uC/GUI 在 Cortex-M3 内核上的移植

屈环宇--嘉兴学院

一、简介

是一种专为嵌入式系统设计的图形界面支持系统。它的代码全部由标准 C 编写，模块化的设计，具有很强的可移植性。uC/GUI 适应大多数的黑白或彩色 LCD 的应用，还提供一个可扩展的 2D 图形库及占用极少 RAM 的窗口管理体系。

二、 要求

目标系统(硬件)

你的目标系统必须具备如下几点：

[1].CPU(8/16/32/64 位)

[2].必要的 RAM 和 ROM 存储

[3].LCD 显示器(任何类型及分辩率的)

对于内存的需求取决于你选用的 UCGUI 的功能模块以及你所使用的目标系统上的编译器的效率。 内存的占用量无法估计准确的值， 下面就一些的数值适用于多数的目标系统。

小型系统(不含窗口管理功能)

[1].RAM:100 字节

[2].堆栈:500 字节

[3].ROM:10~25K(取决于选用的 UCGUI 功能模块)

大型系统(包含窗口管理及各种窗体控件功能)

[1].RAM: 2-6 kb (决于选用的应用中建立窗口的数量)

[2].堆栈: 1200 bytes

[3].ROM: 30-60 kb (决于选用的 UCGUI 功能模块)

还要注意 ROM 的需求量随着你在应用程序中使用的字体数目而增长,以上的所有值都是粗糙的估计,，并不准确。

三、移植前的概述

本次选用的目标系统是基于 cortex-M3 内核的 stm32f103rb 微处理器。选用的是 uC/GUI3.90a 版本。LCD 是 ILI93XX 控制的 TFT 彩色液晶显示屏。

打开 UCGUI390a\GUI，以下列出各个文件夹的作用

表格 uc/GUI 目录结构

Config	配置文件目录
GUI\AntiAlias	搞锯齿支持
GUI\ConvertMono	灰度色彩转换支持
GUI\ConvertColor	色彩转换支持
GUI\Core	核心文件
GUI\Font	字体文件
GUI\JPEG	JPEG 格式图像显示支持
GUI\LCDDriver	LCD 驱动文件
GUI\MemDev	存储设备支持
GUI\MultiLayer	多层画图支持
GUI\Widget	构件库
GUI\WM	窗口管理器

其中 AntiAlias、ConvertMono、ConvertColor、MemDev、Widget、WM 均是可选的配置组件。μc/GUI 提供一些配置选项可在编译时排除某些组件，但是有时候需要手动将 C 文件从工程文件中移除以避免编译文件。最接近用户使用的组件为构件库、窗口管理器、LCD 驱动以及字体文件。

四、uC/GUI 接口函数的设计

移植 uC/GUI 的前提是 LCD 的驱动函数已经设计完成，因为在嵌入式体系结构中，uC/GUI 所在的是操作系统层，其与硬件的交互是通过设备驱动层来实现的。

uC/GUI 与设备驱动层的接口是以下3个函数

```
Void    LCD_DrawPoint(u16    x, u16    y, u16    color);
U16    LCD_ReadPoint(u16    x,u16    y);
```

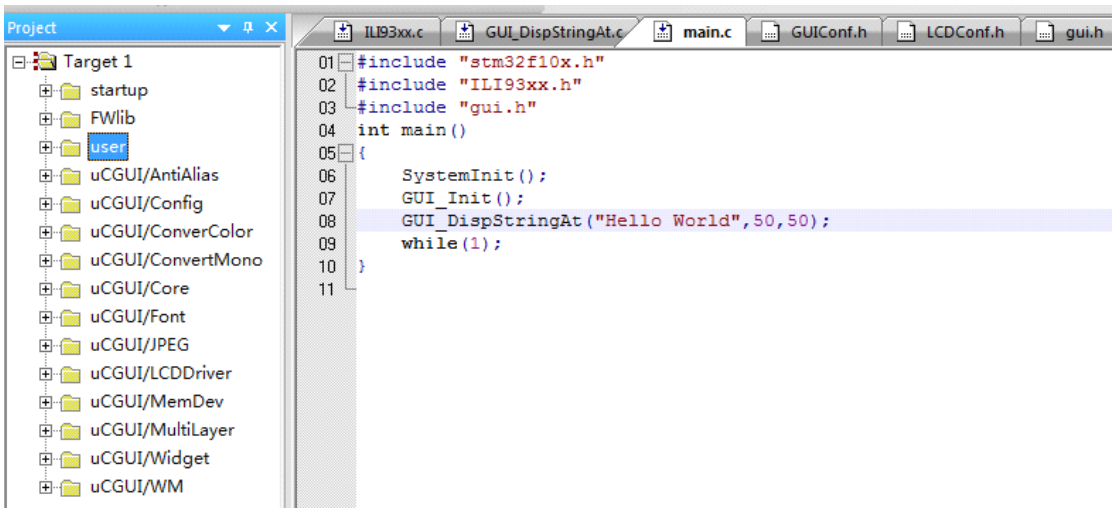
Void　LCD_MyInit();

注意 LCD 的初始化函数名不能用 LCD_Init(); uC/GUI 中已有如此定义的函数，会导致重定义。

接口函数的设计是跟处理器和 LCD 相关的,在移植 uC/GUI 前必须完成以上 3 个函数的设计，即能用 LCD_MyInit() 实现 LCD 的初始化，LCD_ReadPoint(u16　x,u16　y)返回对应点的颜色，LCD_DrawPoint(u16　x, u16　y, u16　color)在 LCD 上显示对应点的颜色。

五、uC/GUI 的移植

将 uGUI 文件夹中的所有文件都加入到工程如图所示



Startup 文件夹中包含处理器的启动文件及内核文件

FWLib 文件夹中包含固件库文件

User　文件夹中包含主函数文件及相关的用户定义的文件

细心的朋友会发现，多了一个 uGUI/Config 文件夹。这个就是 uC/GUI 相关的配置文件。其中包含了以下 3 个文件

GUI_X.c　　UCGUI390a\Sample\GUI_X

由于 uC/GUI 提供了与 uC/OS 相应的接口函数，仅移植 uC/GUI 会因为有些函数是不存在而导致编译错误，GUI_X.c 的作用就是申明这些函数防止编译错误。

```
U32 GUI_X_GetTaskId(void){return 0;}
void GUI_X_Lock(void){;}
void GUI_X_Unlock(void){;}
void GUI_X_InitOS (void){;}
void GUI_X_Log         (const char *s) { GUI_USE_PARA(s); }
void GUI_X_Warn        (const char *s) { GUI_USE_PARA(s); }
void GUI_X_ErrorOut(const char *s) { GUI_USE_PARA(s); }
```

在 GUI_X.c 中加入这些函数的申明防止编译错误

GUIConf.h　　UCGUI390a\Start\Config

此文件用于 uC/GUI 的相关应用配置，介绍如下

```
#ifndef GUICONF_H
#define GUICONF_H

#define GUI_OS                                  (0)   //不支持操作系统
#define GUI_SUPPORT_TOUCH                       (0)   //不支持触摸屏
#define GUI_SUPPORT_UNICODE                     (1)   //支持 ASCII 编码

#define GUI_DEFAULT_FONT                 &GUI_Font6x8   //字体大小
#define GUI_ALLOC_SIZE                   5000        //提供给窗口管理的内存大小

#define GUI_WINSUPPORT              1   // Window manager package available
#define GUI_SUPPORT_MEMDEV          1   //Memory devices available
#define GUI_SUPPORT_AA              1   // Anti aliasing available
#endif   /* Avoid multiple inclusion */
```

LCDConf.h　　UCGUI390a\Start\Config

此文件用于 LCD 的相关应用配置，介绍如下

```
#ifndef LCDCONF_H
#define LCDCONF_H
#define LCD_XSIZE               (240)
#define LCD_YSIZE               (320)
#define LCD_CONTROLLER          (9320)      //控制器编号
#define LCD_BITSPERPIXEL        (16)        //16 位点显示格式
#define LCD_FIXEDPALETTE        (565)       //对应红绿蓝为 565 位
#define LCD_SWAP_RB             (1)
#define LCD_INIT_CONTROLLER()   LCD_Init();
#endif /* LCDCONF_H */
```

接下来就是函数接口的匹配了，打开 GUI\LCDDriver 中的 LCDDummy.c 文件。

将其中的 LCD_L0_Init 函数定义如下

```
int    LCD_L0_Init(void)
{
   LCD_MyInit();
   return 0;
}
```

将其中的 LCD_L0_SetPixelIndex 函数定义如下

```
void (int x, int y, int PixelIndex)
{
   LCD_DrawPoint(x,y,PixelIndex);
}
```

将其中的 LCD_L0_GetPixelIndex 函数定义如下

```
unsigned int LCD_L0_GetPixelIndex(int x, int y)
{
   return LCD_ReadPoint(x,y);
}
```

至此，uC/GUI 移植完毕。如上图所示执行程序后，在 LCD 的(50,50)处将显示 "Hello World"，即表示 uC/GUI 移植移植成功。

六、总结

uC/GUI 的移植不需要对 Cortex-M3 内核及其中断机制有所了解，也不会涉及到硬件底层需要汇编实现的部分。难点在于设备驱动函数的编写及 GUI_X.c 中与 uC/OS 的接口设计。

这是今年暑假移植的，刚刚翻译完 uC/OS-III 用户手册，并移植了 uC/OS-III 到 stm32 处理器上，我就想把 uC/GUI 也放到网上去，希望能帮到大家。

——屈环宇

——2011 年 11 月 6 号晚