

## MPLAB® Harmony 3 之基础篇（07）

### — 如何使用 MHGC 设计图形界面

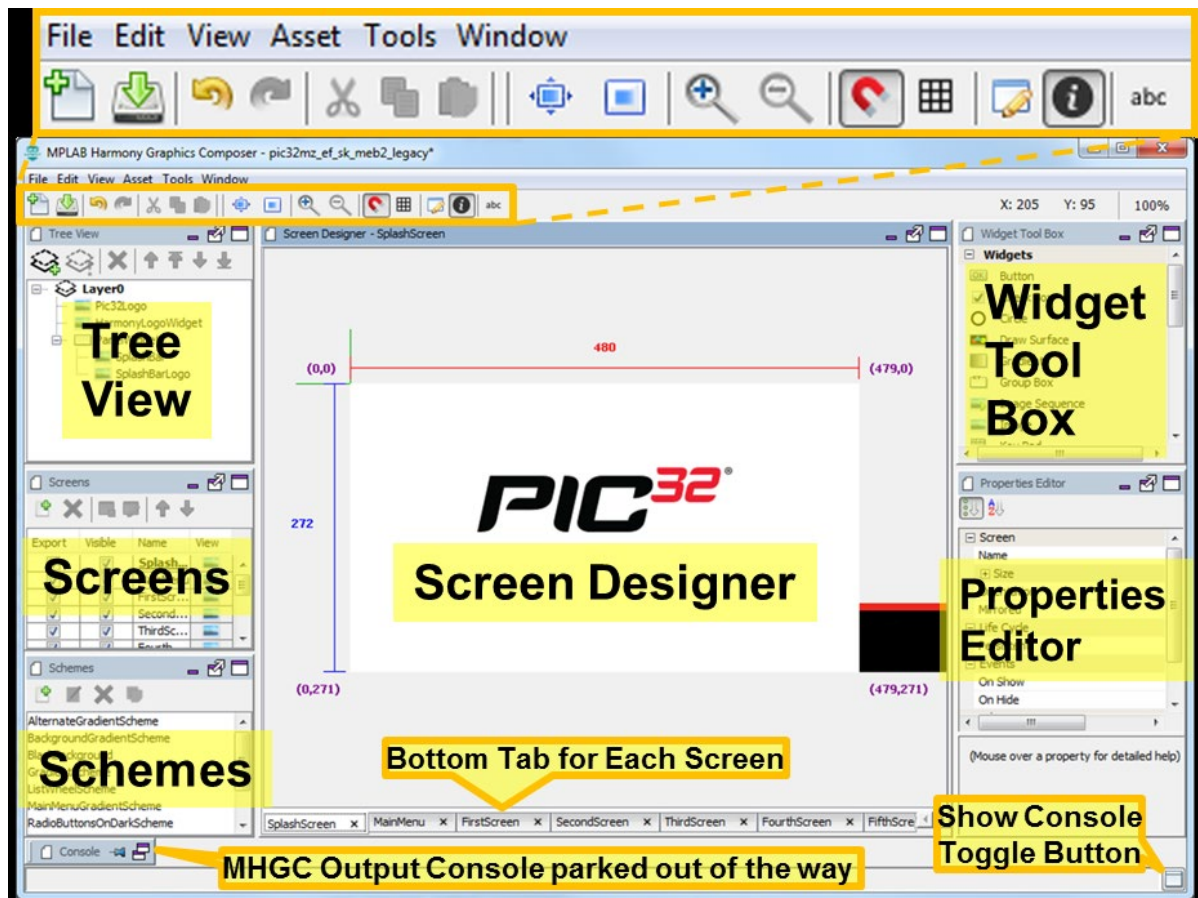
Microchip Technology Inc.  
MCU32 产品部

#### 一、 简介

在上一篇文章中，介绍了利用 Harmony 3 创建图形工程的完整流程，本文章将介绍如何使用 MHGC(Microchip Harmony Graphic Composer)设计图形界面，包含创建页面，控件使用，设置背景和图片颜色，和创建按钮的事件响应。

#### 二、 MHGC 介绍

MHGC 主界面如下图。



主要窗口如下：

Screens: 包含当前工程所有的页面。

**Tree View:** 包含当前页面下的所有的层。

**Schemes:** 包含当前工程所有的可以使用风格。

**Widget Tool Box:** 包含了所有可以使用的控件。

**Properties Editor:** 允许用户调整页面，层和显示对象的参数。

在 MHGC 中主要的图元（图形元件）如下：

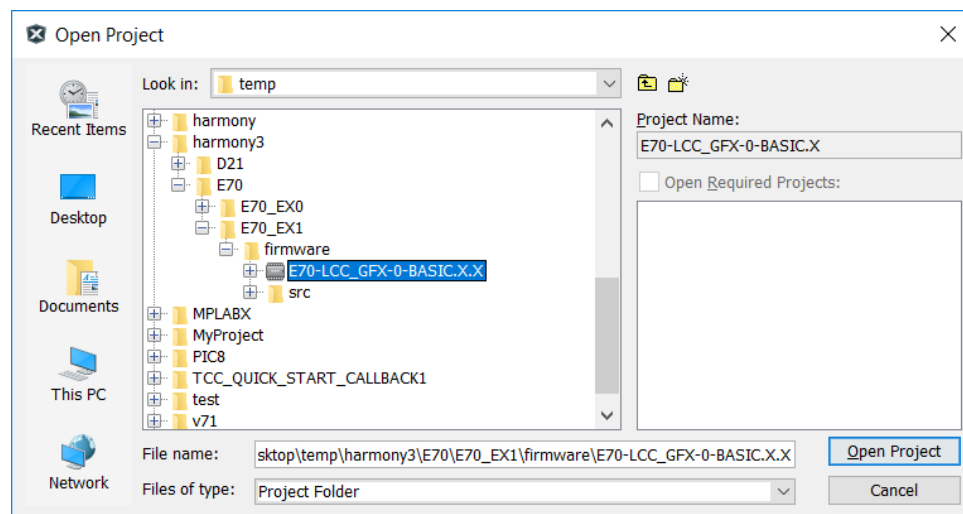
- **Screen/页面** – 每个页面包含了 1 至 3 个层,层与层之间可以做 alpha 混合, E70 LCC 方案只支持 1 个层。
- **Layer/层** – 层相当于容器, 上可以放置各种不同的控件。
- **Widget/控件** – 如按钮, 文本框等基础图元。
- **String/字符串** – 字符的集合, 比如按钮的文本等。
- **Scheme/风格** – 颜色的集合, 比如按钮的风格包含文本颜色, 背景颜色等。

### 三、 准备工作

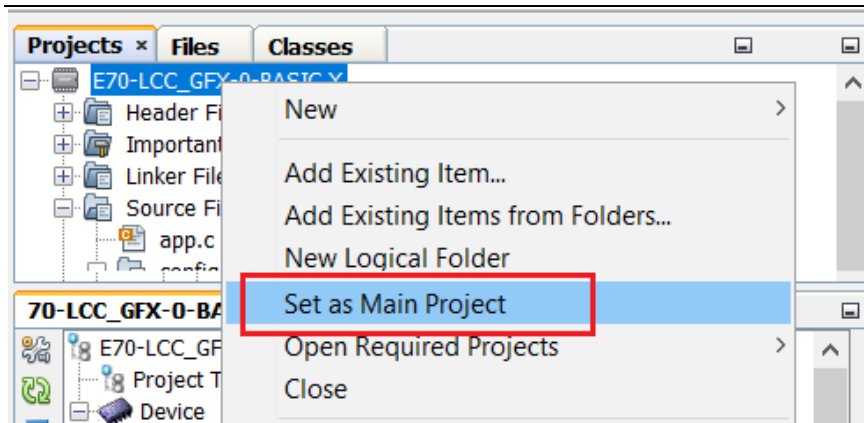
*注: 以下 MHC 配置里没有特别标注出来的地方, 说明使用的是默认选项。*

#### (一) 打开之前的 E70-LCC\_GFX-0-BASIC 项目

点击 MPLAB X IDE 的主菜单 File -> Open Project..., 在弹出的对话框中选择工程 E70-LCC\_GFX-0-BASIC 所在的路径, 再点击 Open Project 按钮。

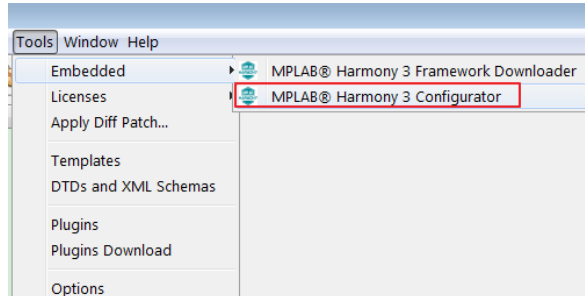


右键工程, 选择 Set as Main Project 来设置主工程。如果已经是主工程则不需要再设置。

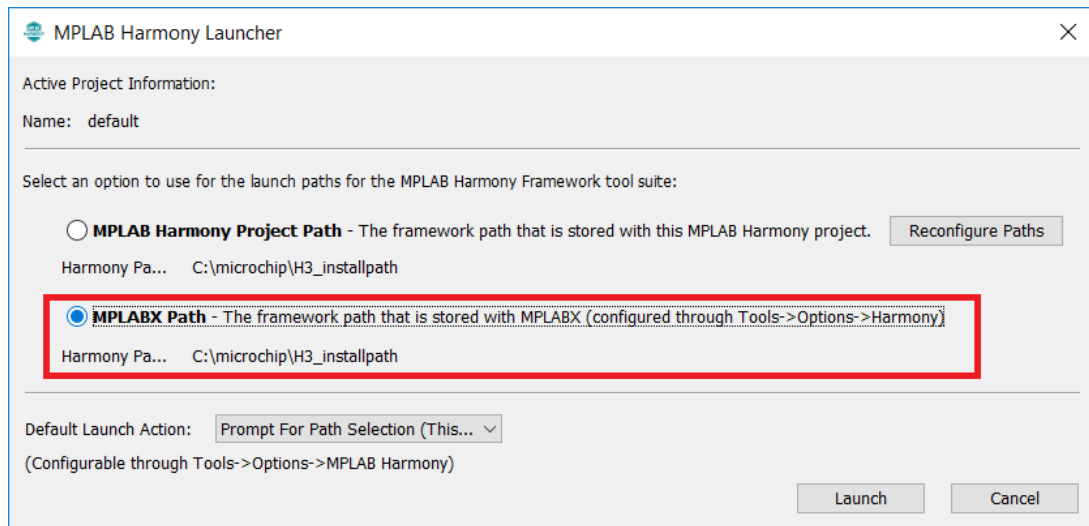


## (二) 启动 MHC

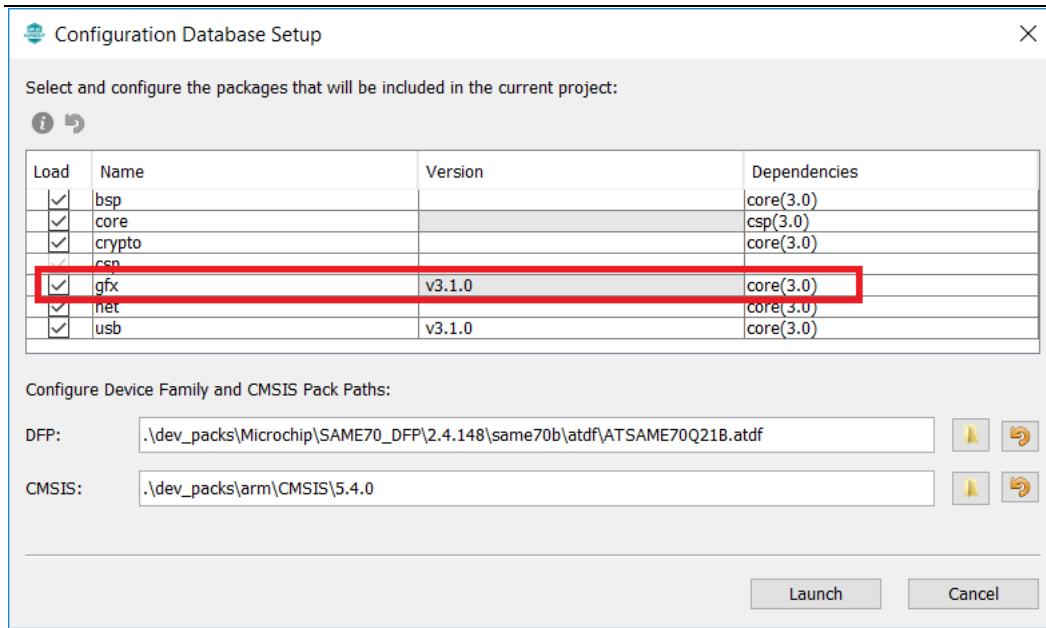
在 MPLAB X IDE 里点击 Tools -> Embedded -> MPLAB Harmony 3 Configurator 来启动 MHC。



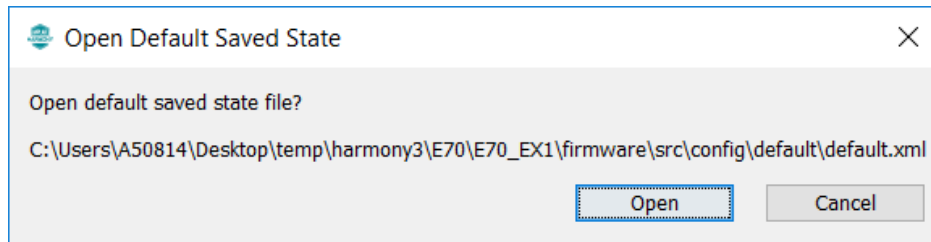
在弹出的对话框中确认本工程用到 MPLAB Harmony Project Path 位于哪个目录，如果工程从其他电脑拷贝而来，应该选择第二个选项，然后点击 launch 按钮。



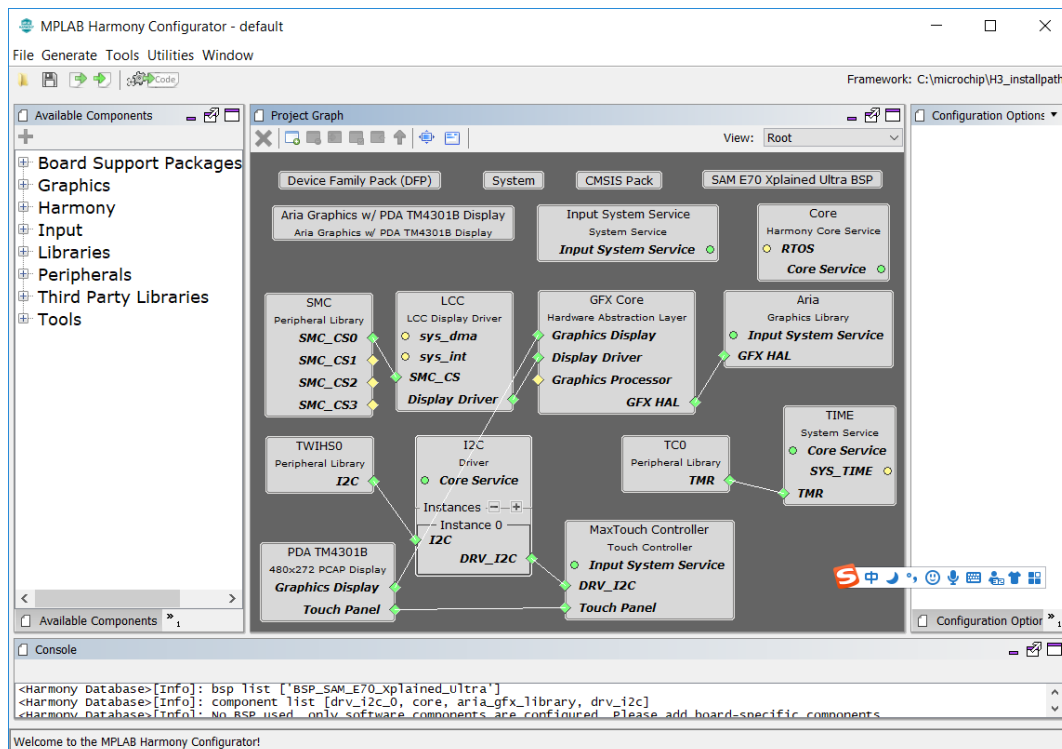
确认 gfx 勾选，点击 launch 按钮。



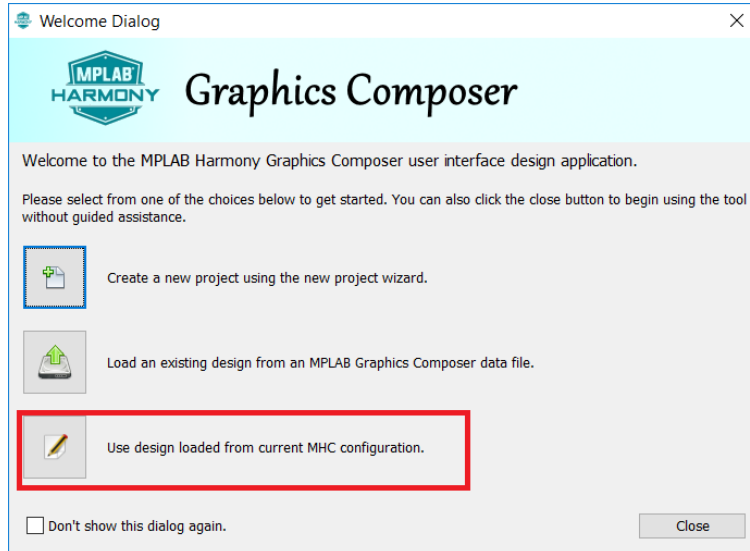
点击 Open 按钮。



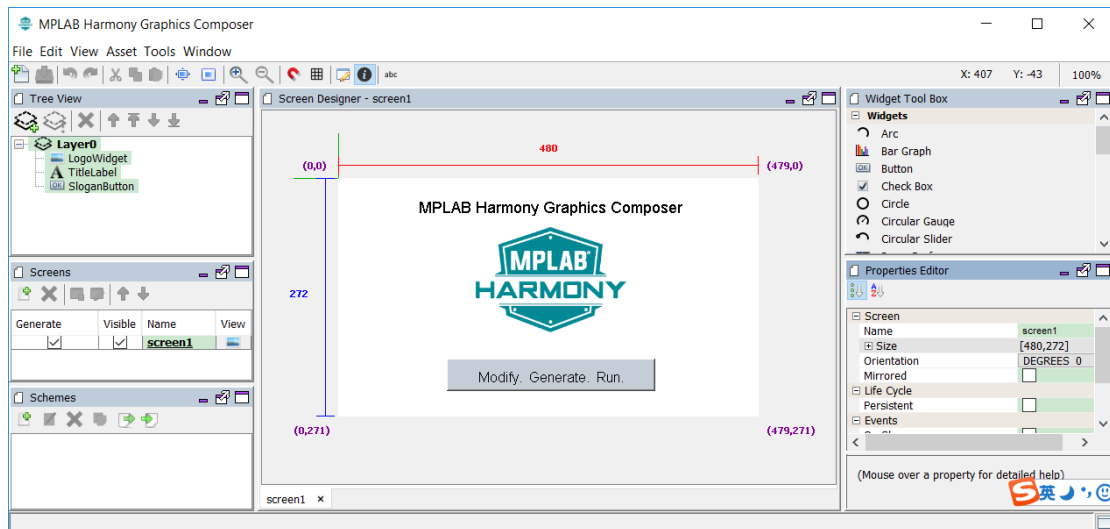
成功打开如下图所示。



点击主菜单 **tools-> Graphic Composer** 来启动 MHGC。该工程已经创建过 MHGC 项目了,在弹出的 **Welcome Dialog** 中选择第三个选项来加载已有的 MHGC 项目。



MHGC 加载项目后如下图。



#### 四、 设计图形界面详细步骤

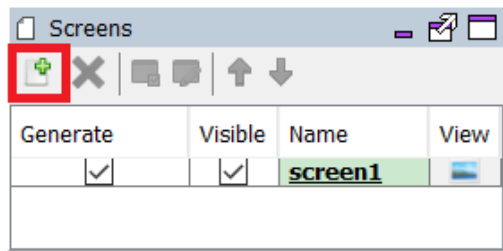
本图形界面实现的功能, 以及相应的学习内容如下:

- 两个页面, 一个页面名为 **screen1**, 另外一个页面名为 **screen2**。其中 **screen1** 为主页面 (**primary screen**)。程序运行时默认显示的页面就是主页面。  
学习内容: 页面的创建和设置, 以及页面和层的关系。
- 主页面带有一个按钮, 通过点击按钮实现页面切换。  
学习内容: 控件的创建和设置, 以及层和空间的关系。
- 页面 **screen1** 带有一个 **Image** 控件, 用于显示图片。  
学习内容: 创建图片资源和创建 **Image** 控件。

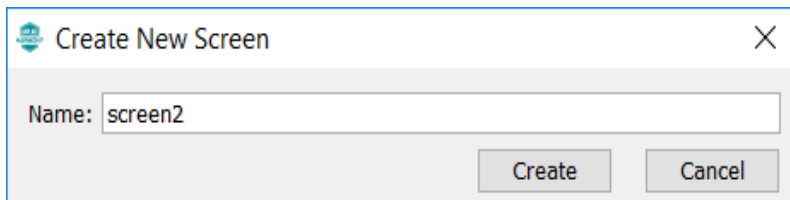
- 页面和按钮设置不同的背景颜色和字体颜色。  
学习内容：图元的属性设置，以及风格的使用。

### (一) 创建页面和设置背景颜色

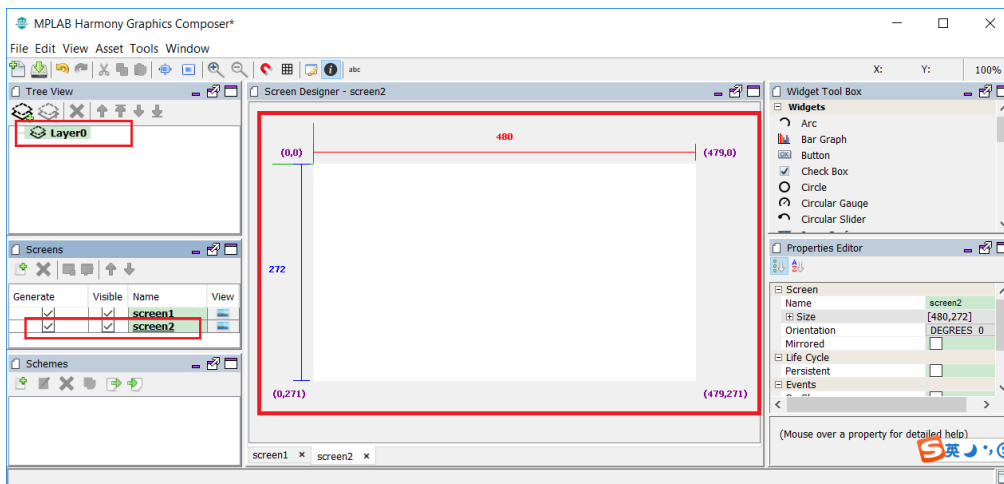
在 screens 窗口中，点击 Create New Screen 按钮来创建页面。



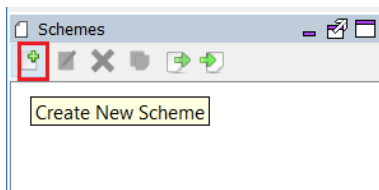
在弹出的对话框中可以设置 screen 的名称，点击“Create”按钮。



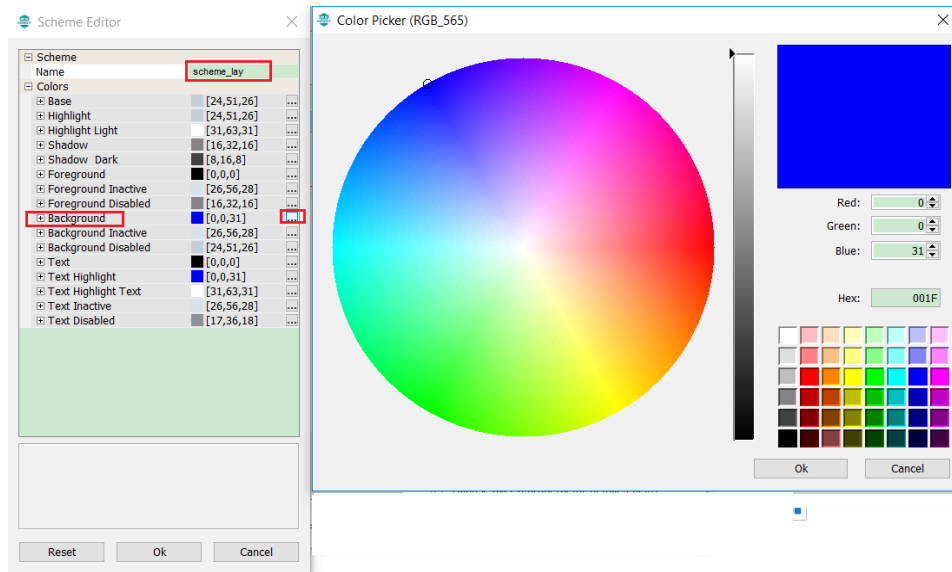
页面 screen2 创建后，MHGC 自动切换到该页面，相应的 Tree View 和 Screen Designer 窗口也切换到该页面，在该页面下，默认创建了一个名为 Layer0 的层。



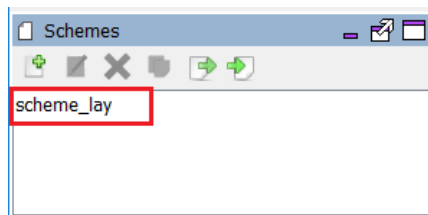
在 Schemes 窗口，点击“Create New Scheme”按钮来创建一个风格。



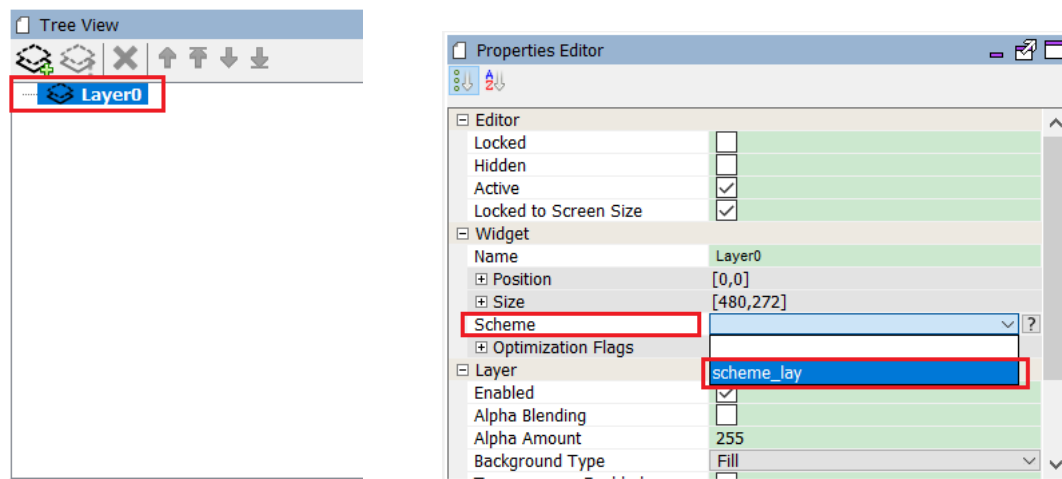
设置名称和背景颜色，之后点击 OK，完成创建风格。



风格创建完成如下图。

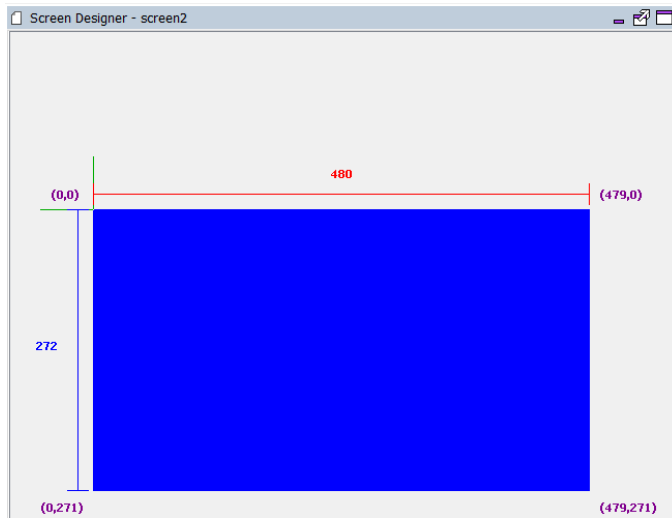


在 Tree view 中选中 Layer0，Layer0 的属性信息会在 Properties Editor 中显示出来，修改 Scheme 为之前创建的风格 scheme\_lay。



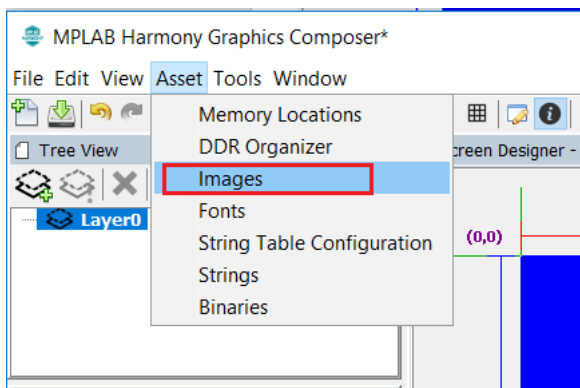
Screen designer 中显示页面的背景如下。



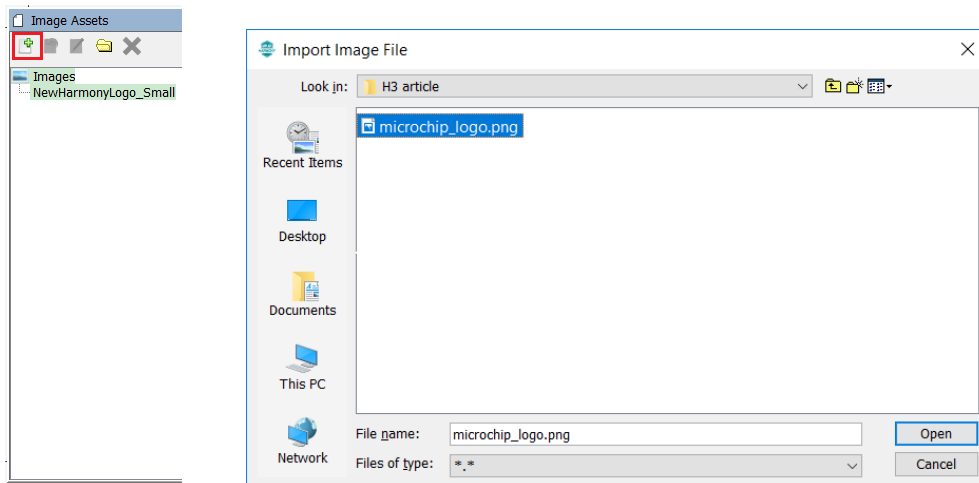


## (二) 添加 Image 控件

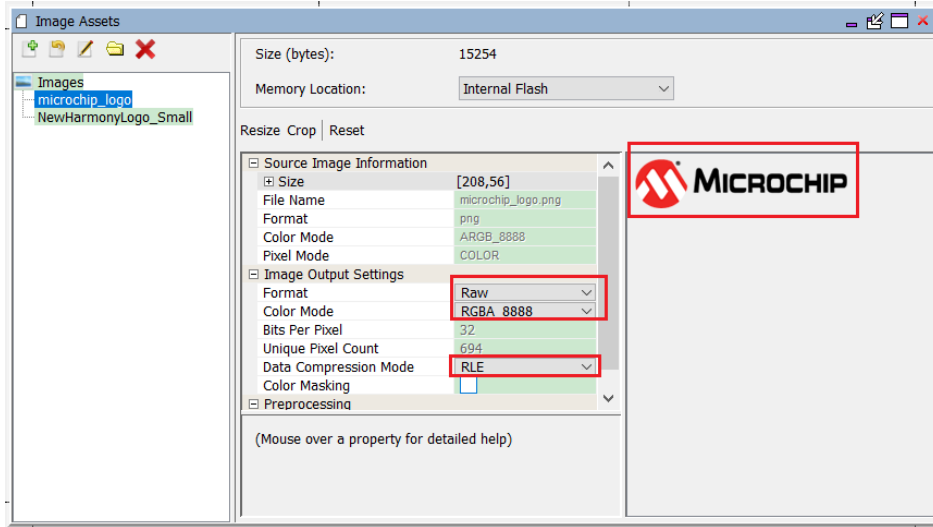
首先添加图片资源 (Image Asset)，选择 MHGC 主菜单->Asset -> Image 来启动 Image Assets。



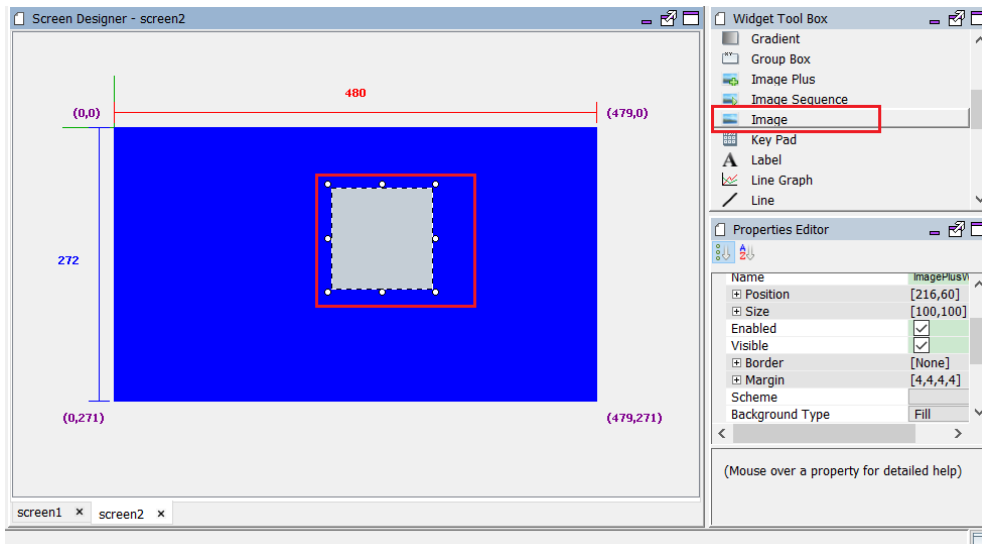
点击 Add Image From File 按钮来导入图片，在弹出的对话框中选择图片文件。



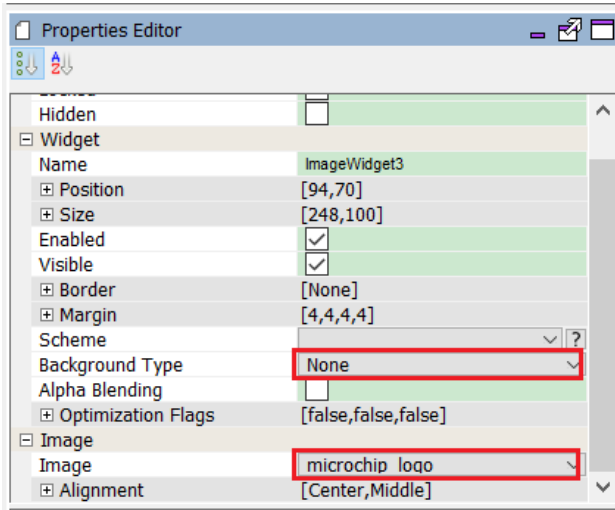
导入成功后，对图片做如下设置，完成后关闭 Image Assets 窗口。



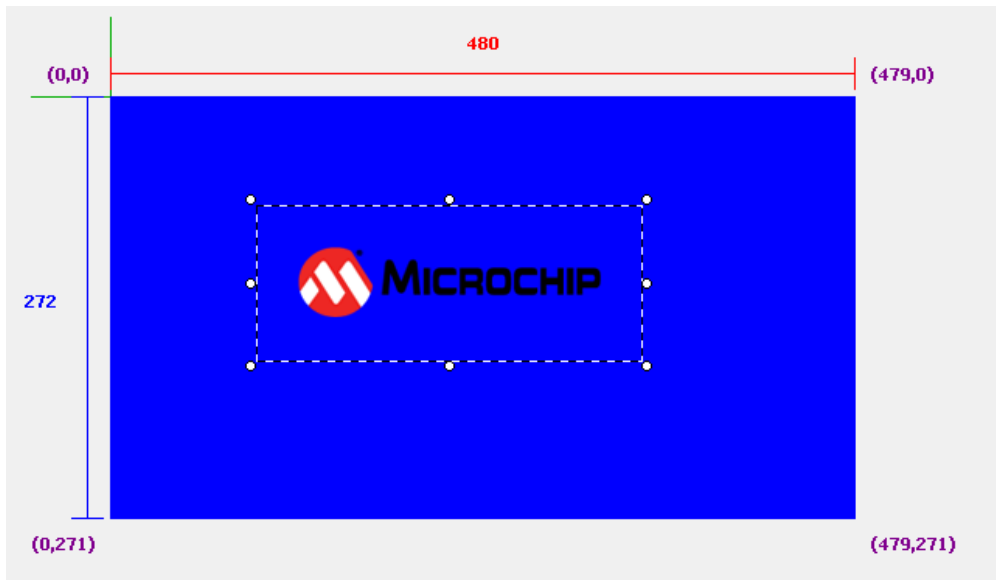
将 image 控件从 Widget Tool Box 拖放到 Screen Designer 窗口。



设置图片控件的属性如下图。

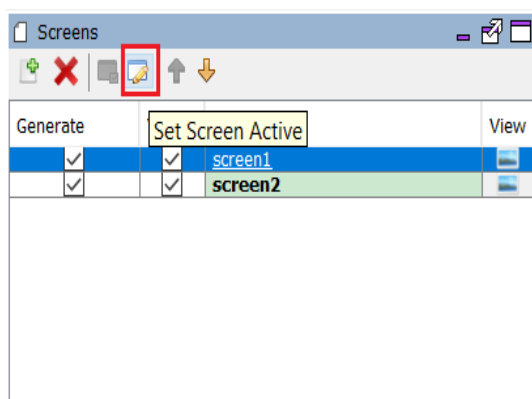


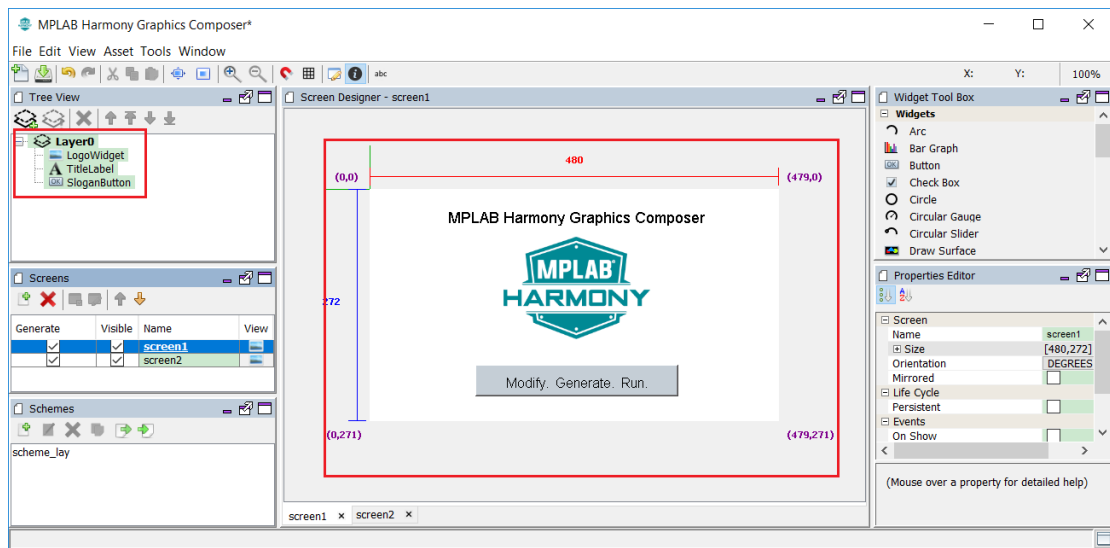
设置完成后，图片控件显示如下。



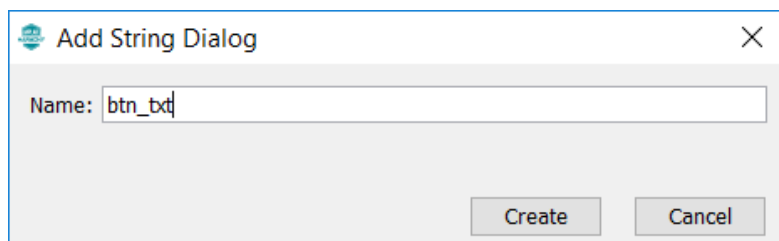
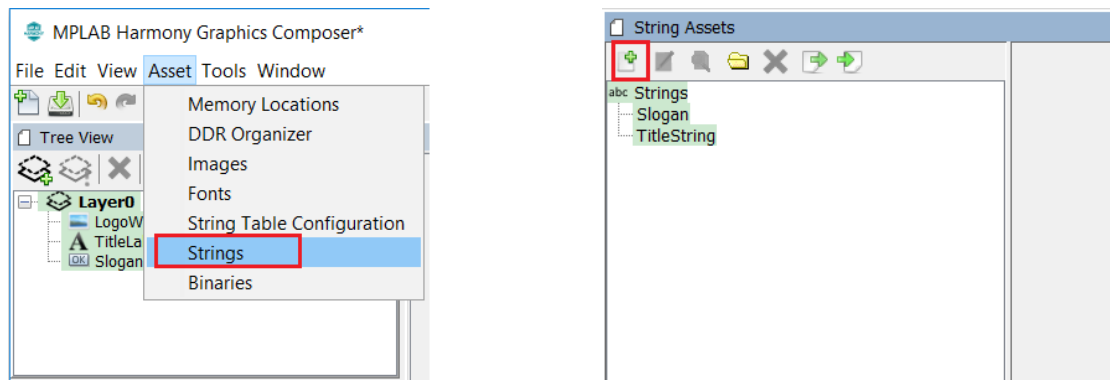
### (三) 创建按钮和相应按钮消息

在 Screens 窗口中，选择 screen1，然后点击 Set Screen Active，这样就切换到了 screen1 的相关界面。

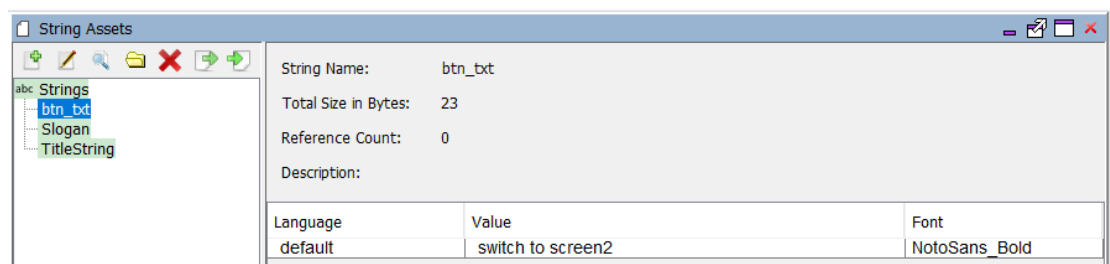




创建按钮上的文字，点击主菜单的 Asset->Strings，在弹出 String Assets 对话框中点击 Add New String 按钮，设置 Name 为 btn\_txt，点击 Create 按钮。



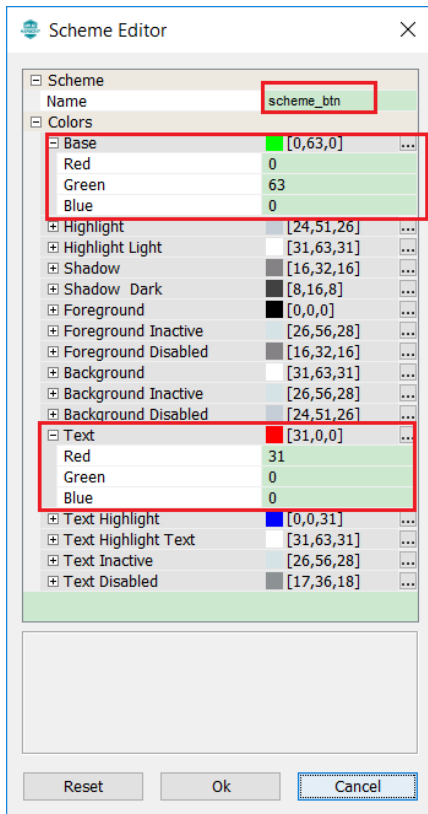
设置 value 为"switch to screen2"，选择 Font 为 NotoSans\_Bold。完成后关闭。



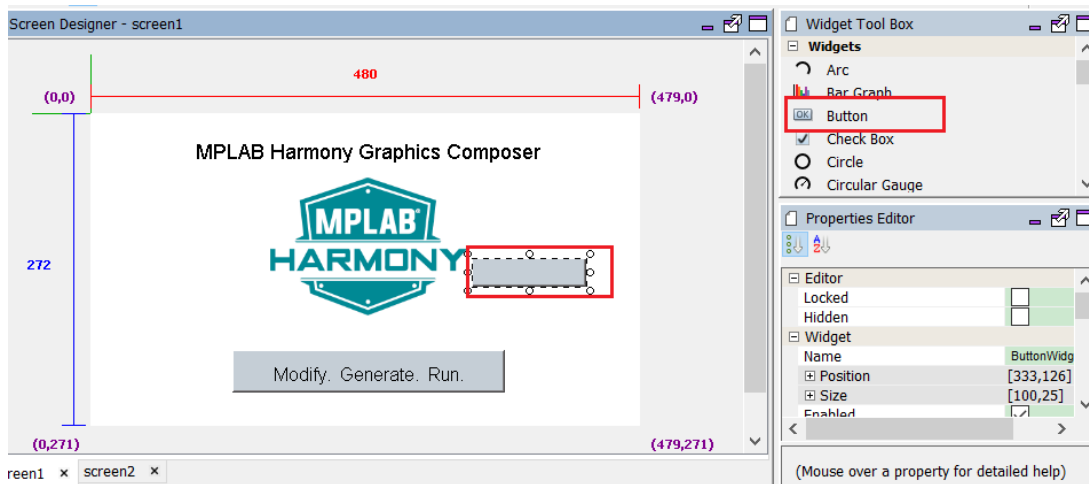
创建按钮使用的风格，在 scheme 窗口中点击 Create New Scheme 按钮。



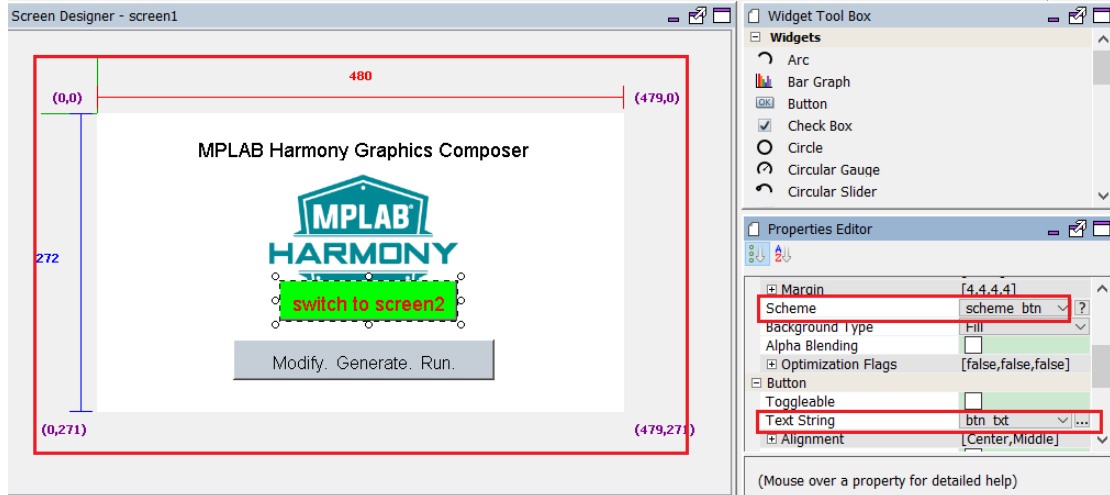
风格设置如下图，点击“OK”完成。



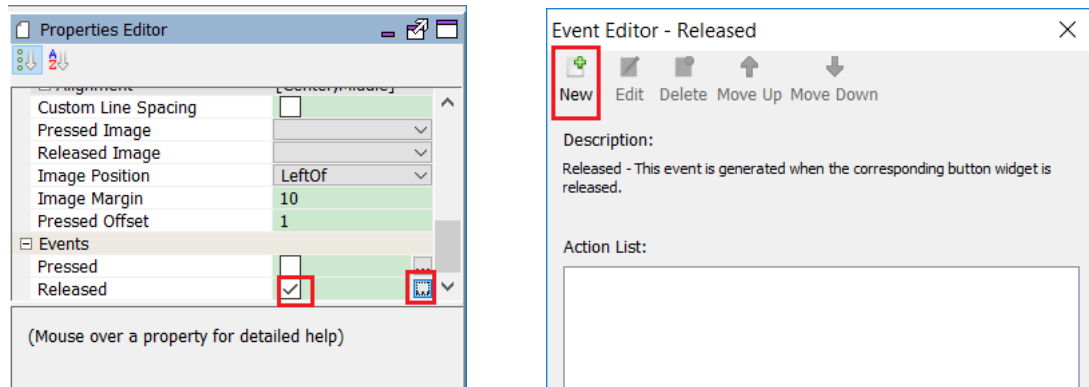
创建按钮，从 widget Tool Box 拖放 Button 控件到 Screen Designer。



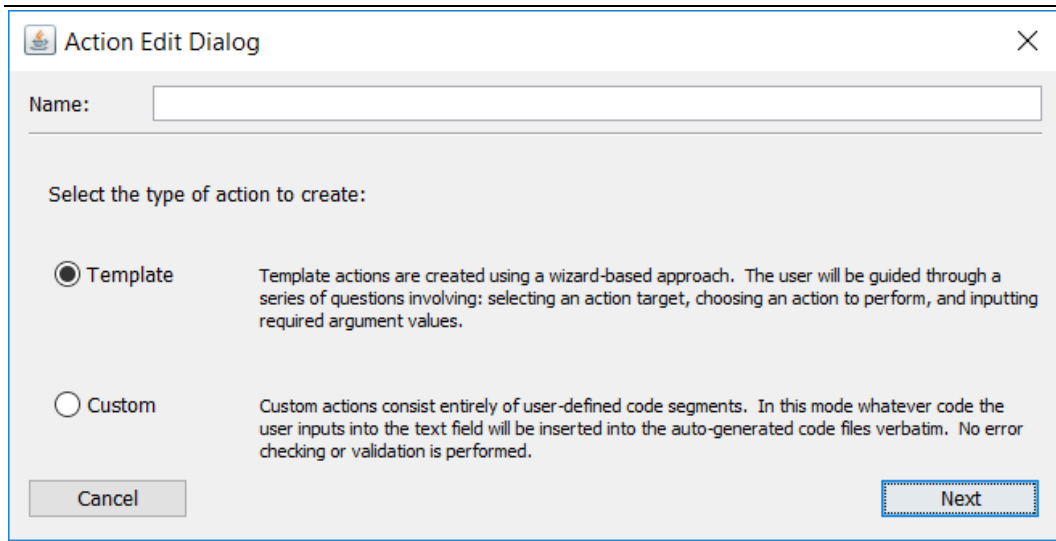
调整按钮大小和位置，设置按钮的文本和风格。最终效果如下图。



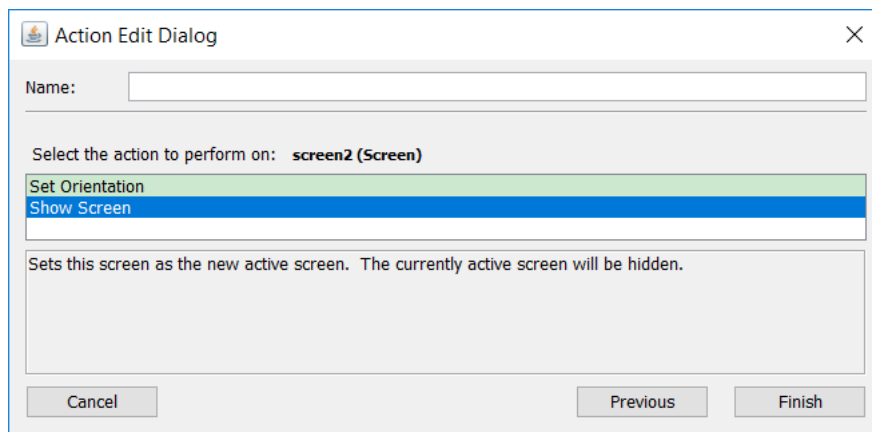
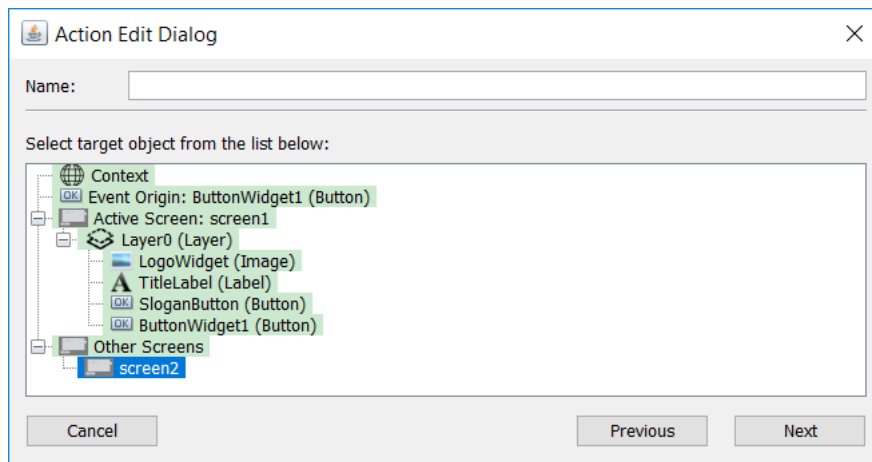
处理按钮的释放事件响应，在 Properties Editor 窗口中，勾选 Released Event 的复选框，并且点击右边按钮。在弹出的 Event Editor 对话框中点击 Create New Action 按钮。



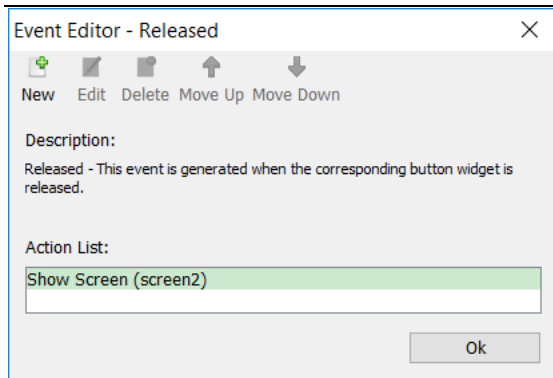
选择第一个选项，点击 Next 按钮。



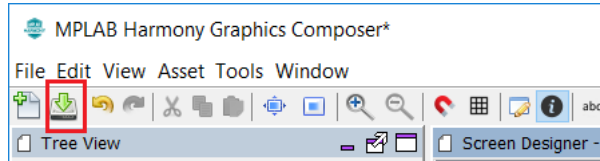
选择操纵对象是 screen2，点击 Next 按钮，选择执行动作是 Show Screen，点击 Finish 按钮。



创建事件成功，如下图，点击 OK 按钮完成。这表示当按下页面 screen1 下的这个按钮，程序就会切换到页面 screen2 显示。

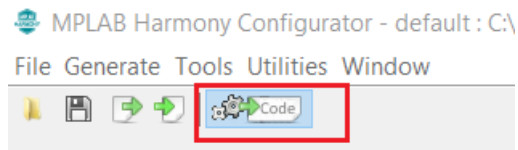


点击 MHGC 主菜单的保存按钮，完成后关闭 MHGC。

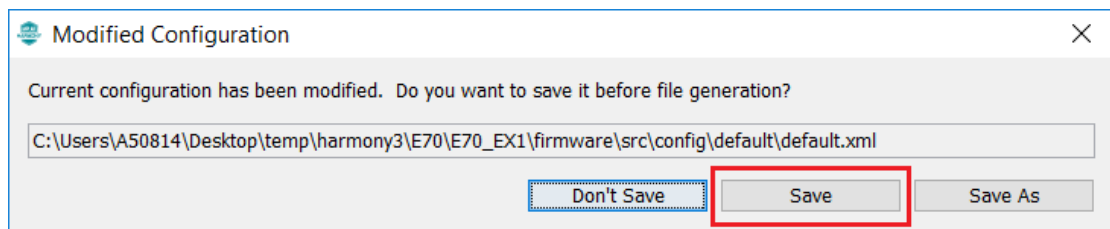


#### (四) 生成和编译代码，烧录程序和测试

回到 MHC 界面，点击 generate code 按钮来生成代码。

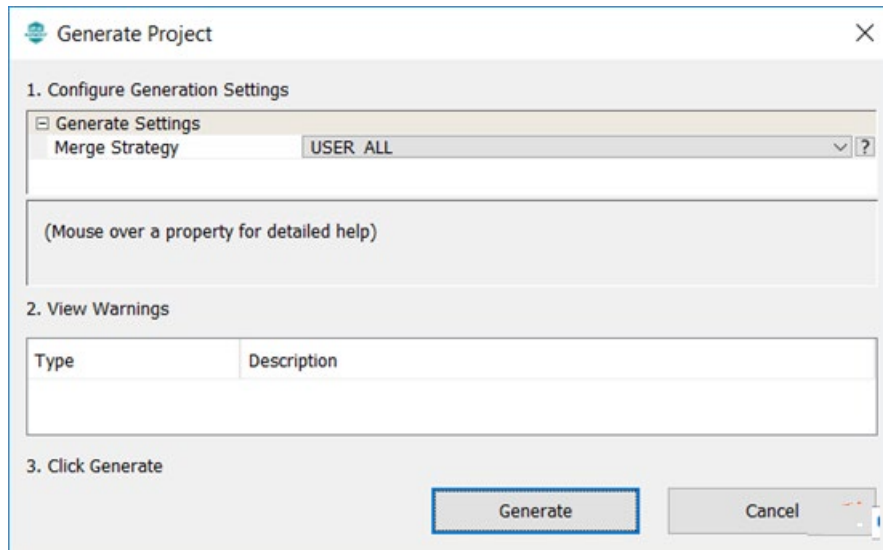


是否保存配置，点击 Save 按钮。



点击 generate 按钮。



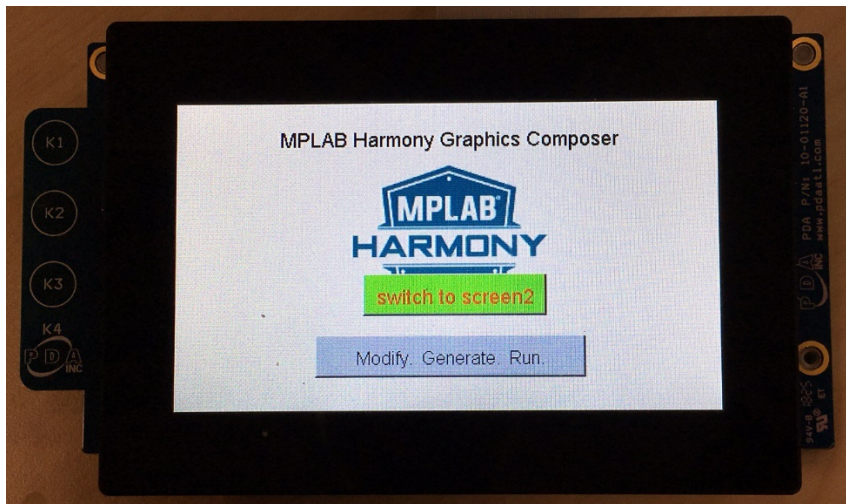


代码生成结束后，可以关闭 MHC。

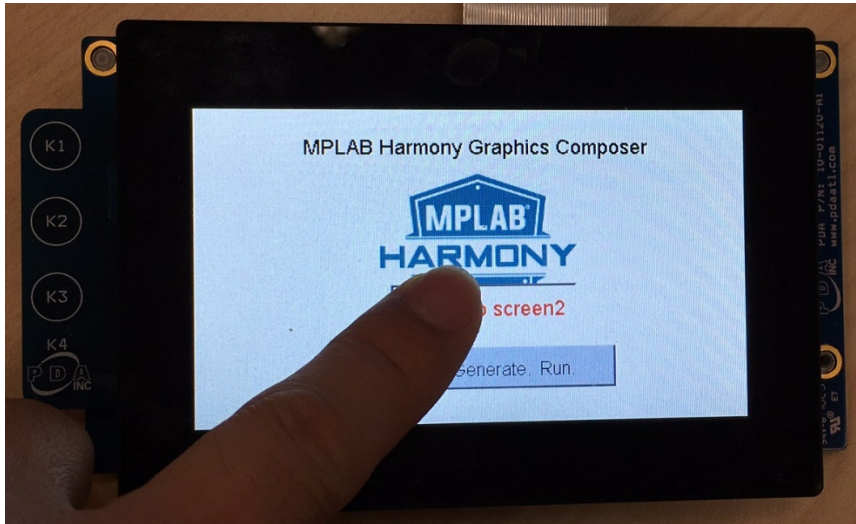
点击工具栏的“Make and Program”来完成编译，烧录和测试。



烧写成功后，程序运行如下图。



点击 switch to screen2 按钮。



成功进入 screen2 页面。



## 五、 总结

本文介绍了使用 MHGC(Microchip Harmony Graphic Composer)设计图形界面，介绍了图元的概念，包含了控件的使用等。

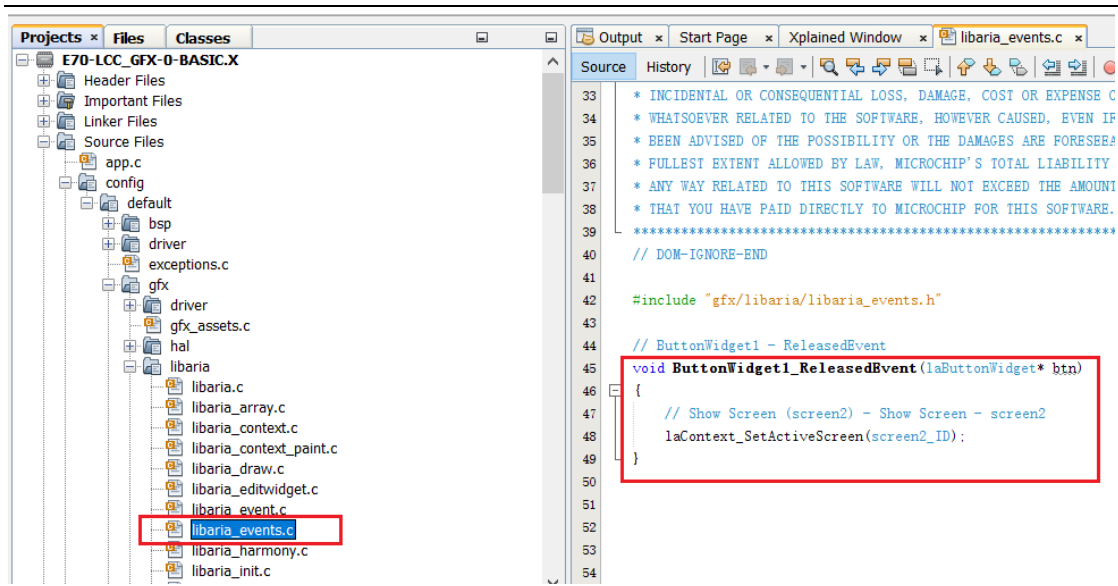
可以在本工程的基础上，在页面 screen2 上添加按钮，按钮事件响应是切换到页面 screen1。

更多图形示例工程和帮助文档请参考：

Harmony3 安装目录\gfx\_apps

Harmony3 安装目录\gfx\_apps\doc\help\_apps\_gfx.chm

附：关于按钮的事件响应，其对应的代码可以从如下位置看到。



The screenshot displays an IDE interface for a project named "E70-LCC\_GFX-0-BASIC.X". The left sidebar shows a project tree with the following structure:

- Header Files
- Important Files
- Linker Files
- Source Files
  - app.c
  - config
    - default
    - bsp
    - driver
    - exceptions.c
    - gfx
      - driver
      - gfx\_assets.c
      - hal
      - libaria
        - libaria.c
        - libaria\_array.c
        - libaria\_context.c
        - libaria\_context\_paint.c
        - libaria\_draw.c
        - libaria\_editwidget.c
        - libaria\_event.c
        - libaria\_events.c**
        - libaria\_harmony.c
        - libaria\_init.c

The right pane shows the source code for "libaria\_events.c". The code includes a copyright notice, a DOM-IGNORE-END comment, and an include statement for "gfx/libaria/libaria\_events.h". The function "ButtonWidget1\_ReleasedEvent" is defined as follows:

```
33 * INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE C
34 * WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF
35 * BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEA
36 * FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY
37 * ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT
38 * THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS SOFTWARE.
39 *****
40 // DOM-IGNORE-END
41
42 #include "gfx/libaria/libaria_events.h"
43
44 // ButtonWidget1 - ReleasedEvent
45 void ButtonWidget1_ReleasedEvent(laButtonWidget* btn)
46 {
47     // Show Screen (screen2) - Show Screen - screen2
48     laContext_SetActiveScreen(screen2_ID);
49 }
50
51
52
53
54
```