

一种嵌入式GUI软件结构实现方案

詹瑾瑜* 熊光泽 孙明

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】综合比较了嵌入式GUI的几种实现方式,结合嵌入式系统的特点,研究了嵌入式GUI中的关键技术,分析了嵌入式GUI与普通GUI系统的不同之处,提出一种通用的嵌入式图形用户界面系统的设计思想和体系结构,这种嵌入式GUI实现方案具有轻型、占用资源少、可剪裁等特点。

关键词 嵌入式系统; 图形用户界面; 消息事件驱动机制; 消息队列

中图分类号 TP316.2

文献标识码 A

Implement Method of Graphical User Interface Framework Based on Embedded System

Zhan Jinyu Xiong Guangze Sun Ming

(College of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract The Embedded System has wide application, in order to use the Graphical User Interface in different field, this paper compares with many implement methods of the Graphical User Interface based on Embedded System, combines with the characteristic of Embedded System, investigates some pivotal technologies of the GUI based on Embedded System, analyzes the difference from the GUI based on Embedded System and the one based on common system and specifies an idea and a framework of a common and transplantable Graphical User Interface System based on Embedded System. The GUI based on Embedded System made by this idea has the characteristic such as light-duty, short-resource and convenient-clip.

Key words embedded system; graphical user interface; message and event drive mechanism; message queue

一个优秀的操作系统应该提供良好的图形用户界面(Graphical User Interface, GUI),否则将给用户的操作带来烦琐、不直观等问题,也将使程序开发人员很难在此操作系统上快速、有效地设计出界面友好的应用程序,所以图形用户界面影响着一个操作系统的发展。

当前,以信息家电为中心的嵌入式系统蓬勃发展,它广泛地应用于办公自动化、消费电子、通信设备、智能仪器和军事电子设备等各个领域。随着信息家电、手持设备、无线设备等迅速发展,相应的硬件和软件也得到迅速发展,许多设备都配有Intel、MIPS、Motorola等公司生产的32位微处理器,甚至还使用了液晶显示器,手持式设备的硬件条件的提高,嵌入式系统对良好GUI系统的需求会越来越迫切。与一般系统上的图形用户界面相比较,嵌入式系统下的图形用户界面要求有轻型、占用资源少、高性能、高可靠性、可配置等特点,嵌入式GUI的开发正处于起步阶段,有许多技术难题亟待解决。本文根据嵌入式系统的特点提出一种嵌入式GUI软件结构的实现方案,通过本方案设计实现的嵌入式GUI系统不仅具有很好的通用性,还具有灵活、可方便剪裁、简捷、高效等特点。

2002年4月29日收稿

*女 24岁 博士生 主要从事嵌入式实时操作系统及软硬件协同验证方面的研究

1 嵌入式GUI的设计思想和技术特点

图形用户界面是指计算机与其使用者之间的对话接口^[1,2],是计算机系统的重要组成部分,它基于操作系统之上,为用户提供丰富的图形编程接口,使其能够方便快速地编制界面友好的应用程序。

1.1 基于嵌入式系统的特殊考虑

与通用系统相比,当前嵌入式系统普遍存在CPU运算速度慢、内存容量小等特点^[3,4],所以嵌入式系统对图形用户界面系统的基本要求有:

- 1) 图形算法简洁、快速,占用资源少;
- 2) 功能强;
- 3) 可靠性高;
- 4) 模块结构,便于移植和定制。

目前国内外已推出了多种用于嵌入式环境的图形用户界面系统,其实现方法如下:

- 1) 照需求开发满足自身特定需要的GUI系统;
- 2) 将GUI作为一个软件层从应用程序中剥离,GUI的支持逻辑由应用程序自己负责;
- 3) 设计一个支持大多数常见的GUI对象的应用编程接口库,使其具有与其他通用开发工具相类似的调用方法(如Win32)的GUI系统^[5]。

在上述实现方法中,方法1)能够设计实现符合特定需要的嵌入式GUI,具有简捷、高效等特点,但它存在专用性强和开发成本高的缺点,每当需求有改变时都要重新开发,对于飞速发展的嵌入式系统硬件,这种方法不可行。方法2)属于一种临时解决方案,利用这种手段编写的程序,无法将显示逻辑和数据处理逻辑划分开来,程序结构不便于调试,而且还导致大量的代码重复,由于嵌入式系统本身就具有CPU运算速度慢、内存容量小等不足,如果想设计和实现简捷、高效的嵌入式GUI,那么这种方法也不可行。方法3)将嵌入式GUI以应用编程接口库的形式出现,使其能很快地被具有编程经验的人员所接受,同时还可以根据需求作适当的剪裁进而转变成如方法1)所实现的专用嵌入式GUI,也可根据需求对原有的应用编程接口库进行扩充,适合于大多数用户的开发,很多成功的嵌入式GUI即通过这种方式实现,如 DeltaGUI、MiniGUI、MicroWindows 等。因此本文使用方法3)设计和实现了一种灵活的嵌入式GUI软件结构方案。

1.2 嵌入式GUI系统的工作模式

嵌入式系统是为了满足特殊需求而被简化了的计算机系统。在嵌入式系统发展初期,嵌入式GUI的工作模式是单进程方式,采用单消息队列,实现机制简单,但其可靠性低,而且仅允许GUI进程来对GUI对象进行操作,而其他进程(如浏览器进程)想在操作过程中创建一个GUI对象(如对话框)就不可能,而且单进程方式很容易在用户进行窗口之间相互剪切过程中出现屏幕花屏的问题。随着嵌入式系统的迅速发展,简单的单进程工作模式已经不能适应新的需要,便出现了客户/服务器进程方式,它采用多消息队列,优缺点正好和单进程方式相反,实现机制比较复杂,允许任何进程调用GUI对象,而且可靠性相对较高。

基于以上的分析,本系统采用客户/服务器方式,系统中同时存在服务器进程和客户进程,服务器进程负责接收和发送消息给客户进程,客户进程调用自己的消息处理函数来响应消息。当服务器进程创建好自己的图形界面以后,进入消息循环,等待输入及客户请求,如果没有输入事件及客户请求,服务器进程阻塞;如果有输入事件发生,判断客户进程是否存在,存在则将消息传递给客户进程,如果有客户请求,响应客户请求,并把响应结果传送给客户。客户进程是从服务器进程中创建的(如点击某个应用图标,启动应用图标所对应的客户进程),客户进程创建好后,系统就让服务器进程退出屏幕显示,即服务器进程在客户进程存活期间不对屏幕进行操作。客户进程得到控制后,进入自己的消息循环,等待从服务器进程传递来的消息,如果没有消息需要处理,客户进程阻塞。当用户要退出客户进程时,服务器进程要恢复显示。

1.3 消息事件驱动机制

本嵌入式GUI采用的工作模式是客户/服务器方式,即是一个多进程环境,进程之间有信息采集、交互和同步的需求,而且GUI与一般程序的结构不同,不仅只有一个入口和若干个出口的顺序执行过程,而是一直处于等待使用者输入的循环中,常规的方法是适应这种复杂需要的,在此提出采用消息事件驱动机

制来管理这些进程间的通信，并为每一个进程分配一个消息队列。

使用者通过键盘或触摸屏等输入设备向系统发送一个消息，服务器接收到此消息，或者自己处理，或者传递给客户进程处理(如果客户进程存在)，客户进程在自己的消息循环中，如果消息队列为空，则等待从服务器传递来消息，如果没有消息，客户进程进入阻塞状态；否则如果有消息从服务器进程传递过来，客户进程恢复执行，将接收到的消息放入自己的消息队列。当消息队列中有消息时，客户进程取出消息，并执行相应的消息处理函数将消息解析为事件，根据事件的类型做出相应的响应，完成指定的功能^[6]。

1.4 内存分配

嵌入式系统的内存容量极为有限，所以合理地利用有限的内存成为嵌入式系统设计中一个必须考虑的问题^[7~9]。

对熟悉CRTL库(C Run - Time Library)工作原理的程序员比较了解在C程序中，如果频繁使用malloc和free申请和释放数据空间不仅非常耗时，而且还会使CRTL库的碎片化，进而导致内存分配失败。而在GUI中通常需要大量而且是频繁的从CRTL库中申请/释放数据块。为了避免频繁使用malloc和free直接从CRTL库申请和释放数据块，嵌入式GUI引入了一种新的机制，即在初始化时建立一个私有的堆，这个私有堆实际是由一些空闲待用的数据块组成的。当再遇到要分配数据块时，就从这个私有堆中分配，而不再需要从全局堆中来分配。只有当私有堆数据空间已经分完时，才用malloc函数在全局堆中申请分配。

1.5 实现平台无关性

嵌入式系统应用的领域不同，要求的GUI底层设备也不同，而GUI最底层的画点画线函数多是通过调用硬件设备的驱动来完成，如果GUI没有实现平台无关性，将出现每更换一种硬件设备都要重新编写GUI系统最底层的画点画线函数的问题，不但使编程变得繁琐，而且通用性不好，所以应设计隔离硬件设备设备硬件抽象层，它位于硬件设备和软件之间，实现对显示输出设备(如LCD显示器)、用户输入设备(键盘、触摸屏)的控制，向GUI上层软件提供统一的编程接口，通过这个设备硬件抽象层隔离具体的物理实现，以实现GUI硬件无关性。

2 嵌入式GUI的体系结构

根据上面的分析，嵌入式GUI系统采取分层设计的结构，体系结构图如图1所示。图中，API提供操作各种GUI对象(如窗口、菜单等)的应用编程接口函数；Core提供核心的图形操作功能，如消息机制、图形设备接口、字体、窗口与桌面等的管理功能，核心图形操作层Core详细模块划分如图2所示。IAL和GAL指硬件设备输入抽象层和图形输出抽象层，与底层输入输出设备接口，便于GUI挂接不同的输入输出设备，实现GUI系统良好的可移植性和通用性。

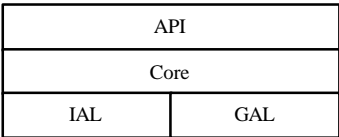


图1 嵌入式GUI分层结构图



图2 核心图形操作层结构图

2.1 核心机制模块

核心机制模块是嵌入式GUI中最重要的组成部分，包括消息机制和事件驱动机制、桌面操作、预定义堆、初始化操作、定时器操作等几个子模块，下面介绍几个主要子模块的功能：

- 1) 消息机制和事件驱动机制子模块：这个子模块主要负责消息的接受、分发和路由等操作；
- 2) 桌面子模块：桌面是GUI系统中非常重要的一个部分，相当于一个最底层窗口，其他主窗口都覆盖其上，维护了GUI中很多总体全局的操作；
- 3) 初始化子模块：它是GUI系统中最基本的一个模块，每次GUI运行之初都要通过这个模块的API函数

对整个GUI环境参数进行设定和初始化操作；

4) 定时器子模块：定时器是GUI中非常重要的组成部分，当用户需要定时的处理某项操作将会使用该模块，在系统中也有使用，如编辑框控件中光标的闪烁，进度条控件进度块的前进等。

2.2 资源和字体模块

资源指的是GUI中使用到的系统图标和系统图片，如单选按钮、关闭按钮等系统按钮的图片；对话框上警告信息的图标以及标题栏上的图标等。一般的GUI系统通常将这些系统图片和系统图标以一定的文件格式存放，当需要显示某个图片或图标时将对应的文件调出，分析之后再显示在屏幕上，但是从调入、分析到显示的过程要花费大量的时间，影响系统图片和系统图标的显示速度和效果，所以在嵌入式GUI提供的API库中先将所需的系统图片和图标进行分析，并将分析后的结果以数组的形式存放，这样嵌入式GUI在使用这些系统图片和图标时就可以直接调入而不必进行分析，进而精简了GUI所需的工作，并保证了显示的速度和效果。

字体模块作为GUI系统支持的一部分，主要为GUI其他模块所调用，处理文本输出和分析，GUI采用了面向对象的技术实现多字体和多字符集的支持。对某个特定的字符集，在内部使用该字符集完全一致的內码表示。然后通过一系列抽象的接口，提供对某个特定字符集文本的一致分析接口，该接口既可以用于对字体模块，也可以用来实现多字节字符串的分析功能，如果要增加对某个字符集的支持，只需要实现该字符集的接口即可。

2.3 图形设备接口

图形设备接口(Graphics Device Interface, GDI)是GUI图形操作的中间件，主要功能是支持与设备无关的图形操作，GDI将上层应用和不同输出设备的特性隔离开来，使编制的上层应用能够毫无困难地在任意一种图形输出设备上运行。它向上层应用(最终用户或者系统其他上层组件)提供了四类基本的服务：位图(bitmap)、文本(text)、一般绘图(Gen Drawing)和剪切域(clip)。

2.4 GUI对象模块

GUI对象模块包括窗口、菜单、对话框和闪烁光标4个子模块，向上层提供一系列Windows风格的GUI对象的操作函数，具有良好的通用性，如可对窗口进行拖动、缩放、最大化、最小化等操作。

窗口子模块：窗口是GUI系统中一个基本的元素，在这个模块中提供了各种与窗口有关的操作函数(如获取窗口相关属性的函数、改变窗口位置的函数等)，供上层调用；

菜单子模块：菜单是一组列表项，用户可以通过菜单知道可选择的项目和所要做的操作。菜单分为系统菜单、主菜单和弹出式菜单三种，此模块就提供了这三种菜单的各种操作；

对话框子模块：本嵌入式GUI系统只支持模态对话框，即当某个对话框被打开时，只有这个对话框及其上的控件能够接收消息，不能在对话框与其他GUI对象间进行任何的切换，若切换，则必须先结束该对话框；

闪烁光标子系统：在此闪烁光标指的是编辑框控件中的闪烁光标，可以清楚地指明当前的编辑位置，提供有关创建、显示及隐藏闪烁光标的各种API函数。

2.5 控件类模块

一般地，GUI系统都会预先定义一些控件类，当利用某个控件类创建控件之后，所有属于这个控件类的控件均会具有相同的行为和显示。利用这些技术，可以确保一致的人机操作界面，而程序员可以像搭积木一样地组建图形用户界面。本嵌入式GUI系统使用了控件类和控件的概念，可以方便地对已有控件进行重载，使其拥有一些特殊效果。如需要建立一个只允许输入数字的编辑框时，可以通过重载已有编辑框而实现，而不需要重新编写一个新的控件类。本模块提供了常用的预定义控件类，包括按钮、静态框、列表框、进度条、编辑框等。

2.6 输入法接口模块

GUI必须支持中文或其他语言的输入，所以必须有灵活的机制使其能挂接不同的输入法。在GUI中输入法是以主窗口的形式存在，可象编制应用程序一样来编写输入法窗口，并且由桌面(Desktop)来维护输入法的挂接。桌面模块中定义了这样一个全局变量来表示输入法窗口的句柄。输入法窗口具有一些特殊的属性，如总在其他窗口之上，或者对于某些手持设备有自己特定的外观等，不同的输入法具体情况有所不同，因

此本嵌入式GUI通过本模块提供标准的挂接接口函数，用户可以按照特定需要将输入法挂接到系统上。

3 结 束 语

本文提出了一种灵活的嵌入式图形用户界面的解决方案，本方案简单、高效、可剪裁、可扩充，其突出特点是向上层软件和下层的硬件设备提供了各种统一的编程接口，使GUI系统独立于上层的软件产品和下层的硬件设备，具有很好的通用性和可移植性。本方案在设计 and 开发过程中，吸取了同类产品的优点，已经开发出来并得到了验证，与同类产品相比具有代码量小、执行效率高的特点。

参 考 文 献

1 董文海. 计算机用户界面及其工具[M]. 北京: 科学出版社, 1994

2 孙家广. 计算机图形学[M]. 北京: 清华大学出版社, 1998

3 罗从难, 耿增强, 李小群, 等. 嵌入式的图形用户界面[J]. 测控技术, 2000, 19(4): 12-14

4 丁茂顺. 用户接口技术与交互系统构造方法[M]. 北京: 科学出版社, 1992

5 William D, Hurley S, John L, *et al.* Modeling user interface application interaction[J]. IEEE Software, 1989, 6(1): 62-70

6 Richard NT. A component- and message- based architectural style for GUI software[J]. IEEE Trans on Software Engineering, 1996, 22(8): 390-406

7 华庆一, 郝克刚. 一个基于CORBA的图形用户界面体系结构及实例[J]. 计算机学报, 1999, 22(1): 79-85

8 华庆一, 葛 玮, 鱼 滨, 等. 一个面向对象的交互式图形工具箱OOIGT[J]. 计算机学报, 1998, 21(2): 154-161

9 华庆一. 用户界面模型与形式规格说明研究[J]. 西北大学学报(自然科学版), 1997, 27(5): 369-374

编 辑 徐培红

· 成果与专利 ·

微波预处理包裹型复合铂钯矿技术

微波预处理包裹型复合铂钯矿技术是一种微波预处理包裹型复合铂钯矿技术，它是将大功率微波能通过由大功率环形器、销钉调配器、波导组成的微波传输系统对炉体内的包裹型复合铂钯矿加热处理，改善矿物后续浸出性能，再采用传统的湿法浸出、分离回收铂钯等贵金属。采用该技术，使我国已发现的大型贫铜、贫的包裹型复合铂钯矿床开发和综合利用成为可能。该技术与传统火法冶炼技术相比，具有高效节能、浸出率高、污染小等优点，可大大改善工人的劳动条件、降低劳动强度，具有极大的经济效益和社会效益。

· 文 争 ·