文章编号:1674-2869(2009)07-0084-04

事件驱动编程机制在嵌入式GUI系统中的实现

冉 全,陈 为,赵世平
(武汉工程大学 计算机科学与工程学院,湖北 武汉 430074)

摘 要:针对嵌入式GUI中使用事件驱动编程机制实现窗口和控件的开发进行了讨论,分析了自主开发的TBGUI的体系结构和特点,提供了一种完全实用的嵌入式系统中的类Windows图形用户界面. TBGUI适合以单片机为主的8位、16位嵌入式系统,也可以扩展到ARM等32位嵌入式平台上.

关键词:事件驱动;TBGUI;抽象层;控件
中图分类号:TP 368.1 文献标识码:A

0 引 言

GUI(Graphics User Interface),即图形用户界面,是用于提高人机交互友好性和易操作性的计算机程序,它是建立在计算机图形学基础上的产物.对于现代通用微机而言,GUI早已经是操作系统的一个组成部分,典型的如微软的Windows以及苹果的Mac OS操作系统,其GUI部分已经可以用华丽来形容,即使是开源的Linux系统,其所支持的GUI功能也已经非常强大.

随着嵌入式系统的日益发展,目标产品对GUI的需求越来越迫切,特别是高端嵌入式产品,如媒体播放器、游戏机、机顶盒、手机、GPS等,均对图形用户界面有较高的要求.一些需要大量人机对话实现操作的智能设备和仪器仪表也开始使用嵌入式GUI,以获得更好的使用效果.

受限于GUI所需求的硬件资源,一般认为,嵌入式GUI,只能在32位高档嵌入式系统上使用,如使用较多的MiniGUI和μCGUI主要应用在32位平台,而不能有效使用在8位嵌入式系统当中.国内具有自主软件著作权的ZLGGUI功能有限,只能实现简单的画图功能[1].因此,简要介绍一种自行研发的嵌入式GUI TBGUI,能在一片增强型8051单片机上运行,不用扩展ROM和RAM,不仅能支持单色LCD和键盘,还能支持TFT LCD和触摸屏,能实现和Windows相似的窗口和控件,其核心采用了事件驱动编程机制.

1 TBGUI的体系结构

TBGUI使用层次化的体系结构,通过抽象层(Abstract Layer)实现可移植性,移植TBGUI时只需要修改抽象层程序以及通用数据类型定义头文件(TG_TYPES.H)即可,TBGUI的体系结构如图1所示.

图1 TBGUI的体系结构

Fig.1 Architecture of TBGUI

1.1 运行时库

运行时库(Run-Time Library,简称RTL或者RT)由抽象层(Abstract Layer)和设备驱动程序(Drivers)两部分组成.RTL是TBGUI的底层,它独立于GUI的核心层和上层,使得TBGUI有良好的可移植性.抽象层的主要C语言模块和头文件如表1所示.

表1 抽象层文件列表

Table 1 Abstract layer file list

模块	源程序/头文件
图形抽象层	TG_GRAL.C/TG_GRAL.H
颜色抽象层	TG_CRAL.C/TG_CRAL.H
字体抽象层	TG_FNTAL.C/TG_FNTAL.H
键盘设备抽象层	TG_KDAL.C/TG_KDAL.H
指点设备抽象层	TG_PDAL.C/TG_PDAL.H
多任务抽象层	TG_MTAL.C/TG_MTAL.H

TBGUI通过抽象层实现可移植,对于不同的

收稿日期:2009-03-16
基金项目:国家自然科学基金资助项目(NO.60802046)
作者简介:冉 全(1968-),男,四川忠县人,副教授,硕士.研究方向:嵌入式系统应用与开发.

MCU 构建的嵌入式系统，移植 TBGUI 时只需要修改抽象层各模块源程序以及通用数据类型定义头文件(TG_TYPES. H)即可.

设备驱动程序包括 LED、LCD 显示驱动程序、键盘和触摸屏输入驱动程序等，可以在无需修改抽象层的情况下，使 TBGUI 在同一类型 MCU 的系统中支持不同的外设.

挂接于抽象层之下的设备驱动程序是可选的，如果不使用设备驱动程序，抽象层也可以直接驱动硬件设备. 这可以简化程序结构，代价是抽象层即使在同一类 MCU 的嵌入式系统中也不再具有通用性.

1.2　核心层

核心层(Core)是 TBGUI 的主体，负责文本和图形的处理以及实现事件驱动机制. 核心层主要 C 语言模块/头文件列表如表 2 所示.

表 2　核心层文件列表

Table 2　Core file list

模块	源程序/头文件
图形接口	TG_GI. C/TG_GI. H
文本功能	TG_TEXT. C/TG_ TEXT. H
事件驱动	TG_MSG. C/TG_MSG. H

图形接口提供了绘制点、水平直线、垂直直线、任意直线、矩形、填充矩形、圆、椭圆等 API 函数，用户只需根据 TG_GI. H 头文件中的定义，确定参数值，再调用相应的 API 函数，就能实现各种基本图形功能.

文本功能提供了兼容西文和中文字符的文本输出 API 函数和限制输出区域的扩展文本输出 API 函数，这两个函数均带有字体名称和字体大小参数，以支持不同字体和大小的远东字符(汉字)以及西文混合显示，前提是存储区域有足够的容量能够容纳多种字体的西文和中文字符集.

1.3　基本控件和基本控件管理器

基本控件和基本控制管理器是 TBGUI 的上层，实现了以控件为中心的 GUI 设计，也是程序员直接使用的主要部分. 这一层由基本控件(Basc Controls)和基本控件管理器(Base Control Manager)两部分组成，其中基本控件管理器只有一个模块. 基本控件和基本控件管理器主要 C 语言模块/头文件如表 3 所示.

除了通用数据类型定义头文件(TG_TYPES. H)以外，TBGUI 中还有 3 个非模块的头文件：TG_ICM. H、TG_ICTL. H 和 TG_IWND. H，分别是控件管理器、控件和窗口的接口定义头文件. 使用 TBGUI 的 C 语言源程序应该包含核心层以及基本控件和基本控件管理器各模块的头文件，特殊情况下，也可以直接包含抽象层甚至 RTL 各模块头文件，以直接使用抽象层功能.

表 3　控件模块列表

Table 3　Controls module list

模块	源程序/头文件
基本按钮控件	TG_BSBTN. C/TG_BSBTN. H
基本菜单控件	TG_BSMNU. C/TG_BSMNU. H
基本文本框控件	TG_BSTXT. C/TG_BSTXT. H
弹出式窗口	TG_POPWN. C/TG_POPWN. H
基本控件管理器	TG_BSCM. C/TG_BSCM. H

随着后续版本的推出，TBGUI 的上述各层将进一步扩展，添加更多的模块，实现更多的 GUI 功能. 考虑到嵌入式系统的资源和用户应用程序的实际需求，各模块的可裁减特性也是必须考虑的，以实现更好地利用有限的存储空间.

2　事件驱动机制在 TBGUI 中的实现

事件驱动机制其实是面向对象概念的延伸. 事件是事件驱动机制的灵魂，程序的执行取决于对事件的触发，换句话说，系统的发生、发展与并发、空闲及终止的全部过程都是由事件来控制的，这里的事件是指用户采取的行动，如键盘活动、对触摸屏的点击动作等，都是事件的例子. 事件驱动的程序是一种“被动”式程序设计方法，程序运行时，处于等待用户输入事件状态，然后取得事件并作出相应反应，处理完毕又返回并处于等待事件状态.

事件驱动机制的含义是，程序的流程不再是只有一个入口和若干个出口的串行执行线路；而是程序会一直处于一种循环状态，在这个循环当中，程序从外部输入设备获取特定的事件(比如鼠标的移动)，然后根据这些特定事件产生特定的消息，程序再根据消息作出特定响应，完成特定的功能，这个循环直到程序接受到特定消息终止为止[2]. 事件驱动的底层基础，就是消息队列和消息循环. 消息是报告有事件发生的通知[3].

TBGUI 支持完全的事件驱动编程，这意味着，只要设置各个基本控件的相应事件的处理过程，例如，对于‘OK’按钮控件，编写好触摸屏点击事件的处理函数 OKBaseBtn_Click()并设置控件事件函数指针指向该函数，如下：

OKBaseBtn. OnClick＝OKBaseBtn_Click

那么该按钮点击事件引发时，如同 VB、Delphi 或者 Java 控件一样，引发相应的事件处理过程，执行 OKBaseBtn_Click()函数. 该类过程处理函数的原型如下，此处仍以 OKBaseBtn_Click()函数为例.

```
static void OKBaseBtn_Click(void PTRMS * Container, void PTRMS * Sender, void PTRMS * EventParams) REENTRANT
{
    ……
}
```

所有事件处理函数的参数固定为3个，其中Container参数保留便于以后使用，Sender参数指向引发事件的控件变量，EventParams参数指向某些事件必需的事件参数数据结构.

需注意的是：不要随意省去事件处理过程中的PTRMS和REENTRANT宏，否则可能导致与Keil C51的不兼容. PTRMS宏和REENTRANT宏的定义如下：

```
#define PTRMS xdata
#define REENTRANT reentrant
```

因为TBGUI的消息、事件、控件处理程序大量使用函数指针实现间接调用，如果将TBGUI移植到51内核单片机中，因为Keil C51编译器的特殊性，通过函数指针调用函数默认使用了3个寄存器传递指针参数，因为寄存器的数量限制，所以这里要使用内存指针[4]. PTRMS宏可以定义为data，idata或者xdata，此处定义为xdata，即将所有参数指针放在外部XRAM中.

另外，TBGUI中的函数如果在抢占式多任务操作系统中使用，例如μC/OS-II，必须是可重入的[5]，因此，如果将TBGUI移植到51系列单片机上，使用Keil C51编译器编译，同时使用抢占式多任务操作系统，因为Keil C51编译器缺省情况下将函数编译成不可重入，所以必需在定义和实现函数时加入reentrant关键字，因此将宏REENTRANT定义为关键字reentrant.

TBGUI事件驱动编程的核心实现，仍然使用了消息和消息循环，不过TBGUI虽然实质上使用了消息，但除非需要进行消息的自定义处理，大多数情况下，无须显式使用消息，也无须显式实现消息循环. 这一点也与VB、Delphi或者Java相似，即用户程序的编程以事件为中心，而不是直接以消息为中心. 因为等待事件到来的消息循环过程在核心层的事件驱动模块中已经实现，无需用户直接定义和处理，这将大大降低编程的复杂性和对消息数据变量的使用，节约了内存空间[6].

图2是事件驱动机制下的程序流程，用户实现GUI界面，必须使用控件管理器，由控件管理器统一管理同一用户界面上的控件，用户界面的重绘也是由控件管理器完成的，这样TBGUI就可以做到无论是否使用窗口都可以使用控件.

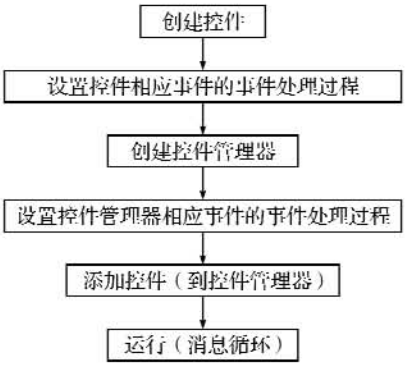

图2 事件驱动机制的程序流程图

Fig. 2 Program flow chart of event-driven mechanism

创建所需的各种控件，并编写控件对应的各事件的处理函数，它们不包含在本流程中. 然后使用回调函数的形式设置好这些事件的处理过程，接下来创建控件管理器，同样使用回调函数形式设置好控件管理器对应事件，主要是重绘界面，然后添加上述控件到控件管理器中. 运行该控件管理器，即开始绘制图形用户界面并加载各种控件，进入消息循环，等待事件驱动.

事件驱动机制下的TBGUI包含典型的类Windows GUI元素，例如窗口、按钮、文本框、对话框、滑动块、弹出式多级菜单等，均可使用指点设备，此处采用触摸屏作为显示终端和输入指点设备. 图3是包含了弹出式菜单、按钮、文本输入框等内容的测试界面.

图3 部分测试界面

Fig. 3 Part of test interface

测试表明，与MiniGUI和μCGUI相比，在图形和窗口功能上基本相同，但因为使用事件驱动的类Windows机制，且体系结构层次分明，所以占用空间小，扩展性更强，有更好的硬件无关性. 与ZLGGUI相比，占用空间略大，但大部分功能后者均不具备，例如各种控件和事件的响应方式，因而先进许多.

3　结　语

相对其它嵌入式 GUI 系统，例如 μCGUI、MiniGUI 或者 ZLGGUI，TBGUI 的特点是支持完全的事件驱动，它使用标准的 C 语言实现了原来只有使用 C++、VB 或者 Delphi 才能实现的事件驱动编程机制. 这种编程机制使得 TBGUI 的程序设计体系结构十分清晰，还使得 TBGUI 具有向可视化开发过渡的能力，实际上这个工作正在进行当中. 经过测试，在 51 内核平台上，TBGUI 自身耗用最小不到 15 kB 的程序空间，在只有 1 kB 的外部 RAM，20 MHz 以上的增强型 51 内核嵌入式系统中已经能够流畅地运行.

参考文献：

[1] 周立功. ARM 嵌入式系统软件开发实例[M]. 北京：北京航空航天大学出版社，2004：527-640.

[2] 方丰平. 嵌入式环境下高性能可配置 GUI 系统设计[D]. 杭州：浙江大学，2006.

[3] 刘黎志，刘军. 一个基于消息通知的工作流管理系统[J]. 武汉工程大学学报，2008，30(2)：94-97.

[4] 徐爱钧，彭秀华. Keil Cx51V7.0 单片机高级语言编程与 uVision 应用实践[M]. 北京：电子工业出版社，2004：497-506.

[5] Labrosse J J. 嵌入式实时操作系统 MicroC/OS-II[M]. 邵贝贝译. 北京：北京航空航天大学出版社，2007.

[6] 邹连英. 单周期 8 位微控制器的时序及流水线设计[J]. 武汉工程大学学报，2008，30(1)：103-108.

Event-driven programming mechanism implementation in the embedded GUI system

RAN Quan, CHEN Wei, ZHAO Shi-ping

(School of Computer and Engineering, Wuhan Institute of Technology, Wuhan, 430074, China)

Abstract: For development of windows and controls using of event-driven programming mechanism in embedded GUI system is disscussed, the independently development TBGUI's architecture and the characteristic are analyzed, and one kind of Windows GUI completely applied in embedded system has been provided. TBGUI fits to embedded systems based on 8-bit, 16-bit MCU, and also be extended to 32-bit embedded platforms such as ARM.

Key words: event-driven; TBGUI; abstract layer; controls

本文编辑：陈晓萍