



卓越工程师培养计划

■ 嵌入式系统 ■

白永斌 张汉光 黄 强 编著



基于STemWin

的 STM32
开发与实践



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

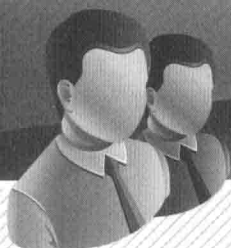


卓越工程师培养计划

■ 嵌入式系统 ■

<http://www.phei.com.cn>

白永斌 张汉光 黄 强 编著



基于STM32

的STM32
开发与实践

电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

本书全面、深入地介绍了 STemWin 图形用户界面系统 (GUI) 的强大功能和使用方法。内容涵盖 STemWin 的基础知识、移植方法, BMP、JPEG、GIF、PNG 等图片文件的解码显示和所有控件的使用方法, 并提供了完善的汉字显示方案。本书采用“理论+实验”的编写方式, 配套大量的实验例程, 图文并茂, 内容通俗易懂, 可以帮助读者快速掌握 STemWin 的应用技术。

本书适合从事软件研发、GUI 研究的工程技术人员阅读使用, 也可作为高等学校相关专业及培训机构的教学用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

基于 STemWin 的 STM32 开发与实践 / 白永斌, 张汉光, 黄强编著. —北京: 电子工业出版社, 2015.5

(卓越工程师培养计划)

ISBN 978-7-121-25972-2

I. ①基… II. ①白… ②张… ③黄… III. ①微控制器 IV. ①TP332.3

中国版本图书馆 CIP 数据核字 (2015) 第 089443 号

策划编辑: 张 剑 (zhang@phei.com.cn)

责任编辑: 徐 萍

印 刷: 北京京科印刷有限公司

装 订: 北京京科印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 25.5 字数: 653 千字

版 次: 2015 年 5 月第 1 版

印 次: 2015 年 5 月第 1 次印刷

印 数: 3 000 册 定价: 68.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

随着智能化时代的到来，越来越多的电子产品将人机交互作为产品核心的一部分。为了适应这一发展趋势，很多在校学生和电子开发人员开始学习 GUI 的相关知识和应用。但是，目前关于 GUI 方面的书籍很少，而且相关书籍的理论知识过多，实践和实际应用内容较少。本书介绍的 GUI 采用的是 SEGGER 公司授权给 ST 公司的 STemWin。这款 GUI 非常适用于小型嵌入式系统，而且如果用户使用的是 ST 公司的 MCU，那么就可以免费使用 STemWin。本书的大部分内容都是作者实际工作经验的总结，通过本书的学习，读者可以由浅入深地掌握 STemWin 的移植方法，BMP、JPEG、GIF、PNG 等图片文件的解码显示，完善的汉字显示方案以及 STemWin 大部分控件的使用方法，从而在实践中提高自身的实验能力、实际操作能力、独立分析和解决问题的能力。

本书的主要特点是理论和实践相结合，着重于实践，以尽量简单的语言和翔实的步骤讲述 STemWin 的重点，由简到繁地揭开 STemWin 的神秘面纱，使读者对 STemWin 有一个较全面的认识，并基本掌握 STemWin 的应用。另外，本书所有的实例代码都可以通过我们的网站 www.armfly.com 获取。

本书由白永斌、张汉光和黄强编著。在编写过程中，宋亭亭对本书的语法和截图进行了反复的修改，在此表示衷心的感谢！同时，还要感谢武汉安富莱电子有限公司的所有工作人员。

由于作者水平有限，本书在某些方面难免有疏忽、不恰当甚至错误的地方，恳请各位同行提出批评和建议。

编著者

目 录

第 1 章 STemWin 基础知识	1
1.1 STemWin 概述	1
1.1.1 STemWin、emWin、μCGUI 之间的关系	1
1.1.2 SEGGER 公司介绍	2
1.1.3 STemWin 介绍	3
1.1.4 STM32F103 和 STM32F407 跑 STemWin 性能测试	8
1.2 STemWin5.xx 的移植	11
1.2.1 关于 STemWin 移植的重要说明	11
1.2.2 移植前的准备工作	12
1.2.3 STemWin 的裸机移植	13
1.2.4 STemWin 带 RTOS 的移植	22
1.3 RA8875 屏触摸校准自适应	23
1.3.1 触摸校准基础知识介绍	23
1.3.2 触摸屏校准步骤	23
1.3.3 触摸参数自动加载	24
1.4 μCGUIBulder4.0 和 GUIBulder5.22 的使用方法	25
1.4.1 GUIBulder5.22 的使用步骤	25
1.4.2 μCGUIBulder4.0 的使用	29
1.4.3 将生成的代码移植到 STM32-V5 开发板	30
1.5 emWin5.24 模拟器与查看器	30
1.5.1 emWin5.24 模拟器	30
1.5.2 emWin5.24 查看器	32
第 2 章 STemWin 文本与数值显示	34
2.1 文本显示	34
2.1.1 简单文本显示	34
2.1.2 文本显示模式	35
2.1.3 GUI_DispStringInRectWrap 函数的使用	36
2.1.4 文本显示实例——自动换行的实现	37
2.2 数值显示	38
2.2.1 十进制数的显示	39
2.2.2 十六进制数和二进制数的显示	43
2.2.3 浮点数的显示	43
第 3 章 2D 图形库	45
3.1 基本绘图	45
3.1.1 画笔设置	45

3.1.2	基本绘制	46
3.1.3	Alpha 混合	46
3.2	绘制流位图	49
3.2.1	生成 C 流文件	49
3.2.2	移植到开发板	50
3.3	绘制图形	53
3.3.1	绘制多边形	54
3.3.2	绘制圆	56
3.3.3	绘制椭圆	57
3.3.4	绘制弧线	57
3.3.5	绘制线图	58
3.3.6	绘制饼图	59
第 4 章	STemWin 图片显示	60
4.1	BMP 图片显示	60
4.1.1	BMP 图片支持	60
4.1.2	绘制已经加载到存储器的 BMP 图片	63
4.1.3	绘制无须加载到存储器的 BMP 图片	66
4.2	JPEG 图片显示	69
4.2.1	JPEG 图片支持	70
4.2.2	绘制已加载到存储器的 JPEG 图片	71
4.2.3	绘制无须加载到存储器的 JPEG 图片	73
4.3	GIF 图片显示	75
4.3.1	GIF 图片支持	75
4.3.2	绘制已加载到存储器的 GIF 图片	78
4.3.3	绘制无须加载到存储器的 GIF 图片	80
4.4	PNG 图片显示	83
4.4.1	PNG 图片支持	83
4.4.2	绘制已加载到存储器的 PNG 图片	85
4.4.3	绘制无须加载到存储器的 PNG 图片	87
第 5 章	STemWin 字符显示	89
5.1	STemWin 支持字体	89
5.1.1	字体类型	89
5.1.2	字符集	91
5.1.3	标准字体	93
5.2	STemWin 汉字显示	96
5.2.1	汉字显示基础知识	96
5.2.2	汉字显示方式	99
5.3	STemWin 外语支持	116
5.3.1	Unicode	116

5.3.2	阿拉伯语支持	121
5.3.3	泰语支持	123
5.3.4	Shift JIS 支持	125
5.4	外置字体 (XBF)	125
5.4.1	XBF 格式字体生成方法	126
5.4.2	XBF 移植	126
5.5	系统独立字体 (SIF)	131
5.5.1	SIF 格式字体生成方法	131
5.5.2	SIF 移植	132
5.6	矢量字体 (TrueType)	136
5.6.1	矢量字体概述	136
5.6.2	STemWin 对矢量字体的支持	137
5.6.3	在模拟器上运行矢量字体	137
5.6.4	矢量字体移植	141
第 6 章	STemWin 光标	145
6.1	STemWin 支持的光标效果	145
6.2	在模拟器上演示光标	145
6.3	在模拟器上演示动态光标	149
第 7 章	STemWin 支持的颜色	150
7.1	预定义颜色	150
7.2	固定调色板及其说明	151
7.3	在模拟器上演示颜色条测试例程	152
第 8 章	Sprite	155
8.1	Sprite 概述	155
8.2	在模拟器上演示 Sprite 实例	155
第 9 章	STemWin 抗锯齿 (AA) 技术	159
9.1	抗锯齿 (AA) 基础知识	159
9.1.1	抗锯齿质量	159
9.1.2	抗锯齿字体	159
9.1.3	高分辨率坐标	160
9.2	抗锯齿 API 函数实例演示	161
9.3	抗锯齿 (AA) 实例	166
9.3.1	实例 1——AA_HiResAntialiasing	166
9.3.2	实例 2——AA_HiResPixels	169
9.3.3	实例 3——AA_Lines	172
第 10 章	STemWin 多任务设计	176
10.1	模拟器多任务设计	176
10.1.1	单任务系统 (超级循环)	176
10.1.2	多任务系统: 一个任务调用 emWin	177

10.1.3	多任务系统：多个任务调用 emWin	177
10.1.4	官方推荐运行方式	178
10.1.5	在模拟器上运行多任务	178
10.2	μCOS-III 多任务设计	183
10.2.1	单任务系统（超级循环）	183
10.2.2	多任务系统：一个任务调用 emWin	188
10.2.3	多任务系统：多个任务调用 emWin	193
第 11 章 STemWin 输入设备		200
11.1	指针输入设备（摇杆）	200
11.1.1	概述	200
11.1.2	指针输入设备（API）	200
11.1.3	五向摇杆操作光标	201
11.2	键盘输入（实体按键操作控件）	204
11.2.1	概述	204
11.2.2	驱动层 API	204
11.2.3	实体按键操作	205
第 12 章 STemWin 窗口管理器		210
12.1	回调机制	210
12.1.1	术语说明	210
12.1.2	回调机制、无效化和渲染	211
12.2	消息类型	213
12.2.1	消息结构	213
12.2.2	消息类型	214
12.2.3	系统类型消息实例	215
12.2.4	指针输入设备（PID）类型消息实例	219
12.2.5	通知代码类型消息实例	223
12.3	定时器使用	227
12.3.1	定时器功能介绍	227
12.3.2	在桌面窗口上使用定时器实例	227
12.3.3	在对话框上使用定时器实例	229
12.4	ToolTips 的使用	233
12.4.1	ToolTips 概述	233
12.4.2	ToolTips 实例	233
12.5	窗口管理器实例	238
12.5.1	用户自定义消息类型实例	238
12.5.2	桌面窗口回调函数实例	242
12.5.3	WM_Redraw.c 实例	247
12.5.4	WM_Sample.c 实例	250
12.5.5	WM_LateClipping.c 实例	255

12.5.6	WM_Video.c 实例	260
第 13 章	STemWin 控件	266
13.1	控件基础知识	266
13.1.1	当前支持的控件	266
13.1.2	控件的重绘机制	266
13.1.3	控件使用方法	266
13.1.4	控件配置选项	267
13.1.5	通用控件 API	268
13.1.6	常用函数	269
13.2	对话框基础知识	270
13.2.1	对话框概述	270
13.2.2	创建对话框	271
13.2.3	对话框 API	274
13.3	按钮控件	274
13.3.1	按钮控件概述	275
13.3.2	按钮控件实例	275
13.3.3	在对话框上使用按钮控件	282
13.4	复选框控件	286
13.4.1	复选框控件概述	286
13.4.2	复选框控件实例——WIDGET_CheckBox 实例	287
13.5	下拉列表控件	291
13.5.1	下拉列表控件概述	291
13.5.2	下拉列表控件实例——WIDGET_Dropdown 实例	292
13.5.3	在对话框上使用下拉列表控件	297
13.6	编辑控件	301
13.6.1	编辑控件概述	301
13.6.2	编辑控件实例——WIDGET_Edit 实例	302
13.6.3	在对话框上使用编辑框控件	306
13.7	框架窗口控件	310
13.7.1	框架窗口控件概述	310
13.7.2	框架窗口控件实例——WIDGET_FrameWin 实例	311
13.8	图形控件	316
13.8.1	图形控件概述	316
13.8.2	图形控件实例——WIDGET_GraphYT 实例	318
13.8.3	使用 GUIBulder 建立 Graph 控件	327
13.9	标题控件	327
13.9.1	标题控件概述	327
13.9.2	标题控件实例——WIDGET_Header 实例	328
13.9.3	使用 GUIBulder 建立 HEADER 控件	334

13.10	列表框控件	335
13.10.1	列表框控件概述	335
13.10.2	列表框控件实例——WIDGET_ListBox 实例	335
13.10.3	在对话框上使用列表框控件	339
13.11	菜单控件	340
13.11.1	菜单控件概述	340
13.11.2	菜单控件实例——WIDGET_Menu 实例	342
13.11.3	使用 GUIBulder 建立 MENU 控件	348
13.12	消息框控件	349
13.12.1	消息框控件概述	349
13.12.2	消息框控件实例	349
13.13	多行文本控件	353
13.13.1	多行文本控件概述	353
13.13.2	多行文本控件实例——WIDGET_Multiedit 实例	354
13.13.3	使用 μ GUIBulder 建立多行文本控件	356
13.14	多页控件	357
13.14.1	多页控件概述	357
13.14.2	多页控件实例——WIDGET_Multipage 实例	358
13.14.3	使用 GUIBulder 建立多页控件	361
13.15	进度条控件	362
13.15.1	进度条控件概述	362
13.15.2	进度条控件实例——WIDGET_Progbar 实例	362
13.15.3	使用 μ GUIBulder 建立进度条控件	365
13.16	单选按钮控件	365
13.16.1	单选按钮控件概述	365
13.16.2	单选按钮控件实例——DIALOG_Radio 实例	366
13.16.3	使用 GUIBulder 建立单选按钮控件	369
13.17	滚动条控件	369
13.17.1	滚动条控件概述	369
13.17.2	滚动条控件实例——DIALOG_Scrollbar 实例	370
13.17.3	使用 GUIBulder 建立滚动条控件	373
13.18	滑块控件	373
13.18.1	滑块控件概述	373
13.18.2	滑块控件实例——DIALOG_SliderColor 实例	374
13.18.3	使用 GUIBulder 建立滑块控件	378
13.19	文本控件	379
13.19.1	文本控件概述	379
13.19.2	使用 GUIBulder 建立文本控件	379
13.20	窗口控件	379

13.20.1	窗口控件概述	380
13.20.2	窗口控件实例——WIDGET_Window 实例	380
13.21	SPINBOX 控件	381
13.21.1	SPINBOX 控件概述	381
13.21.2	SPINBOX 控件实例——WIDGET_Spinbox 实例	381
13.21.3	使用 GUIBulder 建立 SPINBOX 控件	384
13.22	LISTVIEW 控件	385
13.22.1	LISTVIEW 控件概述	385
13.22.2	LISTVIEW 控件实例——WIDGET_ListView 实例	386
13.22.3	使用 GUIBulder 建立 LISTVIEW 控件	391
13.23	在对话框上绘制 2D 图形的方法	393
附录 A	安富莱 STM32-V5 开发板套件	395
A.1	简介	395
A.2	硬件配置	395
A.3	文档教程	396
A.4	技术支持	396

第1章 STemWin 基础知识

本章主要介绍 STemWin 的基础知识，其实确切地讲应该叫 emWin 的基础知识，由于教程使用的开发板是 ST 的微控制器，所以就将其统一命名为 STemWin（为什么叫 STemWin，在下面会有详细的讲解）。



1.1 STemWin 概述

1.1.1 STemWin、emWin、μCGUI 之间的关系

这个放在最开头予以说明，是因为很多的初学者比较迷惑。

对于一些刚学 GUI 的用户来说，知道 μCGUI 的比较多，而不怎么知道 emWin 或者 STemWin。这个并不奇怪，主要是因为大部分人只知道 SEGGER 公司做的 JLINK，而不知道他们还有 RTOS 和相关的中间件（中间件的意思就是基于 RTOS 的文件系统、GUI、USB 主机和设备协议栈等）。

μCGUI 在国内前几年比较火的原因有三点。

- ③ μCOS-II 在国内的推广：自从 Micrum 公司出的那本《嵌入式实时操作系统 μCOS-II》发布之后，国内关于 μCOS-II 的资料可以说是铺天盖地，再加上各种培训机构和开发板的推广，μCOS-II 就在国内火起来了。μCOS-II 火的同时，它配套的中间件，特别是 μCGUI 就跟着在国内火了起来。
- ③ 国内的 μCGUI 论坛：这个论坛在国内的 μCGUI 方面应该算是做得最好的，特别是其站长在 μCGUI 方面的研究，这位站长对于 μCGUI 在国内的发展功不可没。
- ③ μCGUI 是带有源码的：很多时候可以通过修改部分的源码实现一些特殊的功能，现在网上流传的 μCGUI 的源码已经不知道经过多少人的手被修改过，最原始的代码已经在官网上找不到了。

【emWin 和 μCGUI 的关系】

首先要明白，emWin 和 μCGUI 这两个 GUI 是同一个东西。最初这个 GUI 是 SEGGER 公司的，然后以什么方式授权给 μCGUI 就不清楚了。现在 SEGGER 公司是这个 GUI 的主要推动者，已经将其授权给了多个芯片生产厂家。

【STemWin 和 emWin 的关系】

STemWin 是 SEGGER 公司授权给 ST（意法半导体）的，使用 ST 芯片的用户可以免费使用 STemWin。其实不光授权给了 ST，还有 NXP、Energy Micro 等。凡是使用这些芯片厂商生产的处理器都可以免费使用 emWin。但是出于一定的保护措施，使用 STemWin 的库是不能用在其他芯片厂商的处理器上的。因为在工程初始化 STemWin 前要使能 CRC

校验。如果没有使能, STemWin 是启动不起来的。KEIL MDK 的安装目录里也带有 emWin 软件包, 这个软件包也不可以直接使用, 用户需要给 KEIL MDK 注册 RL-ARM 才可以使用。

这里 STemWin 还针对 ST 的微控制器做了专门的优化, 比如在使用 ST 的 F4XX 微控制器带 FPU 的芯片时, STemWin 在需要浮点处理的地方专门做了优化。

【emWin5.xx 版本和以前版本的不同】

emWin 发展到 5.0 版本以后产生了很大的更新, 特别是底层驱动方面。emWin5.xx 版本向下完全地兼容低版本, 当然包括 μ CGUI5.xx 以下的版本, 也就是说如果用户有在 μ CGUI5.xx 以下版本建立的工程, 完全可以用在高版本上面(条件是没有修改过源码)。这里不建议初学者修改源码, 因为修改过后会破坏现有的机制, 在以后的使用中会养成不好的习惯, 只要某些功能无法实现就去修改源码, 随着修改的增多会严重地破坏现有的机制。

emWin5.xx 以后的版本只有库, 没有源码。对于一些想研究源码的, 可以看早期的版本, 了解一下通信机制。不过对于大多数从应用角度出发的, 完全没有必要学习源码, 源码内容太多。对于一些早期版本无法实现的功能, 在 emWin5.xx 上面得到了很大的改善, 基本不需要修改源码。如果通过各种方法仍然无法实现, 可以使用 emWin 支持的自定义控件设计做一个符合要求的。

1.1.2 SEGGER 公司介绍

SEGGER 公司应该算是一家老牌的调试工具和 RTOS 及其中间件的生产商。SEGGER 公司成立于 1997 年, 到 2015 年已经成立有 18 年了。这家公司主要有两个 Office, 一个是在德国的 Hilden, 另一个是在美国的 Massachusetts。官网还有一个工作环境的照片, 环境很好, 见图 1-1。



图 1-1 SEGGER 公司的工作场所

SEGGER 公司的产品主要有三个方向, 分别如下。

☺ RTOS 及其中间件: SEGGER 公司的 RTOS 是 embOS, 在国内知道的人可能比较少。SEGGER 做的 embOS 和中间件都是以库的形式供用户下载的, 除非购买了使用权。产品主要如下:

- ☞ embOS (Real Time Operating System)
- ☞ embOS/IP (TCP/IP Stack)
- ☞ emWin (Graphic Software & GUI)
- ☞ emFile (File System)
- ☞ emUSB Device (USB Device Stack)
- ☞ emUSB Host (USB Host Stack)
- ☞ emModbus (Modbus Stack)

emModbus 是 2014 年发布的。

☺ J-Link 调试工具: J-Link 大家应该都不陌生, 它是由 SEGGER 公司设计的。具体的

J-Link 产品有好几款, 如下所示:

- ☞ J - Link PRO ☞ J - Link
- ☞ J - Link ULTRA+ ☞ J - Trace Cortex-M3
- ☞ J - Link PLUS ☞ J - Trace ARM

☺ Production Programmers: 这个工具在国内用得比较少。主要有如下几款产品:

- ☞ Flasher ARM ☞ Flasher RX
- ☞ Flasher STM8 ☞ Flasher 5
- ☞ Flasher ST7 ☞ Flasher 5 PRO

上面的三项是 SEGGER 公司的主营产品, 更详细的内容可以去官网 www.segger.com 上了解。

1.1.3 STemWin 介绍

emWin5.xx 版本设计出来的界面还是非常漂亮的, 为了使大家有一些感性的认识, 下面先来欣赏几张相关的界面设计图片。

1. STemWin 设计界面

图 1-2 是官方提供的应用界面设计效果。

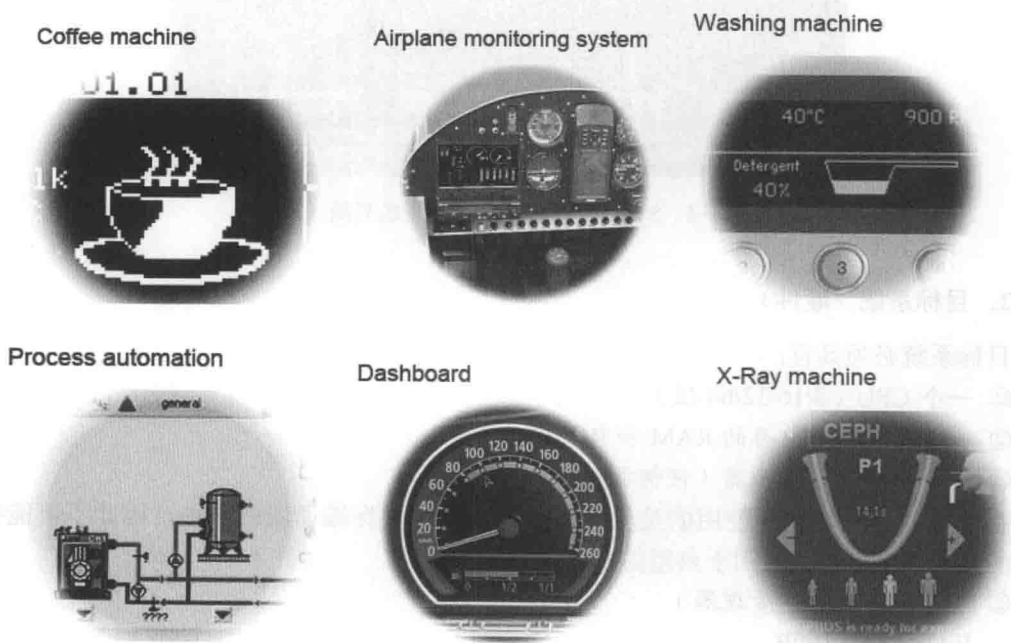


图 1-2 STemWin 设计界面

整体来说, 这些界面还是非常漂亮的, 不过这些界面不是用专门的控件显示出来的, 而是用 2D 绘图配合内存设备管理实现的。

☺ 在 STM32-V5 开发板上实现的界面: 如图 1-3 ~ 图 1-5 所示, 总的来说这些界面是非常漂亮的。关于 STM32-V5 开发板更详细的资料可以参考如下两个地址:

<http://bbs.armfly.com/read.php?tid=1285>

<http://bbs.armfly.com/read.php?tid=1139>



图 1-3 STM32-V5 开发板上实现的界面 (1)

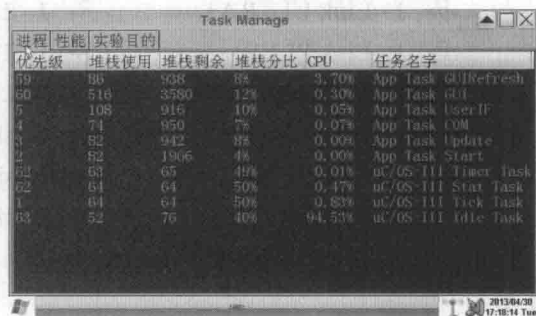


图 1-4 STM32-V5 开发板上实现的界面 (2)

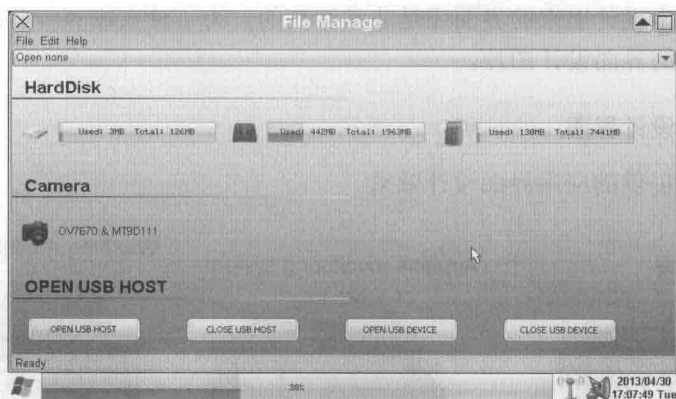


图 1-5 STM32-V5 开发板上实现的界面 (3)

2. 目标系统 (硬件)

目标系统必须具有:

- ☺ 一个 CPU (8/16/32/64 位)
- ☺ 一个具有最小内存的 RAM 和 ROM
- ☺ 一个完整图形显示器 (任何类型和任何分辨率)

存储器的要求取决于使用的是软件的哪部分以及目标编译器的效率, 因此不可能指定精确的值, 但是以下值适用于典型的系统。

- ☺ 小系统 (无窗口管理器)

- ☞ RAM: 100B
- ☞ 堆栈: 600B
- ☞ ROM: 10 ~ 25KB (取决于所使用的功能)

- ☺ 大系统 (包含窗口管理器和小工具)

- ☞ RAM: 2 ~ 6KB (取决于所需的窗口数)
- ☞ 堆栈: 1200 ~ 1800B (取决于所使用的功能)
- ☞ ROM: 30 ~ 60KB (取决于所使用的功能)

【注意】如果应用程序使用了很多字体,则对 ROM 的要求会提高。上述所有值都是粗略估算值,不保证能正常使用。

3. 特性

emWin 设计用于提供高效且独立于处理器和显示控制器的图形用户界面,用于任何使用图形显示进行操作的应用。它与单任务和多任务环境、专用操作系统或具有任何商业 RTOS 兼容,emWin 的发货形式为 C 语言源代码。它可适用于任何尺寸的、具有任何显示控制器和 CPU 的物理和虚拟显示器。其特性包括以下内容。

【一般特性】

- ☺ 支持使用任何控制器的任何(单色、灰度或彩色)显示器(如果有正确的驱动)。
- ☺ 在较小显示器上无显示控制器也可运行。
- ☺ 使用配置宏可支持任何接口。
- ☺ 显示尺寸可配置。
- ☺ 可在显示器上的任何点(而不仅仅是在偶数位字节地址)上写入字符和位图。
- ☺ 已针对尺寸和速度优化了各种例程。
- ☺ 利用编译时间切换可进行不同优化。
- ☺ 对于较慢的显示控制器,可在存储器中缓存显示,将存取操作减到最少,从而获得非常高的速度。
- ☺ 结构清晰。
- ☺ 支持虚拟显示;虚拟显示可大于实际显示。

【图形库】

- ☺ 支持不同色深的位图。
- ☺ 可使用位图转换器。
- ☺ 绝对无浮点使用。
- ☺ 快速线/点绘制(不使用浮点)。
- ☺ 圆形/多边形绘制速度非常快。
- ☺ 不同的绘图模式。

【字体】

- ☺ 基本软件配备多种不同的字体:4*6、6*8、6*9、8*8、8*9、8*16、8*17、8*18、24*32,以及像素高度为8、10、13、16的比例字体。
- ☺ 可以定义新的字体并只需简单链接。
- ☺ 只有应用程序使用的字体才实际链接到生成的可执行程序,从而使 ROM 使用最小。
- ☺ 字体可分别在 X 和 Y 方向完全缩放。
- ☺ 可使用字体转换器;主机系统(Microsoft Windows)上所有的任何字体都可以转换。

【字符串/值输出例程】

- ☺ 例程可以十进制、二进制、十六进制、任何字体形式显示值。
- ☺ 例程可以十进制、二进制、十六进制、任何字体形式编辑值。

【窗口管理器(WM)】

- ☺ 完整的窗口管理操作,包括裁剪。窗口客户区以外的区域不可能被覆盖。
- ☺ 窗口可以移动和调整大小。

- ☺ 支持回调例程（可选择是否使用）。
- ☺ WM 使用最小的 RAM（每个窗口大约 50 字节）。

【PC 界面外观的可选小工具】提供各种小工具（窗口对象，也称为控件）。它们通常自动操作并且简单易用。

【支持触摸屏和鼠标】对于按钮小工具等窗口对象，emWin 提供触摸屏和鼠标支持。

【PC 工具】

- ☺ 模拟脉冲查看器。
- ☺ 位图转换器。
- ☺ 字体转换器。

4. RTOS 对 STemWin 多任务的支持

基本上常见的 RTOS 都支持 STemWin 的多任务，移植起来也比较简单，下面这些是官方提供的移植，如图 1-6 所示。

名称	修改日期	类型	大小
GUI_VNC_X.c	2014/3/20 17:22	C Source file	3 KB
GUI_VNC_X_Keil.c	2014/4/16 9:57	C Source file	8 KB
GUI_VNC_X_StartServer.c	2014/3/20 17:22	C Source file	8 KB
GUI_X.c	2014/3/20 17:22	C Source file	4 KB
GUI_X_CMSIS_RTOS.c	2014/3/21 10:13	C Source file	6 KB
GUI_X_embOS.c	2014/3/20 17:22	C Source file	7 KB
GUI_X_Exc.c	2014/3/20 17:22	C Source file	5 KB
GUI_X_Joystick.c	2014/5/26 12:31	C Source file	4 KB
GUI_X_RTE.c	2014/3/21 10:13	C Source file	7 KB
GUI_X_RTX.c	2014/3/20 17:22	C Source file	6 KB
GUI_X_Touch.c	2014/7/9 12:24	C Source file	6 KB
GUI_X_Touch_Analog.c	2014/3/20 17:22	C Source file	3 KB
GUI_X_Touch_StoreState.c	2014/3/20 17:22	C Source file	4 KB
GUI_X_uCOS.c	2014/3/20 17:22	C Source file	7 KB
GUI_X_uEZ.c	2014/3/20 17:22	C Source file	5 KB

图 1-6 STemWin 的 RTOS 接口文件

5. STemWin 基本的内存需要（见表 1-1）

表 1-1 STemWin 基本的内存需要

	Example	Hello world	Window application
ROM	GUI core	5.9KB	43KB
	Fonts	1.8KB	12.5KB
	Application	38B	2.7KB
	Startup code	300B	300B
	Library	0.1KB	1.5KB
	Total	8.1KB	60KB
RAM	GUI core	62B	5.2KB
	Application	—	40B
	Stack	272B	1.4KB
	Total	334B	6.6KB