

# Qt/E 中文环境的搭建及对常用输入设备的支持

程 龙, 刘彦明, 鲁 刚, 李小平

(西安电子科技大学通信工程学院, 西安 710071)

**摘要:**就开发基于嵌入式 Linux 平台的 GUI 应用程序时遇到的问题, 阐述了如何搭建 Qt/Embedded 中文环境实现中文显示, 以及通过修改 Qt/E 源代码使得 Qt/E 库同时支持触摸屏、鼠标和标准键盘的使用。

**关键词:** 嵌入式 Linux; Qt/Embedded; 中文环境; 输入设备

## Deployment of Qt/E Chinese Environment and Support for Common Input Devices

CHENG Long, LIU Yan-ming, LU Gang, LI Xiao-ping

(College of Telecommunication Engineering, Xi'an Electronic and Technology University, Xi'an 710071)

**【Abstract】** Against problems faced when developing GUI application on embedded linux platform, this paper discusses how to deploy Chinese environment for Qt/E Chinese display and how to make the Qt/E library support touch panel, mouse and keyboard at the same time by revising source code.

**【Key words】** embedded Linux; Qt/Embedded(Qt/E); Chinese environment; input devices

嵌入式 Linux 目前在信息家电、手持设备及移动电话领域都出现了成熟的产品, 因此, 基于嵌入式 Linux 平台设计 GUI 应用程序有广阔的应用前景。

Qt/Embedded 是著名的 Qt 库开发商 Trolltech 公司开发的面向嵌入式系统的 Qt 版本, Qt/Embedded 类库完全采用 C++ 封装。丰富的控件资源和较好的可移植性是 Qt/Embedded 最为优秀的一方面, 使用 X 下的开发工具 Qt Designer 可以非常方便地开发基于 Qt/Embedded 的 GUI(图形用户操作接口)界面。

在开发基于嵌入式 Linux 平台的 GUI 应用程序时, 首先要解决中文显示、中文输入法, 并且要求能同时使用触摸屏、鼠标和键盘等常用输入设备。针对以上应用需求, 本文讨论了 Qt/E 应用程序的中文显示、移植中文输入法及同时支持触摸屏、鼠标和键盘的解决办法。

本文讨论的 Qt/E 中文环境及常用输入设备的支持针对 Qt/E-3.x 版本。

### 1 中文显示的实现

Qt/E 图形系统内部支持 Unicode, 并提供了实现 Qt 国际化编程的相关工具, 如 lupdate、lrelease 等, 利用这些语言工具处理后, 即可显示中文(具体内容请参阅 Qt 帮助文档)。

Qt 国际化编程是指在显示文本的地方不直接输入本地字符, 用英文代替, 对于需要翻译的地方, 首先是在该文本处用 tr() 函数标识, 程序编写完成后, 把文本提取出来翻译, 同时制作出 .qm 文件, 在程序中通过 Qtranslator 加载 .qm 文件即可。但这只是实现了静态中文显示, 例如要从文件中读取中文字符串时, 这时就需要调用 QTextCodec 类转换字符编码, 如 QTextCodec::codecForName("GB2312"), (GB2312、GBK、GB18030 为我国颁布的中文信息编码标准)。由于 Qt 默认的字体系 "Helvetica", 此时并不能支持中文显示, 所以还需要设置 "unifont" 字体。方法有 2 种: (1) 在应用程序中

设置如 a.setFontt( QFont ("unifont" 16, 50) ); (2) 在启动应用程序时设置命令行参数 "-font unifont"。虽然中文可以显示了, 但 Qt/E 自带的 unifont 字体在显示时效果不是很理想, 字形大小不一。

为了解决此问题, 参考 Qt 文档(Fonts in Qt/Embedded), Qt/E 可以支持以下 4 种格式的字体:

- (1) TrueType (TTF)
- (2) Postscript Type1 (PFA/PFB)
- (3) Bitmap Distribution Format fonts (BDF)
- (4) Qt Pre-rendered Font (QPF)

其中, TTF 格式的字体应用最广泛, 因此, 可以利用 Windows 下的提供的 TrueType 字体来改善显示效果。以宋体为例, 将 simfang.ttf 复制到 lib/fonts/目录中, 并编辑该目录中 fontdir 文件, 添加 "simfang simfang.ttf FT n 50 140 su"(格式含义请参考 Qt 文档 Fonts in Qt/Embedded), 这样就可以在应用程序中使用 simfang.ttf 字体了。但 TTF 字体比 QPF 字体占用空间大的多, 这在资源受限的嵌入式系统中并不是 Qt/E 所推荐使用的字体, 为此, Qt/E 还提供了一个十分方便的工具 makeqpf 来将 TTF 字体转换成 QPF 字体。makeqpf 通过读取 \$QTDIR 中的 lib/fonts/fontdir 来获取需要转换的字体信息, 生成相应的 QPF 字体。更为简洁的方法是通过修改 Qt/E 库中/kernel/qapplication\_qws.cpp 源代码: "bool qws\_savefonts = TRUE;" 或在启动应用程序时设置命令行参数 -savefonts, 当非 QPF 字体被使用时, Qt/E 会将相应的 QPF 字体自动保存, 这样下次就可以使用 QPF 字体了。

中文显示中需要注意如下 2 方面:

**作者简介:** 程 龙(1982 - ), 男, 硕士研究生, 主研方向: 计算机通信与网络; 刘彦明, 副教授; 鲁 刚, 硕士研究生; 李小平, 教授  
**收稿日期:** 2006-07-15 **E-mail:** calf9007@163.com

(1)嵌入式 Linux 文件系统所提供的 glibc 软件包最好是 2.3 版本以上的,笔者在调用 QTextCodec::codecForName()时,由于 glibc 是 2.2.3 版本,程序下载到板子上不能正确执行,对 glibc 升级后问题得以解决。

(2)无论是使用 Qt 国际化编程来调用 QTextCodec 类以转换字符编码,还是使用 TrueType 字体,在编译 Qt/embedded 库之前,都需要先确定配置文件中的配置对以上功能的支持。即编译时如果不是 -qconfig full(默认)设置,则要在配置文件如 /src/tools/qconfig-large.h 中注释掉"Qt\_NO\_CODECS"、"Qt\_NO\_TRANSLATION"、"Qt\_NO\_TRUETYPE" 等宏定义。

## 2 常用输入设备的支持

常用输入设备包括触摸屏、鼠标、键盘等,笔者遇到的问题是在编译 Qt/embedded 库时,如何使之同时支持触摸屏、鼠标、键盘(因为默认方式是触摸屏与鼠标取其一)。

通过分析源代码理清相关函数的调用关系,如图 1 所示。

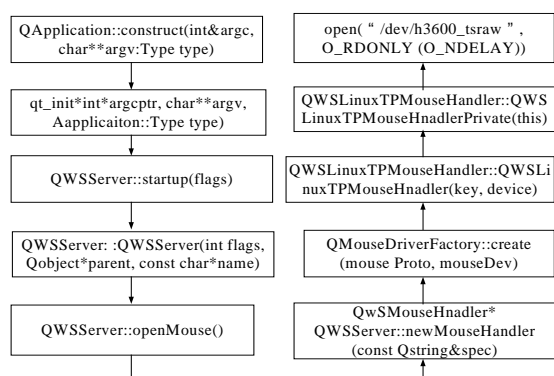


图 1 打开触摸屏及鼠标设备文件时相关函数的调用关系

问题的关键是如何在 QWSServer::openMouse()中同时将触摸屏和鼠标对应的设备文件打开

```
void QWSServer::openMouse() // 位于
/kernel/qwindowsystem_qws.cpp 文件中
{
    QString mice = getenv("QWS_MOUSE_PROTO");
    if ( mice.isEmpty() ) {
#ifdef Qt_QWS_CASSIOPEIA
        mice = "TPanel:/dev/tpanel";
#endif
    }
}
```

```
if ( mice.isEmpty() )
    mice = defaultMouse;
}
closeMouse();
bool needviscurs = TRUE;
if ( mice != "None" ) {
#ifdef Qt_NO_STRINGLIST
    QStringList mouse = QStringList::split(" ", mice);
    for ( QStringList::Iterator m=mouse.begin(); m!=mouse.end();
        ++m ) {
        QString ms = *m;
    }
#else
    QString ms = mice; // Assume only one
    {
#endif
        QWSMouseHandler* h = new MouseHandler(ms);
    }
}
}
```

解决办法如下:

在调用 QWSServer::openMouse() 之前先设置 QWSServer::setDefaultMouse("Auto Linuxtp : /dev/h3600\_tsrw")或直接修改 static const char \*defaultMouse 的值为 "Auto Linuxtp : /dev/h3600\_tsrw"或在运行应用程序时输出 "export QWS\_MOUSE\_PROTO="Auto Linuxtp" "即可解决问题。需要注意的是:"Auto"与"Linuxtp"之间一定要有空格分隔,原因是"Auto"在这里代表鼠标设备,"Linuxtp"代表触摸屏设备,程序中是用空格来区分不同的设备。

## 3 结束语

本文介绍的是笔者在实际项目中如何解决所遇到的问题,即如何搭建 QTE 中文环境及支持常用输入设备,希望能给读者一些参考。

### 参考文献

- 1 Trolltech Inc., Qt Reference Documentation(Qt 帮助文档)[Z]. 2005.
- 2 吴伟清. 基于 QTE 的嵌入式 Linux 中文环境解决方案[J]. 计算机工程, 2005, 31(2): 87-88.

(上接第 266 页)

表 2 存储区域的区分

分区	地址范围	用途
Vivi	0x00000000 ~ 0x0001ffff	存放 Bootload
param	0x00020000 ~ 0x0002ffff	存放 Linux 参数
Kernel	0x00030000 ~ 0x0012ffff	存放 Linux 内核
root	0x00130000 ~ 0x03ffffff	存放根文件系统

## 5.2 文件系统及烧写

用 busybox 制作根文件系统,通过 depmod -broot 命令进行驱动模块的依赖性检测,并把商标识别程序加入根文件系统。修改根文件系统的 rcs 启动文件,使 Linux 系统启动时能加载指定的驱动程序并自动运行商标识别程序。通过 Jtag 口烧写 vivi,使用 Uart 接口烧写 Linux 内核和根文件系统。

## 6 结论

本文详细介绍了基于 ARM 的图像识别系统的设计和实现,所设计系统在实验运行过程中,能稳定和长时间地正常运行,对商标的识别速度大约为 1.5s/次,识别率可以达到 85% 左右。

### 参考文献

- 1 刘 森. 嵌入式系统接口设计与 Linux 驱动程序开发[M]. 北京: 北京航空航天大学出版社, 2006.
- 2 张天序. 成像自动目标识别[M]. 武汉: 湖北科学技术出版社, 2003.
- 3 苏彦华. Visual C++数字图像识别技术典型案例[M]. 北京: 人民邮电出版社, 2004.