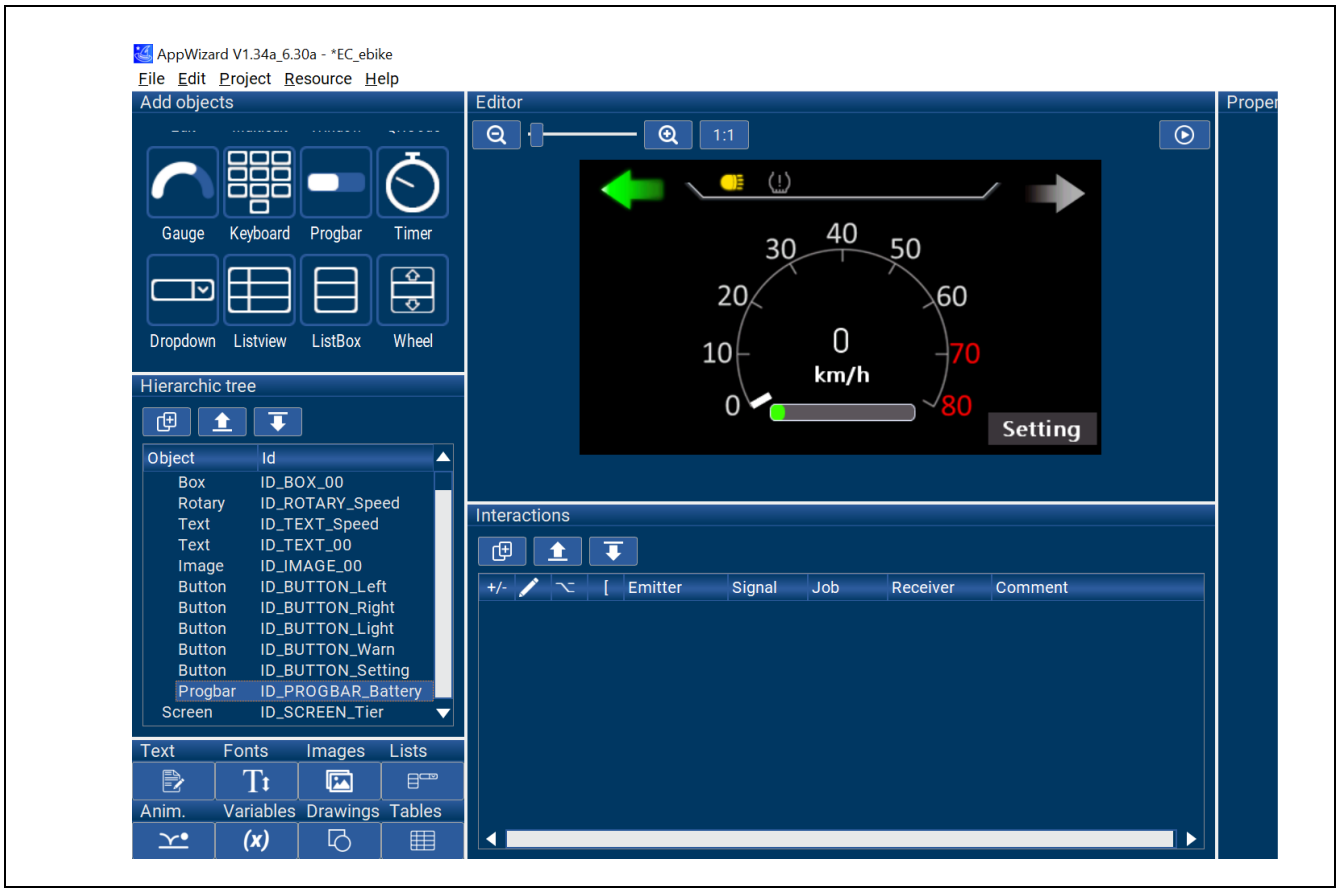


图 1-37 以 Progbar 完成电量显示条

20. 使用 TEXT，新增电池说明文字、单位，请依照步骤 11 操作，此范例相关设定可参考图 1-38，完成后如图 1-39 所示。

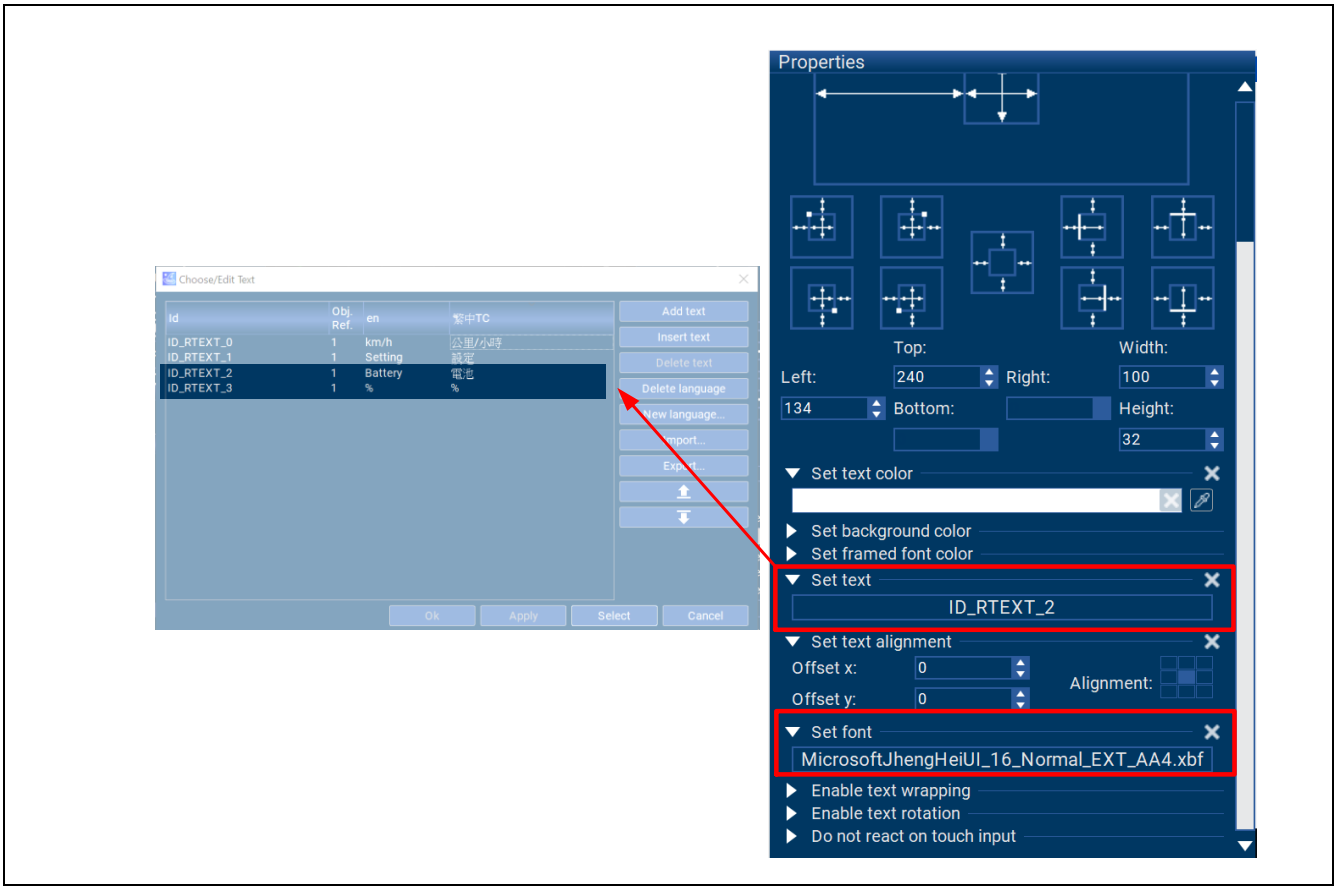


图 1-38 TEXT 属性设定

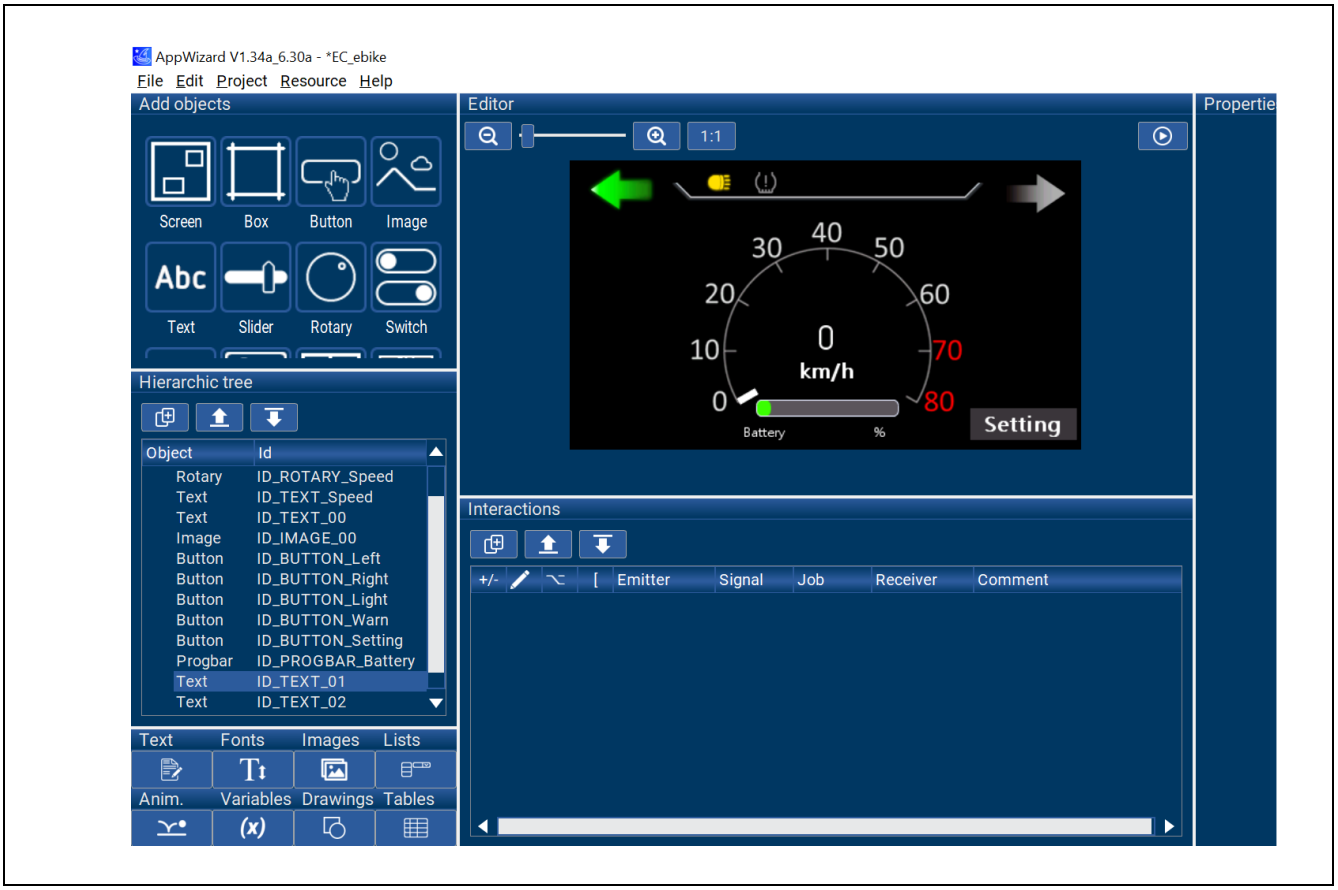


图 1-39 以 TEXT 完成电池文字与单位显示

21. 使用 TEXT，新增电池电量数值显示，请依照步骤 9 ~ 步骤 10 操作，此范例相关设定可参考图 1-40，完成后如图 1-41 所示。

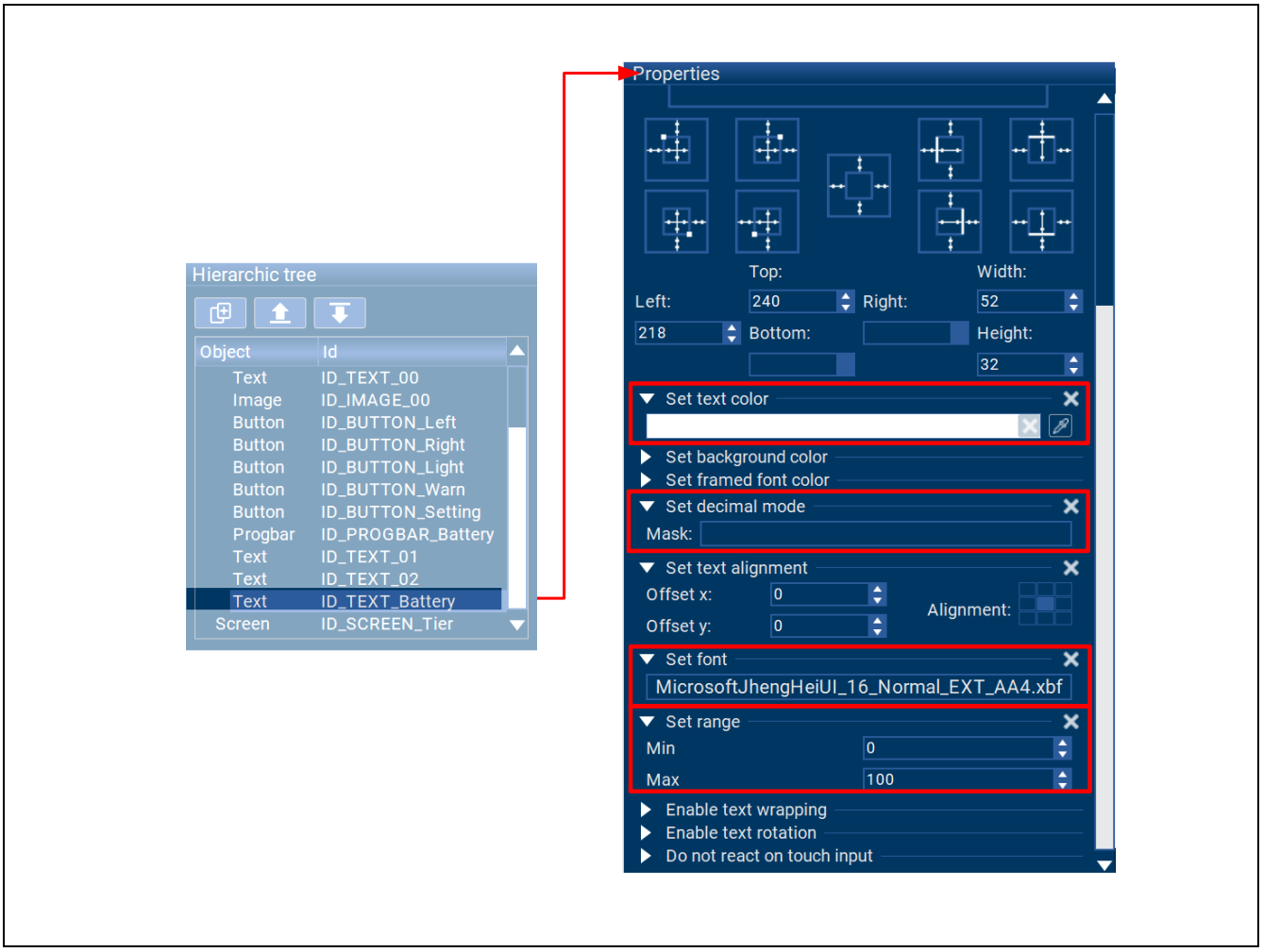


图 1-40 TEXT 属性设定

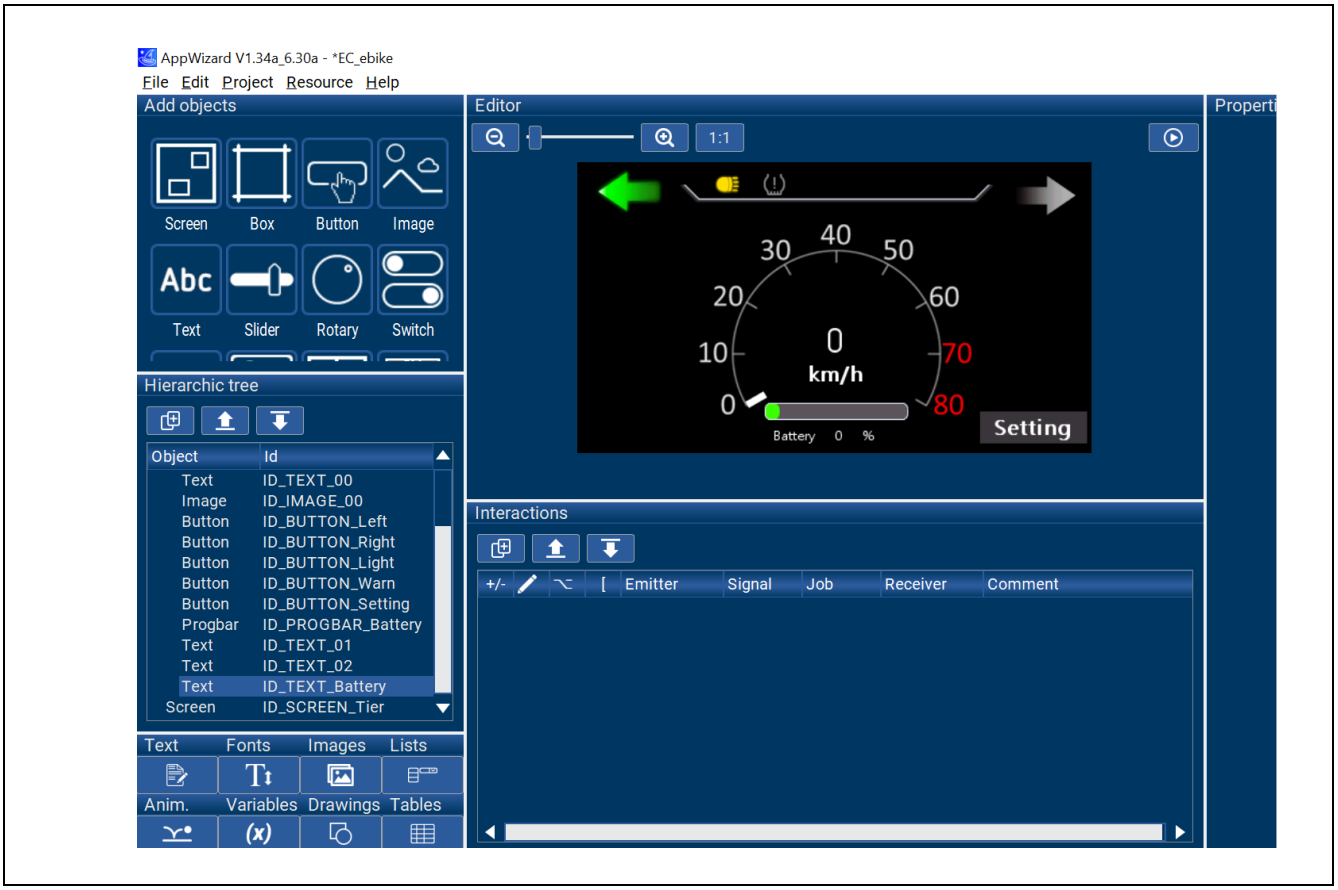


图 1-41 以 TEXT 完成电池电量数值显示

22. 开始编辑胎压页面，于 Hierarchic tree 中点选 ID_SCREEN_Tire。

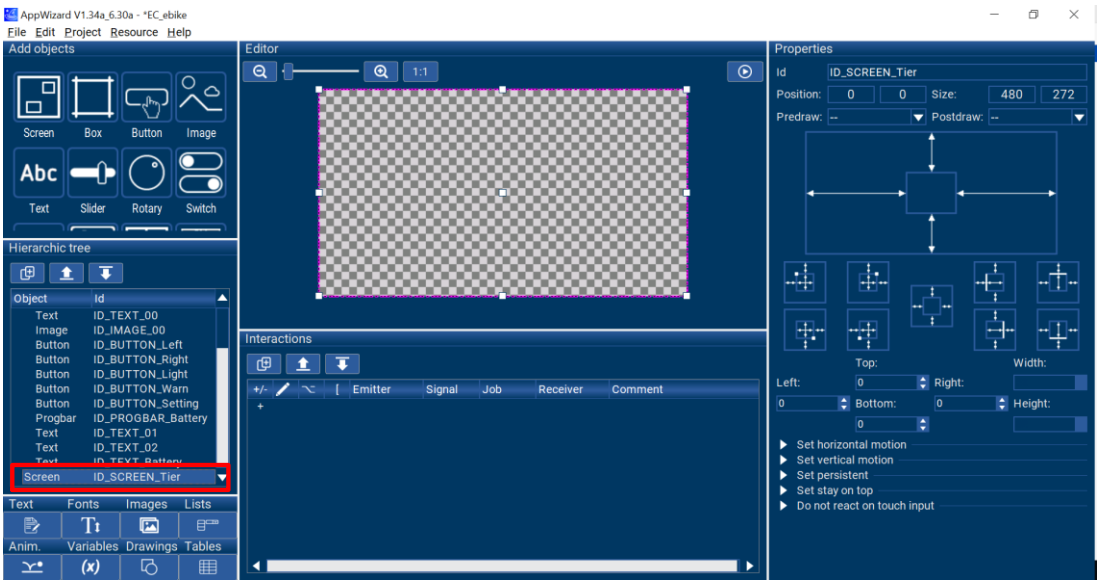


图 1-42 点选 ID_SCREEN_TIRE 以编辑胎压页面

23. 使用 Image 物件，汇入一张大小 480x272 的图当背景，在 Add objects 中，点选 Image。

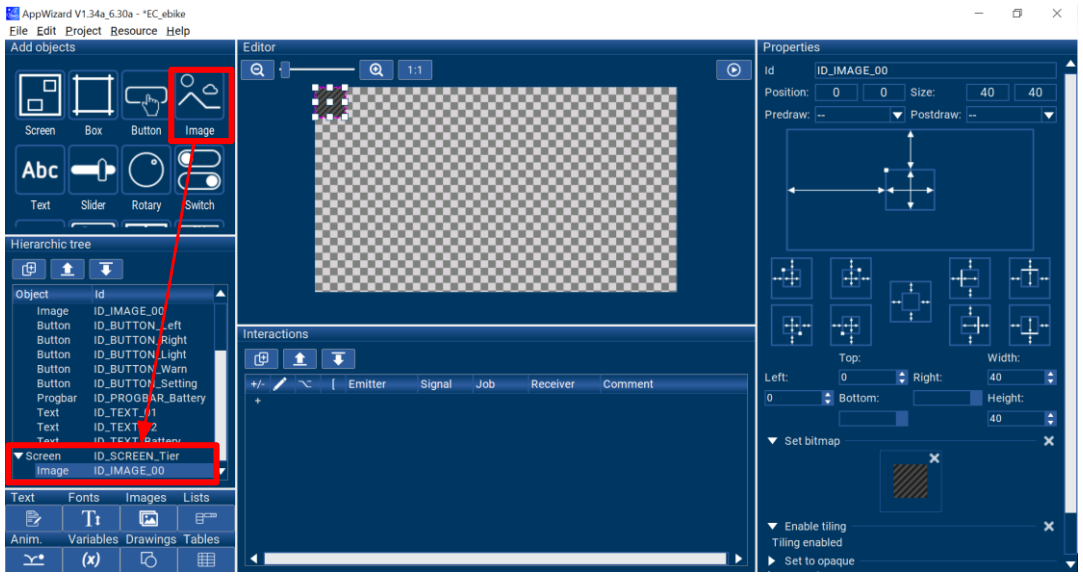


图 1-43 使用 Image 汇入背景图片

24. 依照步骤 12~13，选择图片，完成后如图 1-44 所示。

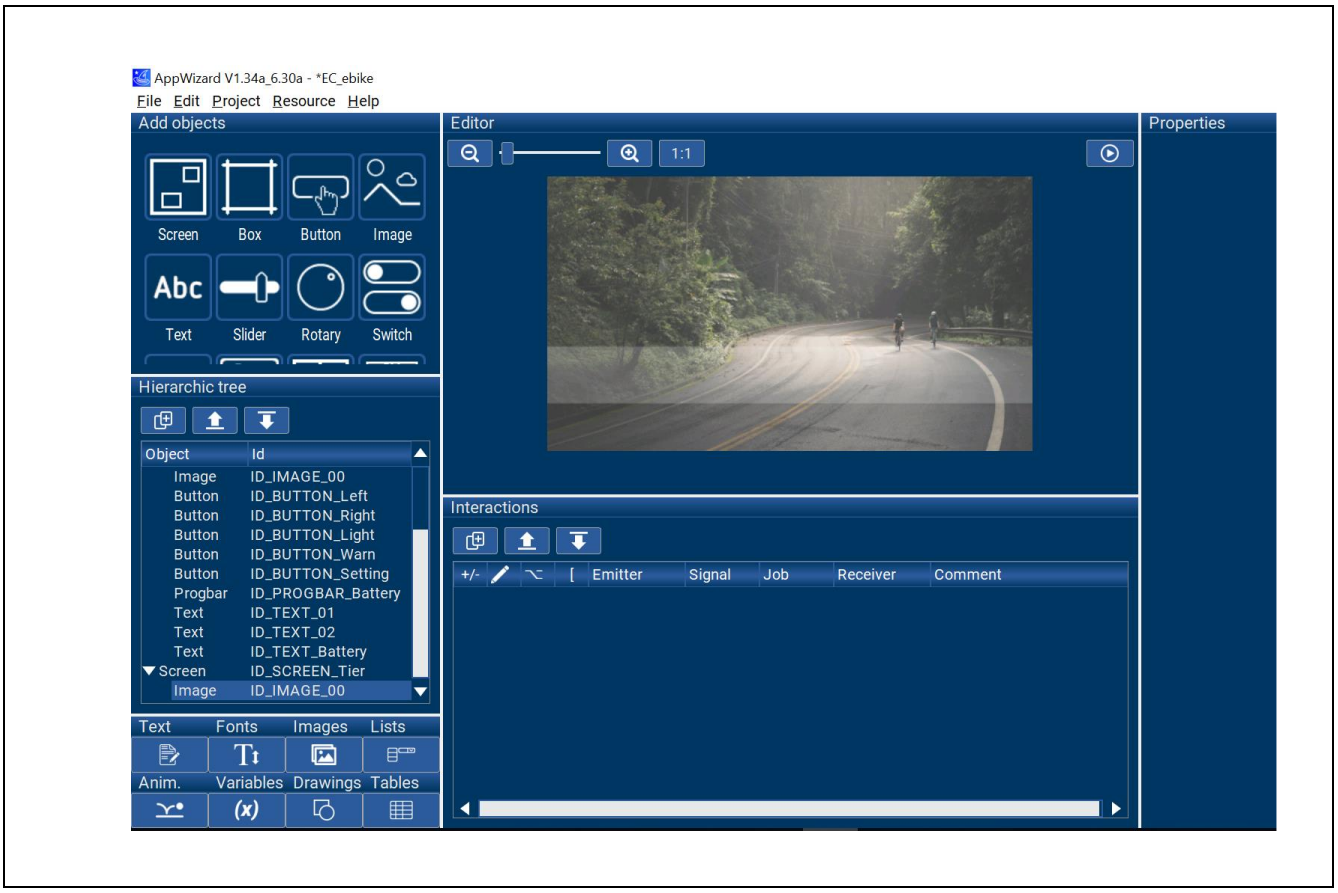


图 1-44 以 Image 完成背景图片

25. 选择脚踏车图片，并依照步骤 12~13，使用 Image 汇入，完成后如图 1-45 所示。

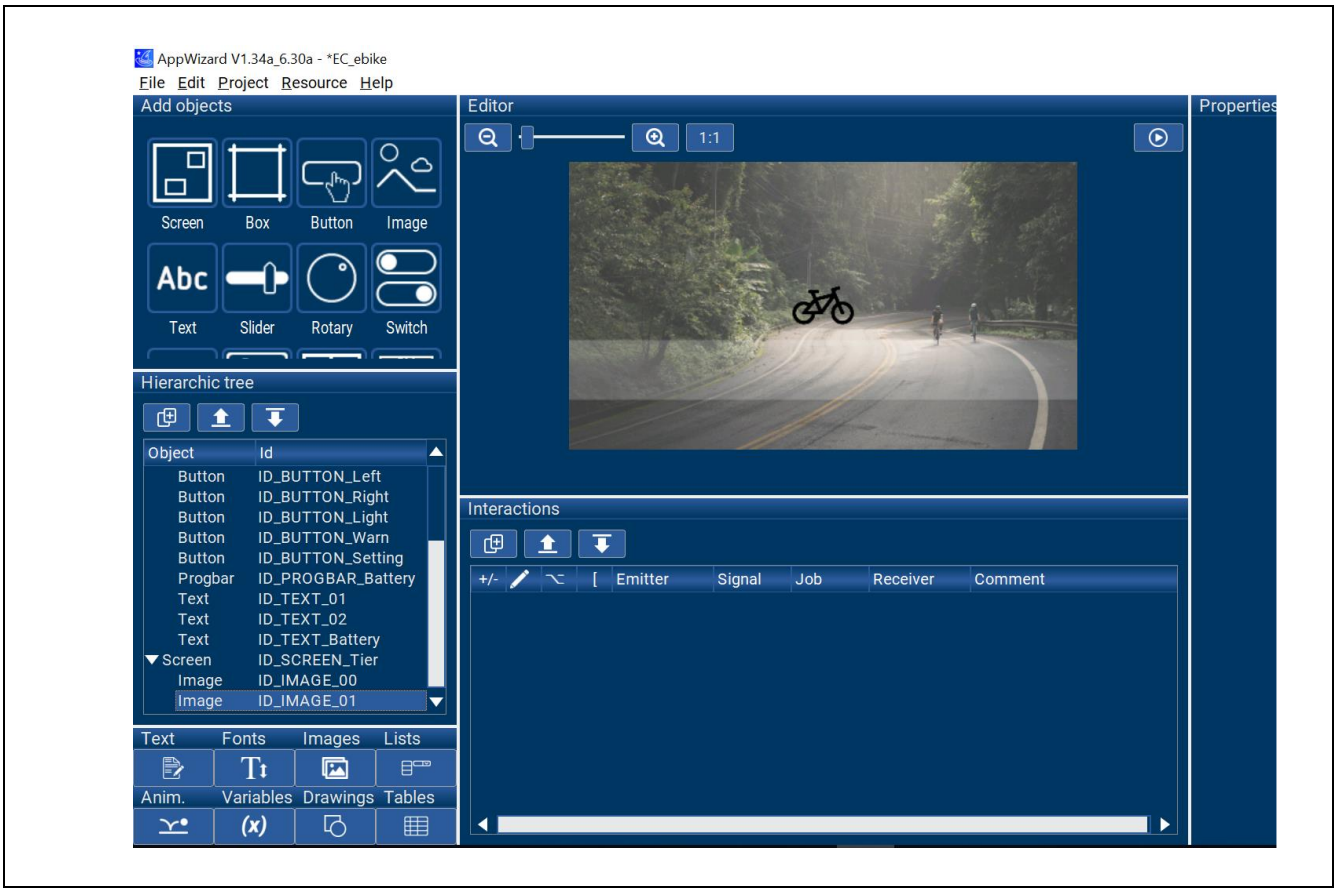


图 1-45 以 Image 完成脚踏车图片

26. 使用 TEXT 物件，新增标题、胎压文字与单位，请依照步骤 11 操作，此范例相关设定可参考图 1-46，完成后如图 1-47 所示。而胎压单位，如 psi、kPa、bar 等，亦可使用相同方式设计显示于画面中。

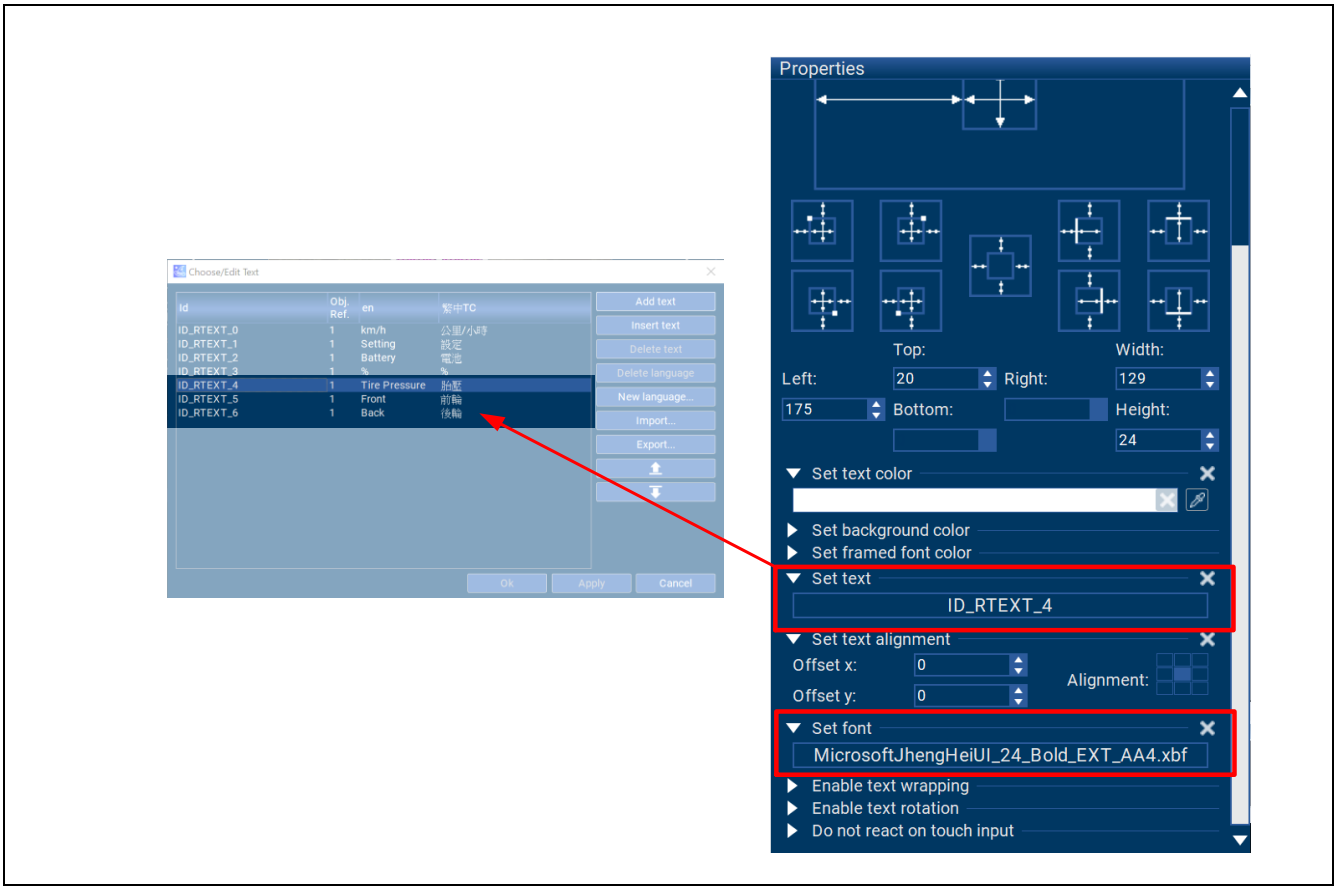


图 1-46 TEXT 属性

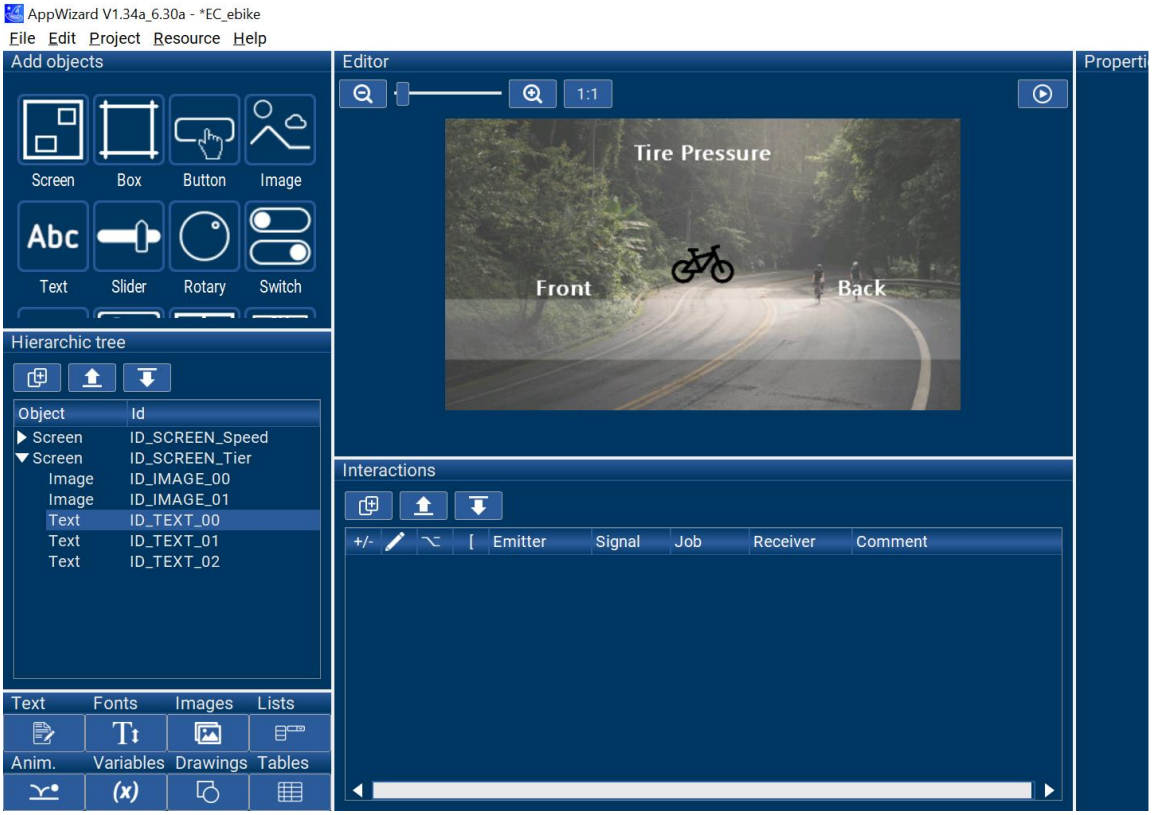


图 1-47 以 TEXT 完成胎压文字显示

27. 使用 TEXT，新增胎压数值显示，请依照步骤 9 ~ 步骤 10 操作，此范例相关设定可参考图 1-48，完成后如图 1-49 所示。

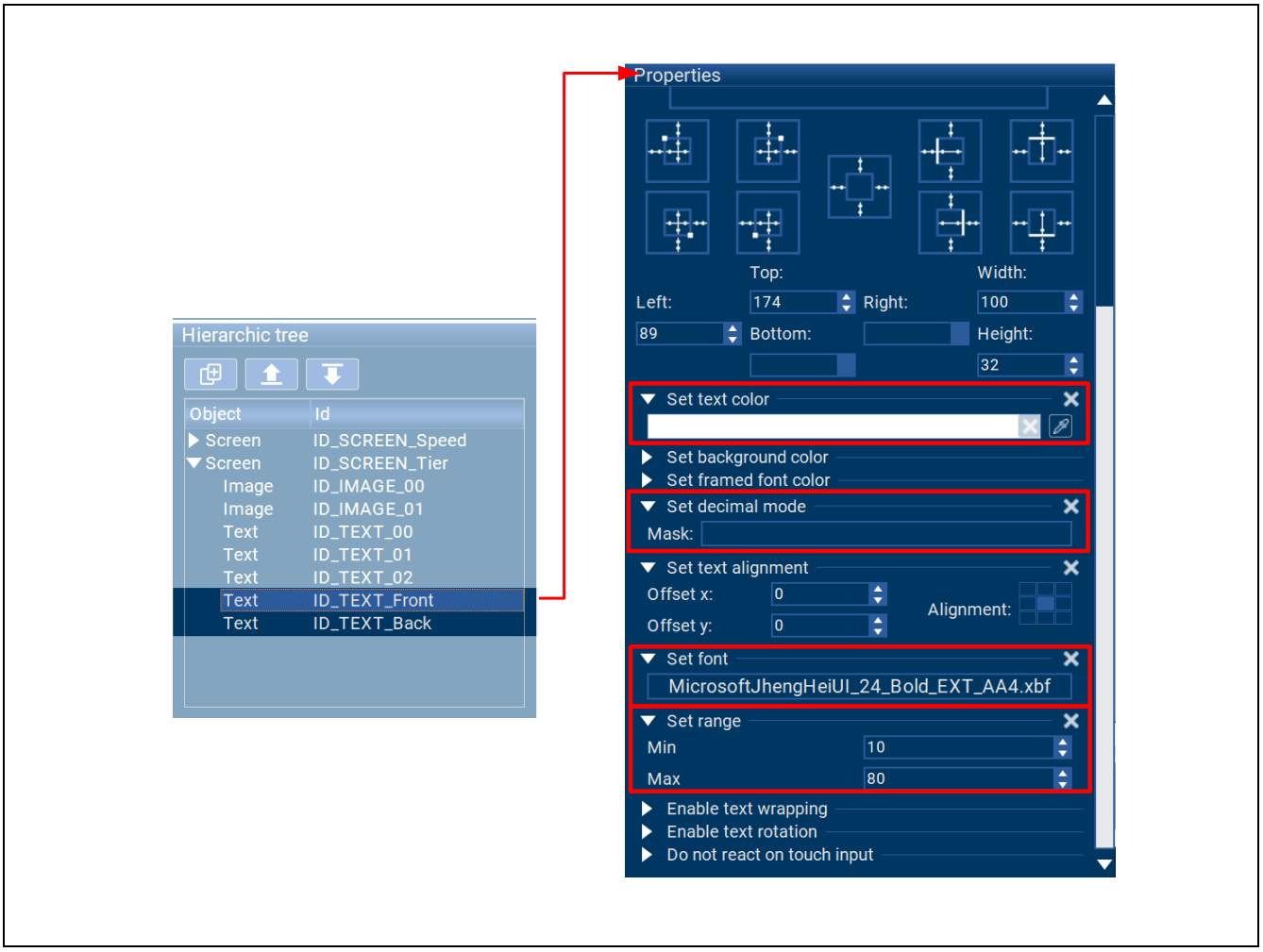


图 1-48 TEXT 属性设定

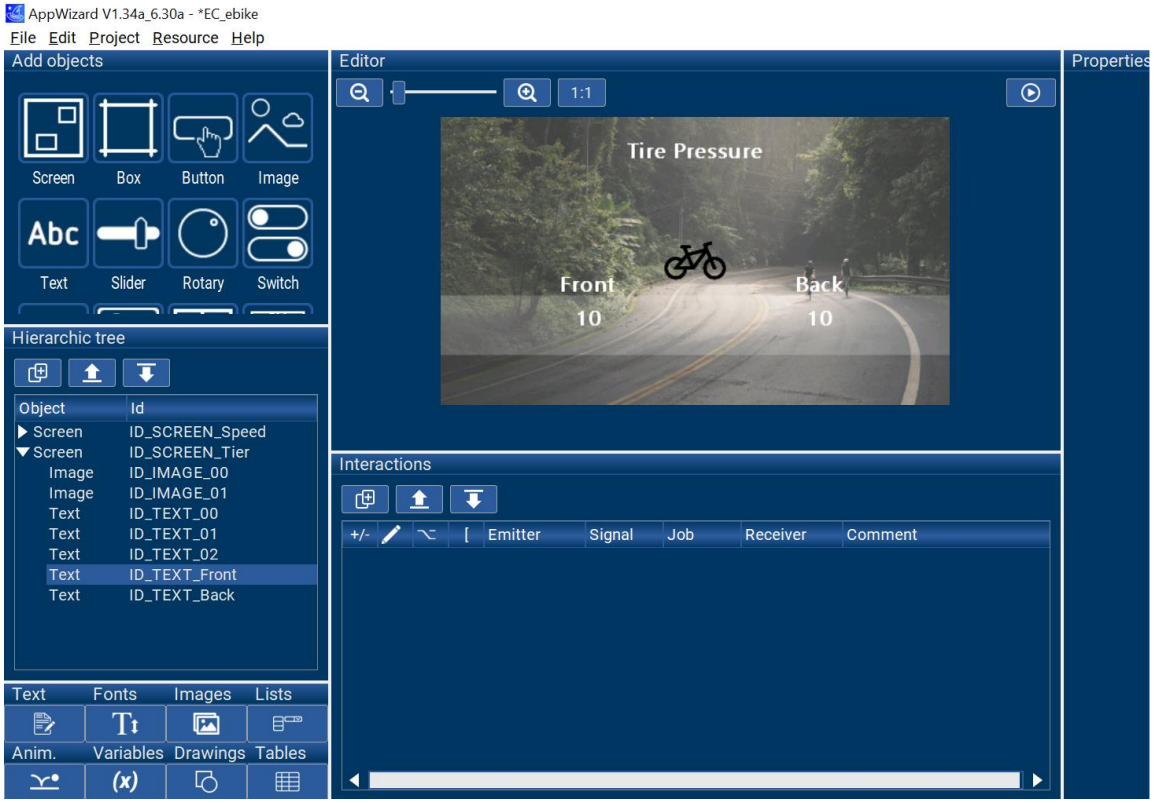


图 1-49 以 TEXT 完成胎压数值显示

28. 依照步骤 16，新增”返回前页”之按钮，完成后如图 1-50 所示。

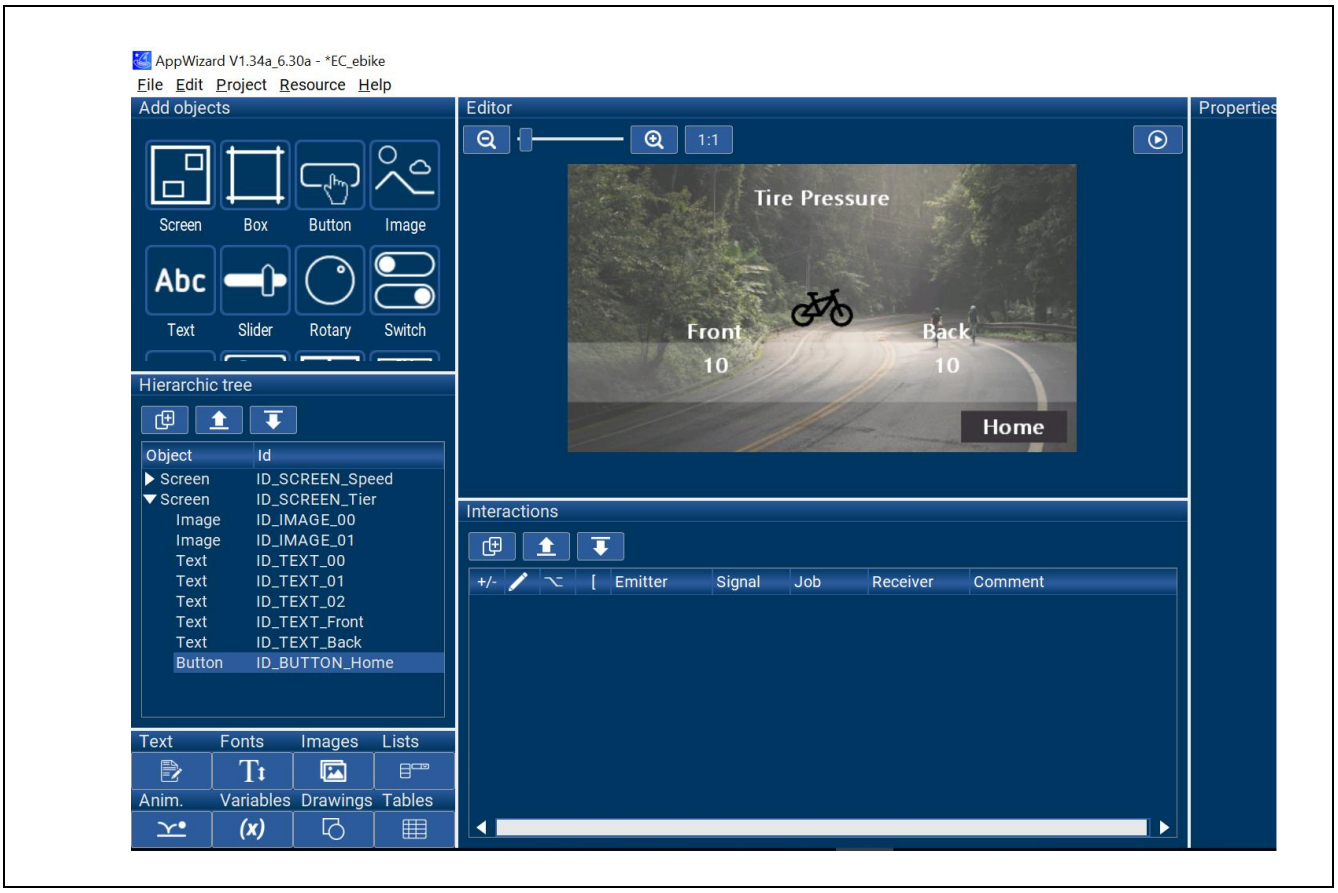


图 1-50 以 Button 新增返回前页按钮

29. 依照步骤 16，新增语言切换”中文”“EN”之按钮，完成后如图 1-51 所示。

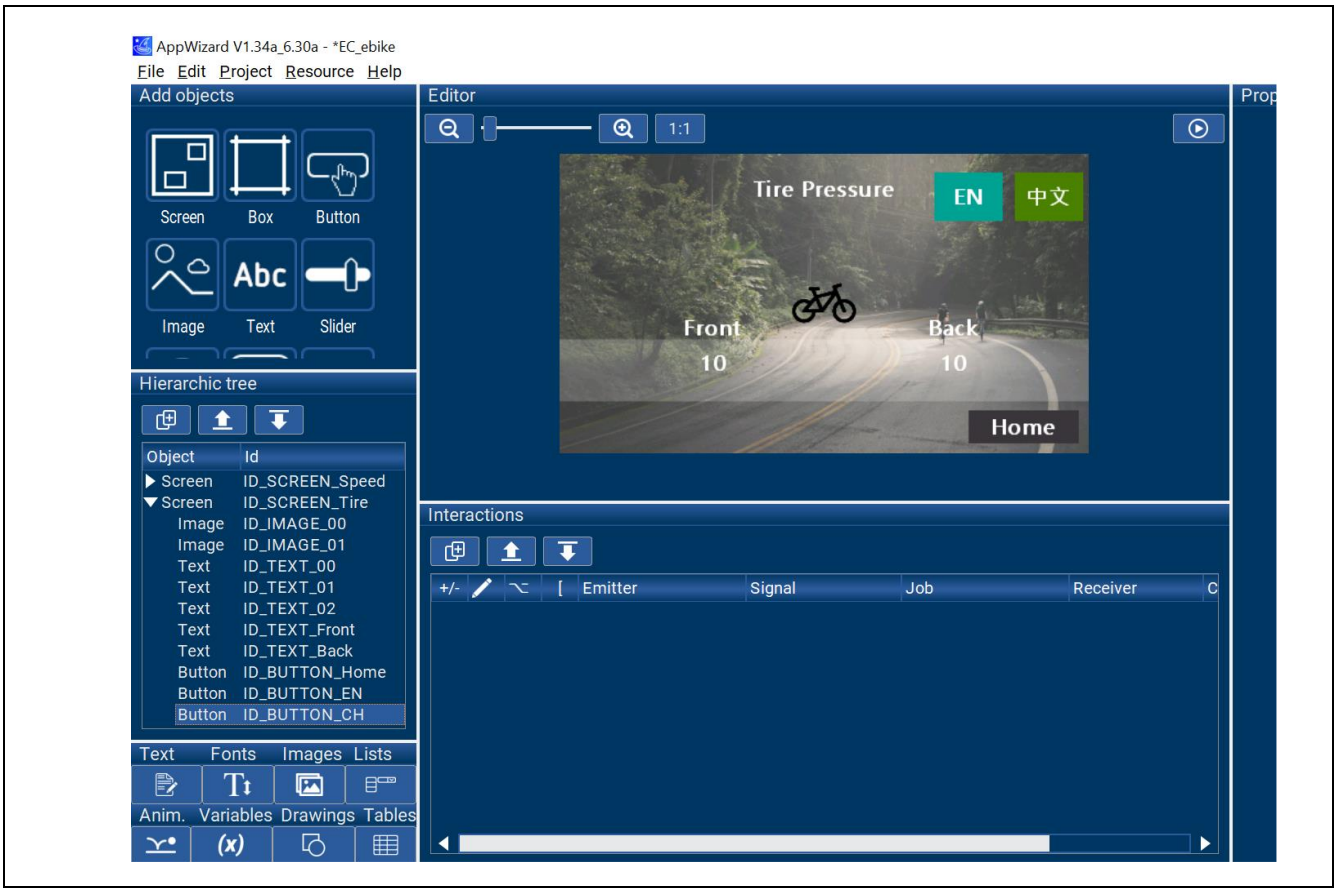


图 1-51 以 Button 新增语言切换按钮

30. 开始设定互动(Interactions)功能：

1) 使用按钮做为页面返回与切换，点选 ID_SCREEN_Tire 底下的 ID_BUTTON_Home，于 Interactions 字段设定如图 1-52 所示，即设定完成页面 Tire 切换至页面 Speed；由页面 Speed 按钮切换至页面 Tire 亦然，如图 1-53。

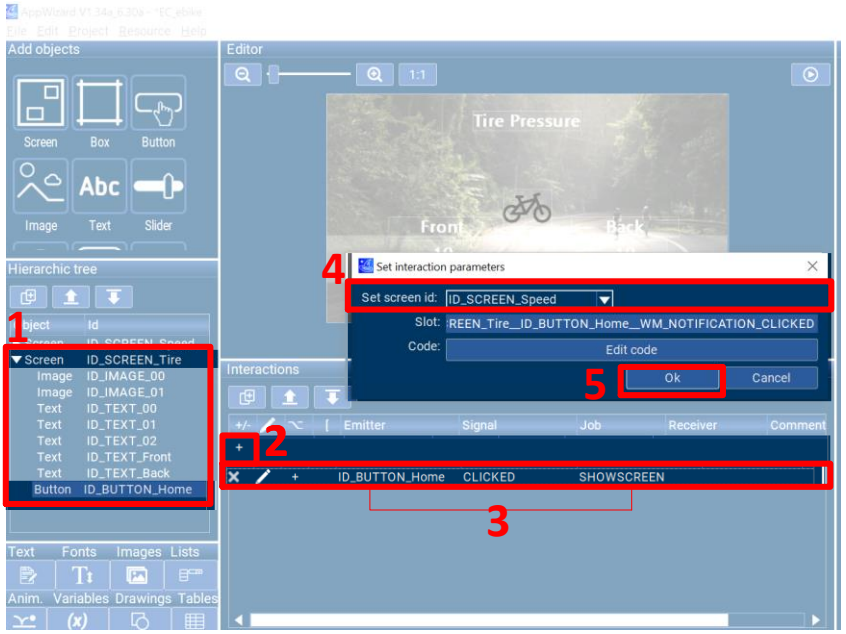


图 1-52 Interaction 设定-页面切换

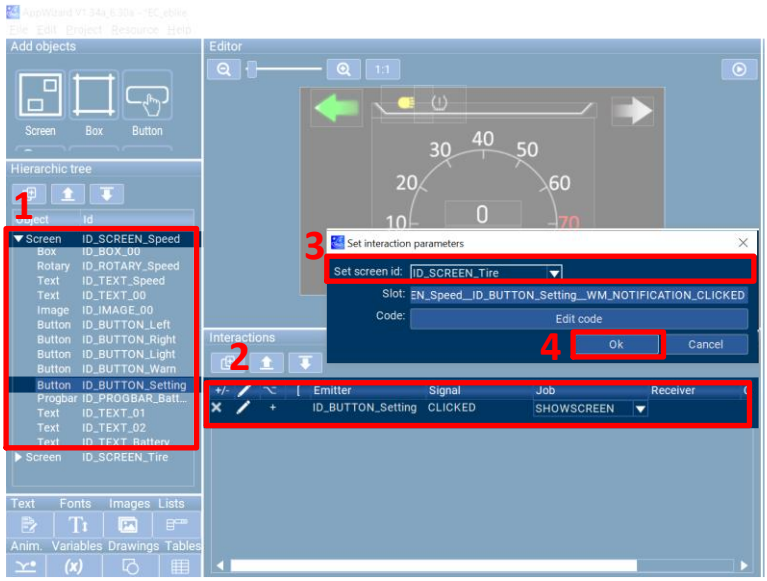


图 1-53 Interaction 设定-页面切换

2) 使用按钮做为语言切换功能，点选 ID_SCREEN_Tire 底下的 ID_BUTTON_EN 设定步骤如图 1-54 所示，即设定完成按下”EN”按钮显示英文功能；显示中文功能亦然，设定步骤如图 1-55 所示。

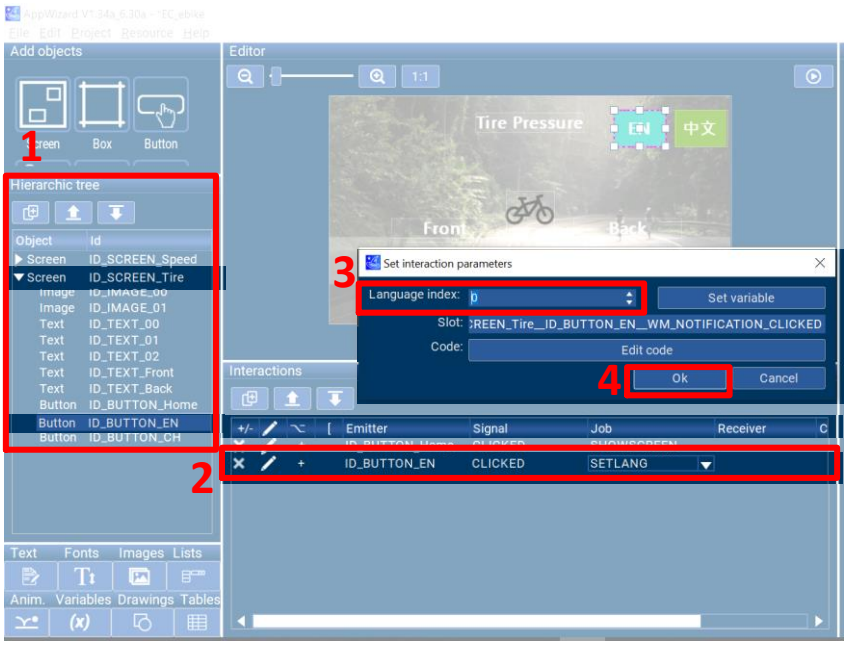


图 1-54 Interaction 设定-语言切换

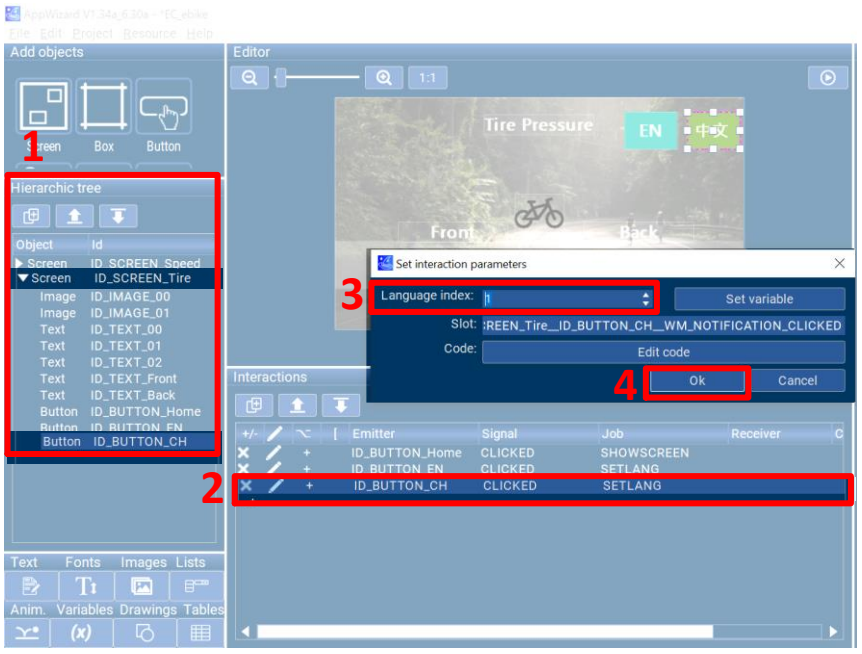


图 1-55 Interaction 设定-语言切换

3) 为了从 MCU 计算完后设定数值给电量、前后轮的胎压，需要使用 Variable 功能，并于程序代码中赋予 Variable 值，再使用 Interactions 功能将 variable 值赋予给欲显示设值的 TEXT，流程如图 1-56 所示，实际步骤请依照图 1-57~图 1-59 操作，程序代码中步骤请参考 2 代码介绍。

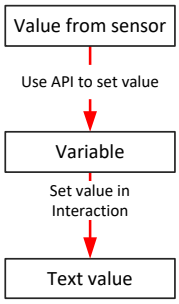


图 1-56 显示 MCU 计算的数值于画面上的流程

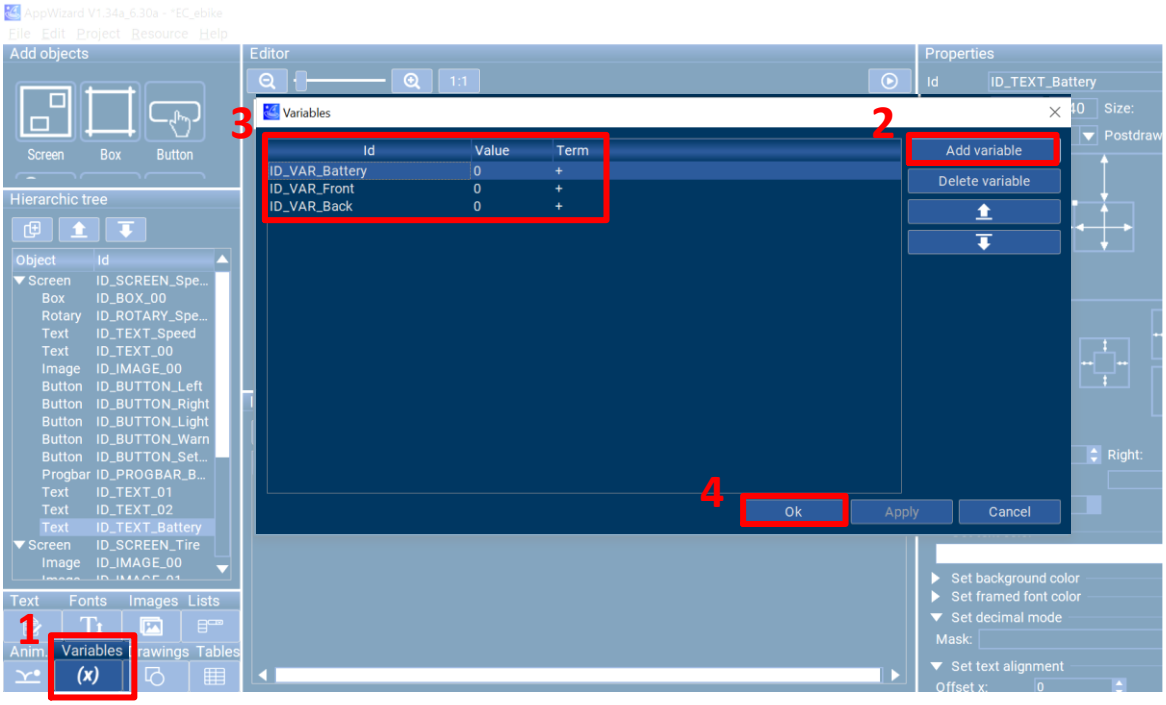


图 1-57 Variable 设定

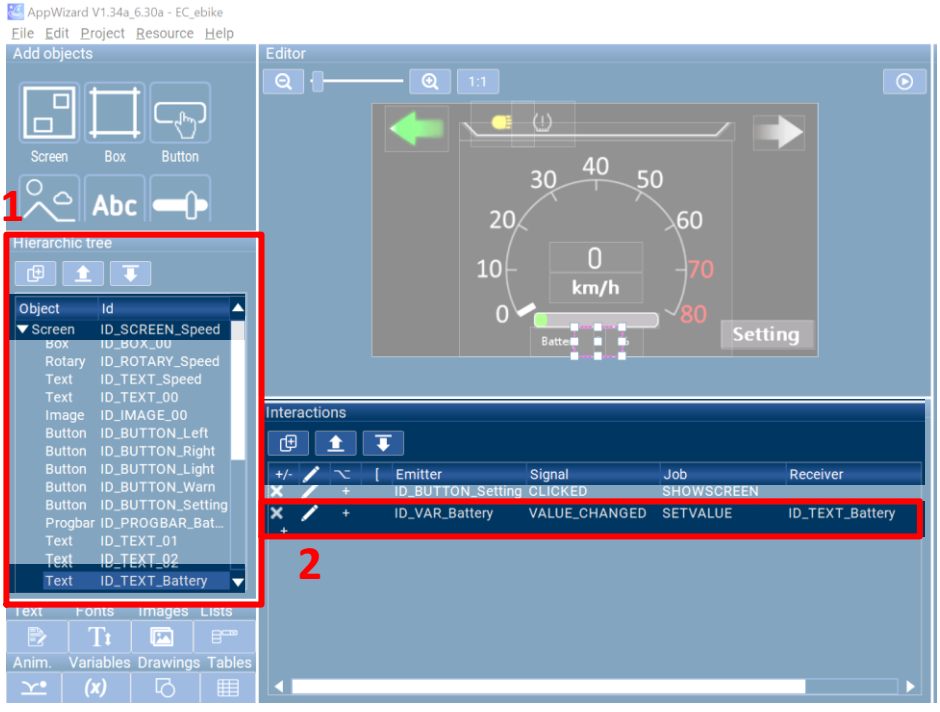


图 1-58 Interaction 设定 - Variable

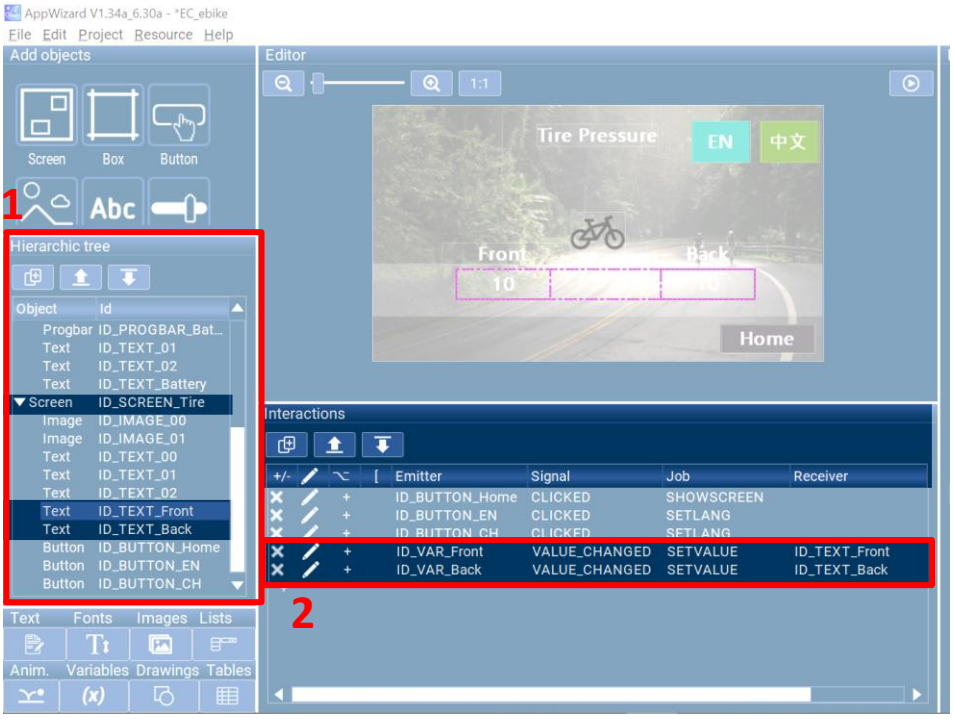


图 1-59 Interaction 设定 - Variable

4) 使用动画 animation 功能，模拟速度仪表变化，请依照图 1-60~图 1-63，设定速度数值自动变化，接着依照图 1-64 设定速度仪表随着速度数值变化而转动。

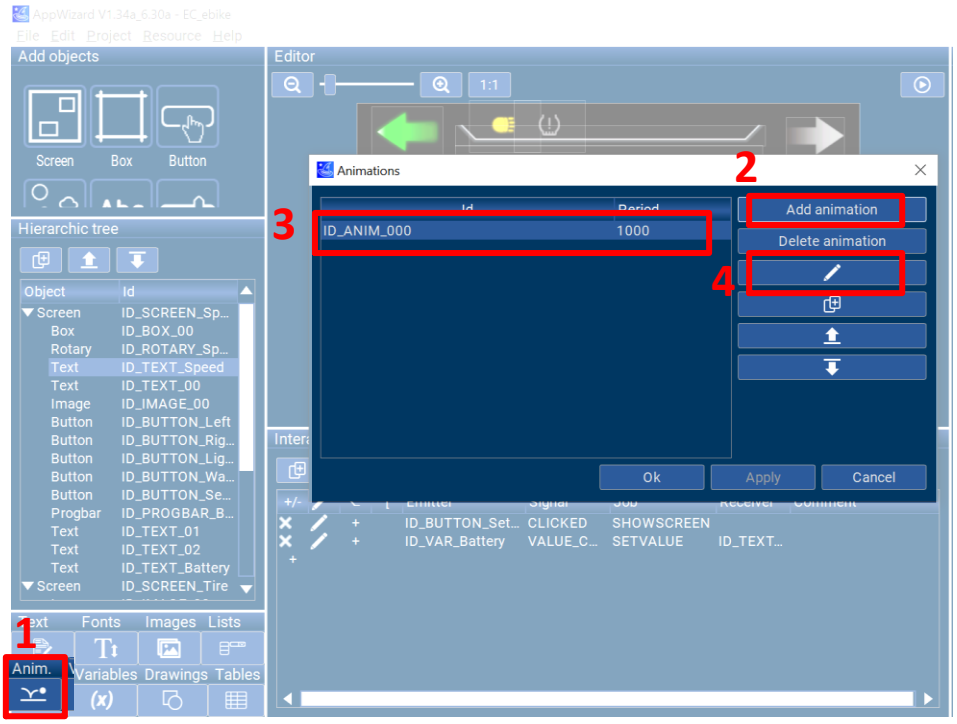


图 1-60 Animation 设定

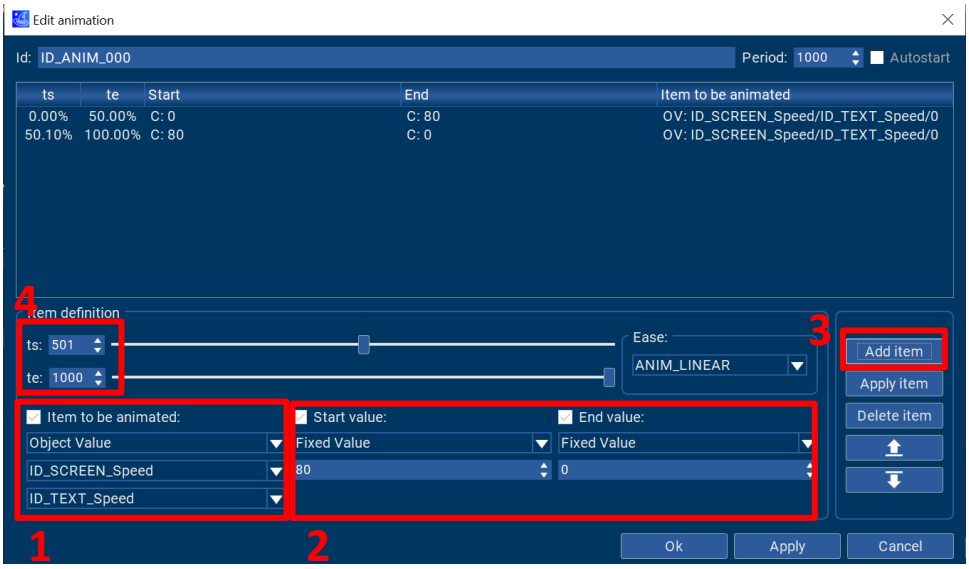


图 1-61 Animation 设定

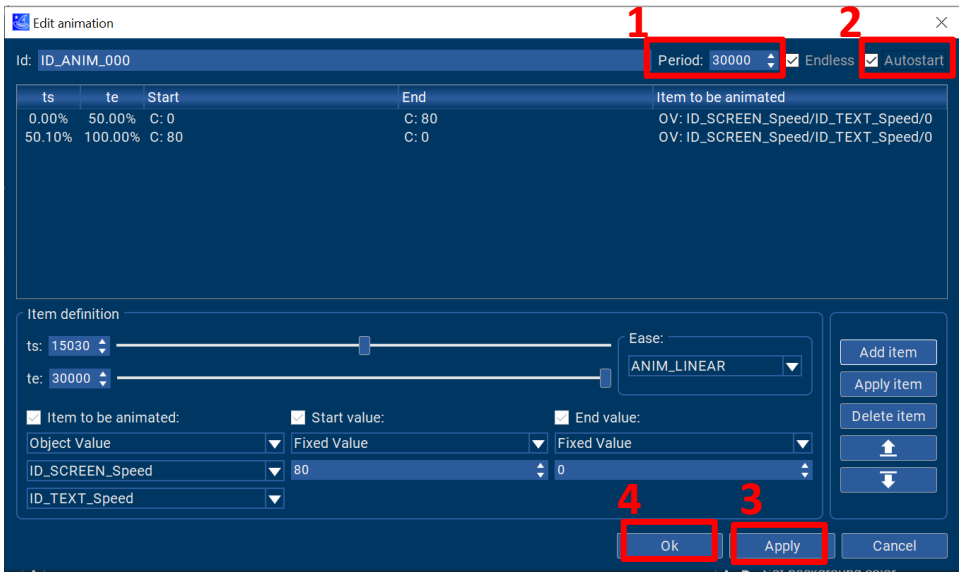


图 1-62 Animation 设定

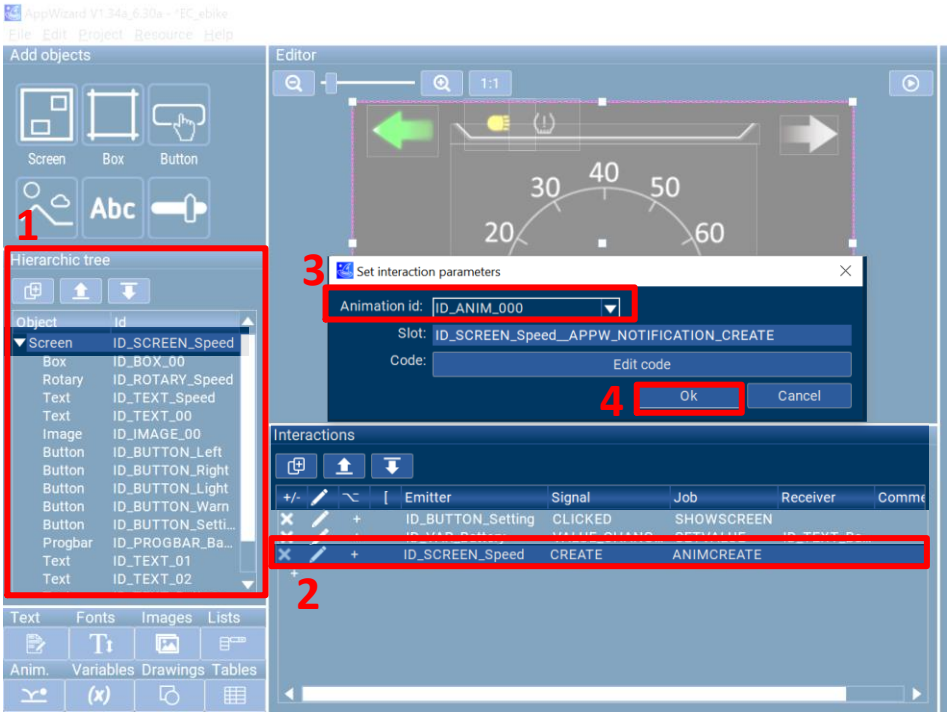


图 1-63 Interaction 设定-开启 Animation

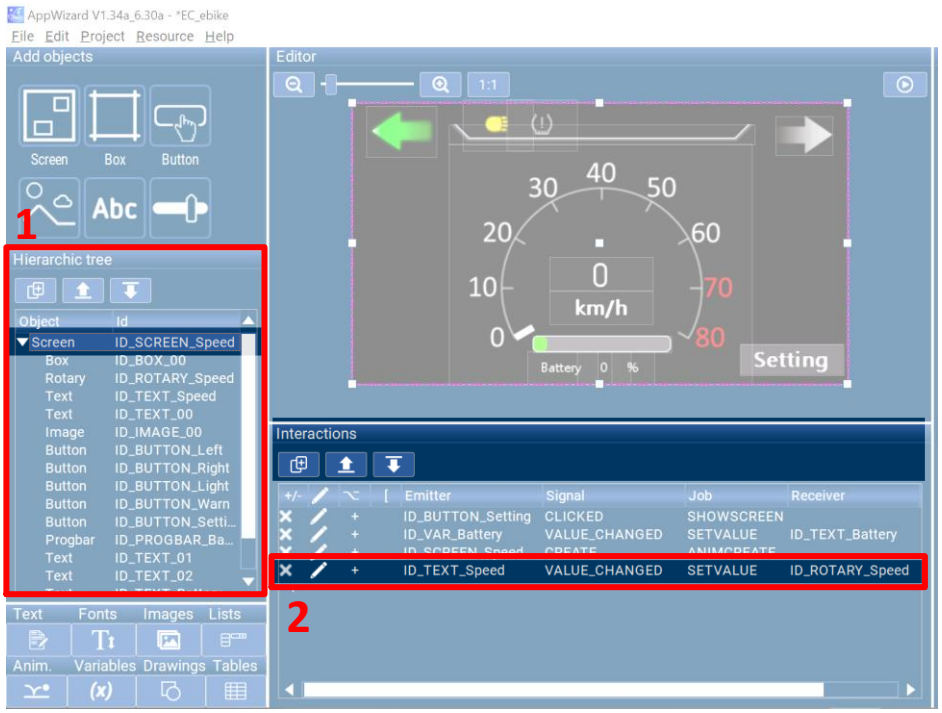


图 1-64 Interaction 设定 – 动作连动

5) 设定电量数值与电量显示图连动，设定方式如图 1-65。

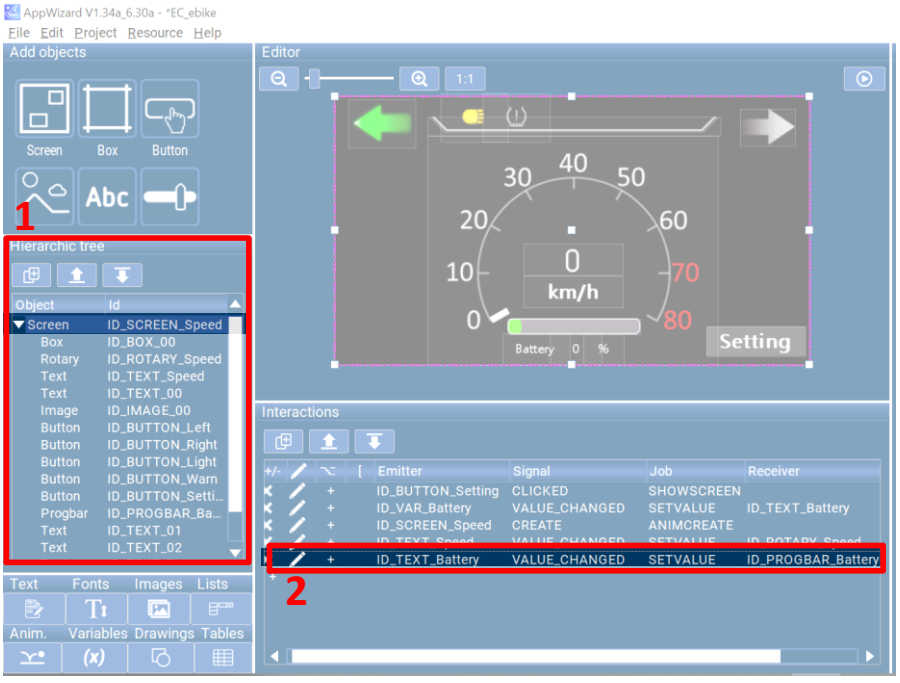


图 1-65 Interaction 设定 – 动作连动

31. 设定以滑动页面切换画面，先于 Hierarchic tree 点选欲切换画面的 Screen ID，再于 Properties 中设定水平或垂直动作，操作步骤如图 1-66。

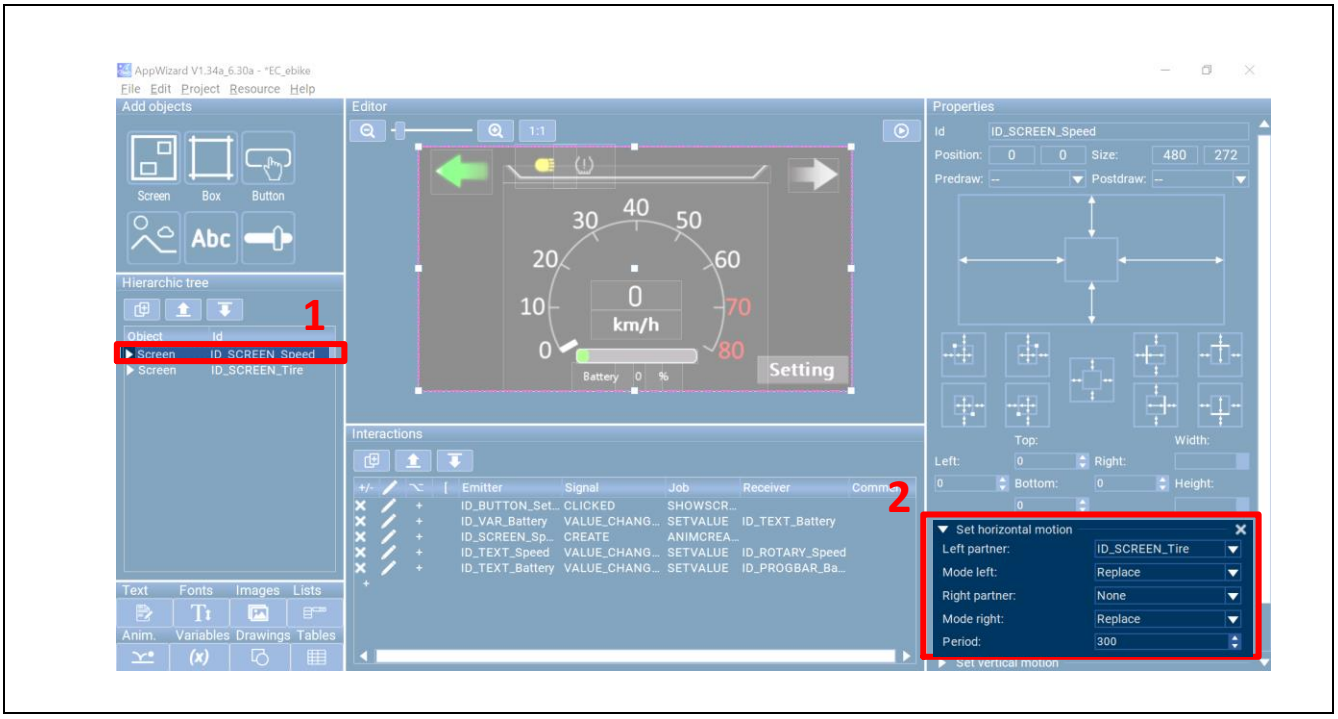


图 1-66 Screen 属性设定

1.2.3 程序代码编辑

1. 设计完成即可于 File-> Export & Save 输出程序代码，如图 1-67。此范例所使用之项目文件名称为 EC_ebike.zip，档案路径可参考图 4-1，在解压缩后，可以使用 AppWizard 直接开启该项目档。

图 1-67 输出程序代码

2. 于 Project -> Open containing folder 开启 AppWizard 专案文件夹，复制 Resource 与 Source 文件夹，于路径:

   EC_M467_AppWizard_Ebike_V1.00\SampleCode\ExampleCode\M467_AppWizard_Ebike\Application 中覆盖原本档案，步骤如图 1-68~图 1-69 所示。

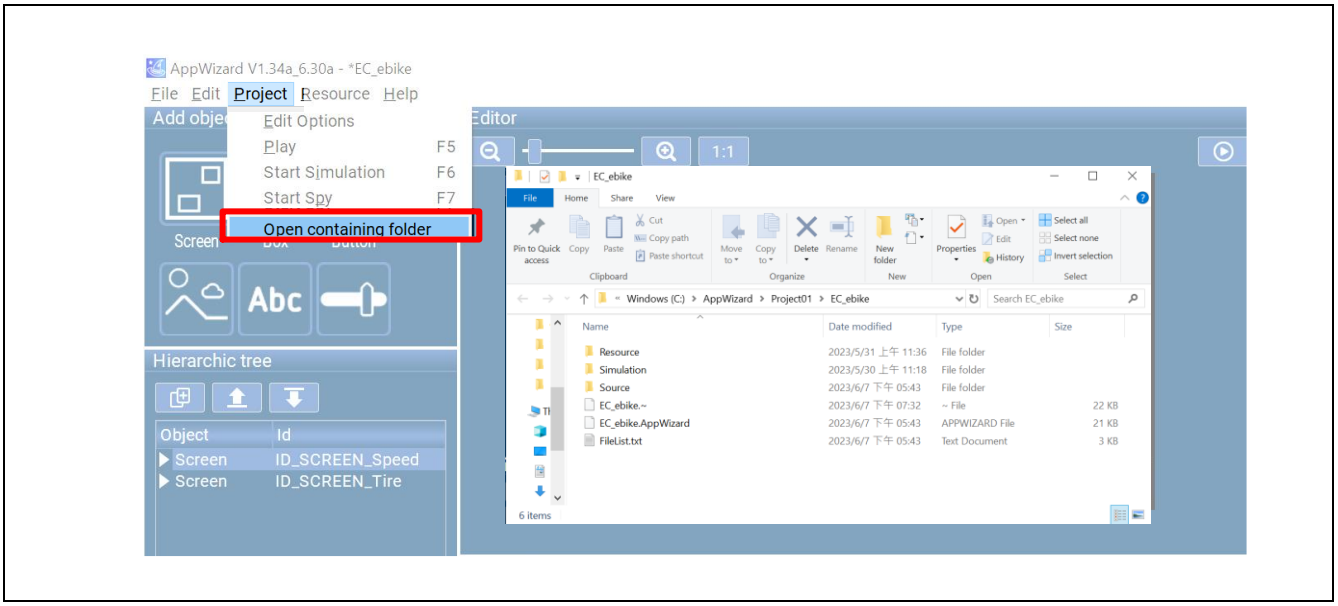


图 1-68 开启专案文件夹

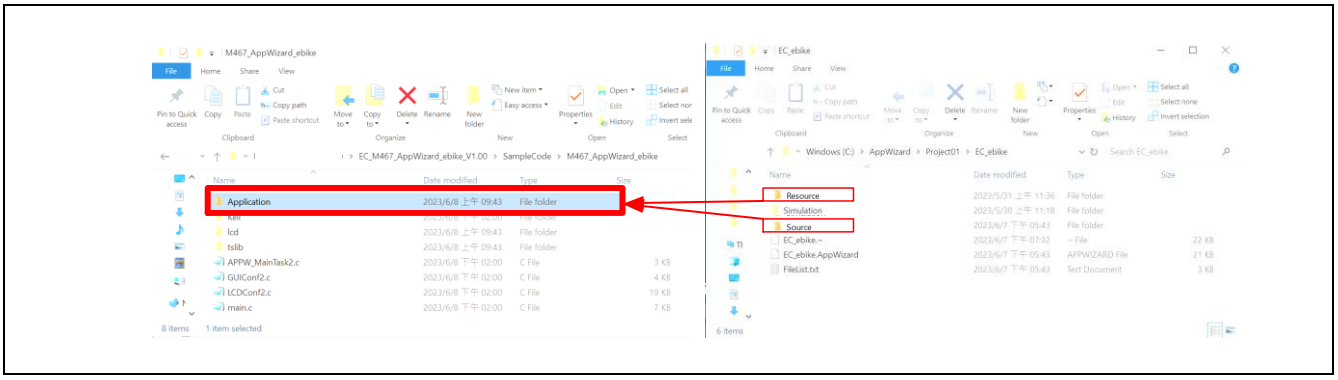


图 1-69 移植至程序代码项目中

主程序代码请参考 2 代码介绍。

1.2.4 烧录运行在装置中

实际运行在装置后，仍可对整体执行效能做调整，相关优化设定如下所述

- 于 AppWizard 使用的图片 format，需设定为符合屏幕格式，此范例设定为 “High color with alpha (565)”。

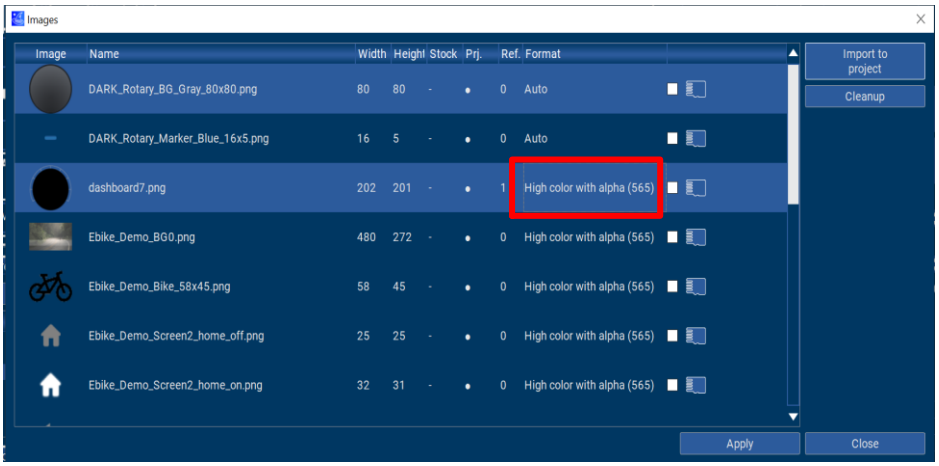


图 1-70 Image 设定

1.3 执行结果

本范例代码执行编译和烧录的结果如图 1-71 所示。

图 1-71 执行结果

2. 代码介绍

主程序由 *main.c* 进入，包括 3 个部分如下：

- 系统时钟与引脚初始化：sys_init
- 界面初始化与开启：UART、SPIM、Timer、Touch 等
- 应用主程序：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 接口，是为了将占较大空间的图片、文本文件资源存至外部 SPI Flash，利用 SPI Synchronous Serial Interface Controller (SPI Master Mode)，在 Direct Memory Mapping Mode(DMM)的模式下，SPI Flash 可被视为 ROM Module，直接到指定地址读取 SPI Flash，并在内建的 cache 帮忙下，以及使用四线模式(quad read)，提升整体读取效率。

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

```
void SPIM_init(void){
    uint8_t     idBuf[3];
    SYS_UnlockReg();              /* Unlock protected registers */
    SPIM_SET_CLOCK_DIVIDER(2);    /* Set SPIM clock as HCLK divided by 2, 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，亦须编辑 Scatter file，以让编译程序知道 SPI Flash 的位置与内容。于 KEIL 环境中，依照路径 Options for Target-> Linker->Scatter File-> Edit，开启.sct 档案并编辑内容如下。

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

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

;SPI flash
LOAD_ROM_2  0x100000
{
    SPIM.bin  0x100000
    {
     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)
    }
}
```

程序进入应用主程序 MainTask2()后，将执行 AppWizard 相关函数，在 while 循环中会执行画图的动作，此范例程序同时新增 APPW_SetVarData()函数，将值从 MCU 传出给 AppWizard 中的 Variable，再透过如步骤 3)所述之方法，数值便会呈现在画面上。

```
void MainTask2(void)
{
    // Setup configuration dependent pointers
    APPW_X_Setup();
    // Initialize AppWizard
    APPW_Init(APPW_PROJECT_PATH);
    // Create 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 执行的画图内容，会从 ID_Screen_xxx_Slots.c 中读取，可以试着在步骤 16 按钮按下时的程序代码中加入 printf("Button Click\n")，在实际 Speed 画面，若按下 Setting 按钮跳至 Tire 页面，UART 会于 PC 印出”Button Click”内容。

```
/*
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. 软件与硬件需求

3.1 软件需求

- 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 硬件需求

- 电路组件
  - NuMaker-HMI-M467
- 线路示意图
  - 如图 3-1，将 Nu-Link2-Me 与子板之屏幕 USB connector 以 2 条 USB 线与 PC 相连。

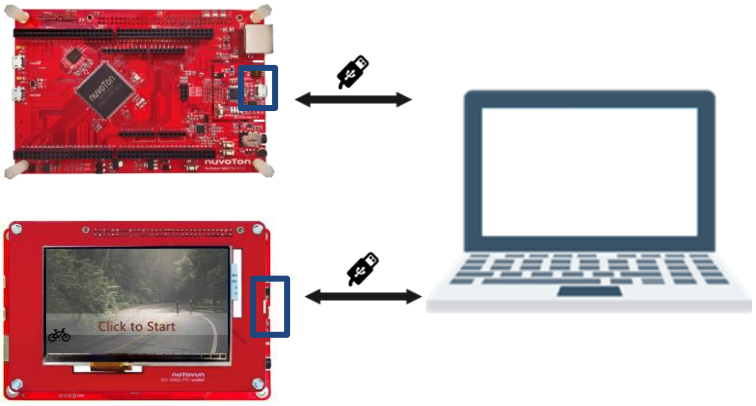


图 3-1 以 USB 线将 NuMaker-HMI-M467 与 PC 相连

4. 目录信息

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.

图 4-1 目录资讯

5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *M467_AppWizard_Ebike.uvprojx*。
2. 进入编译模式接口
   - 编译
   - 下载代码至内存
   - 进入 / 离开除错模式
3. 进入除错模式接口
   - 执行代码

6. 修订纪录

Date	Revision	Description
2023.05.31	1.00	初始发布。

Important Notice

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

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

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

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