

文章编号：1005-8451（2005）12-0023-04

## 基于嵌入式 Linux 的列控系统车载人机界面的实现

任亚飞，唐 涛

（北京交通大学 电子信息工程学院，北京 100044）

**摘要：**随着铁路的高速发展，传统的车载人机界面已不能满足铁路安全至上的要求。所以尝试引入运行于嵌入式 Linux 下嵌入式图形系统 MiniGUI 作为新的图形开发工具。介绍嵌入式 Linux 的特点以及环境建立的方法，然后详细地介绍嵌入式图形系统 MiniGUI，最后对车载人机界面进行规划，并设计在 MiniGUI 上的实现方法。

**关键词：**列车运行控制系统；嵌入式 Linux；车载人机界面；MiniGUI；ERTMS

中图分类号：TP39 文献标识码：A

### Implementation of Train Operation Control System on board MMI based on embedded Linux

REN Ya-fei, TANG Tao

(School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

**Abstract:** With the rapid development of railway, traditional MMI(man-machine interface) could not still meet the requirement of the railway. So we were here to attempt embedded linux. It was firstly introduced the characters of embedded linux and the methods of building linux, then detailedly depicted MiniGUI, at last, designed the onboard MMI, and planed the implement methods based on MiniGUI.

**Key words:** Train Operation Control System ; embedded linux; MMI; MiniGUI; ERTMS

列车运行控制系统（简称列控系统）是由列控中心、闭塞设备、车载速度控制设备等构成的用于控制列车运行速度保证行车安全和提高运输能力的控制系统。列车在工作环境下工作的列控系统，对

收稿日期：2005-07-01

作者简介：任亚飞，在读硕士研究生；唐 涛，教授。

系统的安全性、可靠性、可用性的要求非常高，必须保证系统能安全、可靠、不间断地工作。

列车运行控制系统车载人机界面是车载设备与驾驶员进行信息交互的平台，是列控系统的一个重要组成部分。车载人机界面在机车运行的过程中向驾驶员或维护人员传达着有关机车状态的重要信

夹、结算信息文件夹、结算信息备份文件夹，用于存储和备份数据。

#### 3.2.2 传输格式

制票点数据采用预付款抵用信息和货票摘要信息共享文件格式传输，铁路局转发数据采用大客户资金信息和票据结算信息共享文件格式传输。文件命名格式避免了文件名冲突带来的风险，文件内容设计预留了扩充的余地。两种数据传输过程中，都是以一个文本文件承载两块业务信息，并采用特定的格式设计，使程序能把各业务信息分别读取到相应表中。

#### 3.3 数据访问技术

数据存储平台采用 Oracle 数据库管理软件，结

算机上需要安装 Oracle 客户端软件，并按应用程序的要求配置 Oracle SQL\*Net。

应用程序通过 ADO 技术中连接对象 ADO.Connection 与数据库建立连接。数据库连接字符串通过参数文件实现灵活的配置。

## 4 结束语

目前，该系统的车站级子系统按照本文设计思路已经完成编程、集成测试工作，并为铁道部运输局、财务司、结算中心、信息中心领导和相关部门进行了演示。下一步将选择车站进行试点，在优化完善的基础上实施推广。

息，通过车载人机界面，司机可以实时地得到有关列车和线路的相关状态和数据，对车载设备发出的命令和警告及时地做出响应，因此对列车的安全运行具有极高的价值。

本论文首先按照系统的需求搭建了硬件平台，然后参照欧洲铁路运输管理系统（ERTMS）中的操作界面信息对人机界面进行了规划和设计，最后选用了一种优秀的图形界面系统 MiniGUI 作为图形开发工具并进行了实现。

## 1 系统整体设计方案

系统的目地是搭建一个人机接口即人机界面，司机的操作任务主要是通过人机接口完成司机与车载系统的信息交换。

人机界面运行的硬件平台是一款 arm9 开发板，采用的是三星 S3C2410X 嵌入式 ARM9 处理器，带有 64 M 的 SDRAM，64 M 的 Nand flash，2 M 的 Nor flash 以及带有触摸屏的 LCD 显示器（用于显示人机界面）。开发板通过串口和车载设备进行通信，司机通过触摸屏输入数据。

为在车载 LCD 上实现人机界面的显示，整个系统将由 Linux 操作系统负责底层硬件的初始化和管理，并向上层提供应用接口，MiniGUI 来完成人机界面的绘制，并通过 Linux 提供的串口接口与车载设备完成通信功能。

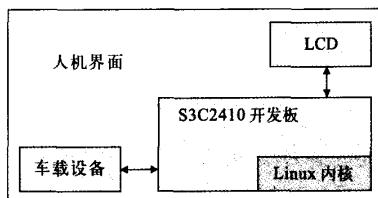


图 1 人机界面整体设计方案

## 2 操作系统的选择

### 2.1 操作流程比较

车载人机界面要求上电时间短以及列车自身的特点决定了在此只能选择嵌入式操作系统作为人机界面运行的平台。目前广泛流行的嵌入式系统主要有 3 种：VxWorks、WinCE 和嵌入式 Linux。这 3 种系统都能满足系统对于稳定性和可靠性的要求。在此选择嵌入式 Linux 主要是基于 Linux 的自身特点：(1) 完全开放源代码，这样嵌入式系统的设计者就可以对嵌入式 Linux 进行二次开发，只保留必须的操作系统功能，并可以根据实际应用的需要优化操作系统的代码；(2) 成本低，嵌入式 Linux 的使用费用相对来说较低；(3) 丰富的实用软件支持<sup>[1]</sup>。

本系统中采用了 Mizi Linux 嵌入式系统，它提供了一整套的针对 ARM 的开发工具，便于应用程序的开发。

### 2.2 Mizi Linux

MIZI Linux 是一种嵌入式 Linux 操作系统，包含有一套完整的应用程序包。它被设计用于各种数字设备，例如电话，SMS，网页浏览器，eMail 客户端，PIMS，媒体播放器，文档管理器，游戏，以及其它由我们自己开发的 GUI 应用。

## 3 车载人机界面的开发

### 3.1 GUI 的选择

由于本系统主要是人机界面的显示，所以图形用户界面系统的选地也是非常重要的。目前比较成熟的，同时得到开发人员认可的针对嵌入式系统的图形用户系统主要有 MicroWindows，MiniGUI，OpenGUI，QT/Embedded 等 4 个系统，各个系统的性能如表 1 所示。

其中面向基于 Linux 的实时嵌入式系统，使用成熟的图形引擎，实现了简化的类 Windows 98 风格的图形用户界面，具有轻型、占用资源少，高性能，高可靠性，可配置的特点。

考虑到列车运行控制系统的高可靠性，高安全

表 1 几种主要的嵌入式 GUI 的比较

名称	API (完备性)	函数库容量 / kB	移植性	系统消耗	操作系统支持
MicorGUI	Win32 (很完备)	300	很好	小	Liunx
MicroWindow	x-Win32 (不完备)	300	很好	较大	Liunx.DOS.ELKS
OpenGUI	私有 (很完备)	300	只支持 X86 平台	最小	DOS.Liunx.QNX
Qt/Embedded	Qt (C++) (很完备)	600	较好	最大	Liunx

性以及它对实时性的需求，在此采用了 MiniGUI。

### 3.2 MiniGUI 的交叉编译

在此使用的 MiniGUI 是 1.3.3 版，它包含资源文件压缩包 minigui-res-1.3.3.tar.gz、函数库文件 libminigui-1.3.3.tar.gz。

(1) 首先应安装交叉编译器，在此使用的是 arm 交叉编译器工具链：cross-2.95.3.tar.bz2，安装完后，修改 PATH 变量，增加 /usr/local/arm/2.95.3/bin。

#### (2) 安装 MiniGUI 的资源文件

```
#tar -xvf minigui-res-1.3.3.tar.gz
```

```
#cd minigui-res-1.3.3
```

```
#make install
```

#### (3) 安装库文件

在前面已经安装过了交叉编译环境，所以这里要使用交叉编译器：

```
#CC=arm-linux-gcc
```

```
./configure --target=arm-linux --prefix=/usr/local/
arm/2.95.3/arm-linux --bulid=i686-linux
```

在此指定了 arm-linux-gcc 为它的交叉编译器，

```
#make
```

```
#make install
```

至此，MiniGUI 的安装就完成了。

### 3.3 人机界面的设计和实现

应用程序是最上层的开发，其交互界面直接通过 MiniGUI 图形系统的 API 接口函数实现。MiniGUI 屏蔽了对底层显示、输入设备编程的细节，使设计者更能专注于车载人机界面的功能特点上，从而缩短了编程投入时间。

#### 3.3.1 软件规划

此人机界面是参照 ERTMS 中的操作界面信息来规划和设计的。ERTMS 的操作界面规定了信息在显示器上的画面显示和安排的要求，从而使提供给司机的信息能够清晰地显示，有助于更好地了解要完成的任务。这样可以提高速度和精确性，从而减少人为失误的可能性。其基本结构如图 2。

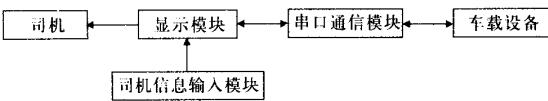


图 2 人机界面软件基本结构

显示模块分为两个部分：司机信息显示界面和主显示。在系统上电后，首先将进入的是司机信息输入界面，司机在根据调度信号起动列车之前，将

通过信息输入界面，将司机 ID，列车编号，列车长度和最高速度等信息通过触摸屏输入，并保存起来，输入完信息后，进入主显示界面。

列控系统通过主显示界面向司机实时返回行车目标速度、目标距离、当前速度等信息显示，在此，我们把相关的信息放在一起进行显示，分成了 5 个部分；它们的具体分布见图 3 所示。速度信息包含列车目标速度，允许速度，缓解速度和当前速度等相关速度信息；计划信息提供司机选定的距离区域内的提示信息，例如车站的位置，坡度信息，起始位置信息，速度曲线相关信息和地理信息，并且能够显示最大限制曲线。列车进行制动时，监督距离信息类提供目标停车点的预告距离，干涉预告时间信息。车载设备向司机发出的命令以及司机进行的人工干预，通过辅助驾驶信息和监测信息来实现。这 5 部分负责各自的信息显示，在主人机界面中分成了 5 个相对集中的部分。

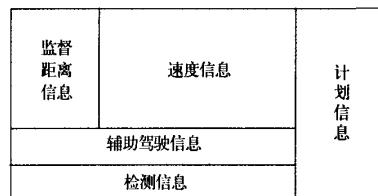


图 3 主显示界面规划

由于在显示模块中将信息分为 5 部分，所以在串口通信中将一帧信息分为 5 个信息包：速度信息包，计划信息包，监督距离信息包，辅助驾驶信息包和检测信息包。每个包都有自己的包头作为包的标识，主显示模块通过串口接收帧数据，通过包标识来识别出各个信息包，然后将不同的信息送到各自对应的部分。

#### 3.3.2 软件的实现

本界面采用的是 MiniGUI 的 Threads 技术，其中司机信息输入部分采用 MiniGUI 的对话框模版，主显示界面使用主窗口方式，并采用定时技术来进行串口信息查询，从而及时地将信息传递给司机。

首先建立的是司机信息输入部分，它采用对话框模版，程序入口点是 MiniGUIMain 函数，系统初始化后就会自动找到该函数。在该函数中通过下面两条语句将对话框与控件以及它的处理函数联系起来。

```
DlgMsgInput.controls = CtrlMsgInput;
DialogBoxIndirectParam (&DlgMsgInput,
HWND_DESKTOP, DialogBoxProc, 0L);
```

注：DlgMsgInput 定义为对话框本身，CtrlMsgInput 在此用来定义控件，DialogBoxProc 为对话框处理函数。

在此司机信息输入界面中，由于最后采用的是通过触摸屏输入，所以只能采用通过 10 个代表 0~9 的按钮控件来实现输入。下面给出了数字输入处理函数的示例：

```
// 数字输入处理函数
static void number_proc(HWND hwnd, int id, int nc, DWORD add_data)
{
    HWND hWnd1, hWnd2, hWnd3, hWnd4;
    hWnd1 = GetDlgItem(GetParent(hwnd), IDC_DRIVER); // 获得IDC_DRIVER的句柄
    hWnd2 = GetDlgItem(GetParent(hwnd), IDC_TRAIN);
    hWnd3 = GetDlgItem(GetParent(hwnd), IDC_LENGTH);
    hWnd4 = GetDlgItem(GetParent(hwnd), IDC_SPEED);
    if((id <= 110) && (id >= 101) && (nc == BN_CLICKED) && ((hWnd_Edit == hWnd1) || (hWnd_Edit == hWnd2) || (hWnd_Edit == hWnd3) || (hWnd_Edit == hWnd4))){
        char buff[] = "0123456789";
        switch(id)
        {
            case 101:
                SendMessage(hWnd_Edit, MSG_CHAR, buff[0], 0L); // 向 hWnd_Edit 输入数字 '0'
                SetFocusChild(hWnd_Edit); // 确定输入光标
                break;
            case 102:
                SendMessage(hWnd_Edit, MSG_CHAR, buff[1], 0L); // 向 hWnd_Edit 输入数字 '1'
                SetFocusChild(hWnd_Edit);
                break;
            .....
        }
    }
}
```

在对话框建立好后，下面要建立的是主显示界面。同样它的入口函数也是 MiniGUI 函数，在此按

照界面的静态部分和动态部分来划分：也就是固定不变的界面信息由一个专门的函数来负责，然后将再在此函数中将按照上面的划分把固定信息划分为 5 个部分，并由 5 个不同的函数来负责；动态部分也就是经常变化的信息，也由一个专门的函数来处理，它使用定时器，每隔一定的时间就刷新一次动态信息，在此，定时器采用 MiniGUI 提供的 SetTimer() 函数，当定时器的时间消耗完时，它会向窗口发送 MSG\_TIMER 消息，在此消息处理函数中，将调用串口接收函数来接收数据，并对其进行处理。

接下来简要介绍一下串口通信部分。Linux 通过设备文件访问串口。为访问串口，仅需打开相应的设备文件即可。车载人机界面的串口规则简要说明：(1) 约定串口设备文件系统为 /dev/ttyS0；(2) 约定串口通信波特率为 B115200，8 位数据位，1 位停止位，无奇偶校验；(3) 每帧数据采用 CRC 校验，使用国际标准 CCITT-16 的生成多项式作为本信息帧的 CRC 生成多项式。

在程序编写完后，使用交叉编译器编译该程序，并链接 MiniGUI 库文件，例如：

```
#arm-linux-gcc -o EnterMsg EnterMsg.c -lpthread -lminigui -llibtff，即生成可执行文件，然后将资源文件、头文件和生成的可执行文件一起下载到目标板上即可。
```

## 4 结束语

安全一直是铁路发展的主题，在铁路高速发展的今天也不例外。本文尝试将嵌入式 Linux 以及 MiniGUI 引入到列控系统车载人机界面中，根本的原因也正是 Linux 的稳定性能满足目前甚至将来铁路发展的需求。当然，嵌入式 Linux 在铁路上的应用应不仅仅限于人机界面，在将来我们会看到 Linux 在铁路越来越多方面的应用，这是必然趋势。

## 参考文献：

- [1] 古丰. 嵌入式 Linux 系统平台的搭建[J]. 电脑与信息技术, 2003 (6): 54—57.
- [2] 李志刚, 周兴社. 嵌入式 Linux 集成开发环境的设计及实现[J]. 计算机工程及应用, 2003 (35): 92—93.
- [3] 潘明泉, 张侃渝. 嵌入式 Linux 下基于 MiniGUI 的信息终端软件开发[J]. 单片机与嵌入式系统应用, 2004 (4): 29—31, 52.