

VxWorks 系统下人机交互界面软件跨平台兼容的研究与应用

Cross-platform Compatibility of Human Machine Interface Software Under VxWorks System

刘泉晶 朱长发 辜鹏 程耀 (上海航天电子通讯设备研究所,上海 201108)

摘要:随着计算机技术的快速发展,人机交互界面(简称 HMI)已在制造业、服务业、航天航空、军工等领域得到广泛应用。针对 XX 型号舰空导弹武器系统研制中提出的在 VxWorks 系统下人机交互界面软件跨平台兼容性要求,采用模块化、参数化设计,在实际双平台下多次测试,并经过系统验证,满足软件跨平台兼容性要求。

关键词:人机交互界面,软件,跨平台,兼容性

Abstract:With the rapid development of computer technology,human-machine interface (HMI)has been widely used in manufacturing,service,aerospace,military and other fields.In this design,for the cross-platform compatibility requirements of human-machine interface software under the VxWorks system proposed in the development of the XX-type AJK weapon system,this paper adopts modular and parametric design and test many times under the actual dual platform.And it verified to meet the cross-platform compatibility requirements of the software.

Keywords:human machine interface,software,cross-platform,compatibility

人机交互界面的开发在一套软件中是与系统平台联系最为紧密的一部分,其工作量往往占整个开发工作的很大一部分。针对同功能的程序能够运行在不同的系统平台,并保持一致的界面和功能,从而增强程序的灵活性和可移植性,这种程序的跨平台性越来越受到重视。

在本次设计中,针对 XX 型舰空导弹武器系统研制中提出的在 VxWorks 系统下人机交互界面软件跨平台兼容性要求,采用模块化、参数化设计,在实际双平台下多次测试,并经过系统验证,满足软件跨平台兼容性要求。

1 研究背景

在 XX 型舰空导弹武器系统研制中,根据要求,人机交互界面软件需要在新、老两代显控平台(简称 1#台、2#台)上兼容,且保持一致的界面和功能。在通常情况下,对于不同显控平台,由于硬件环境不同,比如底层驱动接口、显示分辨率等不同,导致人机交互界面软件很难在不同平台上正常运行与工作,具体硬件环境对比如表 1 所示。

表 1 1#台、2#台硬件环境对比表

项目	1#台	2#台
显示	显示分辨率: 1280*1024 支持 SVGA 接口,输出的信号能适应各类加固平板显示器	显示分辨率: 1600*1200 采用 ATI M22 独立显示芯片,支持双路 DVI 信号输出
底层驱动加载	VXLibInit()	VXLibInit()
	显示开窗 主显: mainDisplay=XopenDisplay("1",17) 副显: auxDisplay=XopenDisplay("2",17)	主副显: TheDisplay=XopenDisplay("2",17)
	绘图 Xlib.h	Xlib.h
	P 显 Opening 等	ML_GVID_OpenPPI 等
	B 显 XOpenBD 等	ML_GVID_OpenPPI 等
	视频 XOpenTv 等	ML_GVID_OpenTV 等
	按键 mmiHotkeyGet 等	ML_SOCM_NextEvent_N 等
	摸球 auxDevInit 等	CursorInit 等
触摸屏 mmiTouchGet 等	ML_SOCM_Init_N 等	

2 人机交互界面软件跨平台兼容方法的探讨

针对 XX 型号舰空导弹武器系统研制中,要求人机交互界面软件在两台不同硬件环境的显控平台上兼容性问题,本文提出一种 VxWorks 系统环境下人机交互界面软件跨平台兼容控制的方法,即人机交互界面软件可在两台不同硬件环境的显控

台(简称 1#台、2#台)下,通过读取配置文件,初始化相应参数表及底层驱动接口等,实现人机交互界面软件跨平台运行。针对底层驱动不同问题,人机交互界面软件通过读取系统配置文件,识别人机交互界面软件当前运行平台,从而进入不同底层驱动初始化分支,比如显示开窗、按键、摸球、P 显、B 显、触摸屏、电视等初始化工作;针对不同显控平台不同分辨率问题,人机交互界面软件通过读取系统配置文件,识别人机交互界面软件当前运行平台,从而进行相应参数表初始化,以适应不同分辨率显示问题。初始化工作完成后,进行静态界面绘制、BIT 检测、网络报文接收、网络报文发送、界面动态显示刷新、操控响应等相关工作,具体流程如图 1 所示。软件采用模块化、通用化、参数化设计,大大缩短了软件开发周期,具备软件框架简洁、便于移植、通用性强、可维护性高等特点。

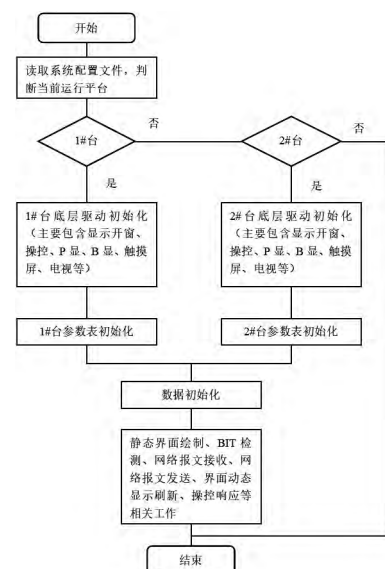


图 1 人机交互界面软件流程图

3 人机交互界面软件跨平台兼容的应用

针对 XX 型号舰空导弹武器系统研制中,要求人机交互界

面软件在两台不同硬件环境的显控平台上相互兼容的问题,且考虑以后其他相似型号研制的可移植性、可扩展性,采用的设计原则是模块化、参数化设计。对于两个显控平台相同的部分,比如界面显示、报文发送、报文接收等功能,进行模块化、参数化处理,对于不同的部分,比如底层驱动接口的加载等,进行分支处理,人机交互界面软件框图如图2所示。

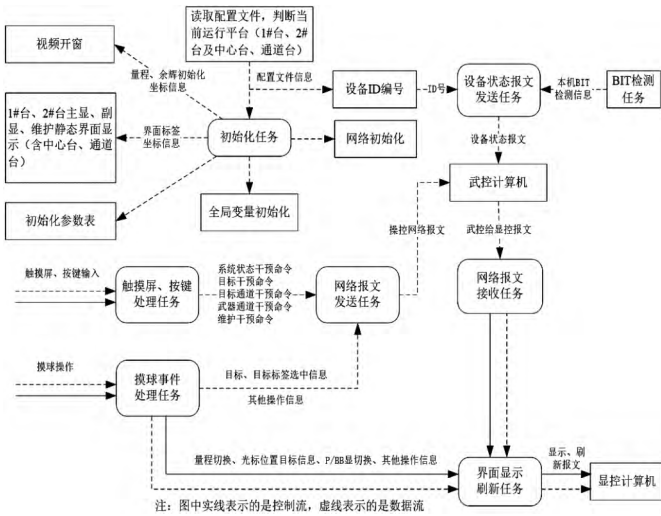


图2 人机交互界面软件框图

具体软件实现步骤:

1) 读取系统配置文件 config.dat, 根据返回值 con 判断当前运行平台, con=11 表示 1# 台中心台, con=12 表示 1# 台通道台, con=21 表示 2# 台中心台, con=22 表示 2# 台通道台;

2) 若为 1# 台, 则进入 1# 台底层驱动初始化 init1(), 主要包含显示开窗、按键、摸球、P 显、B 显、触摸屏、电视等初始化工作, 若为 2# 台, 则进入 2# 台底层驱动初始化 init2());

3) 参数表初始化 table_init(), 参数表主要用于解决显示分辨率不同问题, 由于采用模块化、参数化设计, 只需根据不同的显示分辨率, 对显示内容进行布局, 将尺寸大小、坐标等信息参数化, 填入相应的参数表位置即可; 比如, 要在界面上绘制一个静态的显示标签, 界面显示内容、全局变量定义、参数表初始化及静态界面绘制如图3所示;

4) 全局变量初始化 init_global(), 主显静态界面 drawmain(), 副显静态界面 drawaux(), 维护界面 drawWF(), 网络初始化 NetInit()等;

5) VxWorks 系统多任务处理, 主要包括 BIT 检测、设备状态报文发送任务、网络报文接收任务、网络报文发送任务、操控响应任务(包含按键、摸球、触摸屏)、显示刷新任务等。

4 结束语

目前, 此次开发的人机交互界面软件已在 XX 型号舰空导弹武器系统中得到应用。在实际双平台环境下多次测试, 并经系统验证, 该软件设计合理可行, 满足跨平台兼容控制的要求, 具有软件框架简洁、便于移植、通用性强、可维护性高等特点, 可广泛应用于同类的项目开发中, 可缩短软件开发周期及可靠性。

参考文献

[1]李艳民.基于QT跨平台的人机交互界面的研究与应用[D].重庆:重庆大学,2007
[2]王培铎.计算机人机交互界面的变化与计算机的发展[J].武警学院学报,2001,17(6):59-61
[3]杨萧.基于QT的富客户端软件设计与实现[D].成都:电子科技大学

界面显示内容

Table with 3 columns: 量程, P显, B显

全局变量定义

```
typedef struct TabF/*标签结构体*/
{
    Display *Dis;/*主副显标志*/
    Window win; /*开窗标志*/
    GC GCA; /*GC标志*/
    int x; /*标签左上角x坐标*/
    int y; /*标签左上角y坐标*/
    int w; /*标签单元宽度*/
    int h; /*标签单元高度*/
    int l_num; /*行数*/
    int c_num; /*列数*/
    unsigned long cor; /*设置前景色*/
    unsigned long cor1; /*设置背景色*/
};
TB Tba0;
```

参数表初始化

```
if((con==11)|| (con==12))/*1#台*/
{
    Tba0.Dis=mainDisplay;
    Tba0.win=win1;
    Tba0.GCA=mainGC;
    Tba0.x=50;
    Tba0.y=50;
    Tba0.w=100;
    Tba0.h=30;
    Tba0.l_num=1;
    Tba0.c_num=3;
    Tba0.cor=WHITE;
    Tba0.cor1=BLACK;
}
else if((con==21)|| (con==22))/*2#台*/
{
    Tba0.Dis=theDisplay2;
    Tba0.win=win2;
    Tba0.GCA=mainGC2;
    Tba0.x=60;
    Tba0.y=60;
    Tba0.w=150;
    Tba0.h=40;
    Tba0.l_num=1;
    Tba0.c_num=3;
    Tba0.cor=WHITE;
    Tba0.cor1=BLACK;
}
else
{
}
```

静态界面绘制

```
void table_flag(TB *TBO)/*画标签*/
{
    int i,j;
    for(i=0;i<TBO->l_num;i++)/*行*/
    {
        for(j=0;j<TBO->c_num;j++)/*列*/
        {
            XSetForeground(TBO->Dis,TBO->GCA,TBO->cor);/*设置前景色*/
            XDrawRectangle(TBO->Dis,TBO->win,TBO->GCA,TBO->x+TBO->w*j,TBO->y+TBO->h*i,TBO->w,TBO->h);
        }
    }
    /*字符串所在行*/
    /*字符串所在列*/
    /*dx:字符串左起x偏移量*/
    /*dy:字符串y偏移量*/
    /*str:显示内容*/
    /*num:字下数*/
    /*color1:前景色*/
    /*color1:背景色*/
}
void table_string(TB *TBO,int l,int c,int dx,int dy,char *str,int num,
unsigned long color1,unsigned long color2)/*写字符串*/
{
    XSetForeground(TBO->Dis,TBO->GCA,TBO->cor);/*设置前景色*/
    XSetBackground(TBO->Dis,TBO->GCA,TBO->cor);/*设置背景色*/
    XDrawString16(TBO->Dis,TBO->win,TBO->GCA,TBO->x+TBO->w*(c-1)+dx,TBO->y+TBO->h*(l-1)+dy,str,num);/*写字符串*/
}
void draw_table(void)
{
    int i;
    char *str={"量程","P显","B显"};
    table_flag(Tba0);/*画标签*/
    for(i=0;i<3;i++)
    {
        table_string(&Tba0,i+1,Tba0.w-6*strlen(str[i])/2,(Tba0.h-16)/2,
        &C[i],strlen(str[i]),WHITE,BLACK);
    }
}
```

图3 静态的标签显示示例图

学,2009
[4]呼健.人机交互界面设计与评估技术的研究和应用[D].济南:山东大学,2005