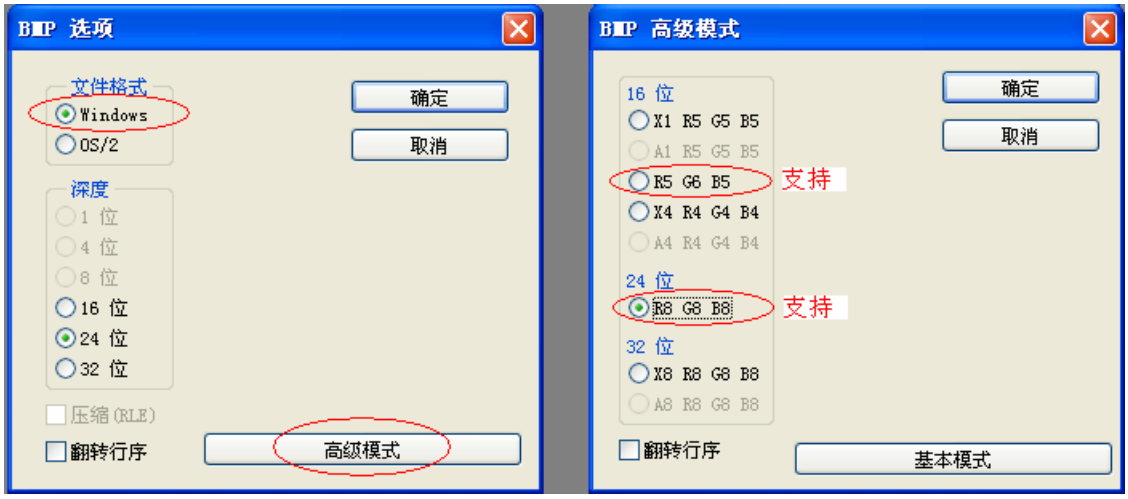
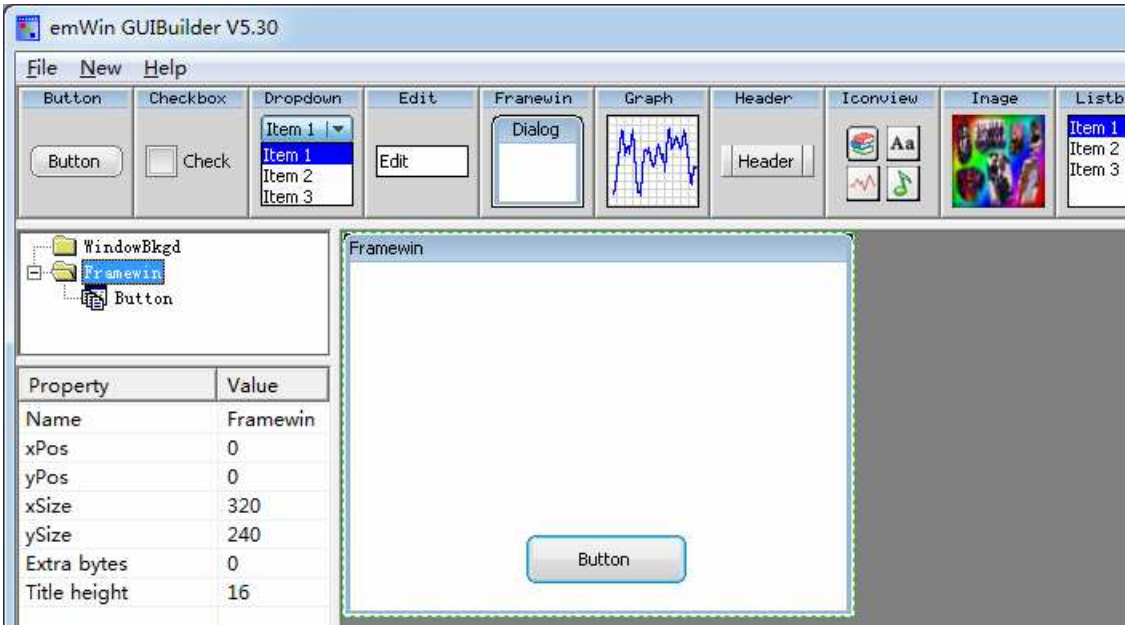
emwin 2-day quick tutorial 009_bitmap display and bitmap skinning

1.Use bitmaps in CodeBlocks

Note:The GLCD screen supports BMP bitmaps in 24bbp(RGB888) and 16bbp(RGB565) formats(bitmaps in other formats are not supported), and 16bbp(RGB565) BMP bitmaps can be saved through photoshop.Therefore,bitmaps in this format are also used in CodeBlocks.Take Photoshop as an example:

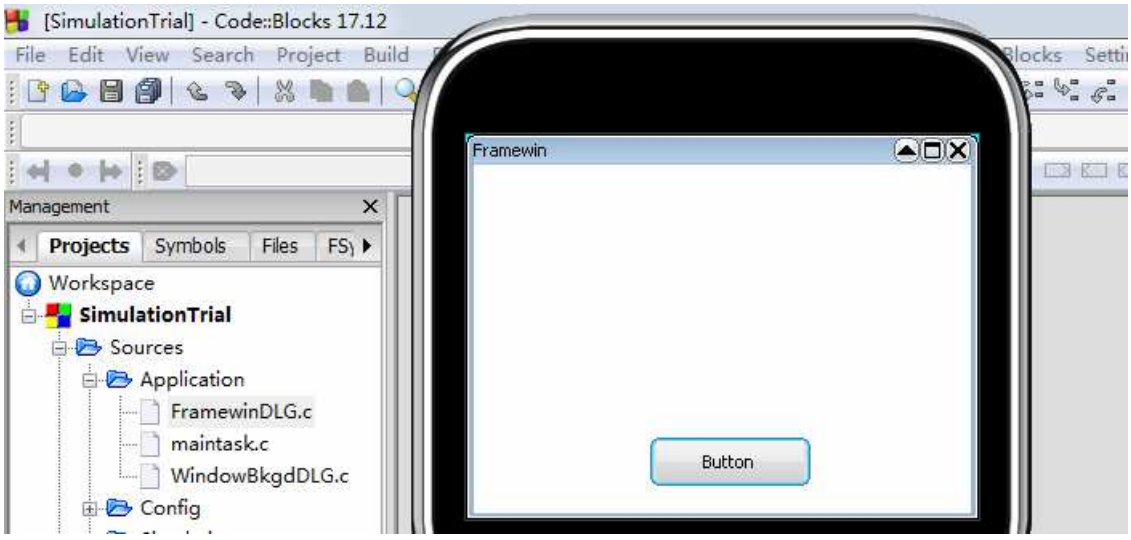


(1)Use GUIBuilder V5.30 to create the following pages and save them as c files,then add them to the CodeBlocks project:



Note:In the C file generated by GUIBuilder,user code is best added between "USER START"and "USER END", don't make any changes except the numbers elsewhere. Otherwise,GUIBuilder will not be able to open this c file again. In addition,when GUIBuilder opens this C file again and edits and saves it,the content between "USER START" and "USER END" will not be changed. Also,do not modify the C code directly when it is opened by GUIBuilder. Otherwise,the modified content will be lost when saved by GUIBuilder.

Compiling and running:

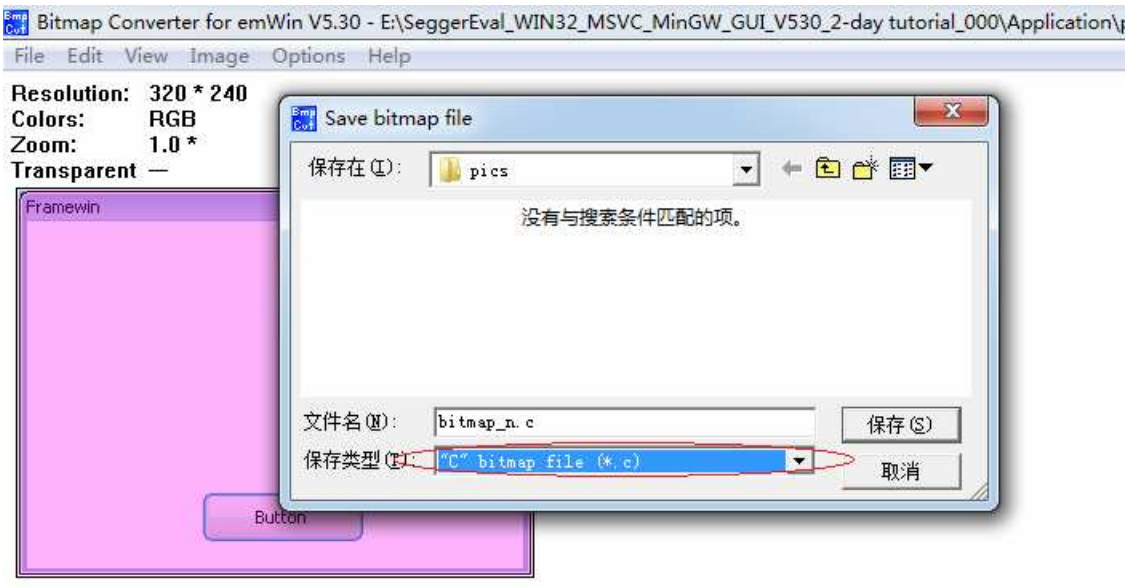


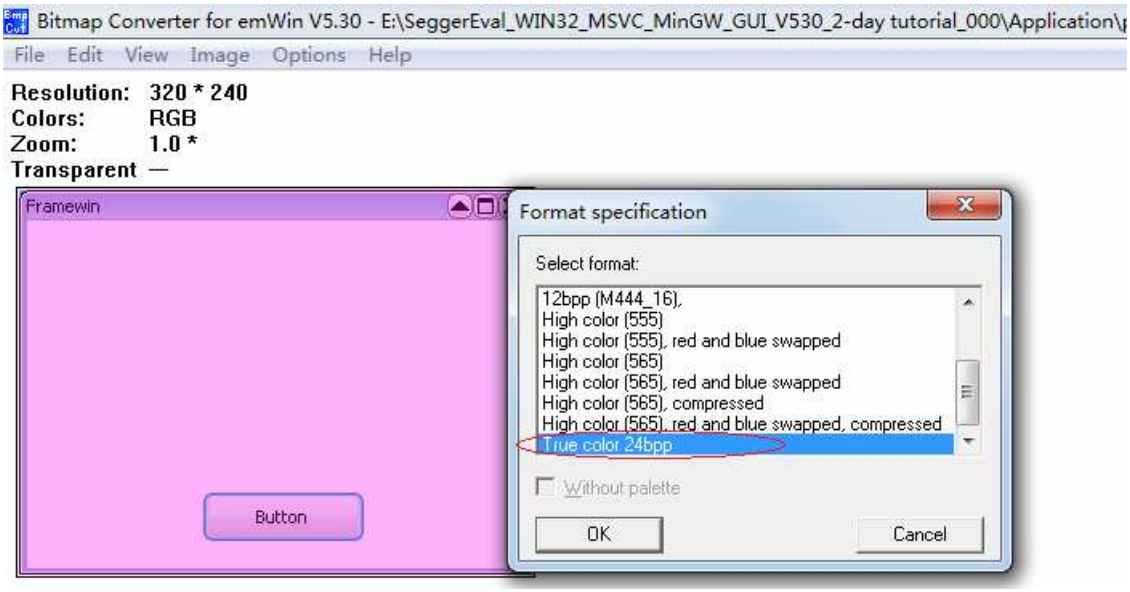
(2)capture the screen of the simulator and save it as two BMP files, one is the normal color interface(In order to distinguish the original interface, the color was modified), the other is the anti-color interface, the file name corresponds to bitmap_n.bmp and bitmap_p.bmp:

The normal color bitmap is used for button released, and the reverse color bitmap is used for button pressed.

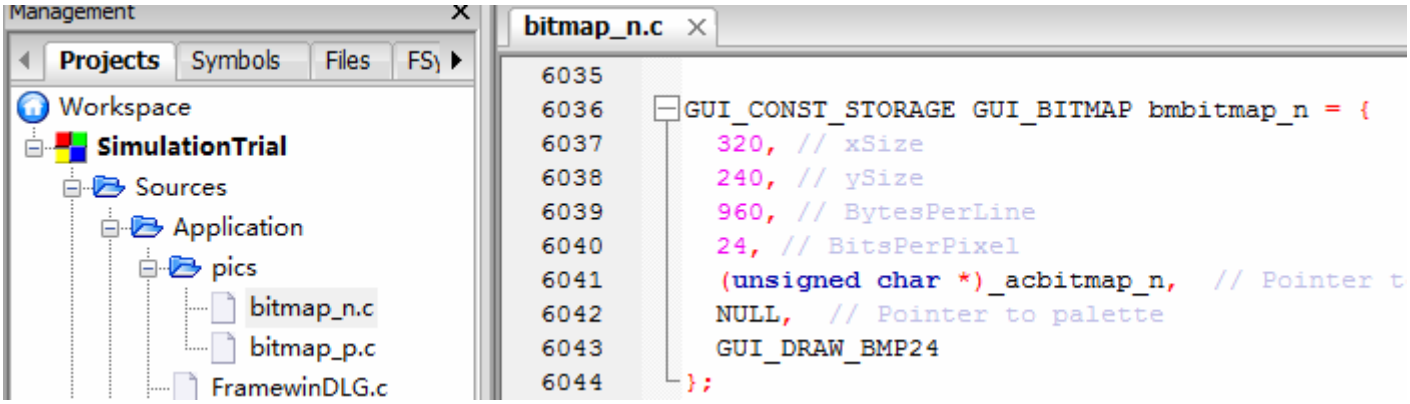
Note:If you want exquisite and gorgeous interface effect,also follow this method:capture the original interface->UI processing->set emWin to use bitmap skin(whole picture,no need to be sliced and decomposed).

(3)Using BmpCvt_V530.exe(under Tool directory) to convert bitmaps into c files: bitmap_n.c and bitmap_p.c





(4)Copy bitmap_n.c and add them to CodeBlocks project:



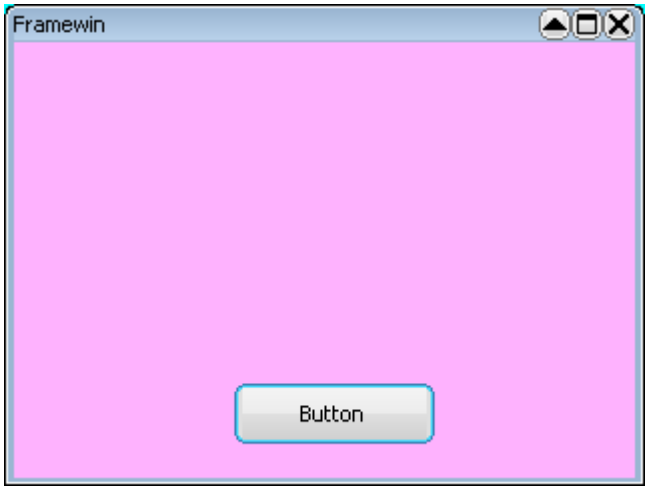
See here, you may think that if it is MCU platform, where is the bitmap data stored? How to read the bitmap data?Storage data can use NAND-FLASH, SPI-FLASH, NOR-FLASH and so on. As for how to read the bitmap data, this is the matter of the underlying driver of emWin. Here we just talk about how to design the interface application of emWin. In GLCD, bitmap is downloaded to Nand-Flash of GLCD through USB, and the calling display method has been done in the underlying driver of emWin in demo project.

(5)Modify the program to display bitmaps as background:

DLG.c

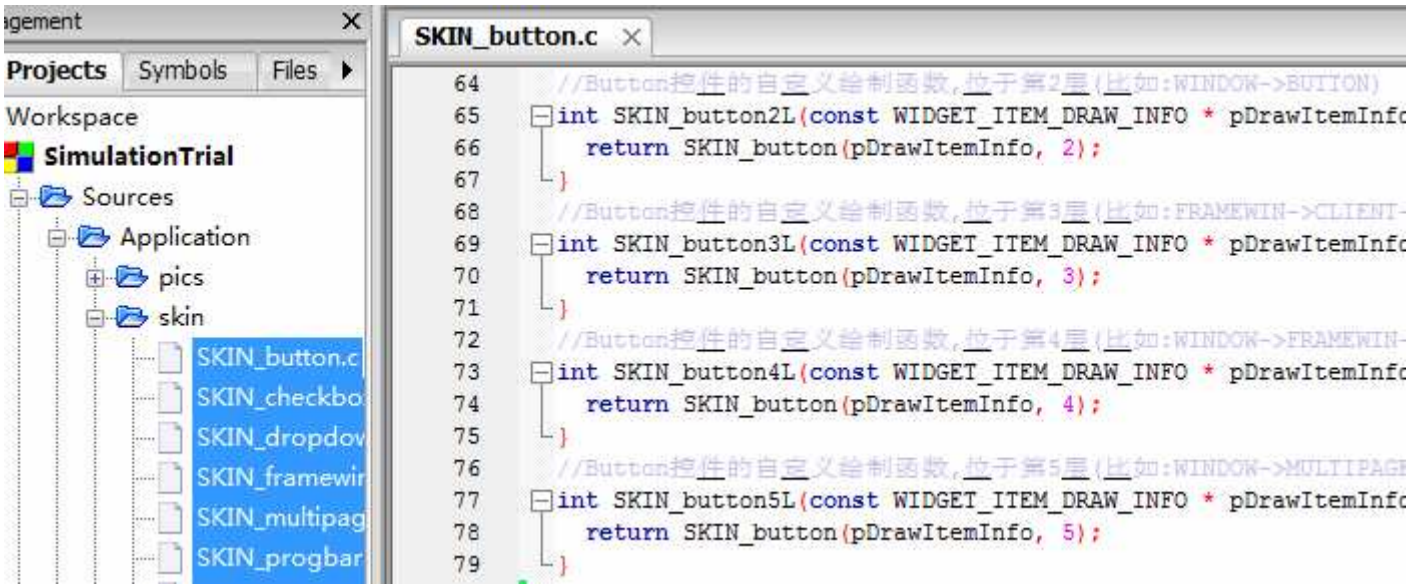
```
// USER START (Optionally insert additional message handling)
//窗口重绘消息,这个比较难说明白,反正在Framewin或Window窗口之中我们一般是用控件,如果
//Window redraw message, this is difficult to explain,anyway,we usually use widget
case WM_PAINT:
  posiX = WM_GetWindowOrgX(WM_GetParent(pMsg->hWin))-WM_GetWindowOrgX(pMsg->hWin);
  posiY = WM_GetWindowOrgY(WM_GetParent(pMsg->hWin))-WM_GetWindowOrgY(pMsg->hWin);
  GUI_DrawBitmap(&bmbitmap_n, posiX, posiY);
  break;
// USER END
default:
```

Note: Because GUI_DrawBitmap() is only displayed in the Client area,using coordinate calculation to offset the border of the bitmap to get posiX and posiY.Of course,you can also make a bitmap without the border and only leave the client area.

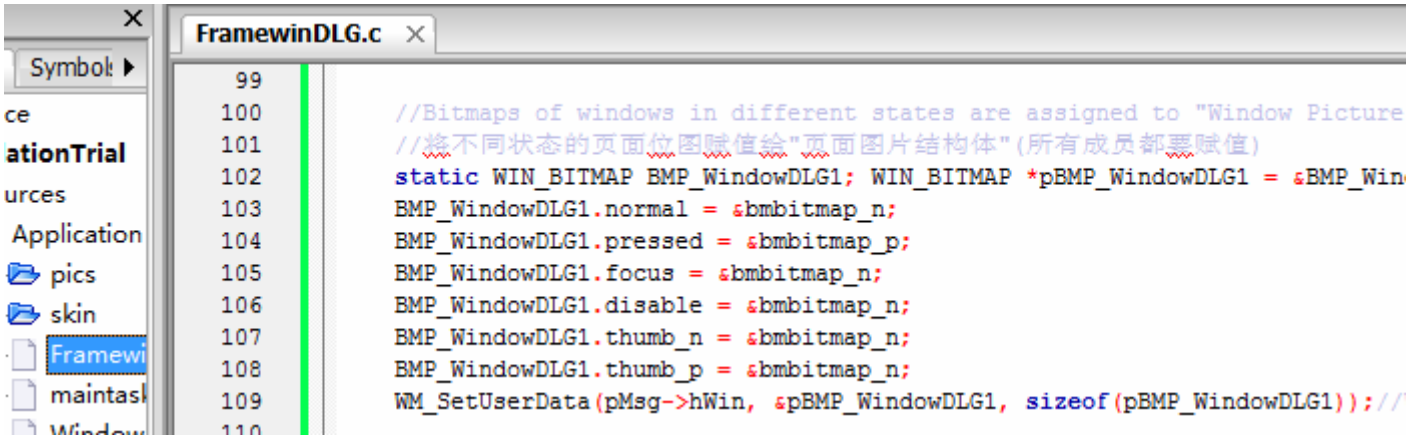


(6)The realization of button bitmap skin:

Copy the Application/skin directory of the skinPRJ_WINemWin530_NoOS_CodeBlocks_GBK.zip demo project(found in the GLCD demo code) to the project directory and add it to the CodeBlocks project.The c files in the skin directory are bitmap skinning source codes written by neqee(The principle of bitmap kinning please see the end of this article.):

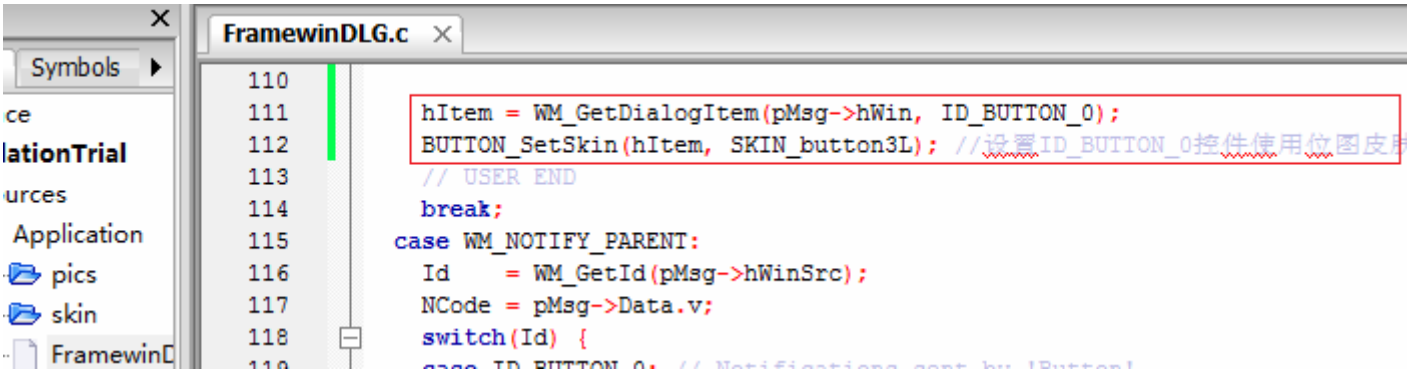


In the WM_INIT_DIALOG message(Initialize messages,can setting some initial parameters of widgets here), the page bitmaps in different states are assigned to "Page Picture Structures":



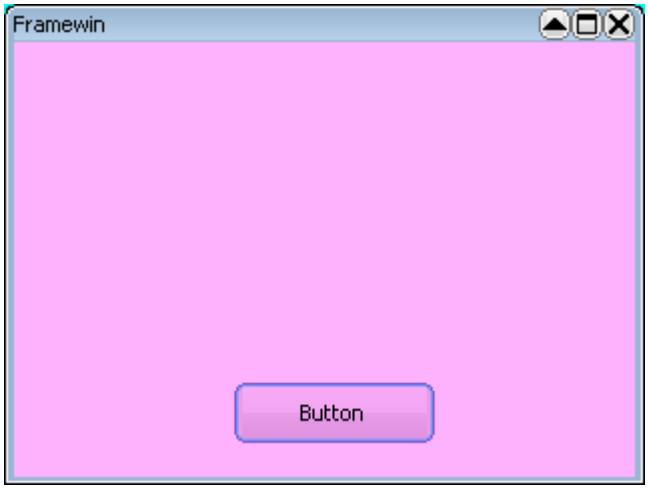
The goal of this step is to store the "pointer" of the skin bitmap of two states into the parent page. All the child controls in the parent page can share this bitmap(this is the principle that the whole btimap does not need to be sliced). If all the child widgets are set to use this skin bitmap,emWin automatically calls the bitmap as the skin(d Don't think about realizing it by yourself).Bitmaps of two states include: normal, press/mark (e.g., checking state of the checkbox widget), There are also "disabled" and "focused" states that are optional.

Setting widget to use bitmap skin:



SKIN_button3L "3L" means the number of layers from the parent window with the bitmap "pointer" to the child widget.Here the BUTTON widget is a child of the Framewin window,so their child-parent relationship is FRAMEWIN->CLIENT->BUTTON, That is, BUTTON is located at the third layer of the framewin window.

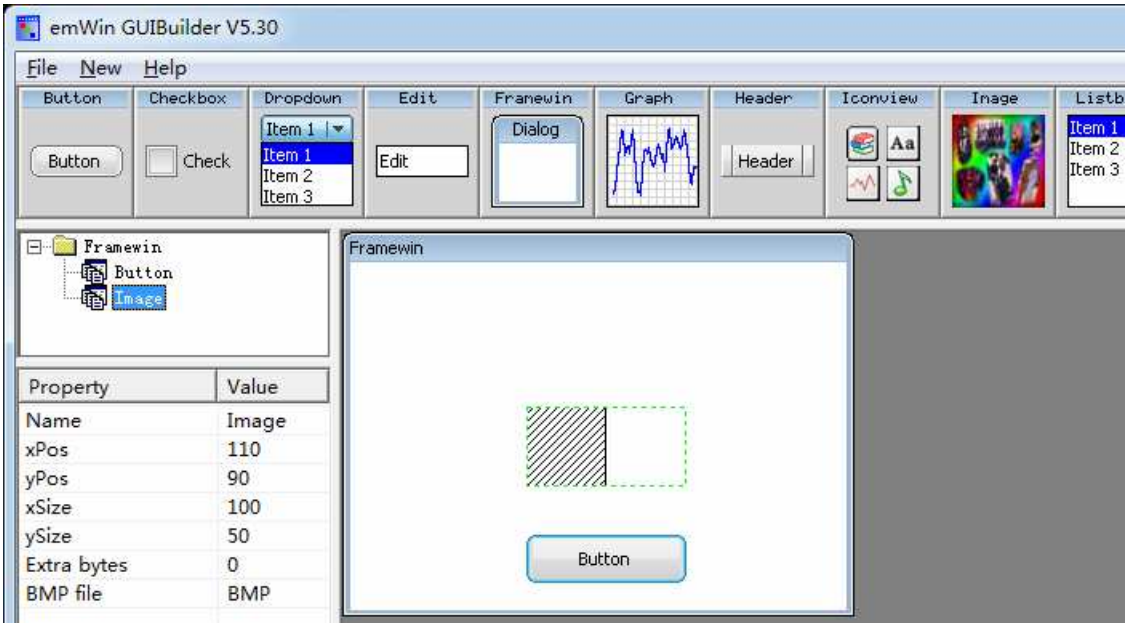
Compiling and running:



Looking at this, you might wonder, What should I do if I want to use bitmap skin even on the Framewin? Then you can use window and make a bitmap skin with borders.

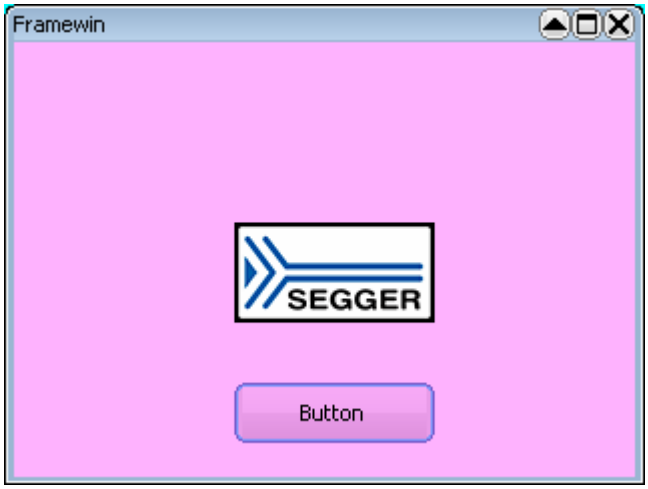
2.Use Image Widget to Display Bitmap:

In addition to being displayed as background and bitmap skin, it is much more convenient to display a bitmap somewhere using the Image widget. Use GUIBuilder V5.30 to open the pre-generated c file, add an Image widget and click Save, User code previously added between "USER START" and "USER END" will not be changed:



Assign the bitmap pointer to the Image widget in the WM_INIT_DIALOG message(Initialize messages, can setting some initial parameters of widgets here):

```
IMAGE_SetBitmap(WM_GetDialogItem(pMsg->hWin, ID_IMAGE_0), &bmlogo);
```



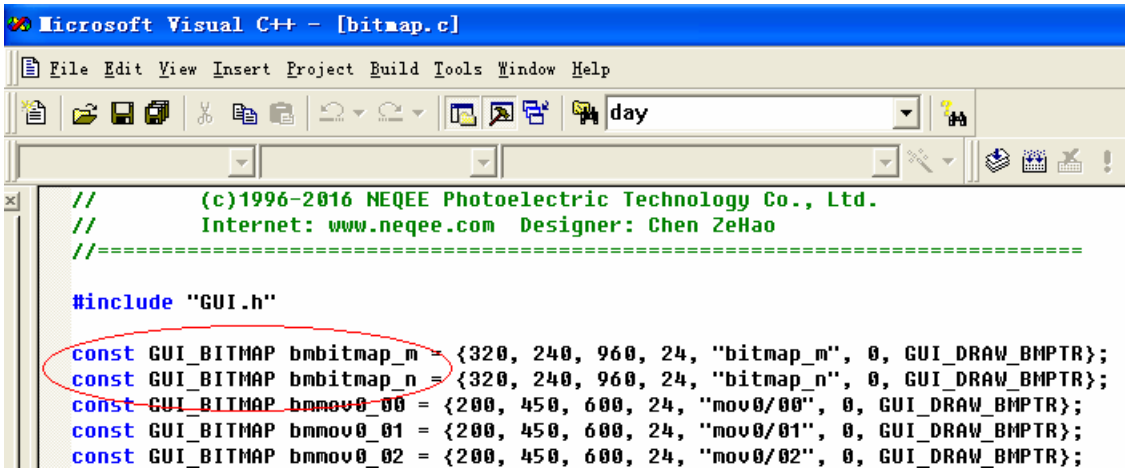
3.Use bitmaps in the GLCD

The method of using bitmap in GLCD is the same as CodeBlocks except for generating bitmap_n.c and bitmap_p.c.

1) Copy bitmap_n.bmp and bitmap_p.bmp to the pics directory of GLCD's Nand-Flash.

2)Double-click bmpGen_v121.exe tool in the Nand-Flash root directory to generate bitmap.c and bitmap.h files.

3)Adding bitmap.c and bitmap.h to the Keil MDK project,then you can use bitmap_n and bitmap_p bitmaps just like CodeBlocks:



4.The principle of emWin realizing bitmap skinning for widgets

①The traditional method of emWin pasting skin bitmap is to call the APP function XXXX_SetBitmap(),but it is very troublesome.

②In order to display the widget,emWin has one widget drawing function for each widget,we change the widget drawing function into our own "custom drawing function",So we can draw this widget as anything we like, emWin is no longer involved in drawing this widget;In this custom drawing function,we do nothing but display the bitmap of the widget,this is the principle of emWin realizing bitmap skinning for widgets.

③Support the whole bitmap mapping,do not need to slice and decompose the whole bitmap,we can do interface skinning very quickly.

As shown in the figure above,(x0, y0) is the display coordinate of the whole bitmap,(x1, y1) and (x2, y2) are the cut display coordinates automatically generated by emWin for the widget.We just need to call the GUI_DrawBitmap() function to display the whole bitmap in (x0, y0).emWin will automatically cut and display the bitmap for skinning of the widget.

The steps of bitmap skin by skinning method:

1) In the WM_INIT_DIALOG message(Initialize messages,can setting some initial parameters of widgets here),the page bitmaps in different states are assigned to "Page Picture Structures":

```
//Bitmaps of windows in different states are assigned to "Window Picture Structures" (all members must be assigned values)
//将不同状态的页面位图赋值给"页面图片结构体"(所有成员都要赋值)
static WIN_BITMAP BMP_WindowDLG1; WIN_BITMAP *pBMP_WindowDLG1 = &BMP_WindowDLG1;
BMP_WindowDLG1.normal = &bmbitmap_n;
BMP_WindowDLG1.pressed = &bmbitmap_p;
BMP_WindowDLG1.focus = &bmbitmap_n;
BMP_WindowDLG1.disable = &bmbitmap_n;
BMP_WindowDLG1.thumb_n = &bmbitmap_n;
BMP_WindowDLG1.thumb_p = &bmbitmap_n;
WM_SetUserData(pMsg->hWin, &pBMP_WindowDLG1, sizeof(pBMP_WindowDLG1));//Write the pointer to the user data
```

2) In WM_INIT_DIALOG message(Initialize messages,can setting some initial parameters of widgets here),change the drawing function of BUTTON widget to the custom drawing function:

```
hItem = WM_GetDialogItem(pMsg->hWin, ID_BUTTON_0);
BUTTON_SetSkin(hItem, SKIN_button3L);//设置 ID_BUTTON_0 控件使用位图皮肤 Set ID_BUTTON_0 to used bitmap skinning
```

Note: The custom drawing function SKIN_button3L() of BUTTON widget is implemented in SKIN_button.c file,and the "page picture structure" is defined in skin.h.

5. Summary:

Using the whole bitmap to realize emWin bitmap skin should be the first method created by neqee in the industry. It can greatly reduce the workload of bitmap skinning for widgets.Conservatively, the efficiency of making bitmap skin is estimated to be at least 10 times higher. If the bitmap corresponding to each widget is sliced down and then converted into a c file, then paste it to the widget,the workload will be terrible.