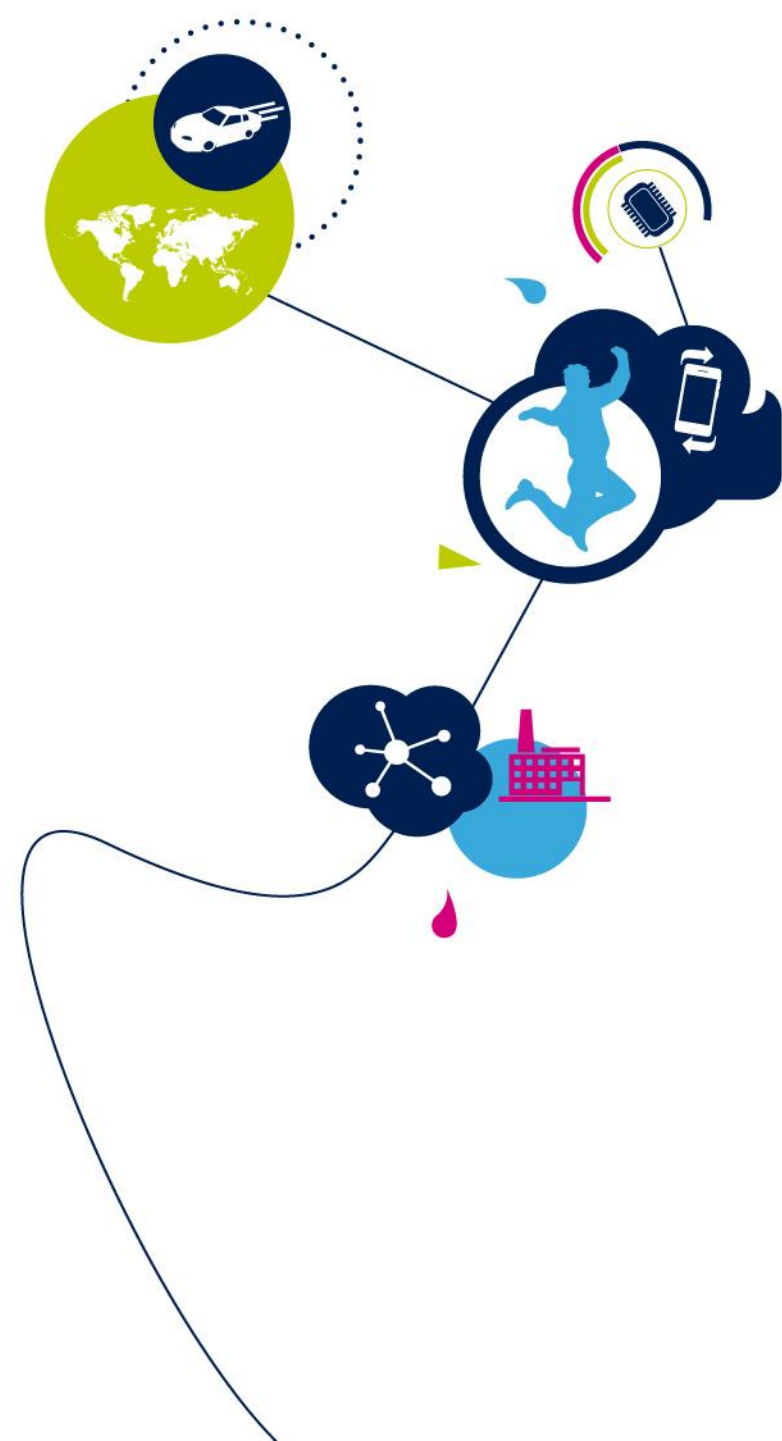


动手实验4

创建一个自定义控件



-可使用TouchGFX Designer的模板

- Application Template
 - STM32F746G Discovery Kit
- UI Template
 - Blank UI

TouchGFX DESIGNER

MY APPLICATIONS ONLINE APPLICATIONS

Create New Application

APPLICATION NAME

MyCustomizedWidgetDemo

APPLICATION DIRECTORY

C:\TouchGFXProjects

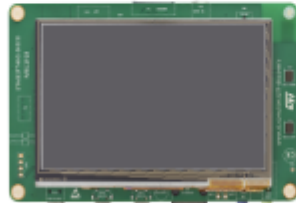
APPLICATION TEMPLATE

STM32F746G Discovery Kit
by STMicroelectronics (v2.0.0)

Board Name
STM32F746G-...

Operating System
FreeRTOS

Resolution
480 x 272



UI TEMPLATE

Blank UI
by STMicroelectronics (v1.1.0)



COLOR DEPTH

24 bit

WIDTH

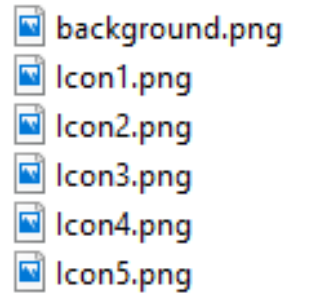
480

HEIGHT

272

CREATE

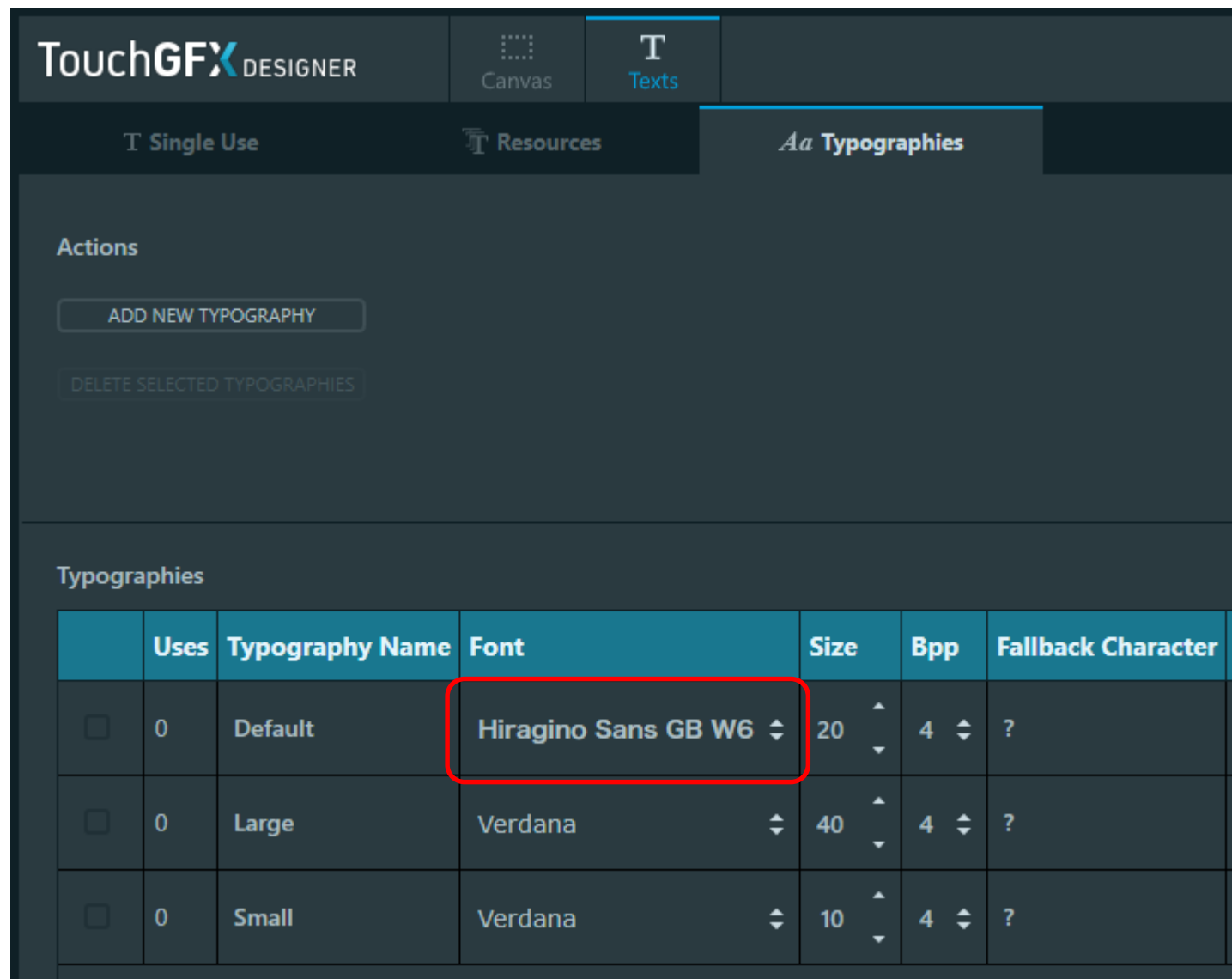
- 将图片icon1,2,3,4,5拷贝到下面目录：
 - C:\TouchGFXProjects\MyCustomizedWidgetDemo\TouchGFX\assets\images\
- 将字体文件拷贝到下面目录并改名：
 - C:\TouchGFXProjects\MyCustomizedWidgetDemo\TouchGFX\assets\fonts\
 - Hiragino Sans GB W6.otf → HiraginoSansGBW6.otf.otf
 - 因为TouchGFX目前不支持路径或文件名带空格



修改默认字体为

4

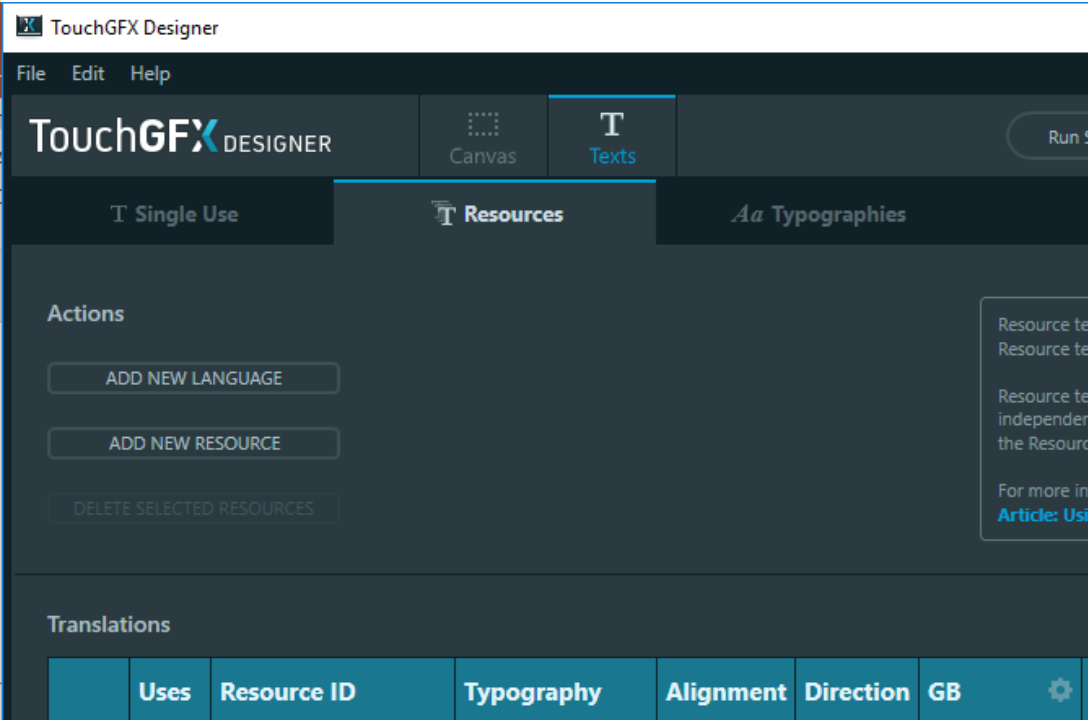
- 修改默认字体为中文字体



The screenshot shows the TouchGFX Designer interface. The 'Texts' tab is selected in the top navigation bar. Below it, the 'Aa Typographies' section is active. The 'Actions' panel on the left contains buttons for 'ADD NEW TYPOGRAPHY' and 'DELETE SELECTED TYPOGRAPHIES'. The 'Typographies' table lists three entries: 'Default', 'Large', and 'Small'. The 'Default' entry is highlighted with a red box, showing the font 'Hiragino Sans GB W6'.

	Uses	Typography Name	Font	Size	Bpp	Fallback Character
☐	0	Default	Hiragino Sans GB W6	20	4	?
☐	0	Large	Verdana	40	4	?
☐	0	Small	Verdana	10	4	?

- 增加字体资源

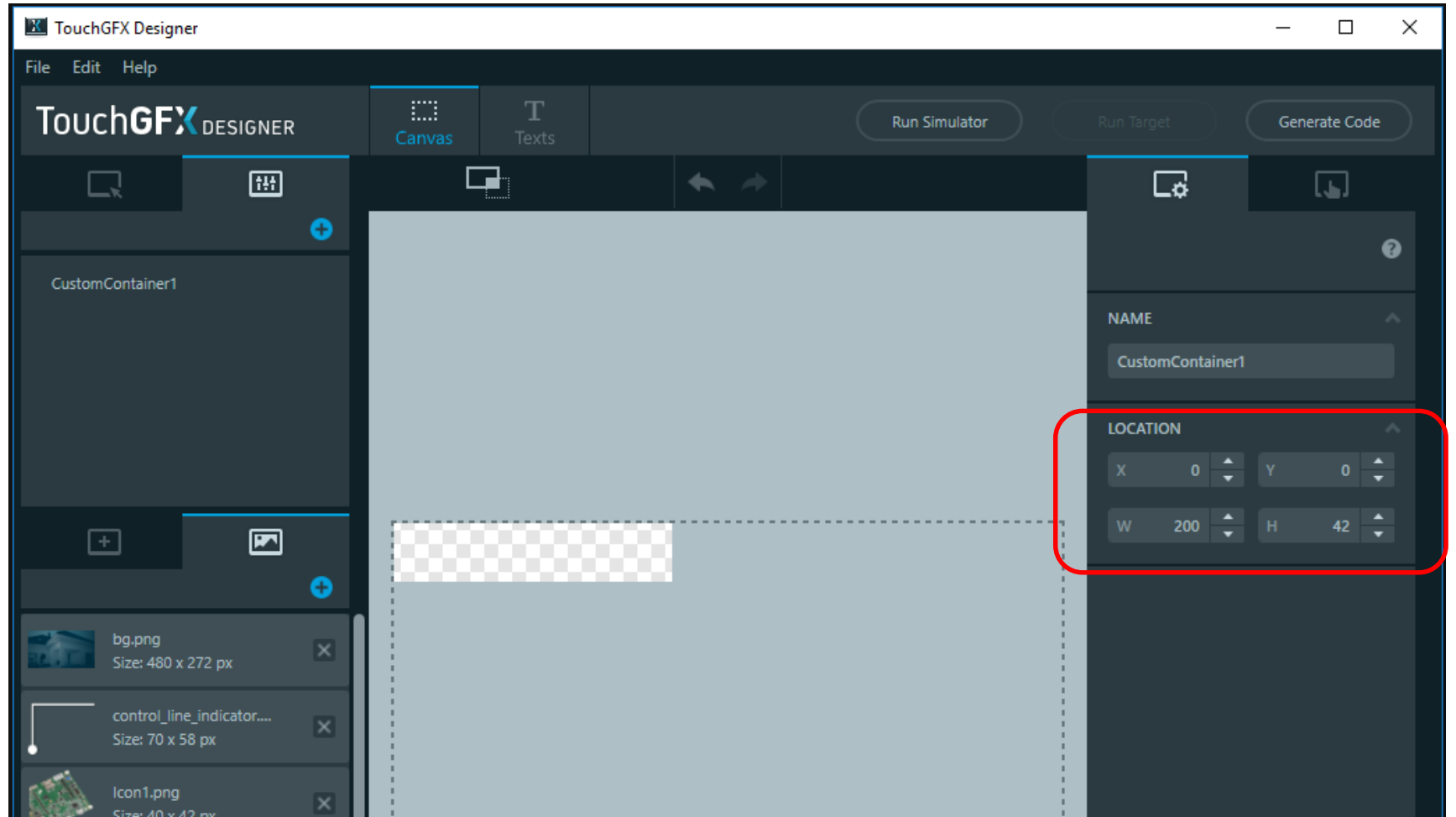


Translations						
	Uses	Resource ID	Typography	Alignment	Direction	GB
☐	0	ResourceId1	Default, 20px			PCB工具
☐	0	ResourceId2	Default, 20px			关灯
☐	0	ResourceId3	Default, 20px			开灯
☐	0	ResourceId4	Default, 20px			法国
☐	0	ResourceId5	Default, 20px			英国
☐	0	ResourceId6	Default, 20px			PCB工具2
☐	0	ResourceId7	Default, 20px			关灯2
☐	0	ResourceId8	Default, 20px			开灯2
☐	0	ResourceId9	Default, 20px			法国2
☐	0	ResourceId10	Default, 20px			英国2

创建 customer container

6

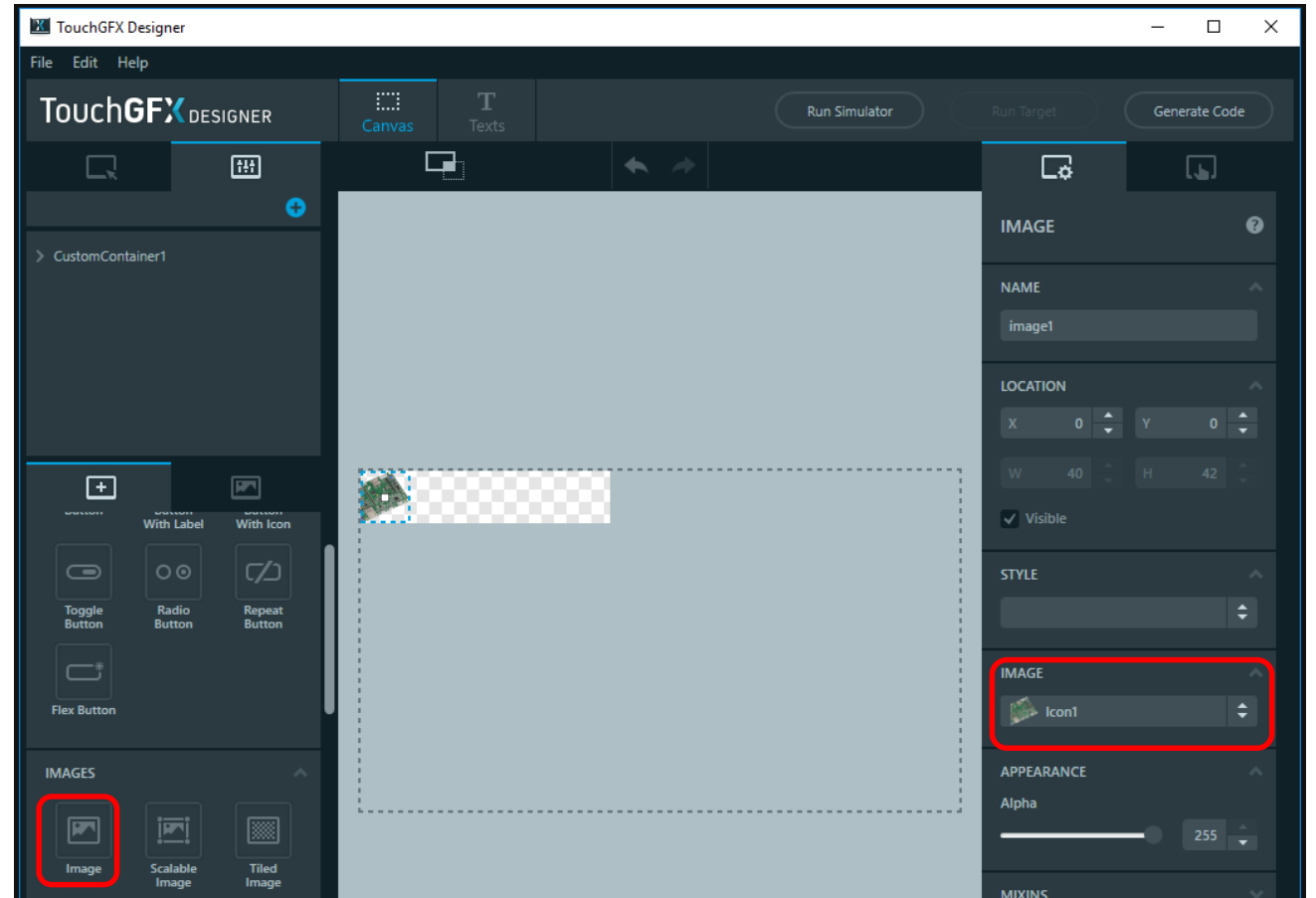
- 调整宽度和高度



创建 customer container

7

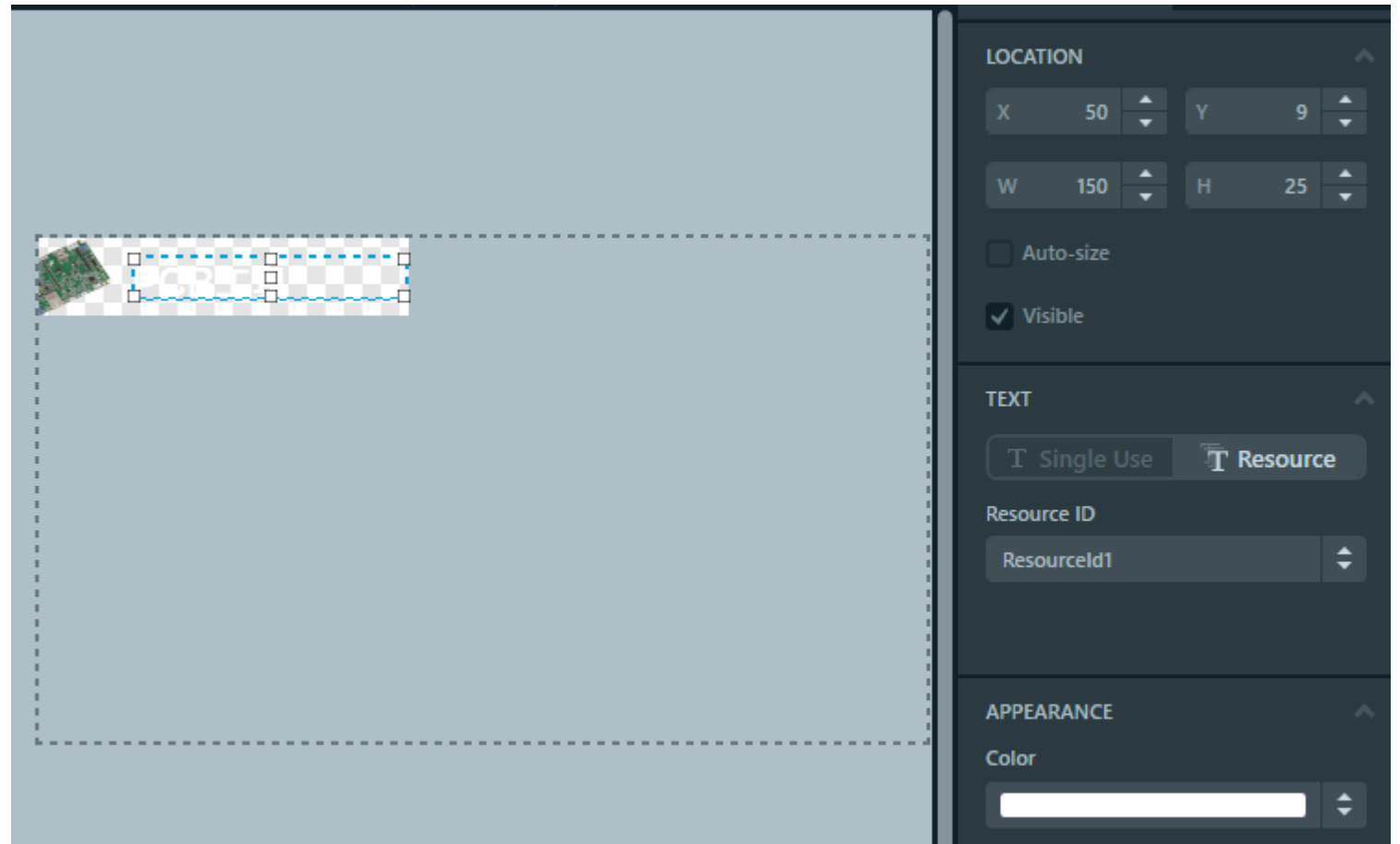
- 增加一个Image控件



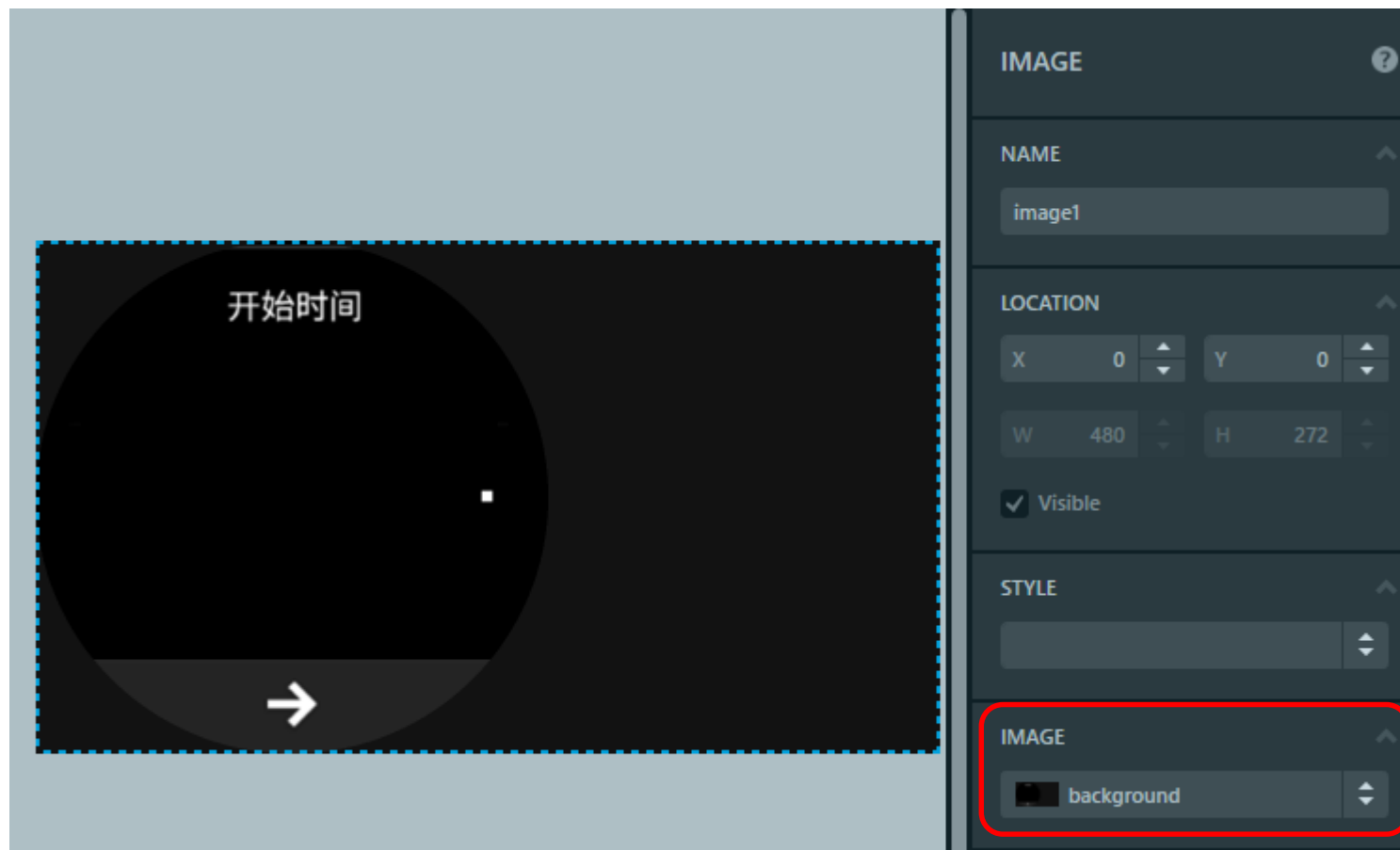
创建 customer container

8

- 增加Text控件
 - 取消Auto-size
 - 使用T Resource
 - 颜色改为白色



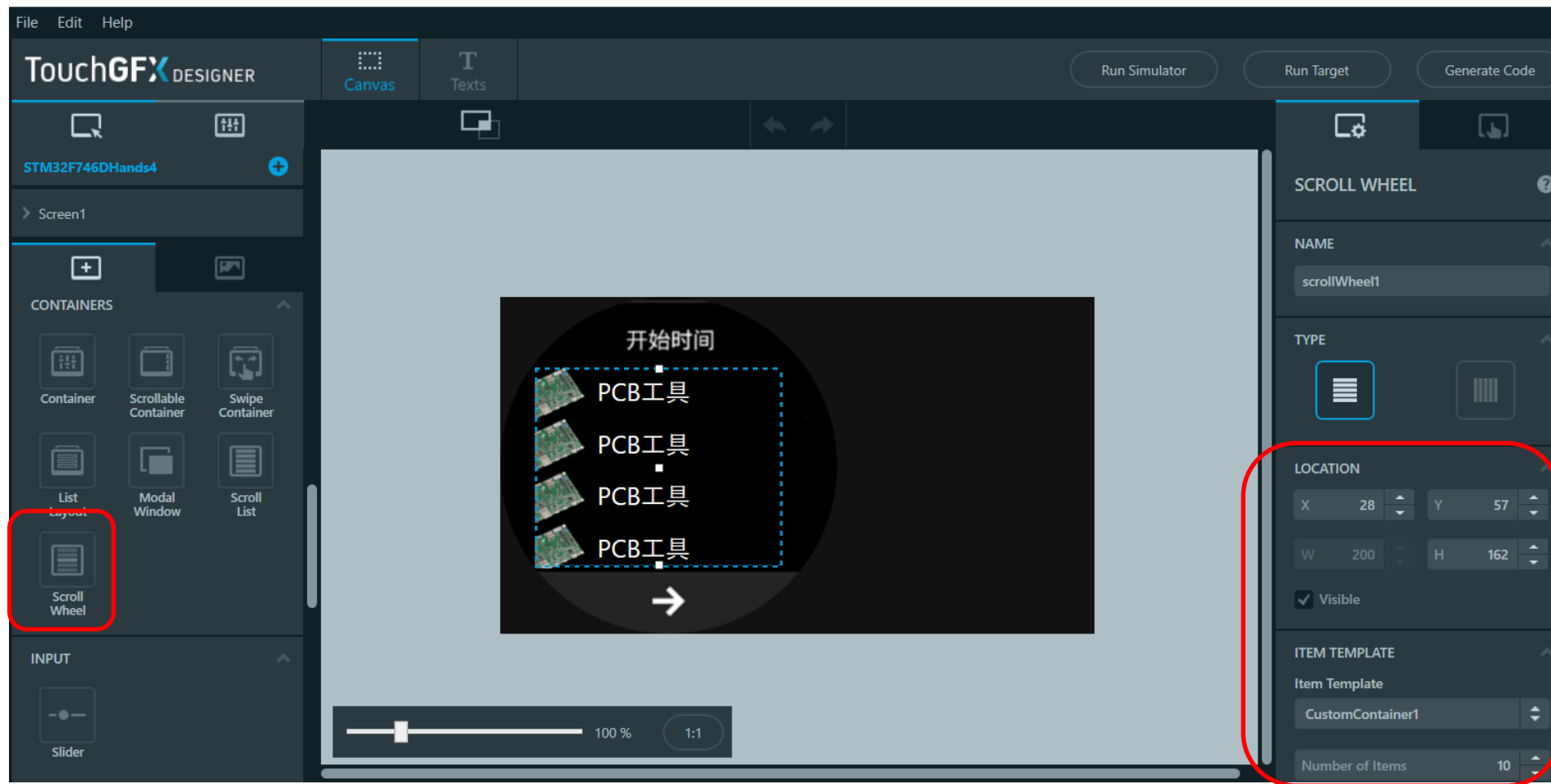
- 添加Image
 - 选中background



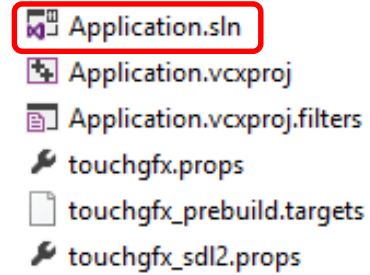
使用customer container

10

- 新建一个ScrollWheel控件并设置customer container作为模板



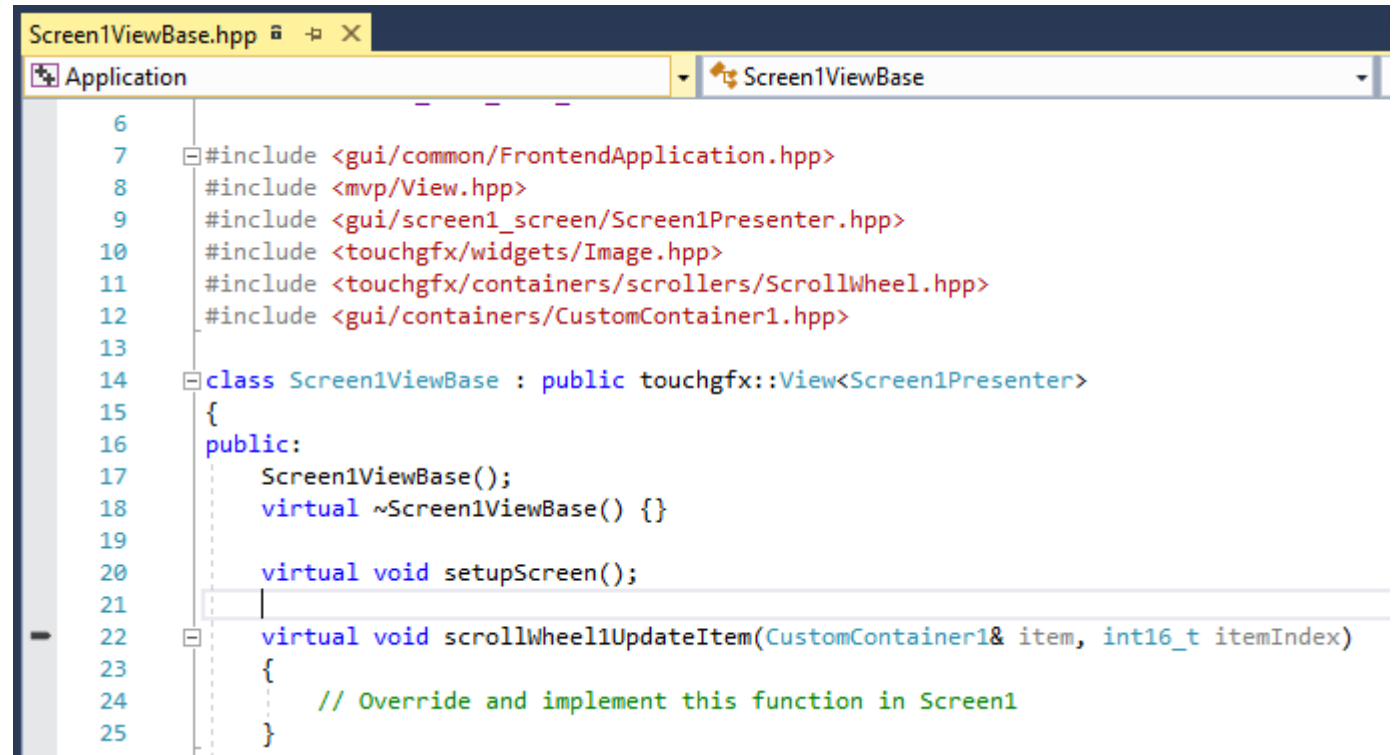
- 生成代码使用Visual Studio打开工程
 - C:\TouchGFXProjects\MyCustomizedWidgetDemo\TouchGFX\simulator\msvs



定义新的更新模式

12

- 找到scroll wheel的更新虚拟函数,复制函数名,需要在Screen1View.cpp中实现
 - Screen1ViewBase.hpp



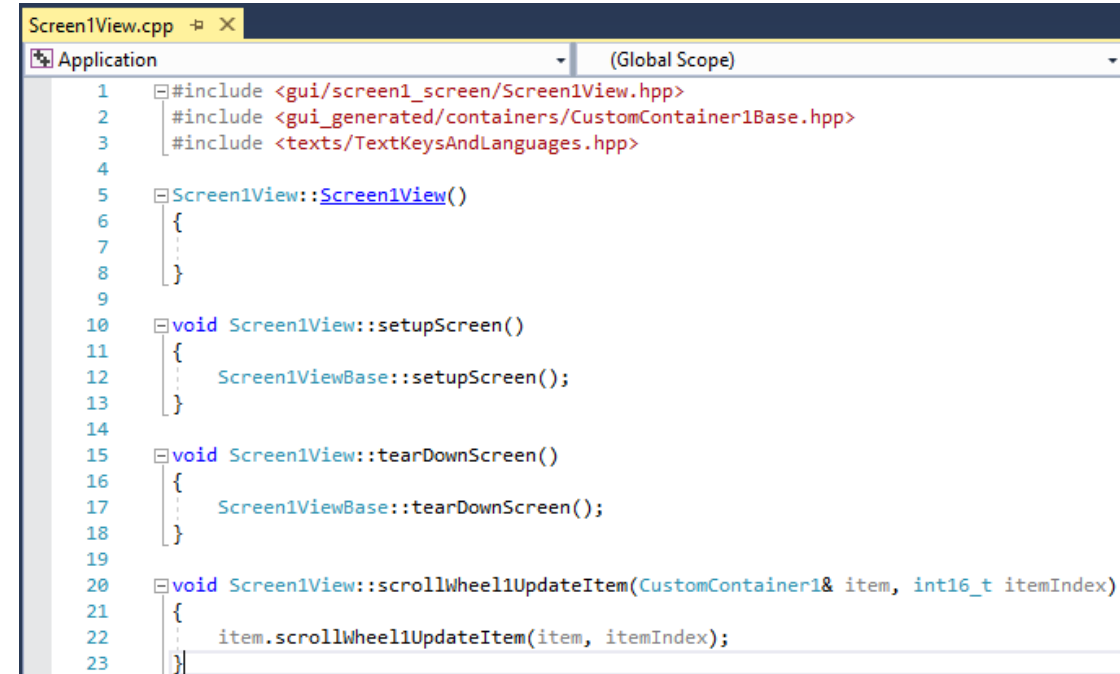
```
Screen1ViewBase.hpp
Application Screen1ViewBase

6
7 #include <gui/common/FrontendApplication.hpp>
8 #include <mvp/View.hpp>
9 #include <gui/screen1_screen/Screen1Presenter.hpp>
10 #include <touchgfx/widgets/Image.hpp>
11 #include <touchgfx/containers/scrollers/ScrollWheel.hpp>
12 #include <gui/containers/CustomContainer1.hpp>
13
14 class Screen1ViewBase : public touchgfx::View<Screen1Presenter>
15 {
16 public:
17     Screen1ViewBase();
18     virtual ~Screen1ViewBase() {}
19
20     virtual void setupScreen();
21
22     virtual void scrollWheelUpdateItem(CustomContainer1& item, int16_t itemIndex)
23     {
24         // Override and implement this function in Screen1
25     }
26 }
```

- 修改Screen1View.hpp
 - 重载虚拟函数
 - virtual void scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex);

```
Screen1View.hpp  X
Application (Global Scope)
1  #ifndef SCREEN1_VIEW_HPP
2  #define SCREEN1_VIEW_HPP
3
4  #include <gui_generated/screen1_screen/Screen1ViewBase.hpp>
5  #include <gui/screen1_screen/Screen1Presenter.hpp>
6
7  class Screen1View : public Screen1ViewBase
8  {
9  public:
10     Screen1View();
11     virtual ~Screen1View() {}
12     virtual void setupScreen();
13     virtual void tearDownScreen();
14     virtual void scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex);
15 protected:
16 };
17
18 #endif // SCREEN1_VIEW_HPP
```

- 修改Screen1View.cpp
 - 增加头文件
 - `#include <gui_generated/containers/CustomContainer1Base.hpp>`
 - `#include <texts/TextKeysAndLanguages.hpp>`
 - 实现scrollWheel1UpdateItem()
 - `void Screen1View::scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex)`
 - `{`
 - `item.scrollWheel1UpdateItem(item, itemIndex);`
 - `}`



```
Screen1View.cpp  Application (Global Scope)
1  #include <gui/screen1_screen/Screen1View.hpp>
2  #include <gui_generated/containers/CustomContainer1Base.hpp>
3  #include <texts/TextKeysAndLanguages.hpp>
4
5  Screen1View::Screen1View()
6  {
7  }
8
9
10 void Screen1View::setupScreen()
11 {
12     Screen1ViewBase::setupScreen();
13 }
14
15 void Screen1View::tearDownScreen()
16 {
17     Screen1ViewBase::tearDownScreen();
18 }
19
20 void Screen1View::scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex)
21 {
22     item.scrollWheel1UpdateItem(item, itemIndex);
23 }
```

定义新的更新模式

15

- 在CustomContainer1.hpp中重载虚拟函数
 - virtual void scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex);
 - virtual void setXY(int16_t x, int16_t y);

```
CustomContainer1.hpp - X
Application (Global Scope)
1  #ifndef CUSTOMCONTAINER1_HPP
2  #define CUSTOMCONTAINER1_HPP
3
4  #include <gui_generated/containers/CustomContainer1Base.hpp>
5
6  class CustomContainer1 : public CustomContainer1Base
7  {
8  public:
9      CustomContainer1();
10     virtual ~CustomContainer1() {}
11
12     virtual void initialize();
13     virtual void scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex);
14     virtual void setXY(int16_t x, int16_t y);
15 protected:
16 };
17
18 #endif // CUSTOMCONTAINER1_HPP
```

- 修改CustomContainer1.cpp
 - 增加头文件
 - `#include "BitmapDatabase.hpp"`
 - `#include <texts/TextKeysAndLanguages.hpp>`
 - 增加位图列表
 - `uint16_t iconArray[10] = {`
 - `BITMAP_ICON1_ID,`
 - `BITMAP_ICON2_ID,`
 - `BITMAP_ICON3_ID,`
 - `BITMAP_ICON4_ID,`
 - `BITMAP_ICON5_ID,`
 - `BITMAP_ICON1_ID,`
 - `BITMAP_ICON2_ID,`
 - `BITMAP_ICON3_ID,`
 - `BITMAP_ICON4_ID,`
 - `BITMAP_ICON5_ID,`
 - `};`

- 修改CustomContainer1.cpp

- 增加资源列表

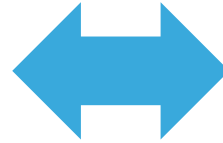
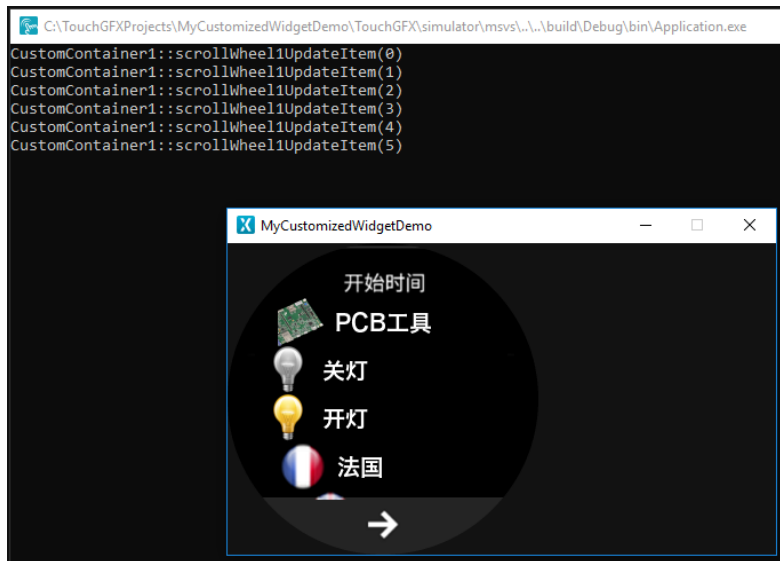
- TEXTS buttonNameArray[10] = {
 - T_RESOURCEID1,
 - T_RESOURCEID2,
 - T_RESOURCEID3,
 - T_RESOURCEID4,
 - T_RESOURCEID5,
 - T_RESOURCEID6,
 - T_RESOURCEID7,
 - T_RESOURCEID8,
 - T_RESOURCEID9,
 - T_RESOURCEID10,
 - };

- 修改CustomContainer1.cpp
 - 实现函数scrollWheel1UpdateItem()
 - `void CustomContainer1::scrollWheel1UpdateItem(CustomContainer1& item, int16_t itemIndex)`
 - `{`
 - `touchgfx_printf("CustomContainer1::scrollWheel1UpdateItem(%d)\n", itemIndex);`
 - `image1.setBitmap(Bitmap(iconArray[itemIndex]));`
 - `textArea1.setTypedText(TypedText(buttonNameArray[itemIndex]));`
 - `}`
 - 实现函数SetXY()
 - `void CustomContainer1::setXY(int16_t x, int16_t y)`
 - `{`
 - `x = x + (y-60)*(y-60)/300; //用于实现按照弧线布局`
 - `Drawable::setX(x);`
 - `Drawable::setY(y);`
 - `}`

期望实现结果

实验完成期望的运行结果如下所示：

- 1.在Visual Studio 上仿真
- 2.在STM32F746Discovery板上运行





Thank You !