

# 基于 QNX 实时操作系统的图形控制界面设计

石峰, 胡大斌, 胡锦晖

(海军工程大学 船舶与动力学院, 湖北 武汉 430033)

**摘要:** 介绍了在 QNX 实时操作系统图形界面开发环境 PhAB 下软件设计的特点, 并结合船舶动力装置控制系统的具体要求, 设计了船舶动力装置控制系统图形控制界面和程序。

**关键词:** QNX; PhAB; 船舶动力装置; 图形界面

中图分类号: TP274

文献标识码: A

## Design of graphics control interface based on QNX RTOS

SHI Feng, HU Da Bin, HU Jin Hui

(College of Naval Architecture and Power, Naval University of Engineering, Wuhan 430033, China)

**Abstract:** The characteristics of software design in QNX RTOS's photon application builder(PhAB) are described in this paper. Combining with detail requirements of control system on marine power, design the photon interface and program of this system.

**Key words:** QNX; PhAB; marine power; photon interface

在计算机控制系统中, 采用图形界面对被控对象实施控制具有结构清晰、操作方便、界面友好等优点。但目前工业控制等领域采用的图形控制界面开发平台多为通用的组态软件, 这种图形控制软件依赖 Windows 平台运行, 稳定性和可靠性不高, 在工作环境恶劣、可靠性要求高的系统中应用较少。实时操作系统是专门用于实时控制的一类操作系统, 相比于常用的分时操作系统, 实时操作系统大多拥有微内核, 并且通过对结果返回的时间限制来实现程序运行的可预测性, 因此具有较高的稳定性和可靠性。近年来, 随着实时操作系统在一些可靠性和实时性要求较高的领域的成功应用, 实时操作系统得以快速发展。为适应硬件的发展和用户的需求, 各大实时系统开发商都开发出适合该系统的图形控制界面, 如 VxWorks 的 WindML 和 QNX 的 PhAB。本文在船舶动力装置控制系统设计中, 以 QNX 实时操作系统为基础, 研究了图形控制界面设计的特点, 设计了系统控制程序。

### 1 图形界面开发环境

PhAB(Photon Application Builder)是 QNX 集成开发环境(QNX Momentics IDE)附带的图形界面编辑器。PhAB 承接了 QNX 微内核和可优化裁减的优点, 使用 PhAB 编写出的系

统除用于核心内存保护的微内核和一些必要的核心管理器之外, 其他功能都是可选择的。这种设计不仅保证了作为实时系统的高可靠性和实时性, 也在很大程度上提高了使用 PhAB 所开发的图形界面控制系统的应用范围<sup>[1]</sup>。

PhAB 采用所见即所得的控制界面开发模式, 并带有常用控件数据库, 在控制界面设计时可以直接在界面编辑区域创建工作控件并进行所需设置, 而无需程序编写。

基于 QNX 的控制界面和程序设计支持主机——目标机的开发模式, 主机用于界面和程序的设计编写, 目标机用于运行编译好的程序。主机可以是装有 Windows 或 Linux 等常用操作系统的 PC, 且支持多人同时对同一系统进行开发。本系统的控制界面和程序就是在装有 QNX Momentics IDE 4.0.1 的 Windows XP 操作系统的 PC 上进行设计的。

### 2 软件设计

系统软件设计分 2 个部分: 一部分是系统控制界面设计, QNX 提供了多种方式对控制界面进行个性化设置, 可以在 PhAB 中创建编辑, 也可以使用 QNX 提供的函数进行编写; 另一部分为控制系统程序设计, QNX Momentics IDE 支持多种语言对基于 QNX 的应用程序进行开发, 包括 Java、C 和 C++ 等。

## 2.1 界面设计

控制界面部分的设计主要在 PhAB 中完成。PhAB 以类的方式定义每一个控件，共有 76 个控件类，其中大部分在 PhAB 中都有相应的图标与之对应，控件的总父类为 PtWidget。每个控件类由多个不同的资源(resources)来定义，例如在文本控件(PtText)中，字体的类型、大小、颜色都是该控件的资源。值得注意的是，PhAB 中不允许两个控件拥有同一个名字，并且一个控件被创建后，系统将为该控件指定一个由前缀“ABN\_”加控件名组成的局部变量名和一个由前缀“ABW\_”加控件名组成的指向该控件的指针，当需要调用该控件时，可以直接使用为其分配的局部变量名或指针。

以资源的方式定义控件类提高了控件编辑的效率。QNX 提供了多个函数对控件的资源进行读写操作，常用到的如 PtSetResources()和 PtGetResources()就是对控件类的单个或多个资源进行读写操作的函数。系统中各种控件状态变化频繁，需要经常对控件的一个或几个资源进行更改，如下所示为系统中使某个阀门标识背景图形变换的程序片段：

```
PtArg_t arg[2];           // 定义控件资源的数据结构
PhImage_t *image;
...
/* 以下是对控件资源的读取和设置 */
```

```
PtSetArg(&arg[1],Pt_ARG_ARM_IMAGE,&image,0);
PtGetResources(ABW_Button1,1,&arg[1]);
PtSetArg(&arg[1],Pt_ARG_ARM_IMAGE,image,0);
PtSetResources(ABW_Button2,1,&arg[1]);
PtBkgdHandlerProcess();
...
```

在 PhAB 中，不仅可以将不同的控件组合成一个只具有它们共同属性的组合控件，还可以将个性化的控件类定义为自己的模板，下次需要使用同类型控件时，只需直接调用即可。同时，PhAB 还提供了一个叫做类型转换(Change Class)的强大功能。顾名思义，类型转换就是可以任意改变控件所属的类，而控件的类又是控件所有属性的集合。使用此功能不仅可以随意改变控件的类型，还可以方便地制作形状不同的各类控件。如若制作一个特殊形状的按钮控件，只需把控件的形状在绘图软件上画好后导入 PhAB 中，再使用类型转换的功能将其定义为一个 PtButton 控件就可以了。

图 1 为系统控制界面的截图，图中线条、方框和各种按钮都作为一个或几个控件类而存在，控制面板对系统相关部分进行控制，被控对象的不同状态将以不同的颜色在界面的对应标识上表示。

PhAB 为每个控件提供十余种程序调用方式，常用到的有按下调用(Armed)、点击调用(Activated，点击为一

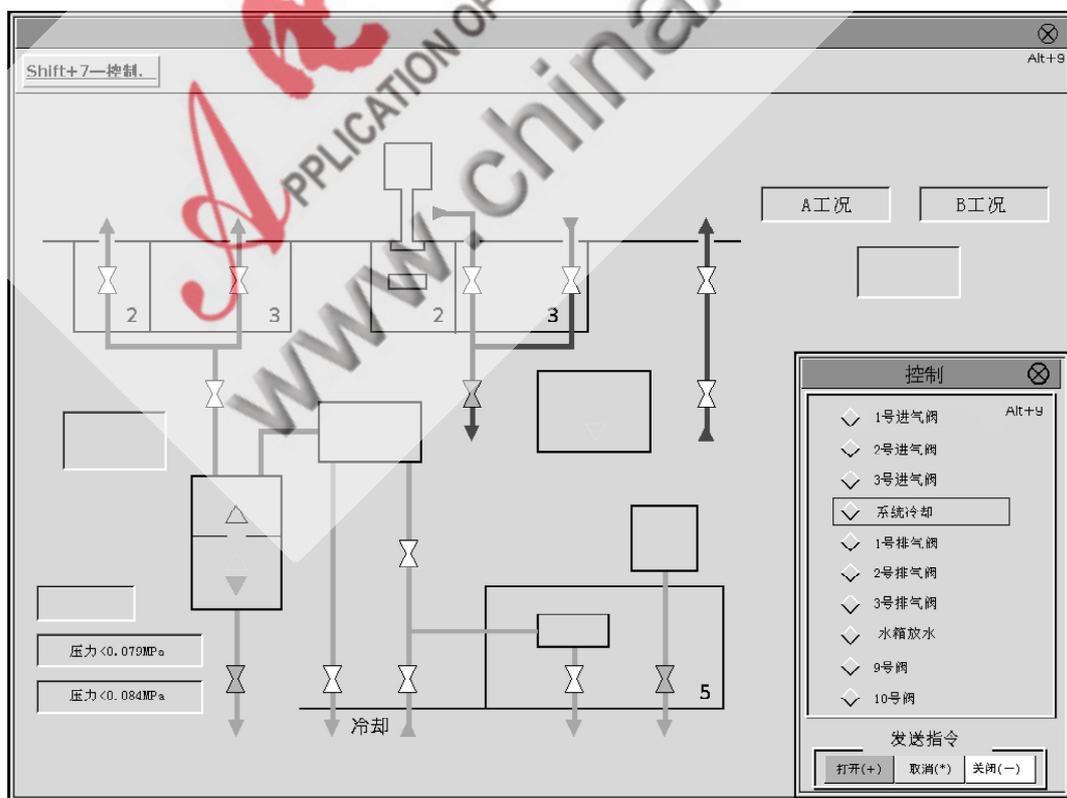


图 1 控制界面截图

## 软件天地 Software Technology

个在同一控件上按下然后松开的过程)、热键调用(Hotkey)和鼠标右键按下调用(Menu)等,系统默认它们分别调用Pt\_CB\_ARM、Pt\_CB\_ACTIVATE、Pt\_CB\_HOTKEY和Pt\_CB\_MENU函数,如果需要调用自定义的函数,可以直接在提供的对话框中输入需调用的函数名和所在文件。当触发的事件是显示一个窗口或对话框时,还可以指定调用是在窗口构建之前还是之后。

### 2.2 程序设计

QNX Momentics IDE支持C/C++语言对系统进行设计,并且兼容C++标准模板库(STL),熟练的C/C++程序员可以很快地掌握在QNX Momentics IDE中进行程序编写的技巧。值得一提的是,在QNX集成开发环境下对图形界面控制系统的主程序进行编写并不用直接对main()函数进行编辑,而是将编写的程序挂载到main()函数中,并可以选择程序是在图形界面生成前运行还是之后运行。系统主程序的方框图如图2所示。

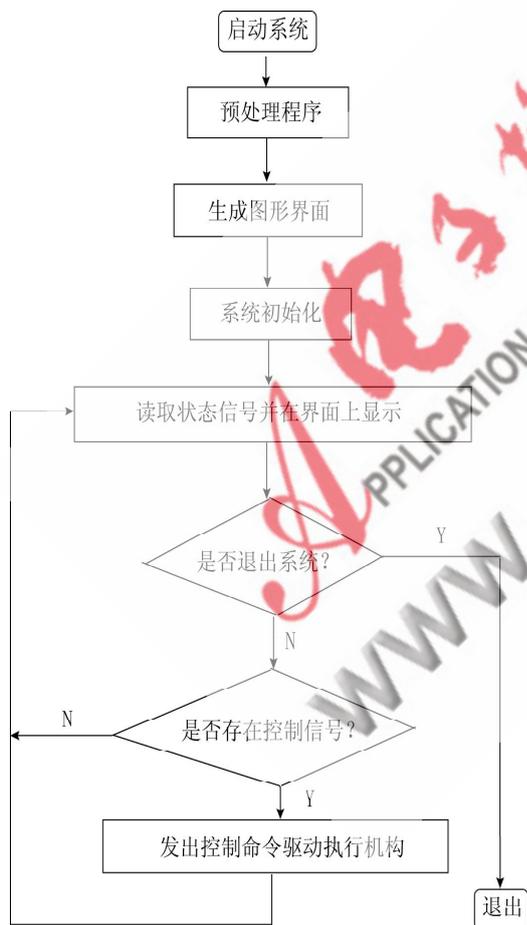


图2 主程序方框图

在程序运行时,预处理命令先于生成图形界面部分执行,它与系统初始化及以后的程序不在同一个编写的函数中,但都是main()函数的一部分。读取并显示状态信号部分程序在执行中会自动检测信号所标识状

态是否达到或超过警报值和危险值,当读取到警报值信号时,系统将自动发出报警;存在危险值信号时,系统将按编写的程序自动处理。

在控制系统运行时,需要同时进行多项运算。如对某个阀发出开阀的控制命令时,系统不仅要显示需更改阀图标的变换(以标识该阀正在进行相关操作),还要进行数据采集、逻辑运算等。若此时某项指标达到警戒值,还要能及时进行相关处理。因此,除了应用中断的方式提高程序运行效率外,还有必要进行多线程编程。

QNX为多线程编程提供了丰富的函数。如线程创建函数pthread\_create()、ThreadCreate\_r()和ThreadCreated()、线程终止函数pthread\_exit()、ThreadDestroy()和ThreadDestroy\_r()等。当然,各个函数的使用有差别,以线程创建函数为例,相比于pthread\_create()函数,ThreadCreate\_r()和ThreadCreated()都是从内核直接创建线程,并且它们的使用方法很相似,ThreadCreate\_r()函数与ThreadCreated()函数的唯一区别在于创建线程出错时的返回值不同。下面是系统中用pthread\_create()函数创建的一个信号处理线程:

```

void* chuli( void* arg )           // 新创建的线程
{
    /* 信号处理程序 */
    return( 0 );
}

int buttondown()
{
    ...

    pthread_attr_t attr;
    /* 线程初始化并进行参数设置 */
    pthread_attr_init( &attr );
    pthread_attr_setdetachstate( &attr,
    PTHREAD_CREATE_DETACHED );
    pthread_create( NULL, &attr, &chuli, NULL );
    // 创建新线程
    ...
}
  
```

在多线程程序的编写过程中,需要使用互斥体来保护线程的数据访问,以保证公共数据不被其他线程修改破坏。在QNX中,可以使用函数pthread\_mutex\_init()来创建一个互斥体,使用函数pthread\_mutex\_lock()和pthread\_mutex\_unlock()来锁住一个互斥体和为一个互斥体解锁。

本系统运行的硬件平台为一台SBS公司的VP9 6U VME Single Board Computer,一块由Acromag公司生产的AVME9668 VME bus 6U Non-intelligent IP Carrier Cards和IP408 High Voltage Digital Input/Output信号采集模块,单板机和信号采集卡都插在基于VME64x总线协议的6U

## 软件天地

Software Technology

标准机箱中运行。长时间的调试运行表明,本文所设计的基于 QNX 操作系统的图形控制软件运行稳定、可靠,实现了系统功能,达到了既定要求。

### 参考文献

- [1] QNX官方网站.QNX SOFTWARE SYSTEMS Ltd[EB/OL].Http://www.qnx.com,2008-10-10.

[2] 曹冰冰.基于QNX实时操作系统的测试系统的开发[N].西安:西北工业大学,2002.

[3] 王冬霞,王茂,单家方.基于QNX的分布式采集控制系统[J].核聚变与等离子体物理,2007,27(4):334-338.