

demo 工程中的 emWin 版本更新方法

Updating emwin version of demo project

目前 emWin 分以下几个版本 Currently emWin has the following versions:

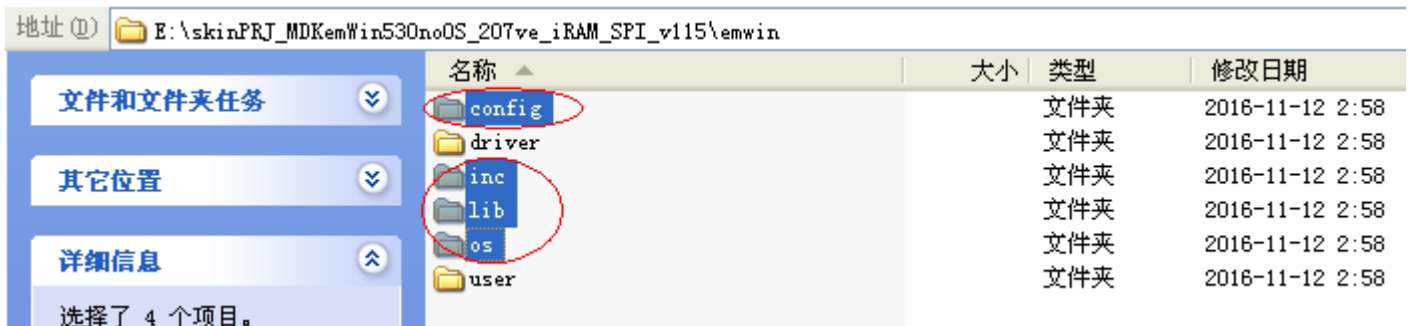
- 1) X86 版本, 在电脑上基于 CodeBlocks/VC 开发环境的模拟仿真界面程序.
 - 2) ST 版本 (STemWin), 只能运行于 ST 的单片机 (比如 STM32 系列), 在 ST 单片机上无需授权免费使用.
 - 3) NXP 版本 (NXPemWin), 能运行于所有单片机, 在 NXP 单片机上无需授权免费使用.
 - 4) MDK 版本 (MDKemWin), emWin 通用版, 位于 MDK 软件 \ARM\Pack\Keil\MDK-Middleware 目录下, 能运行于所有单片机, 授权情况不太清楚.
- 1) X86 version, simulation interface program based on CodeBlocks/VC development environment on computer.
 - 2) ST version (STemWin), can only run on ST microcontrollers (such as STM32 series), on ST microcontrollers without authorization and free use.
 - 3) NXP version (NXPemWin), can run on all MCUs, in the NXP MCUs without authorization for free use.
 - 4) MDK version (MDK emWin), emWin general version, located in the MDK software \ARM\Pack\Keil\MDK-Middleware directory, can run on MCUs, the authorization situation is not clear.

若 GLCD 提供的演示例程中 emWin 版本并不适合客户使用, 可以通过以下方法更新其他版本的 emWin 图形库 (无 RTOS 操作系统):

If the emWin version of the demo provided by GLCD is not suitable for customers, other versions of emWin (without RTOS operating system) can be updated by the following methods:

1. 在下面目录中, 删除旧版本并复制新版本的所有文件 In the following directory, delete the old version and copy all the files of the new version:

- [1] inc 文件夹下面的文件在不同版本 emWin 中可能有差异, 全部删除旧的, 并将新的复制过来即可.
 - [2] os 文件夹下面的文件对于无 RTOS 操作系统的应用应该使用 GUI_X.c 才对, 但编译通不过, 因此使用了 GUI_X_Ex.c, 实测暂时没有问题.
- [1] The files under the Inc folder may differ in different versions of emWin. Just delete old and copy the new.
 - [2] The files under the OS folder should use GUI_X.c for applications without RTOS operating system, but the compilation is not complete, so the GUI_X_Ex.c is used, and the actual test has not found any problems for the time being.



2. 修改以下文件 Modify the following files:

2.1 GUIConf. c:

```
#include "GUI.h"

/*****
 *
 *      Defines
 *
 *****/
//
// Define the available number of bytes available for the GUI
//
#define GUI_NUMBYTES (1024*64)

/*****
```

2.2 GUIConf. h:

```
#ifndef GUICONF_H
#define GUICONF_H

/*****
 *
 *      Multi layer/display support
 */
#define GUI_NUM_LAYERS 1 // Maximum number of available layers

/*****
 *
 *      Multi tasking support
 */
#define GUI_OS 0 // Compile with multitasking support

/*****
 *
 *      Configuration of touch support
 */
#define GUI_SUPPORT_TOUCH 0 // Support a touch screen (req. win-manager)

/*****
 *
 *      Default font
 */
#define GUI_DEFAULT_FONT &GUI_Font6x8

/*****
 *
 *      Configuration of available packages
 */
#define GUI_SUPPORT_MOUSE 0 // Mouse support
#define GUI_WINSUPPORT 1 // Use Window Manager
#define GUI_SUPPORT_MEMDEV 0 // Use Memory Devices
#define GUI_SUPPORT_DEVICES 0 // Enable use of device pointers

#endif // Avoid multiple inclusion
```

2.3 LCDConf.c:

```
// The display size should be adapted in order to match the size of
// the target display.
//
#define XSIZE_PHYS 800
#define YSIZE_PHYS 480

//
// Color conversion
// The color conversion functions should be selected according to
// the color mode of the target display. Detailedds can be found in
// the chapter "Colors" in the emWin user manual.
//
#define COLOR_CONVERSION GUICC_M888

//
// Display driver
// GUIDRV_WIN32 is for use only within the emWin Simulation
// environment. In order to use the target display controller, the
// according emWin display driver should be configured as it is
// described in the chapter "Display Drivers" in the emWin user
// manual. Beyond that sample configuration files can be found in
// The folder "Sample\LCDConf\%DISPLAY_DRIVER%\".
//
#define DISPLAY_DRIVER &GUIDRV_Template_API

/*****
*
* Configuration checking
*/
```

2.4 GUI.h:

```
extern const GUI_BITMAP_METHODS GUI_BitmapMethodsTR;
#define GUI_DRAW_BMPGL &GUI_BitmapMethodsGL
#define GUI_DRAW_BMPTR &GUI_BitmapMethodsTR

#endif /* ifndef GUI_H */

/***** End of file *****/
```

2.5 GUI_Type.h:

```
DECLARE_FONT(PROP_GL);

extern const tGUI_ENC_APIList GUI_ENC_APIList_GB2312;
extern const tGUI_ENC_APIList GUI_ENC_APIList_GBK;
extern const tGUI_ENC_APIList GUI_ENC_APIList_BIG5;
extern const tGUI_ENC_APIList GUI_ENC_APIList_UTF8;

typedef struct GUI_FONT_PROP_GL {
    U16P First; //First character
    U16P Last; //Last character
    const unsigned char GUI_UNI_PTR * pXsize; //Address of first character
    const struct GUI_FONT_PROP_GL GUI_UNI_PTR *pNext; //Pointer to next
} GUI_FONT_PROP_GL;

#define GUI_FONTTYPE_PROP_GL \
    GUIPROP_GL_DispChar, \
    GUIPROP_GL_GetCharDistX, \
    GUIPROP_GL_GetFontInfo, \
    GUIPROP_GL_IsInFont, \
    (GUI_GETCHARINFO *)0, \
    &GUI_ENC_APIList_UTF8, \
    //&GUI_ENC_APIList_GB2312, \
    //&GUI_ENC_APIList_GBK, \
    //&GUI_ENC_APIList_BIG5, \
    //&GUI_ENC_APIList_UTF8

typedef struct GUI_FONT_GL {
    GUI_DISPCHAR * pfDispChar;
    GUI_GETCHARDISTX * pfGetCharDistX;
    GUI_GETFONTINFO * pfGetFontInfo;
    GUI_ISINFONT * pfIsInFont;
    GUI_GETCHARINFO * pfGetCharInfo;
    const tGUI_ENC_APIList* pafEncode;
    U8 YSize;
    U8 YDist;
    U8 XMag;
    U8 YMag;
    union {
        const void GUI_UNI_PTR * pFontData;
        const GUI_FONT_MONO GUI_UNI_PTR * pMono;
        const GUI_FONT_PROP_GL GUI_UNI_PTR * pProp;
        const GUI_FONT_PROP_EXT GUI_UNI_PTR * pPropExt;
    } p;
    U8 Baseline;
    U8 LHeight; //Height of a small lower case character (a,x)
    U8 CHeight; //Height of a small upper case character (A,X)
    U8 Encode; //1:gb2312 2:gbk 4:big5 8:unicode(monospaced) 16:u
    const U8 GUI_UNI_PTR * pFont; //Library of [ascii]:[gb2312/gbk/big5/unicode/utf-
} GUI_FONT_GL;

#endif /* GUITYPE_H_INCLUDED */

/***** End of file *****/
```