

基于 VxWorks 的 Zinc 人机界面的设计与实现

Design and Realization of Graphic User Interface based on Zinc of VxWorks

陈 广 王福友 袁赣南 (哈尔滨工程大学自动化学院 哈尔滨 150001)

王佳佳 (中国人民解放军 92132 部队 青岛 266405)

摘要:本文主要对 VxWorks 的 Zinc 这个工具进行了探讨,介绍了它在嵌入式实时环境下(RTOS)开发人机界面的方法,并给出工程开发样例。文章着重讨论了 Zinc 的层次框架、事件模型、对象、人机界面的设计与实现方法。

关键词:VxWorks Zinc 嵌入式实时环境 人机界面

1 引言

现在,嵌入设备用户也像 PC 用户一样,对图形用户界面(GUI)的需求不断增加。但是,由于嵌入设备的独特的、多变的要求,GUI 方案有时不可行。这些设备可根据它们的多样性表述:它们通常在专门的硬件上开发的,采用有成本效益的经济性模型,并用于专用目的。所以,一个嵌入设备的 GUI 方案必须满足传统的、桌面 GUI 方案的特殊要求。Wind River 的 Zinc 实现了嵌入计算机的图形要求,使开发人员在 Wind River 的 VxWorks 实时操作系统上建立丰富的、全功能的可嵌入 GUI 成为可能,并且这种 GUI 占用系统资源很低,开发周期也很短。

本文针对 Zinc 这个工具进行了探讨,并结合实际使用情况给出了开发过程。

2 Zinc 简介

Zinc 是 WindRiver 公司推出的一个功能强大、跨平台、国际化的图形用户界面开发工具。Zinc 提供给程序员的是许多 C++ 类库形式的 API 及一个可视化设计工具 Zinc Designer。Zinc 既支持图形用户接口 API 的可视化开发,也支持直接编写代码开发。

Zinc 可以很容易地规划和配置,以满足给定应用程序的准确 GUI 要求。Zinc 是针对小内存环境而设计的,它完全可以工作在由 VxWorks 和 Zinc 构成的低于 1MB 内存的环境中。

Zinc 可以运行在多个平台之上,这些平台包括 VxWorks、Microsoft Windows、Unix、X/Motif、MSDOS。

Zinc 采用了先进 il8n (internationalization) 技术^[1],其程序可以显示几种语言的信息,包括中文、英文、日文等。

3 Zinc 层次框架

Zinc 应用框架定义了一个用户接口 API 的抽象层,这个抽象层独立于任何操作系统给用户在任何环境下提供可移植的存取方法,API 会被映射到每个操作的本地功能。

在 VxWorks 中使用 Zinc 要运行 WindML (Wind Media Library) 之上,WindML 支持多媒体运行于嵌入式操作系统之中,主要目的是用来提供基本的图形、视频和音频技术,以及提供一个设计标准设备驱动程序框架。WindML 包括两个组件——软件开发包 (SDK) 和驱动程序开发包^[2] (DDK)。SDK 组件用来开发应用程序,它提供了一个全面的 API 集,包括图形、输入处理、多媒体、字体和内存管理。DDK 组件是用来实现驱动程序的,它提供了一个完整的驱动程序参考集,包括硬件配置和 API 集,已使得开发者能够迅速的引导和使用自己的驱动程序。WindML 为基本的操作系统提供一个抽象层^[3],Zinc 是更高一层的抽象,它提供了类 Windows 风格的接口,有各种控件被封装在内,编程接口很类似于 Windows 编程。

4 Zinc 事件模型

Zinc 的应用程序是在基于事件驱动 (Event-driven) 的机制下运行的^[4]。输入设备和应用程序之间通

过事件(Event)相互作用的。Zinc 的 Event 来源:操作系统本身、外部设备和用户应用程序。例如键盘就是一个典型的外部设备输入。Event 获取之后再通过一个标准的方法包装 Event,然后发送给相应的对象以供处理,如图 1 所示。

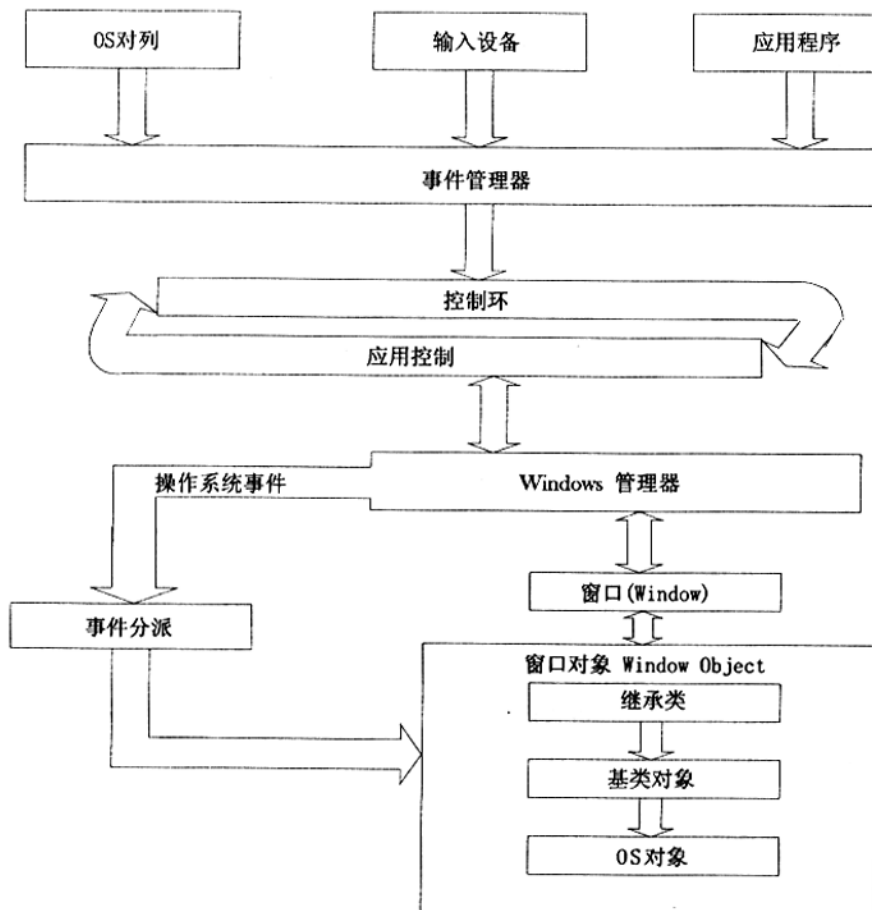


图 1 Zinc 事件模型

5 Zinc 对象

Zinc 提供了一个 C++ 编程接口的、可扩展的、面向对象的类库。在 Zinc 类库里有以下三种类型的对象:

(1) 用户界面对象(User Interface Objects) 用户界面对象就是用户在屏幕上看到的那些控件,如窗体、按钮、对话框等。Zinc 具备 windows 上所有常用的控件。用户界面对象又分为两种:作为容器的和不可作为容

器的。前者由 Zafwindow 类派生得到,可以管理多个子控件,属于这一类的有窗体、组合框、树形列表框等;后者由 ZafWindowsObject 类派生得到,如按钮、文本框等。

(2) 框架对象(Framework Objects)

该类对象构成了 Zinc 应用程序的骨架,它们在屏幕上不可见,负责在幕后完成以下工作:管理用户界面对象;负责用户界面对象与应用程序的通信;管理输入设备,生成事件发送给上层的对象;管理应用程序的资源。下列框架类特别重要:

① 事件管理器类(ZafEventManager)

负责收集来自操作系统、输入设备和应用程序的消息,放入消息队列中,准备提交给系统或用户界面对象。

② 应用程序类(ZafApplication)

ZafApplication 负责初始化和存储应用程序所用到的一些数据。如指向事件管理器的指针、指向窗口管理器的指针、指向显示器的指针等等。同时,它也提供了一个平台无关的应用程序的入口点。

③ 窗口管理类(ZafWindowManager)

负责管理应用程序的窗口和事件的路由。为了让窗口管理器管理某个窗口,利用 Add() 方法把该窗口添加到窗口管理器即可。窗口管理

器可以无需用户的干预,透明地管理窗口。

(3) 数据对象(Data Objects)

数据对象具有字符和数字的识别与存储,当特定的数据对象改变时通知图形窗口对象,当图形窗口内容改变时发出通知的三种功能。Zinc 的绝大多数用户界面对象都有一个数据对象与之关联,当然这些数据对象可独立使用。

6 Zinc 人机界面的设计与实现

用 Zinc 开发图形用户界面,一般有三种方式:

- Zinc Designer。
- 编写代码方式。
- Zinc Designer 与编写代码结合的方式。

6.1 Zinc Designer

Zinc Designer 做为一种设计方法,设计更加直观。其设计方法类似于 VB,打开 Zinc 编辑窗口,其上面有很多控件栏,设计者首先选择一个窗口类,定义好每一个窗口类名,然后在所选窗口添加所需控件,如按钮、工具条、下拉菜单、弹出菜单等。对所添加控件属性进行设置,如控件 ID、控件名称等如图 2。



图 2 Zinc 窗口对象属性设置

用 Zinc Designer 设计完各窗口及其控件时后,以 *.znc 文件保存。接下来生成代码,选择 OptionsGenerate Code 生成 *.cpp、*.hpp、*.inc 文件。用户只需在 *.cpp、*.hpp 添加代码就能实现相应功能。

6.2 编写代码方式

编写代码方式是指完全依靠代码形式实现界面的开发。根据 Zinc 采用 C++ 面向对象的设计思想,完全包含了完整的 C++ API,用户设计界面时,首先定义类,然后再填写代码。对于界面设计简单的可以采用此方法。

6.3 Zinc Designer 与编写代码方式相结合设计界面

此种方式是以 Zinc Designer 设计出界面并生成代码后,对界面上的对象根据需要定义相应的类,并对其中的成员函数编写代码。根据作者在实际开发界面的经历,此种方法比较好。

因为仅用 Zinc Designer 不仅需要了解 Zinc 的各种工作机制而且有时很难实现其他功能如作为嵌入式时时界面有时免不了要收发串口数据部分,而要把这部分加入 Zinc Designer 设计界面生成的任何类中都不适合,而需要设计者额外设计一个 ComInit 类,而且这对程序封装及可读性都有益处。而仅用编写代码方式工作量往往很大。

根据作者的经历将二者结合起来可以减少工作量,缩短开发周期,这也是人机界面设计的一个关键指标。而且 Zinc 给出的 demos 例子基本上使用的是 Zinc Designer 与编写代码方式相结合来设计界面的。Zinc 中的 demos 例子对于开发者具有很高的参考价值。

6.4 人机界面的实现

Zinc 应用程序设计时,开发环境采用主机/目标机的形式^[5]。当应用程序设计完后,在 Tornado 下建立一个 Download 型工程,加入 Zinc 中的 demos 例子运行以验证设计者配置的 ugl 及 Zinc 是否正确,若正确,加入用户设计的应用程序,编译后下载。为了使程序正确运行应注意以下几点:

(1) 应在应用程序中加入一个 v_app.cpp 文件,作为入口程序文件,可以参考 Zinc 中的 demos 例子中的任意一个 v_app.cpp 文件;

(2) Zinc 的资源文件 *.znc 要和 Tornado 放在同一个目录下;

(3) 将应用程序下载以前,应依次下载 uglPENTIUMgnu.o、zincPENTIUMgnu.o (对于目标机是 PC486 的,依次下载 ugl80486gnu.o、zinc80486gnu.o)。

7 Zinc 的工程开发实例

步骤 1: 用 Zinc Designer 设计好一个界面,并生成相关代码。

步骤 2: 添加控件代码以实现相关功能。以图 3 的时间显示需要 1s 定时器,以及按钮操作即按下按钮进入下一个界面代码为例。

1s 定时器:

```

.....
MainWind * pMainWind; //主界面指针
GPS * pGPSDlg //子界面指针
ZafTimer * g_Timer1s; //定时器指针
Const ZafEventType T_TIME1S = 10090;
.....
{
.....
g_Timer1s = new ZafTimer( D_ON, 1000 ); //1s 定
时
g_Timer1s -> SetNotifyMessage( T_TIME1S ); //发送
消息
g_Timer1s -> Add ( pMainWind );
EventManager( ) -> Add( g_Timer1s );
.....
}
按钮:
.....
Const ZafEventType Button_Open = 10000;
ZafEventType MainWind:: Event ( const ZafEventStruct
&event) // Zinc Designer 设计时界面自动生成的事件
窗口类;
{
ZafEventType ccode = LogicalEvent( event );
Switch ( ccode )
{
case Button_Open:
{
pGPSDlg -> SetOwner( pMainWind ); //激活按钮
控件
WindowManager( ) -> Add( pGPSDlg );
}
break;
.....
default:
//将事件传递到窗口基类加以处理;
ccode = ZafWindow:: Event( event );
break;
}
return( ccode );

```

```

}

```

这样类似操作就能使界面的控件“活”起来。

步骤3: 编译下载即可在目标机上看到设计的界面。

图3给出作者在实际课题中以实时环境下用Zinc开发的一个主界面。

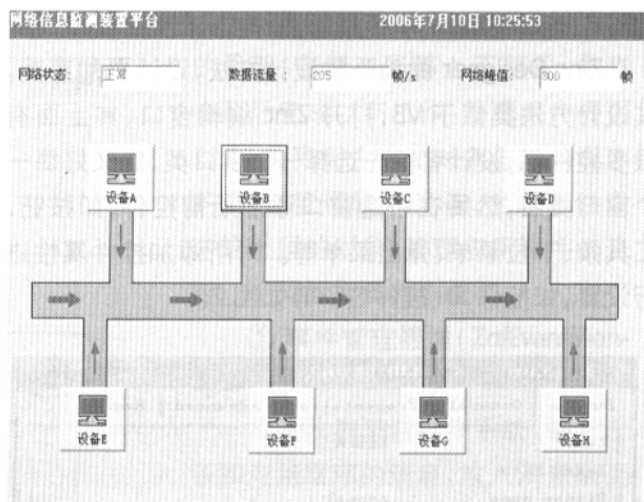


图3 用Zinc开发的网络信息监测平台的主面

8 结束语

以上简单地介绍了Zinc在RTOS环境下人机界面的使用方法。用Zinc可以成功地设计桌面、RTOS环境下的界面。作者用Zinc成功地实现了“某应用网络信息监测平台界面的设计”的课题。在课题中用Zinc开发程序整体结构,完成了收发信息,时时显示信息、报警等功能。界面清晰直观、且可具有动画效果。

参考文献

- 1 Zinc 6.0 Programming's Guide [M], Wind Rive System, Inc, Alameda, U. S. A, 2002: 179 - 180.
- 2 Wind ML Programmer's Guide [M], Wind River System, Inc, Alameda, U. S. A, 2002: 65 - 67.
- 3 UGL Developer's Kit User's Guide [M], Wind River System, Inc, Alameda, U. S. A, 1999: 78 - 79.
- 4 Zinc for VxWorks Getting Started Guide [M], 5.3 Edition, Wind River System, Inc, Alameda, U. S. A, 1999: 25 - 26.
- 5 Tornado User's Guide [M], Wind River System Inc, Alameda, U. S. A, 2002: 14 - 15.