

基于嵌入式 Linux 和 QTE 的中文 GUI 平台实现^①

Design and Implementation of Chinese GUI System Based on Embedded Linux and QTE

师晓敏 朱名日 葛清志 (桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘要: 嵌入式 Linux 是目前流行的嵌入式系统解决方案,而嵌入式 GUI 是嵌入式 Linux 不可缺少的组成部分。随着嵌入式系统的广泛应用,嵌入式系统平台对于中文环境支持的需求日益迫切,文章以自动煮糖控制系统的中文 GUI 平台的实现为例,讨论了嵌入式系统中文环境的现状,阐述了如何在嵌入式 Linux 系统平台上利用 QTE(QT/Embedded)实现中文显示和输入,并构建出符合应用要求的中文环境。从而大大拓宽了嵌入式技术在自动煮糖控制系统中的应用。

关键词: 嵌入式 Linux GUI 国际化 本地化 中文环境 QTE(QT/Embedded) 煮糖控制

1 引言

随着嵌入式系统在信息家电、移动通信、手持信息设备以及工业控制等众多领域的广泛应用,为了提供更好的人机交互,越来越多的嵌入式终端需要一个图形化的人机接口界面(GUI),用户对于嵌入式系统的轻量级 GUI 支持以及中文环境的需求日益迫切。良好的人机接口界面是嵌入式系统设计的一个关键技术,能够极大地提高人机交互的效率。

中文环境作为 GUI 的一个重要组成部分,可以进一步提高人机交互程度。中文平台或者说中文环境对于中文信息处理最基本的要求就是中文显示、中文输入以及中文打印支持。考虑到嵌入式系统通常的应用需求,本文以自动煮糖控制系统的 GUI 平台实现为例,详细讨论中文显示和输入。

2 系统平台介绍

根据系统设计需求,本文目的是实现一个具有中文显示的参数设置界面和中文输入功能的嵌入式终端,该系统使用嵌入式系统设计技术。硬件上,使用基于 Samsung S3C2410A CPU 的目标板,该 CPU 使用 arm920T 内核,其主频可达 200Mhz;在软件上,选择嵌入式 Linux 为操作系统,因为它源码开放,而且稳定性

与安全性较高。

整个系统软件由引导装载程序(vivibootloader)、设备驱动(包括帧缓存 framebuffer)、嵌入式 Linux 内核、文件系统(yaffs)、基于 QT/Embedded 和 Qtopia 的用户图形界面以及应用程序组成,系统平台结构如图 1 所示。

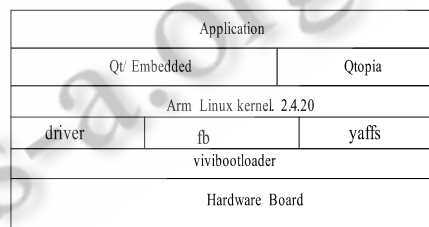


图 1 系统平台结构

3 嵌入式 Linux 下 GUI 现状分析及其开发环境介绍

3.1 嵌入 Linux 下 GUI 现状分析

嵌入式 Linux 是一种流行的嵌入式系统平台,进行嵌入式 Linux 开发,一般有以下几种 GUI 系统可供选择:紧缩的 X Window 系统,MiniGUI, Micro Windows, OpenGUI,QT/Embedded 等。当前嵌入式 Linux 的 GUI 正日益完善,但作为嵌入式 GUI 的一个重要组成部

① 基金项目:广西区科技攻关项目(桂科攻:0428007-9)

分——中文环境的支持,却一直进展缓慢,重要原因之一是在 Linux 设计之初,并未考虑到双字节编码的语言环境,虽然现在国内学者进行了大量的中文平台的研究,但这些都未能从根本上解决问题。而具体到嵌入式 Linux,由于资源有限,不可能搭建传统的如基于 X Window 的中文平台。因此利用嵌入式系统中有限的系统资源实现中文 GUI 平台成为嵌入式应用中亟需解决的问题。本文选用了 QTE,针对如何在嵌入式 Linux 上实现中文 GUI 控制平台进行阐述。

3.2 选择 Qt/Embedded 和 Qtopia 的依据

(1) Qt/Embedded

Qt/Embedded 是 Qt 产品家族中专门面向嵌入式应用开发平台的版本,是一个多平台的 C++ 图形用户界面应用程序框架,其对象容易扩展,可移植性好,支持多个 GUI 平台的交互开发。QT/Embedded 可以支持任何可以运行 framebuffer 的嵌入式设备。而 Linux 内核 2.2 以上的版本都提供了对 framebuffer 的支持。通过支持 framebuffer,QT/Embedded 降低了对运行环境资源的要求,这为它在嵌入式系统中的广泛应用提供了良好的保障。同时,它支持 Unicode 编码,支持任何语言和字符集,遵循 i18n (internationalization) 和 l10n (localization) 标准的发展路线,这都有利于国际化,从而有利于应用的中文化。

因此,本设计选择 Qt/Embedded 来开发嵌入式中文 GUI 控制平台。

Qt/Embedded 的实现结构如图 2 所示:

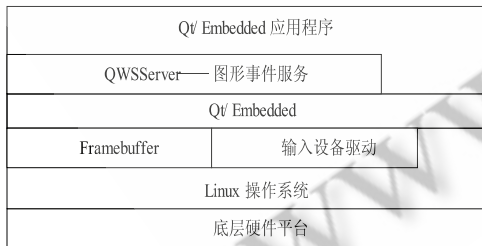


图 2 Qt/Embedded 实现结构

(2) Qtopia

Qtopia 是 Trolltech 公司为采用嵌入式 Linux 操作系统的设备开发的应用平台,它提供给开发者一个面向对象的 API (Application Programming Interface) 来给嵌入式设备编写应用程序,这些 API 与 QT/X11 下是一致的,因而可以把在 PC 上开发出来的软件经过新的交

叉编译后下到嵌入式设备上运行。Qtopia 平台由 Qtopia 库 (Qt/E, libqpe, libqtopia1, qtopiapim) 和 Qtopia launcher/server 组成。Qtopia launcher/server 是控制窗口系统、进程间通信、发起所有应用和其他核心任务的主要服务程序。

4 中文显示

中文显示是设计嵌入式 Qt 中文 GUI 平台最重要的部分之一。最常用的一种是通过 Qt 的国际化来实现中文显示。

4.1 环境设置

在嵌入式 Qt 下解决中文显示问题,环境设置可以分两步完成:

第一步,是解决中文字体(字型)的问题。

嵌入式 Qt 可以支持 TTF、PFA/PFB、BDF、QPF 四种格式。QPF 是 Qt/Embedded 专用的一种适合嵌入式应用的字体,QPF 是一种相当经济的字体存储格式,如果把 BDF 转换成 QPF 格式的话,可以节省一半以上的空间。它属于位图字体,不可以缩放。默认情况下 Qt/Embedded 在 lib/fonts 目录下提供了一种可以显示中文的字体库 UniFont,为 QPF 字体。

第二步,是正确设置处理语系的 codec 引擎(编码译码成 Unicode)。由于嵌入式 Qt 完整地支持 Unicode 编码,所以处理语系的 codec 引擎就变得相当重要,如果缺少处理 GBK 的 codec 引擎,即便有中文字体也无能为力。同样,中文字体的显示也必须依靠 Unicode 编码。此步的实现依靠 QTE 的库编译进对中文 textcodec 的支持,可通过修改 qconfig - qpe. h, 去掉 CODEC 有关的宏,重新 configure 和 make 来实现。

4.2 中文显示的实现

解决中文显示问题是一个软件的本地化过程,是在国际化编程基础上的一个提取翻译信息,翻译软件所使用界面的语言数据的过程。

标准的国际化编程应该分为两个过程。首先在程序中标注待翻译的字符串。Qt 使用了类似 GNU gettext 一样的函数 QObject::tr(), 它用于从 Qt 的翻译信息文件 *.qm 中取出信息。所有在程序中用 tr 标注的字符串,程序在运行时都会动态提取翻译信息进行翻译。如在系统参数设置界面里用到的 TextLabel1 -> setText(tr(" setting system parameters"));

对于绝大多数情况,可以用上述方法标注待翻译字符串。对于一些特殊情况,方法上还是会有所不同,可以参考 QT 参考文档中的 QT 国际化编程一文,这里不再赘述。

其次,是为程序建立翻译信息文件,翻译和动态加载的过程。下面以自动煮糖控制系统中的系统参数设置界面的设计为例,说明该步实现的过程。

(1) 在 `design.pro` 文件里面加入 `TRANSLATIONS = design.ts`。

(2) 用 `lupdate` 操作 `pro`,以提取出 `.ts` 文件供下面的 `linguist` 工具翻译成汉化所需要的 `.qm` 文件。命令是 `#lupdate design.pro`。`*.ts` 文件是翻译源文件,它是基于 XML 语言描述的。通过编辑 `*.ts` 文件,已经指定了对应的翻译信息。这一步是翻译的过程,所谓翻译就是把 `tr("english")` 中的 `english` 提取出来,以翻译成相应的语言,供程序加载使用。

(3) 用 `linguist` 打开刚才的 `design.ts` 文件,`linguist` 是在 `qt` 的 `bin` 的目录下的一个界面工具。如果使用 QT/E 2.3.7 的版本,没有自带这一个工具,可以在 windows 下单独下载 `linguist` 工具进行这一步。在 `linguist` 中用菜单栏 `file -> open` 打开相应的 `.ts` 文件,如该设计中的 `design.ts`。打开后会看到左边是相应的类,右边的上半部是相应的类里面提取出来供翻译的内容,下半部是要翻译的语言的相应的东西,即为需要输入中文的地方。在翻译中,要注意标点符号的翻译最好还是用英文输入状态下的标点符号。

(4) 用 `linguist` 菜单保存翻译好的 `.ts` 文件,接着用 `linguist` 界面工具里面菜单 `file` 里面的 `release...`,点击这个弹出对话框,提示输入 `.qm` 文件的文件名,用默认的即可。此以 `.qm` 为后缀名的文件,即为程序翻译器待用文件。

(5) 最后,把 `design.qm` 文件 `copy` 到 `design` 目录下,在 `main.cpp` 程序里面使用刚才得到的 `.qm` 文件,方法如下:

```
QTranslator translator( 0 ); //生成翻译器
translator.load( " design.qm", "." ); //载入
qm 文件,与刚才得到的 qm 的文件名对应
app.installTranslator( &translator ); //安装翻译器
MyWidget m;
m.setFont( QFont( " unifont", 16 ) ); //设置中文
```

字体

```
app.setMainWidget( &m ); // 将 m 设为当前窗口
```

```
m.show(); //将当前窗口显示出来
```

程序在运行时是动态提取和源语言对应的本地语言信息并显示在用户界面上。设计中一定要在界面出来之前安装翻译器,否则就不能在界面中显示翻译后的界面。

在自动煮糖控制系统的参数设置界面设计中,使用了 `qt-embedded-2.3.7` 中的 `Qt Designer` 完成其 GUI 图形界面的设计,选用了系统自带的 `unifont` 字体,参数设置界面如图 3 所示:

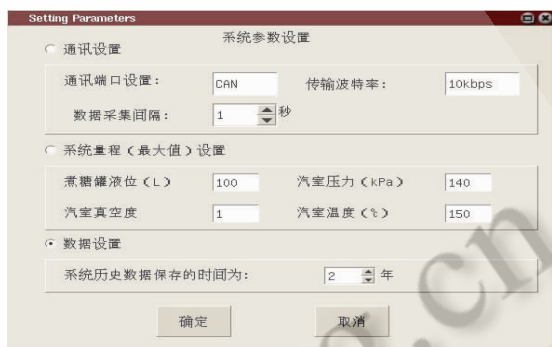


图 3 系统参数设置界面

5 中文输入

相对于中文显示来说,中文输入要复杂的多。

嵌入式 Qt 下中文输入的原理是:服务器进程接收到“键事件”后,首先将“键事件”送到中文输入服务器的键事件过滤器进行“过滤”;然后,由中文输入服务器的事件处理模块进行处理,它根据需要调用中文输入法进行中文字符编码转换,同时在中文输入界面上(包括预编辑区域、汉字候选区域等)显示相应信息;最后将转换后的中文字符编码发送给拥有焦点的窗口,并在焦点窗口中显示相应的中文字符。

目前,已有比较好的开源输入法可供我们移植,如我们平台上的 `Qtopia` 中移植的中文输入法叫 `MurphyPinyin`,是目前为止基于 `Qtopia` 的一个比较好的开源中文输入法。可通过移植此输入法实现中文输入。

`MurphyPinyin` 移植过程如下:

(1) 从网上下载输入法源码 `MurphyPinyin - src`。

tgz 文件到主机,进行解压

(2) 解压后生成了一个名为 `MurphyPinyin` 的文件夹,去掉此文件夹的“只读”属性,并应用到所有子文件夹。这样可避免在下面打软键盘补丁的步骤中失败。

(3) 修改 `MurphyPinyin` 文件夹下的 `Makefile`,使得编译时可加入 `QPEDIR` 下的库。将 `LIBS` 的宏定义改为:
`LIBS = $(LIBS) -L$(QTDIR)/lib -L(QPEDIR)/lib -lqpe -lqte`

(4) 将软键盘的补丁文件 `patch -softkeyboard` 拷贝到 `MurphyPinyin` 目录下,并进入 `MurphyPinyin` 目录下运行如下命令为输入法打上软键盘的补丁:

```
patch -p1 < patch -softkeyboard
```

(5) 为了使输入法的配置更利于管理,可通过修改源码文件 `PinyinFrame.cpp` 中的 `table_file[]` 的读取路径,使三个输入法的配置文件和码表文件的存放更加适合自己的需要。

(6) 根据具体的文件夹和编译工具的安装情况来配置相应的环境变量。

(7) 在 `MurphyPinyin` 目录下运行 `make` 命令,在此目录下生成了一个 `DIST` 目录,里面是 3 个输入法插件的库文件。

(8) 进行输入法往开发板的移植。此步的前提是板上已经移植过 `Qtopia` 系统。将生成的 3 个库文件和码表文件分别拷贝到开发板的相应目录下。

(9) 将实现中文显示的 `qpf` 字体文件拷贝到 `/opt/Qtopia/lib/fonts` 目录下,如 `simsun.qpf`。

(10) 通过脚本启动 `Qtopia`。

(11) 进入 `Qtopia`,此时可看到 `MurphyPinyin` 输入法,在 `Qtopia` 的 `Setting -> Appearance -> font` 中选择相应中文字体,如 `simsun`,以使系统支持中文显示。

图 4 所示即为输入法在 `Qtopia` 移植成功后看到的输入界面。

此中文输入法的输入效率比较高,并且码表可以修改替换以获得更好的输入效率。

6 结束语

嵌入式产品的广泛应用带动了图形用户界面 (GUI) 的迅速发展,嵌入式系统需要一个高性能、高可

靠的 GUI 的支持。该设计选择了嵌入式 `Qt` 作为研究对象,以自动煮糖控制系统的中文 GUI 平台实现为例,讨论了嵌入式系统中文环境的现状,对嵌入式 `Qt` 上如何实现中文 GUI 平台进行了深入研究,为嵌入式产品用于自动煮糖控制系统提供了有力的支持。笔者在基于 `ARMS3C2410` 芯片的嵌入式 `Linux` 平台上已经成功实现了基于 `QTE2.3.7` 的中文平台搭建。该平台初步解决了嵌入式 `Qt` 上的中文显示和中文输入问题,但是对移植的中文输入法导致 `Qtopia` 的启动时间延长的的问题还有待进一步研究。



图 4 输入法输入中文界面

参考文献

- 1 倪继利. `Qt` 及 `Linux` 操作系统窗口设计. 北京: 电子工业出版社, 2006: 212 - 220.
- 2 曹毅, 李德玉. 基于 `Qt/Embedded` 的嵌入式桌面环境的研究与实现. 西南民族大学学报, 2006, 32(6): 1270 - 1274.
- 3 赵卫星, 李敏, 董溯前. 基于嵌入式的智能家居原理与实践. 计算机系统应用, 2007, 16(10): 113 - 116.
- 4 吴伟清, 王磊, 吴朝晖. 基于 `QTE` 的嵌入式 `Linux` 中文环境解决方案. 计算机工程, 2005, (1): 31 - 2.
- 5 郑志国, 赵万春. 基于嵌入式系统键盘驱动的原理及实现. 计算机系统应用, 2007, 16(11): 120 - 122.