

文章编号:1007-2780(2013)02-0261-05

基于 ARM9 平台上 Qt/Embedded 的移植与开发

吴燕燕,贺锋涛*

(西安邮电大学 电子工程系,陕西 西安 710061, E-mail:wuyanyan19981995@163.com)

摘 要: 目前,嵌入式 Linux 操作系统快速发展,并在手持设备、移动设备中得到广泛应用。因此,拥有一个实用友好的图形用户界面就显得尤为重要。文章通过交叉编译源码库和应用程序的方式详细说明了 Qt/Embedded 在嵌入式 ARM9 上的移植流程,解决了应用程序跨平台无法运行的问题,并结合实例对应用的程序开发流程进行了阐述。

关 键 词: 嵌入式 Linux;嵌入式 GUI;Qt/Embedded;移植

中图分类号: TP621 **文献标识码:** A **DOI:** 10.3788/YJYXS20132802.0261

Qt/Embedded Transplantation and Development Based on ARM9 Platform

WU Yan-yan, HE Feng-tao*

(Department of Electrical Engineering, Xi'an University of Posts & Telecommunications, Xi'an 710061, China, E-mail:wuyanyan19981995@163.com)

Abstract: At present, the Embedded Linux operating system is rapidly developed and has wide application in handheld devices and mobile equipment. Therefore, possessing a practical friendly graphical user interface is particularly important. This article illustrates the transplantation of Qt/Embedded to Embedded ARM9 in detailed through the mode of crossing compile source repository and application, solves the problem of the cross-platform application not running, and expounds application combined with the development process.

Key words: embedded Linux; embedded GUI; Qt/embedded; transplantation

1 引 言

随着近几年嵌入式技术的飞速发展和广泛应用,嵌入式设备与用户亲密接触,自然友好、简单便捷的人机交互界面成为嵌入式设备的迫切需要,以 GUI 屏幕为中心的嵌入式图形界面以其灵活的控制方式,舒适清晰的界面给人们很大的亲和力。目前,一些先进的 PDA 在显示屏幕上以实现汉字写入、短消息语音发送、语音拨号上网、

收发电子邮件以及彩色图形、图像等技术的需求和应用不断上升,标志着 GUI 应用在嵌入式消费电子产品中正无孔不入地深入人们的生活,发展前景也将日益广阔^[1]。目前发展比较成熟的主要有以下几种:MiniGUI:图形设备层次过于抽象,图形功能不够完善;Microwindows:源码开放,但其许多图形引擎算法低下,控件或构件的实现还很不完备,系统整体不够完善;OpenGUI:内核采用汇编实现,不支持多进程,目前发展较慢。Qt/

收稿日期:2012-11-16; 修订日期:2013-01-03

基金项目:国家自然科学基金资助项目(No. 60878060)

作者简介:吴燕燕(1988—),女,山东菏泽人,硕士研究生,研究方向为数字信号处理算法与系统。

* 通信联系人, E-mail:344806450@qq.com

Embedded;面向嵌入式系统图形用户界面的应用程序开发框架,其库采用 C++ 封装,完全面向对象,可移植性好。Qt/Embedded 开发过程既完整又独特,界面设计美观,拥有一套比较完整的嵌入式 GUI 解决方案,因此在众多 GUI 里发展较快^[2]。为了解决跨平台的程序运行问题,本文将 Qt/Embedded 作为移植对象移植到目标机上。

2 Qt/Embedded 的实现结构及开发平台

2.1 Qt/Embedded 概述

Qt/Embedded 是以原始的 Qt 为基础,并做了许多出色的调整以适用于嵌入式环境。它采用轻量级窗口系统,在底层完全屏蔽了 Xlib,底层仅采用 Framebuffer 作为图形接口,同时支持键盘、GPM 鼠标、触摸屏以及用户自定义的设备等,并将外部输入设备抽象为键盘、鼠标和触摸屏的输入事件来实现事件驱动对外部事件的响应。它的实现结构图如图 1 所示^[3]。

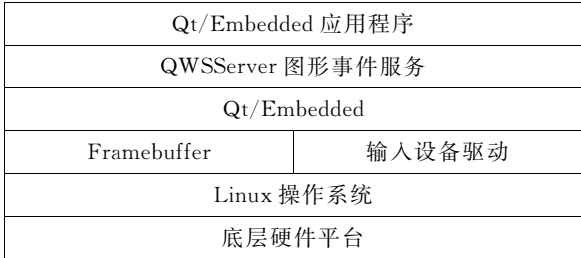


图 1 Qt/Embedded 的实现结构

Fig. 1 Realization structure of Qt/Embedded

2.2 Qt/Embedded 的硬件平台

移植选择 mini2440 为硬件平台,因为其价格低廉且非常实用,在 Linux 系统相关技术上已达到领先水平,并支持多种文件系统。它采用三星公司生产的 S3C2440A 为微处理器,主频 400 MHz,最高 633 MHz。在板 64M SDRAM,128M NAND FLASH(掉电非易失存储器),2M NOR FLASH(掉电非易失存储器),32 位数据总线,1 个 100M 以太网的 RJ-45 接口,3 个串行口,12M 无源晶振,内部实时时钟,并提供了如 LCD 控制器、相机接口、触摸屏接口等 GUI 所必须的硬件接口,为 Qt/Embedded 的移植与开发提供了良好的硬件基础^[4]。

2.3 Qt/Embedded 的软件平台

软件采用嵌入式 Linux 操作系统为开发环

境。嵌入式 Linux 是将 Linux 操作系统进行适当裁剪与修改,使之能在嵌入式计算机系统上运行的一种操作系统。它是一种自由和源码开放的类 Unix 操作系统,支持多用户,多任务,多线程以及多个 CPU,提供字符界面和图形界面。嵌入式 linux 既继承了 Internet 上无限的开放源代码资源,又具有嵌入式操作系统的各种优良特性,深受广大用户的青睐^[5]。因此,嵌入式 linux 是移植 Qt/Embedded 的最佳软件平台。

3 Qt/Embedded 嵌入式图形开发

3.1 Qt/Embedded 开发环境的搭建

由于基于 Qt/Embedded 开发的应用程序最终会移植到装有嵌入式操作系统的开发板上,而在嵌入式开发环境下开发和调试应用程序比较复杂,可采用 NFS 将文件系统挂载在宿主机的方式对应用程序进行调试,调试成功后再将应用程序加载到嵌入式 Linux 的根文件系统上,最后再通过 TFTP 和 nand erase、nand write 等相关命令烧写到到开发板 Flash 存储器上;若没有开发板,也可在 PC 机上直接开发,通过 QVFB 模拟调试结果,调试成功交叉编译后同样可以加载到文件系统中,最后再移植到开发板上。开发环境搭建如图 2 所示。

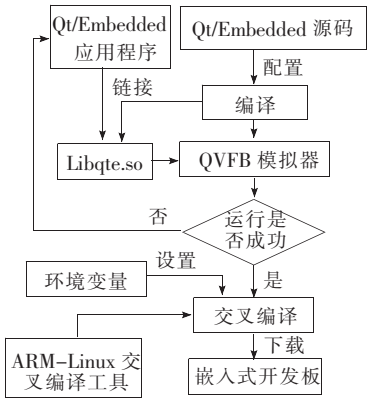


图 2 开发环境搭建流程

Fig. 2 Development environment building process

开发环境的搭建主要包括:(1)X11 平台,用于产生 designer、qvfb 等工具;(2)嵌入式 X86 平台,用于编译在 qvfb 上运行的程序^[6]。文中所使用的 Qt 版本是 Qt-4.5.0,可以在 Trolltech 官网下载 Qt 源码包,开发环境搭建过程如下:

(1)下载源码包 qt-x11-opensource-src-4.5.0.

```
tar. bz2
qt-embedded-linux-opensource-src-4. 5. 0. tar. gz
(2)编译生成 X11 平台 Qt 库
tar -xzf qt-x11-opensource-src-4. 5. 0. tar. gz
cdqt-x11-opensource-src-4. 5. 0
./configure-qvfb
make
make install
```

这一步最重要的是为了得到 qvfb 模拟器,有了 qvfb,嵌入式开发即便是没有实际的开发板也可以开发 Qt 应用程序。编译完成后,生成的库默认安装在/usr/local/Trolltech/Qt-4. 5. 0 下。

在使用 X11 平台的 Qt 库前,首先要通过 export 指令对环境变量进行设置。

```
(3)编译生成 X86 平台 Qt 库。
tar xjvf qt-embedded-linux-opensource-src-4. 5. 0. tar. bz2
cp qt-embedded-linux-opensource-src-4. 5. 0
qt-embedded-linux-opensource-src-4. 5. 0-x86
cd qt-embedded-linux-opensource-src-4. 5. 0-x86
./configure-prefix/usr/local/Trolltech/QtE-
mbedded-4. 5. 0-x86-embedded x86 -qvfb make
make install
```

./configure 命令中的-prefix 指定了生成库的安装路径,embedded x86 指明生成的 Qt 库是 X86 平台的,-qvfb 指明使用该库的 Qt 应用程序在 qvfb 上运行,编译完成后即可生成 X86 平台的 Qt 库。

由于嵌入式 X86 平台的程序在 qvfb 上执行,所以还需要将 X11 平台生成的 qvfb 拷贝到/usr/local/Trolltech/QtEmbedded-4. 5. 0-x86/bin 下。

在使用 X86 平台的 Qt 库前,首先也不要忘记通过 export 指令对环境变量进行设置。

3.2 基于 Qt/Embedded 图形用户界面开发

Qt/Embedded 的开发有两种方式,一种是传统的代码编程方式,是一般程序员开发程序时所使用的;一种是图形编程方式,可以利用 Qt-Designer 开发,这是一种所见即所得的方式。不论是这两种方法的哪一种,都会涉及到 Qt 的元对象系统。元对象系统有两个关键技术:信号和槽,以及内省,其中内省是信号和槽的基础。由于 Qt 通过 moc 来解析 Q_OBJECT 类定义,所以 Qt 的源对象系统可以在任意 C++ 编译器上工作,

这也是开发者觉得源程序编译速度较慢的原因。其编程过程如图 3 所示^[7]。

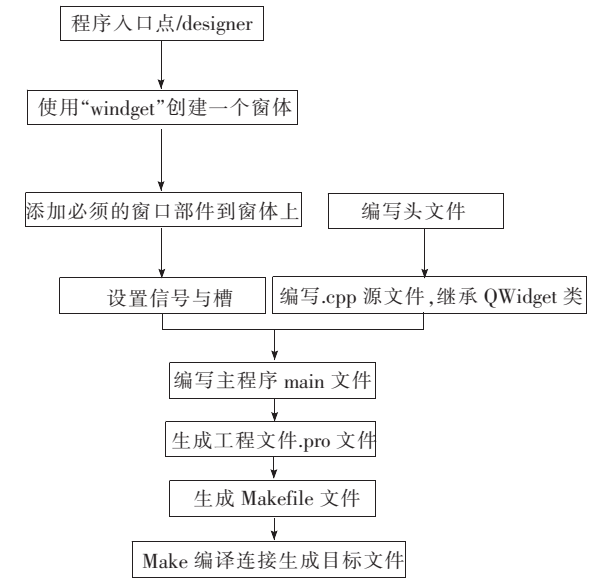


图 3 Qt/Embedded 编程过程
Fig. 3 Qt/Embedded programming process

程序编写完毕后,我们先用 qmake 工具生成 file. pro 文件,设置好环境变量后再用 tmake 生成 makefile 文件。调试程序时,我们可以先在宿主机的 qvfb 模拟器上模拟程序运行的结果,调试成功交叉编译后再把程序下载到目标平台上,这样大大减少了应用程序的开发周期^[8]。

4 Qt/Embedded 的移植与开发

4.1 Qt/Embedded 的移植

在将应用程序移植到目标开发板之前要保证触摸屏、framebuffer 等驱动程序正常运行^[9]。Qt/Embedded 移植过程大概如下:

(1)安装交叉工具链

文中选用 arm-linux-gcc 系列的交叉工具链。安装包为 arm-linux-gcc-3. 4. 1. tar. bz2,对其进行解压,解压完成后生成 usr/local 文件夹,进入该文件夹,将 arm 文件拷贝到/usr/local/下。命令如下:

```
tar zxvf arm-linux-gcc-3. 4. 1. tar. bz
cd usr/local/
cp -rv arm /usr/local/
```

现在交叉编译环境集都在/usr/local/arm/3. 4. 1/bin 文件夹下了。然后修改环境变量,对

.bashrc文件最后添加:

```
export PATH = :/usr/local/arm/3.4.1/bin;
```

```
$ PATH
```

若要使新的环境变量立即生效且不用重启电脑,可执行命令:

```
source /root/.bashrc
```

(2) 移植 tslib

Tslib 程序开源,为触摸屏驱动所获得采样提供滤波、去抖、校准等功能,为上层应用提供统一接口,通常作为触摸屏驱动的适配层^[10]。具体如下:

```
tar -zxvf tslib-1.4.tar.gz
```

```
cd tslib
```

```
./autogen.sh //生成 configure
```

```
./configure--prefix =/usr/local/tslib/--host =
```

```
arm-linux ac_cv_func_malloc_0_nonnull=yes
```

```
make
```

```
make install
```

对 tslib/etc/ts.conf 文件进行相关修改,对 tslib 打包后下载到开发板的/usr/local 文件下,设置好环境变量,执行命令:

```
cd /usr/local/bin
```

```
./ts_calibrate
```

运行后会出现 tslib 的触摸屏五点校准画面,完成后即可生成 pointercal 文件。然后进行链接:ln -s /usr/tslib/etc/pointercal /etc/pointercal

(3) 交叉编译 Qt/Embedded 库

在交叉编译前,需要先修改 Qt/Embedded 的配置文件,再将 GCC,G++ 编译器和链接器设置为当前编译安装的交叉工具链,然后对 Qt/Embedded 库的源代码进行交叉编译生成一个以 ARM 为目标代码的 Qt/Embedded 库。命令如下:

```
mv qt-embedded-linux-opensource-src-4.5.0
qt-embedded-linux-opensource-src-4.5.0-arm
```

```
./configure-prefix/usr/local/Trolltech/Qt_
Embedded-4.5.0-arm -embedded arm -I /usr/
local/tslib/include -L /usr/local/tslib/lib
```

```
make
```

```
make install
```

./configure 命令中-embedded arm 指明生成

的 Qt 库是 ARM 平台的,编译完成后即可生成 ARM 平台的 Qt 库。

(4) 交叉编译 Qt/Embedded 应用程序

设置 X86 平台 Qt 运行环境,编写应用程序代码,在 QVFB 模拟调试运行成功后,再设置环境变量,进入 ARM 平台的 Qt 运行环境,修改 tmake 的配置文件 tmake.conf,将里面的“gcc”改为“g++”,然后执行命令

```
qmake -project
```

```
tmake
```

```
make
```

即可生成可在 ARM 开发板上运行的二进制目标可执行文件。

(5) 发布 Qt/Embedded 应用到 mini2440 开发板

把交叉编译后的 Qt/Embedded 库和应用程序的可执行代码复制到某个目录下,将其制作成某种类型的根文件系统,然后通过 tftp 工具和相关命令将此根文件系统下载到 mini2440 的 Nand Flash 中即可完成移植。

4.1 Qt/E 应用程序开发

应用程序为基于嵌入式 ARM9 平台进行图片浏览器的设计。硬件提供 8.89 cm (3.5 in) TFT LCD,320×240 像素。采用窗口、菜单、按钮和图标为常见的图形元素进行设计 GUI;利用 setGeometry (0,0,320,240)设置窗口大小为 320×240, QPushButton()函数创建新按钮,并可设计其大小,setIconSize(QSize(30,30))实现按钮上添加图标;Qpixmap(“图形文件名”)函数设置界面的背景;QHBoxLayout(水平布局)、QVBoxLayout(垂直布局)和 QGridLayout(表格布局)把各个按钮在界面进行排序^[11]。

应用程序编写完毕,进入该文件夹,设置好 X11 平台的 Qt 的运行环境变量,输入命令:

```
tmake
```

```
make
```

即可生成可执行二进制文件,启动 qvfb 界面,打开另一终端,运行程序:

```
./gui -qws
```

调试成功后,即可下载到开发板上运行,显示结果如图 4 所示。

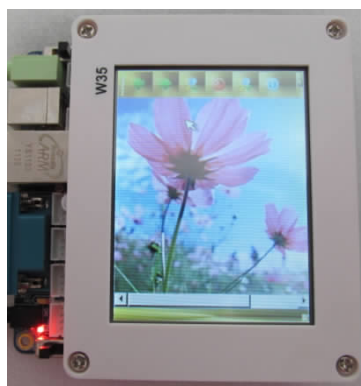


图 4 开发板运行结果

Fig. 4 Development board operation results

5 结 论

随着嵌入式的发展,用于实现与用户交流的嵌入式 GUI 将逐步成为各种应用研究中的重点项目。目前,图形用户界面在嵌入式系统中的需求也正在逐步上升,尤其在一些智能电子产品和工业控制领域。Qt/Embedded 专门针对嵌入式系统开发,拥有 Qt 在桌面系统的大部分功能,并具有丰富的 API 和强大的工具支持,采用其进行人机界面设计,为嵌入式 Linux 系统开发节省极大成本和维护费用,并使系统应用程序开发更加便捷^[12]。

参 考 文 献:

- [1] 刘金星,李洪文. 基于高速单片机的液晶显示模块控制 [J]. 液晶与显示, 2011, 26(1): 88-91.
- [2] 倪继利. Qt 及 Linux 操作系统窗口设计 [M]. 北京: 电子工业出版社, 2006: 10-30.
- [3] 李想,郑喜凤,陈宇. 基于 Linux 下的 OLED 显示模块设计 [J]. 液晶与显示, 2012, 27(1): 103-107.
- [4] 黄亮,杨景常. 基于 SOPC 的 TFT 触摸屏显示系统设计 [J]. 液晶与显示, 2009, 24(5): 719-722.
- [5] 古容江,张珣. 基于 ARM 的图形用户界面软件开发平台的搭建 [J]. 液晶与显示, 2009, 24 (6): 901-906.
- [6] 张伽伟,周安栋,罗勇. ARM11 嵌入式系统 Linux 下 LCD 的驱动设计 [J]. 液晶与显示, 2011, 26 (5): 660-664.
- [7] 何剑锋,李祥,何月顺,等. 基于 S3C2410 的嵌入式 LINUX 下 OLED 模块驱动设计 [J]. 液晶与显示, 2010, 25 (1): 90-93.
- [8] 吴子平,徐爱钧. 基于 Qt/Embedded 的嵌入式 GUI 的研究与构建 [J]. 电脑开发与应用, 2012, 25(1): 13-16.
- [9] 谢芬,潘丽,刘守印. 基于 QT/E 的嵌入式 Linux 系统的软键盘实现 [J]. 电子设计工程, 2012, 20(5): 177-179.
- [10] 华晶,贾晶,何火娇. 基于 Qt/Embedded 在嵌入式 GUI 在 ARM 的移植 [J]. 计算机与现代化, 2009(10): 117-119, 125.
- [11] 彭均键,史步海,刘洋. 基于 Qt 的嵌入式 GUI 开发平台的搭建 [J]. 微型电脑应用, 2010, 26(2): 40-42.
- [12] 汤伟,李强. Qt_E 的嵌入式 Linux GUI 研究与实现 [J]. 计算机应用, 2011, (10): 260-263.