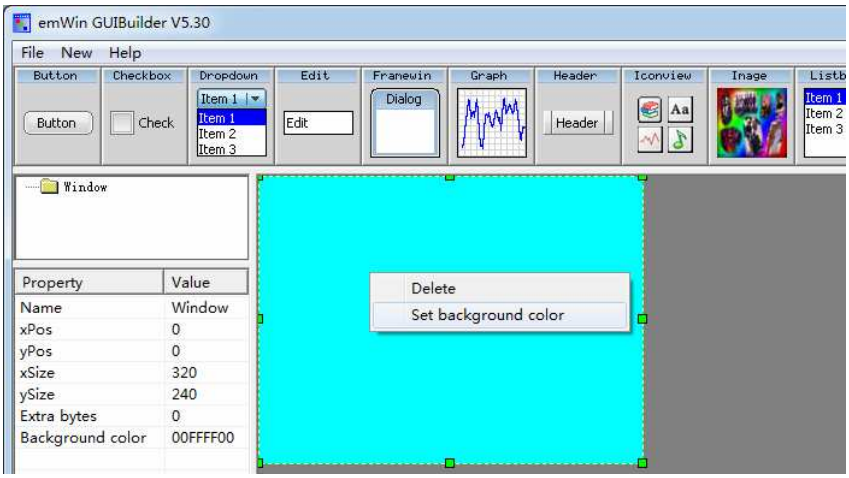
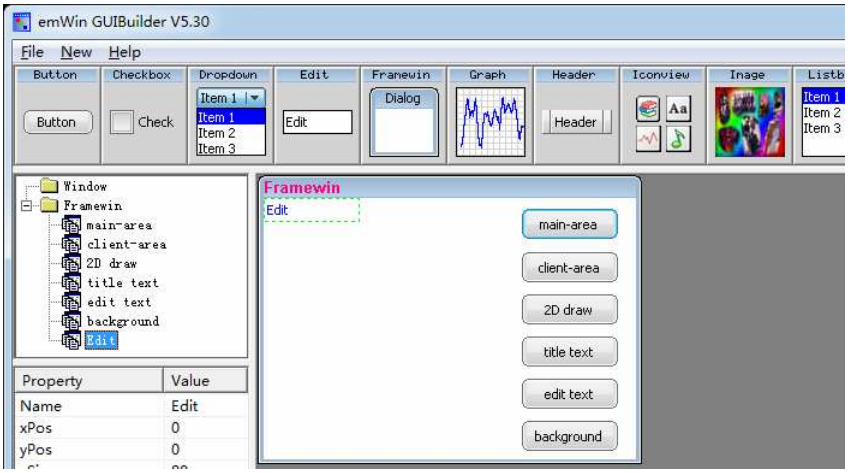
emwin 2-day quick tutorial 002_Framewin 窗口控件及其 Client 区以及 2D 绘图

(1)emWin 的每个界面都需要一个 Framewin/Window 窗口控件作为父窗体，就好像 Framewin/Window 是一个窗体框架，所有控件都放置在这个框架当中,组成父窗体和子控件的连带关系,可以通过父窗体找到子控件,也可以通过子控件找到它的父窗体,Framewin 和 Window 在外观上是有区别的:





(2)Window 窗口控件使用比较简单,就好像一个背景一样,也没有所谓的 Client 区的概念,当然功能和 Framewin 是一样的,只是外观不同而已;这里重点说明 Framewin 窗口控件的使用方法:

上图 Framewin 窗口控件中绿色部分是 Client 区域即客户区,以下是 emwin 说明书关于 Framewin 及其 Client 区的说明:

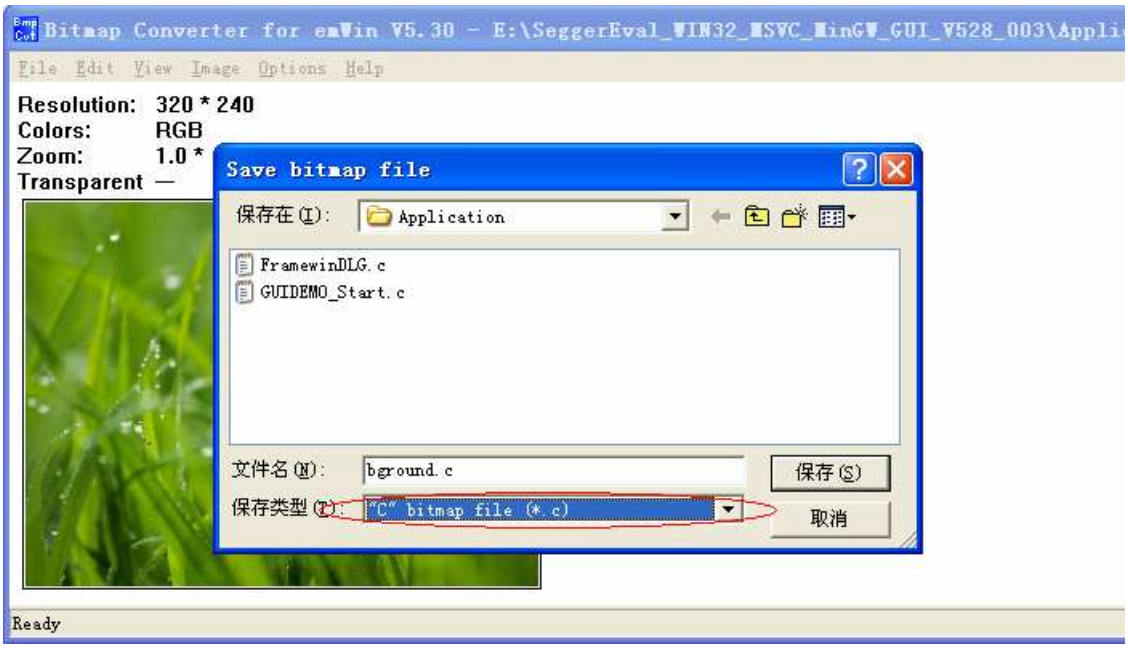
The frame window actually consists of 2 windows; the main window and a child window. The child window is called Client window. It is important to be aware of this when dealing with callback functions:There are

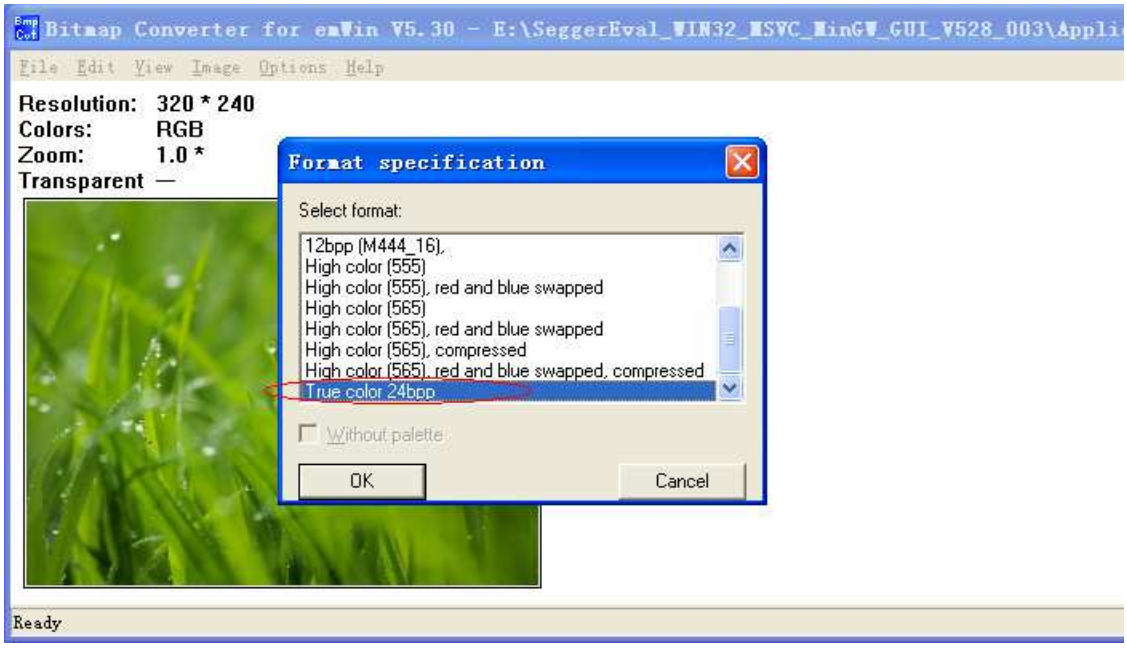
2 windows with 2 different callback functions. When creating child windows, these child windows are typically created as children of the client window; their parent is therefor the client window.
Framewin 窗口控件实际上包含 2 个窗口:主窗口和子窗口,子窗口称为 Client 窗口,在处理回调函数时理解主窗口和 Client 窗口的概念是很重要的:两个窗口可以对应两个不同的回调函数;当创建例如按键、编辑框等控件时,这些控件是作为 Client 窗口的子控件包含在 Client 窗口中的,因此这些控件的父窗口是 Client 窗口而非主窗口.

(3)在 emWin 中,我们要尽量使用控件,比如要在界面某个地方显示文字和图片,我们宁愿使用 Text 和 Image 控件也不要使用 2D 绘图方式显示文字和图片,这是由于使用控件管理起来会方便很多.但有时候必须用到 2D 绘图,比如画线、画圆、画矩形等等,可以在 WM_PAINT 消息中用 API 函数实现,WM_PAINT 消息请看源码注释或看教程 000.
值得注意的是,在程序其他地方修改了绘图的内容,需要使该 2D 绘图区域无效才行,否则 emWin 是检测不到变化也不去刷新该区域,程序自然不会跑到 WM_PAINT,如果使用的是控件,就不需要我们操心了,emWin 会自己处理,这就是前面说的使用控件管理起来会方便很多的原因. 使某 2D 绘图区域无效的 API 函数有三个:

```
void WM_InvalidateWindow(WM_HWIN hWin);//使指定的控件或整个窗体区域无效
void WM_InvalidateArea(const GUI_RECT * pRect);//使指定的矩形区域无效,相对于整个屏幕来说
void WM_InvalidateRect(WM_HWIN hWin, const GUI_RECT * pRect);//使指定的控件或窗体的某个矩形区域无效,相对于某个控件或窗体来说
```

(4)背景图片显示,用 BmpCvt.exe 工具(在 Tool 目录下)将 BMP 图片转成 C 文件,然后将此 C 文件加入 emWin 工程,并在 WM_PAINT 窗口重绘消息中使用 2D 绘图函数 GUI_DrawBitmap()显示图片即可,这个是背景图片一般显示方法,在某个位置显示图片还可以使用 Image 控件显示更为方便.另外 2D 绘图函数还有画线、画圆、画矩形、显示文字(不是控件上面的文字)等等.
如果是 STM32 平台,图片数据存到哪里?怎样读取图片数据?储存数据可以考虑 NAND-FLASH、SPI-FLASH、NOR-FLASH 等等...,至于怎样读取图片数据,这是 emWin 底层驱动程序的事情,在这里我们只说怎样做 emWin 的界面应用程序而已.




Bitmap Converter for emWin V5.30 - E:\SeggerEval_WIN32_MSVC_MinGW_GUI_V528_003\Appli
File Edit View Image Options Help
Resolution: 320 * 240
Colors: RGB
Zoom: 1.0 *
Transparent —
Format specification
Select format:
12bpp (M444_16),
High color (555)
High color (555), red and blue swapped
High color (565)
High color (565), red and blue swapped
High color (565), compressed
High color (565), red and blue swapped, compressed
True color 24bpp
Without palette
OK
Cancel
Ready


Framewin
main-area
maxi-size
Hello!
main-area
client-area
2D draw
title text
edit text
background
client-area
NVIDIA
The Digital Media F