

# MiniGUI 在 VxWorks 中的应用及编程方法研究

姜海峰,王汀,赵政

(北京航天控制仪器研究所,北京 100039)

**摘要:** VxWorks 是风河公司设计开发的一种实时嵌入式操作系统,具有高可靠性和实时性。通常 VxWorks 不包含图形用户界面,这导致系统不易操作,为了实现图形用户界面,本文将 MiniGUI 应用在 VxWorks 中。首先本文对 WindML、MiniGUI 进行了介绍并对编程方法进行了研究和改进,然后实现了一个较复杂的图形用户界面。这种方法使系统易于操作,为 VxWorks 系统的应用带来了便利,而且不影响 VxWorks 系统的实时性能。另外对编程方法的改进可以提高图形用户界面的开发效率,便于代码的维护和二次开发。

**关键词:** VxWorks; MiniGUI; X86; 图形用户界面

中图分类号: TP316.2 文献标识码: A doi:10.3969/j.issn.1674-5558.2013.02.014

## Research on Programming Method to Using MiniGUI in VxWorks

JIANG Hai-feng, WANG Ting, ZHAO Zheng

(Beijing Institute of Aerospace Control Device, Beijing 100039)

**Abstract:** VxWorks is a kind of Embedded Real-Time Operating System (ERTOS) designed by WindRiver which features with high reliability and real-time performance. Generally we can't use Graphic User Interface (GUI) with VxWorks, which makes it uneasy to manipulate the system. In order to implement the GUI this paper takes a practice in using MiniGUI with VxWorks. Firstly this paper introduces WindML and MiniGUI, researches programming method and makes some improvement. Secondly this paper implements a GUI which is a little complex. This method makes it easy to manipulate the system without influencing real-time performance of VxWorks. And it not only enhances efficiency in designing GUI but also makes it easy to maintain and redesign the GUI.

**Key words:** VxWorks; MiniGUI; X86; GUI

## 0 引言

VxWorks 是美国风河公司(WindRiver)于 1983 年设计开发的一种实时嵌入式操作系统(RTOS)。VxWorks 以其良好的持续发展能力、高性能的内核、友好的用户开发环境、高可靠性和实时性被广泛地应用在高精尖技术及实时性要求极高的领域中。VxWorks 的实时性做得非常好,其系统本身的开销也很小,进程调度、进程间通信、中断处理等系统公

用程序精练而有效,延迟很短。VxWorks 提供的多任务机制中对任务的控制采用了抢占式和轮转调度机制,充分保证了可靠的实时性,使同样的硬件配置能满足更强的实时性要求,为应用的开发留下了更大的余地。

通常 VxWorks 不包含图形用户界面,这导致系统不易操作,实现图形用户界面主要有两种方法,一种方法是在其它电脑上实现人机交互功能,再通过网络控制 VxWorks;另一种方法是借助第三方嵌入

收稿日期:2013-03 修订日期:2013-05

作者简介:姜海峰,男,硕士研究生,从事导航、制导与控制专业研究。

式图形界面开发工具在 VxWorks 系统上实现图形用户界面(GUI)。本文中 VxWorks 是运行在 X86 构架的 PC(个人台式机或者工控机)上,具有图形用户界面开发的硬件基础,为了降低系统的复杂性,采用第二种方法。本文成功将 MiniGUI 应用在 VxWorks 中,首先对 WindML、MiniGUI 进行了介绍并对编程方法进行了研究和改进,然后实现了一个较复杂的图形用户界面。由于 VxWorks 提供的多任务机制中对任务的控制采用了抢占式和轮转调度机制,所以只要使图形用户界面任务的优先级低于其它用户任务,便不会对其它任务的运行产生影响。这种方法在不影响 VxWorks 系统实时性的情况下使系统易于操作,为 VxWorks 系统的应用带来了便利,另外对编程方法的改进可以提高图形用户界面的开发效率,便于代码的维护和二次开发。

## 1 配置 WindML

### 1.1 WindML 构架及工作原理

WindML 是风河公司为 VxWorks 系统设计的多媒体函数库,成功配置 WindML 后才可以在运行 VxWorks 的计算机上实现图形用户界面,使系统的应用更加方便。WindML 模块化的设计使得 WindML 具有可移植性,WindML API 函数库提供了访问硬件的一致接口,而与处理器和操作系统无关。WindML 实现了下面的设计目标:

(1)简单:提供了简单的通用的图形函数集合,基本的视频、音频处理功能。

(2)硬件可移植:可以在不同的硬件环境下运行。

(3)操作系统可移植:可以在不同的操作系统环境下运行。

(4)驱动开发简单:为驱动程序的开发提供了通用的机制。

WindML 主要由两部分组成,如图 1 所示。

#### (1) Software Development Kit(SDK)

SDK 是用来开发应用程序的,它提供了一个复杂的 API 函数集合,可用于图形处理、输入处理、多媒体、字体和内存管理,使用它可以开发出与硬件无关的应用程序。

#### (2) Driver Development Kit(DDK)

DDK 用于实现驱动程序,它提供了一个完整的参考驱动的集合,可以实现常用硬件的配置。它还提供了一个 API 函数库可以使用户很快的以现有驱动为基础开发出新硬件的驱动程序。

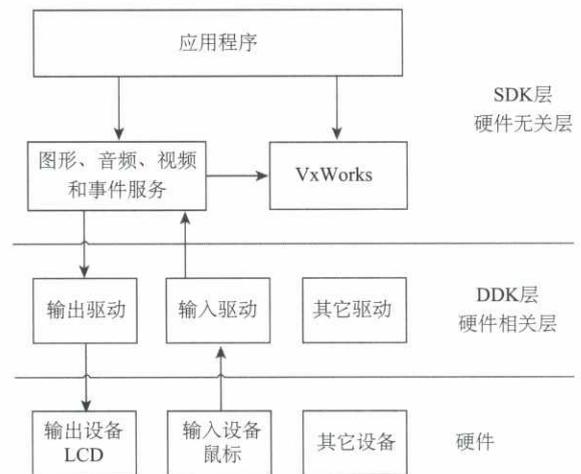


图 1 WindML 层次结构

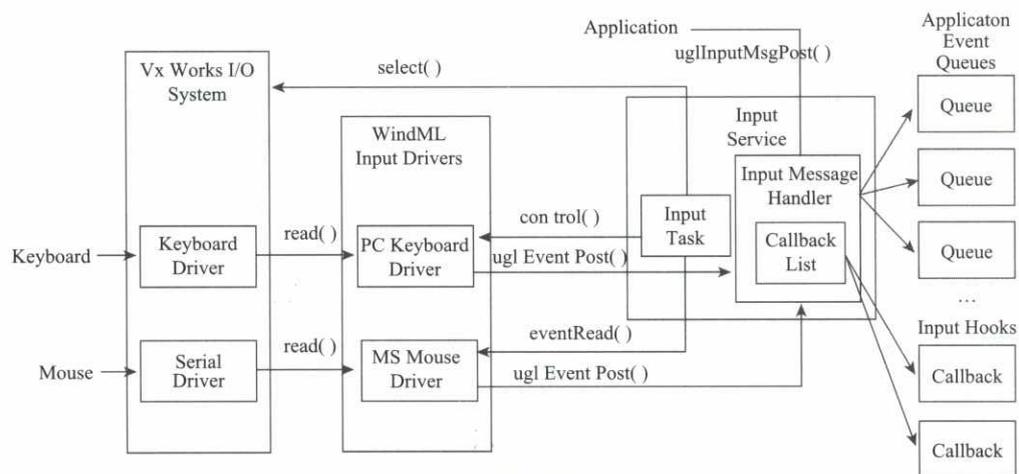


图 2 输入服务

输入服务是 WindML 的核心,输入服务负责管理输入设备,对它们产生的所有消息进行统一处理。输入服务使用一个周期执行的任务从输入设备获取数据,并将这些数据封装成消息以便进一步处理,图 2 展示了输入服务的结构以及应用程序是如何使用输入服务的。

### 1.2 WindML 配置

WindML 的配置是 BSP 设计的一部分,下面介绍 WindML 配置过程:

(1) 在 config. h 文件内添加宏 INCLUDE\_WINDML 的定义,即包含 WindML 组件;

(2) 在 romInit. s 文件内添加#include < ugl/ driver /graphics/pcbios/romInit. h >,即包含一些与 BIOS 有关的图形设备参数和功能函数声明;

(3) 在 i8042KbdMse. c 文件内添加看门狗计时器创建代码,这个看门狗计时器是供键盘鼠标的驱动程序使用的。即 i8042Wdid = wdCreate( );

(4) 编译 WindML。编译时选择 pentium\_VESA-BIOS\_RGB565\_640X480 项目,点击配置 Configure 打开对话框。Build 标签页的选项都勾上。Devices 标签页,分辨率选 800x600,鼠标选择 PS/2,名称为 /pointer/0,键盘选择 PC/AT,名称为 /pcConsole/1。其它两个标签默认,然后保存该配置文件。编译译完后会生成 libwndml. a 文件。

## 2 MiniGUI 运行机制及编程研究

### 2.1 MiniGUI 运行机制

直接使用 WindML 来开发 GUI,整个过程不是所见即所得的,开发的难度大,效率低,所以本文借助第三方嵌入式图形界面开发工具来开发 GUI。目前可以选用的开发工具有多种,本文将它们分为两类,一类是风河公司的产品,如 Zinc 和 Tilcon;另一类是非风河公司的产品,如 MiniGUI、OpenGUI、QT/Embedded 等。Zinc 和 Tilcon 主要是为 VxWorks 设计的,所以与 VxWorks 系统有很好的兼容性,而 Tilcon 比 Zinc 功能强大,也是风河现在主推的图形界面开发工具,缺点是成本高。其它工具最初不是为 VxWorks 设计的,应用在 VxWorks 下需要进行必要的移植工作,这需要开发工具的厂商提供技术上的支持,所以这方面的应用还不成熟,一些成功的应用通常针对具体的硬件平台。本文选用 MiniGUI 来开发 GUI,MiniGUI 是北京飞漫软件公司的产品,选择 MiniGUI 原因是有技术上的支持,而且成本低。

在任何 GUI 系统中,均有事件或消息驱动的概念。在 MiniGUI 中,我们使用消息驱动作为应用程序的创建构架。在消息驱动的应用程序中,计算机外设发生的事件,例如键盘键的敲击、鼠标键的按击等,都由支持系统收集,将其以事先的约定格式翻译为特定的消息。应用程序一般包含有自己的消息队列,系统将消息发送到应用程序的消息队列中。应用程序可以建立一个循环,在这个循环中读取消息并处理消息,直到特定的消息传来为止。这样的循环称为消息循环。一般地,消息由代表消息的一个整型数和消息的附加参数组成。应用程序一般要提供一个处理消息的标准函数。在消息循环中,系统可以调用此函数,应用程序在此函数中处理相应的消息。图 3 是一个消息驱动的应用程序的简单构架示意。

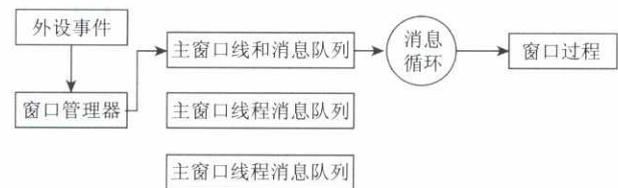


图 3 消息驱动的应用程序的简单构架

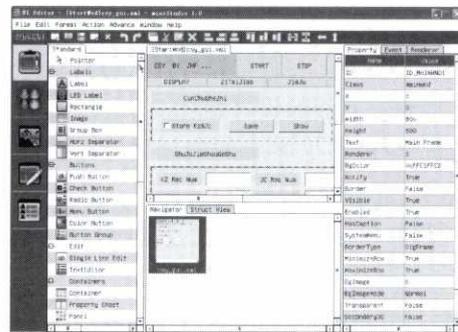


图 4 mStudio IDE

### 2.2 编程方法研究及改进

MiniGUI 的编程是在名称为 mStudio 的 IDE 中进行的,如图 4 所示。mStudio 基于 eclipse 开发环境,使用起来很方便,下面介绍一下编程的方法。GUI 的编程一方面是窗口内各个控件的安排,另一方面是消息函数的编写。使用 mStudio 编写的消息函数主要有两种,下面以具体实例进行介绍:

(1) 不访问其它控件,如 static BOOL Propsheet2\_onCreate ( mWidget \* self, DWORD dwAddData ), Propsheet2 是一个控件的名字,而 mWidget \* self 便是这个控件的指针,在这个函数里只能对这个控件

进行操作。显然这是不能满足编程需要的。

(2) 访问其它控件, 如 `static BOOL sledit1_on_button1_clicked (mSledit * self, mButton * sender, int id, DWORD param)`, `sledit1` 和 `button1` 是两个控件的名字。可以这样理解, `button1` 是消息的产生者, `sledit1` 是这个消息的响应者, 也就是说 `sledit1` 需要对这个消息做出相应的响应。

在消息函数里通常需要访问其它控件, 甚至是多个控件, 若采用上面两种方法将会导致控件之间的关系很复杂, 程序的编写也会很混乱, 很难编写出复杂的 GUI。所以本文借鉴了 C++ Builder 等 Windows 系统编程工具的优点将整个 GUI 中需要响应消息的控件的指针作为全局变量保存起来, 这样就可以在所有消息函数里访问这些控件了。实现的方法如下所示:

```
static BOOL Panel8_onCreate (mWidget * self,
DWORD dwAddData) {
    pCheckButZTJ = (mCheckButton *)_c (self) -
> getChild (self, ID_CHECKBUTTON1);
    return TRUE;}
```

`Panel8` 是一个容器控件, 在它创建的时候将子控件的指针作为全局变量保存起来, 如 `pCheckButZTJ`, 这样在其它消息函数中便可以随意访问这个控件了。

### 3 应用程序设计

#### 3.1 共享数据与代码

GUI 与其它任务一定会有数据和代码需要共享, 本文将这些数据和代码放在 GUI 中进行管理, 这是因为在本文中 GUI 的开发先于其它任务, 而 GUI 需要这些数据和代码来进行功能验证。

在一个多任务环境中, 函数的可重入性是十分重要的。可重入函数是可被多个任务调用的函数, 任务在调用时不必担心数据是否会出错。例如, 许多任务可能调用 `printf()` 子函数, 但是整个系统中只有一份 `printf()` 子程序的代码拷贝。在写函数时只需考虑尽量使用局部变量 (如寄存器和堆栈中的变量); 对于要使用的全局变量 (例如采用关中断和信号量等) 要加以保护, 这样构成的函数就一定是一个可重入的函数。在设计共享数据与代码时这一点要特别注意。

#### 3.2 与后台任务交互

在 `mStudio` 中只能进行 GUI 编程, 不能使用 Vx-

Works 提供的功能和函数, 但在消息函数中必然需要调用底层函数, 而且其它任务也需要使用 GUI 提供的功能。如果在 GUI 中不实现这些功能, 而是在 `tornado` 中进行开发, 对于功能明确的程序来说是可行的, 可一旦需要修改程序将会很麻烦, 因为一旦修改了 GUI, 这些与后台任务的交互需要重新编写。为了解决这些问题, 本文使用模块化编程的理念将 GUI 设计为一个模块, 结构如图 5 所示。

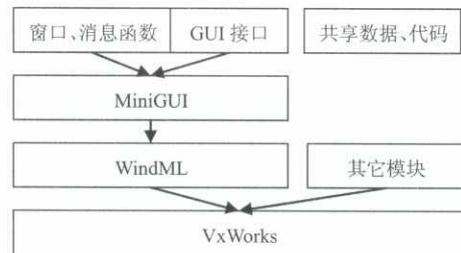


图 5 GUI 结构

图中, 消息函数是在窗口任务中执行的, 它们负责处理用户输入, 并将这些信息转换到共享数据里, 必要时再调用其它模块的函数; GUI 接口是被其它任务调用的, 在其它任务空间执行, 它们主要提供显示功能, 如显示共享数据, 清空文本框等。调用 GUI 接口函数的方法如下面代码所示:

```
int vxShowMsg (char * msg) {
#ifdef INC_GUI
    guiShowMsg (msg);
#endif
}
```

其中 `guiShowMsg (msg)` 是 GUI 接口函数, 这个函数的功能是在 GUI 中显示一个字符串。同时 GUI 也需要调用 VxWorks 或其它模块提供的函数, 调用的方法如下面代码所示:

```
#ifdef INC_VXWORKS
    vxCsyStart();
#endif
```

其中 `vxCsyStart()` 是由其它模块提供的函数。

### 4 实验

本实验实现了一个较复杂的 GUI, 如图 6 所示。实验中, 屏幕的分辨率为 `800x600`, 较低的分辨率导致窗口中不能放置太多的控件, 另外将 `MiniGUI` 应用在 VxWorks 中只能实现单窗口的应用程序。为了满足复杂界面的设计需求, 可以使用属性页控件来实现多个页面窗口, 图 6 中 GUI 便有 6 个页面窗

口。GUI的功能主要有两个,一是负责处理用户的输入,二是显示状态、数据和结果等。实验中 GUI 使用的输入控件有命令按钮、文本框、单选框、复选框;显示控件有静态文本框、文本框和状态灯。经过实验测试,GUI 的键盘、鼠标、控件等都能正常工作。



图6 GUI界面运行效果

## 5 结论

本文成功将 MiniGUI 应用在 VxWorks 中,首先对 WindML、MiniGUI 进行了介绍并对编程方法进行了研究和改进,然后实现了一个较复杂的图形用户界面。这种方法在不影响 VxWorks 系统实时性的情况下使系统易于操作,为 VxWorks 系统的应用带来了便利,另外对编程方法的改进可以提高图形用户界面的开发效率,便于代码的维护和二次开发。经过实验测试可知,这种方法可以实现复杂、实用的图形用户界面。

### 参考文献

[1] 张扬,于银涛. VxWorks 内核、设备驱动与 BSP 开发详解(第2版)[M],北京:人民邮电出版社,2011.  
 [2] 邝(kuang4)坚. Tornado/VxWorks 入门与提高[M],北京:科学出版社,2004.

[3] 李忠民,杨刚,顾亦然,刘尚军. ARM 嵌入式 VxWorks 实践教程[M]. 北京:北京航空航天大学出版社,2006.  
 [4] 吴宁,马旭东,张颖,周芳. 80x86/Pentium 微型计算机原理及应用[M]. 北京:电子工业出版社,2011.  
 [5] 王学龙. 嵌入式 VxWorks 系统开发与应用. 北京:人民邮电出版社[M],2003.  
 [6] VxWorks Programmer's Guide. Wind River Systems Inc., 1999-02.  
 [7] WindML DDK Programmer's Guide 3.0. Wind River Systems Inc., 2002-09.  
 [8] WindML SDK Programmer's Guide 3.0. Wind River Systems Inc., 2002-09.  
 [9] 孔祥营,柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado[M]. 北京:中国电力出版社,2002.  
 [10] 顾亚刚. 基于 VxWorks 的异构多核处理器软件系统的研究与设计[D]. 硕士学位论文. 长沙:国防科学技术大学,2008.  
 [11] 余舰. 星载 PowerPC 最小系统及 VxWorksBSP 设计[D]. 西安:西安电子科技大学,2011.  
 [12] 杨康. 嵌入式操作系统 VxWorks 实时性能研究与测试[D]. 长沙:国防科学技术大学,2009.  
 [13] 陈智育著. VxWorks 程序开发实践[M]. 北京:人民邮电出版社,2004.  
 [14] H. Tokuda, C. W. Mercer, ARTS. a distributed real-time kernel[J]. ACM Operating Systems Review, 1989, 23(3): 29~53.  
 [15] 张辉译, Neville-Neil, G. V. 著. FreeBSD 操作系统的设计与实现[M]. 北京:电子工业出版社,2005.  
 [16] 李贵山,陈金鹏. PCI 局部总线及其应用[M]. 西安:西安电子科技大学出版社,2004.  
 [17] 唐晓平. VxWorks 在 cPCI 高速数据采集系统中的应用研究[D]. 长沙:国防科学技术大学,2008.

(上接第67页)

[7] 张锦江. 仿真转台机械回转精度对测量精度的影响分析[J]. 中国惯性技术学报,2000(2).  
 [8] 谷文韬,曹海旺. 转台运动性能检测系统的设计与实现[J]. 测控技术,2008(2).  
 [9] 任顺清等. 测量角位置误差时自准直仪读数与正多面体棱镜偏差的符号取定[J]. 计量技术,2003(12).  
 [10] 刘文魁,石建玲. 光电旋转编码器在角度测量中的应用[J]. 现代制造工艺,2006(11).  
 [11] 贾明,王振飞,晁代宏. 双轴转台误差对 IMU 标定精度的影响[J],导航与控制,2012(2).

[12] 浦昭邦,涛卫. 角度测量的光学方法[M]. 光学技术,2002(2).  
 [13] Hongliang Zhang, Yuanxin Wu, Wenqi Wu et al. Improved multi-position calibration for inertial measurement units[J]. Measurement Science And Technology, 2010(21).  
 [14] 张红良,武元新,练军想等. 基于转台误差分析的高精度惯测组件标定编排改进[J]. 中国惯性技术学报,2010,18(1).  
 [15] 姜复兴著. 惯导测试设备原理与设计[M]. 哈尔滨工业大学出版社,1998年.