

基于 ARM LINUX 的嵌入式 GUI 的研究和移植

纪竞舟, 付宇卓

(上海交通大学芯片与系统研究中心, 上海 200030)

摘要: 随着嵌入式系统的发展, 用户对嵌入式系统的要求越来越高, 因此用于实现与用户交流功能的嵌入式 GUI 成为嵌入式研究中的一个重点。该文首先介绍了什么是嵌入式系统和嵌入式系统中的 GUI, 并阐述了开发基于嵌入式 Linux 平台的 GUI 系统的必要性。然后详细的介绍了三种常见的嵌入式 GUI 系统(Microwindows、MiniGUI 和 QT)的设计结构和使用上的优缺点等, 并列表进行了比较。最后, 描述了每个系统基于 ARM Linux 平台的移植, 包括编译环境的建立、编译选项的配置和修改以及运行时参数的设置等等。

关键词: 嵌入式; 图形用户界面; 移植

中图分类号: TP311 **文献标识码:** A

Research and Porting of Embedded GUI Based on ARM LINUX

Ji Jing-zhou, Fu Yu-zhuo

(Shanghai Jiaotong University SOC and System Research Center, Shanghai 200030, China)

ABSTRACT: With the development of embedded system and improvement of the requirements from customers, embedded GUI which was used to communicate between users and embedded system became to be a key in researching the embedded system. This article firstly introduced what is embedded system and what is embedded GUI. What's more, the necessary of developing GUI based on embedded linux is also explained. Then three kinds of embedded GUI's (Microwindows, MiniGUI and QT) architectures, traits and disadvantages were also introduced. Finally, it described the porting of each embedded GUI on ARM Linux platform, including the establishment of compile environment, configuring and modifying of compile options and setting of runtime parameters.

KEYWORDS: Embedded; GUI; Porting

1 引言

1.1 什么是嵌入式系统

嵌入式系统一般指以嵌入式微处理器为核心, 有别于 PC 系统, 有计算机的部分功能但又不称之为计算机的设备或器材。它主要是以应用为目的, 系统软硬件于一体, 通常要求具有代码小、响应速度快、可靠性高、低功耗、集成度高等特点。嵌入式系统遍布于各个行业以及人们的日常生活中, 比如掌上 PDA、移动计算设备、电视机顶盒、手机、汽车、数码相机、家庭自动化系统、安全系统、自动售货机、工业自动化仪表与医疗仪器等。

1.2 什么是嵌入式系统中的 GUI

随着后 PC 时代的到来, 嵌入式系统的性能有了大幅度的提高, 应用范围也越来越广, 当初的一些简单的人机交互接口已经无法满足人们的要求。而与此同时, 在台式 PC 机上图形交互界面早已普及并成熟, 于是在嵌入式系统中也逐

渐出现了图形用户界面(Graphic User Interface, GUI), 特别是在一些消费类产品中。嵌入式系统中的 GUI 就是在嵌入式系统中为特定的硬件设备或环境而设计的图形用户界面系统。由于受到当前嵌入式系统本身特点的影响, 并受其发展限制, 所以嵌入式系统中的 GUI 应该有如下特点:

- 1) 占用的存储空间以及运行时占用资源少;
- 2) 运行速度以及响应速度快;
- 3) 可靠性高;
- 4) 便于移植和定制。

Linux 有开放的源码、高效稳定的内核、良好的开发环境以及支持多种硬件平台等特点, 而且由于 Linux 的可配置性和可模块化, 一个 Linux 内核经过配置和裁减可以只占用几百 K 左右的存储空间, 使其在嵌入式系统中也得到广泛应用。然而, 在嵌入式系统中有严格的资源限制, 台式机 Linux 平台下成熟的 GUI 无法直接移植到嵌入式设备上。因此, 嵌入式系统中基于 Linux 的 GUI 愈发成为开发基于 Linux 的嵌入式系统所必须面对的问题。目前, 随着嵌

收稿日期: 2003-11-09

入式行业分工的细化,出现了专门的公司致力开发 Linux 下的 GUI 并给其它公司提供授权,从而缩短了嵌入式系统的开发周期。本文就旨在对这样的几种嵌入式 GUI 进行研究,列表 1 进行比较,并介绍这几种 GUI 在基于 ARM Linux 上的移植。

2 几种比较成熟的嵌入式 GUI 的研究

2.1 Microwindows

Microwindows 是由美国 Century Software 公司开发的开放源码的嵌入式 GUI 项目。它不需要其他图形系统的支持,可以充分利用 Linux 提供的 Framebuffer 机制来进行图形显示。同时在底层提供了对多种芯片的支持,并基本上用 C 语言实现,因此移植性很强。国内也有人参与了这个项目的开发,并编写了 GB2312 等字符集的支持。

Microwindows 是基于分层式设计的,它允许不同的层可以被重新设计以满足系统实现的需要,目前总共分为三层。最底层是屏幕、鼠标或触摸屏和键盘的驱动程序,用于实现系统的显示功能以及满足用户特有的输入需求。中间层实现了一个可移植的图形引擎,以提供对画线、填充、剪切和着色的支持。在最上层,则提供了多种的 API 给应用程序设计者。Microwindows 支持两种 API: ECMA APIW(类 Win32 API)和 Nano-X APIs。这些 API 对 Win32 和 X Window 系统有很好的兼容性,使程序可以很容易地从其它系统上移植过来。其中比较完备的是 Nano-X API。Nano-X 设计成是一个客户端/服务器的环境,在此模式下应用程序与一个客户端的库相连,而这个库则通过一个 UNIX socket 与 Nano-X 服务器相连。每个应用程序都通过 UNIX socket 来传递各种参数,并由服务端来完成客户端的各种请求。当为了满足运行速度以及调试的需求时,还可以通过共享的内存空间来完成传播数据。

由于 Microwindows 为了与 X Window 的兼容,它采用了传统的基于 UNIX socket 的客户端/服务器结构。此是,客户端的画图、窗口建立等操作都是通过套接字与服务器通信。在这种情况下,系统就变得非常依赖于 UNIX socket,而 UNIX socket 的传递都需要通过内核,这样在一些对实时性要求比较高的系统里就不太有利。而且 Microwindows 基本上用 C 语言实现的,虽然增强了可移植性,但也同样导致了系统的运行效率不高。同时 Microwindows 的开发工具 FLNX 也还不够成熟,有待进一步提高。

2.2 MiniGUI

MiniGUI 是我国国内自主开发、比较成熟的一个图形用户界面系统,面向基于 Linux 的实时嵌入式系统,使用现有成熟的图形引擎(SV2GALib/ LibGGI),采用类似 WinCE 的线程机制(目前 MiniGUI Lite 版本改用微客户端/服务器模式来提供对多进程的支持),集成了多字体和多字符集,支持硬件加速能力,充分使用显示内存。

从整体来看,MiniGUI 也是属于分层结构设计(见图 1)。

在最底层, GAL 和 IAL 提供了 Linux 控制台和底层图形接口的支持以及输入设备比如鼠标、键盘等的驱动, PThread 则是提供了内核级线程支持的函数库;中间层 MiniGUI Kernel 则包括了窗口系统的各个模块, MiniGUI 本身是运行在多线程模式下,它的各个模块都是以单独的线程运行,同时它也利用线程来实现多窗口;最上层,则是提供给各种应用程序使用的 API 函数。MiniGUI 中利用线程之间的同步操作,实现了客户线程和服务线程之间的微客户端/服务器结构,因为它的客户和服务器是同一进程的两个不同线程。它的核心实现主要体现在消息队列数据结构的使用上。

多线程有其一定的好处,但弱点则在于不同的线程共享了同一个地址空间,因此,客户线程可能会破坏系统服务器线程的数据,导致系统健壮性不够健全。虽然目前的 MiniGUI Lite 版本抛弃 Linux Thread 库而改用微客户端/服务器模式提供对多进程的支持,但是仍然存在着对多进程支持弱、图形功能不完善、窗口剪切不完备等问题。

2.3 QT/Embedded 和 Qtopia

QT/Embedded 是著名的 QT 库开发商 TrollTech 开发的基于 framebuffer、面向嵌入式系统的 QT 版本。它的特点主要有界面美观、色彩配比好,使用与 QT/Windows 和 QT/X11 完全一致的 API 接口,许多基于 QT 的程序可以非常方便地移植到嵌入式系统;同时它具有丰富的模块,用户可以根据需要选择 QT/Embedded 的特性集合。Qtopia 是 TrollTech 专门为嵌入式 Linux 设备所定制终端用户平台,它包括了 Qtopia 库和 Qtopia 的服务器及客户端应用程序。

QT/Embedded 和 Qtopia 的结构如图 2 所示。QT/Embedded 实现了对下层输入设备的驱动和底层图形接口以及各类和 API。Qtopia 也是以客户/服务器的方式,通过消息机制来管理各种应用程序。QT 的资源相当丰富,基于 QT API 开发的应用程序相当多,像 KDE 这种应用范围极广的 GUI 也是使用 QT API 开发出来的;而且 OPIE 组织也开发了并改进了许多专门基于 QT/Embedded 上的 Qtopia 应用程序及函数库。

不过 QT/Embedded 对于系统的资源要求非常高,需要在较大的存储空间和运行空间上才可以运行;而且它是基于 C++ 来实现的,而这种 C++ 接口对于嵌入式系统中的某些应用来说就显得比较臃肿;另外如若需要修改 QT/Embedded 并在商业产品中发布,则需要分别购买开发授权及运行时授权,所需费用太昂贵。

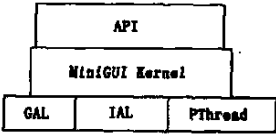


图 1 MiniGUI 体系结构图

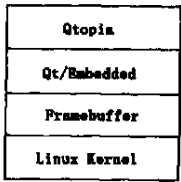


图 2 QT/Embedded 和 Qtopia 的结构

表 1 上述三个 GUI 的比较

	MiniGUI	Microwindows	QT/ Embedded
API	Win32 风格	X、Win32 子集	QT(C++)
函数库大小	500K	600K	1.5M
可移植性	好	很好	较好(但函数库本身的交叉编译较难)
多语种支持	很好	好	使用 UNICODE, 但效率较低, 对中文字体支持几乎没有
系统资源消耗	小	较小	最大(C++ 实现)
操作系统支持	Linux	Linux、ELKS、MSDOS	Linux
硬件平台支持	X86、ARM、MIPS、PowerPC	X86、ARM、MIPS、PowerPC、SPARC	X86、ARM

3 基于 ARM Linux 平台的移植

3.1 编译环境的建立

GUI 的编译通常都是在 PC 机上执行的, 也就是说, 编译器本身能够在 PC 机上执行, 同时编译源代码生成的二进制文件必须能在目标机上执行, 这类编译器通常称为交叉编译器。对于 ARM 平台, 我们安装了 `cross-arm-binutils-*.i386.rpm`, `cross-arm-gcc-*.i386.rpm`, `cross-arm-glibc-*.i386.rpm` 这三个包。这些包都可以从网上免费获取, “*”通常代表版本号。arm-binutils 这个包一般包含了一些针对 ARM 平台的二进制工具, 比如 `arm-strip`、`arm-ar` 等命令; arm-glibc 这个包包含的是标准 C 的函数库的 ARM 的版本以及对应的头文件; arm-gcc 中包含的则是生成 ARM 平台代码的 x86 上的交叉编译器。执行 `rpm` 命令将这些包安装到 PC 机上, 若不在系统默认搜索目录下, 比如安装在 `/opt/cross` 目录下, 则必须将 `/opt/cross/bin` 目录加到系统的 PATH 环境变量中, 这样在每次编译时系统才能找得到编译器。另外需要注意的是, 编译时所用的函数库版本要与目标板上运行时所用的函数库版本一致。经过上述步骤, 就已经建立了交叉编译环境, 接下来的就是选择或修改 GUI 的编译选项, 从而生成所需要的 GUI 系统。

3.2 Microwindows 的移植

Microwindows 的定制和编译主要是通过修改它源程序根目录下的 `config` 这个文件中的选项来实现, 而在 `configs` 这个目录下已经有一些针对不同平台的 `config` 文件, 若目标平台与之定义的相符, 可直接将其拷贝到根目录下覆盖掉原有的 `config` 文件, 然后输入命令 `make` 来开始编译。另外还可以执行根目录下的 `xconfigure` 文件, 它提供了一个可视化的窗口来方便用户进行编译选项配置。下面介绍一下其中主要的一些选项:

1) 目标平台选项。ARCH 表示目标机的平台, 比如 ARCH=LINUX-ARM, BIGENDIAN 表示目标平台是否为大数端;

2) 图像支持选项。HAVE-BMP-SUPPORT=Y、HAVE-JPEG-SUPPORT=Y 两项表示对 BMP 和 JPEG 格式图像的支持(还有其它格式选项)。设置 JPEG 图像选项时必须给出外部

jpeg 解压库库的位置, 例如 LIBJPEG=/usr/lib/libjpeg.a;

3) 字体支持选项。HAVE-FREETYPE-SUPPORT=Y、HAVE-TILIB-SUPPORT=Y 表示支持 TrueType 和 Adobe Type 1 字体, 这些库也必须预先编译并在配置文件中指定其位置, Microwindows 的网站上提供了许多字库函数的下载。另外, Microwindows 还提供了中文 BIG5、GB2312 以及日文和韩文的支持;

4) 输出显示选项。FRAMEBUFFER=Y 表示使用 Linux 上的 framebuffer 来实现显示, 当然 Microwindows 也支持 X11 显示;

5) 输入设备选项。Microwindows 提供了一系列的鼠标(包括触摸屏)和键盘的支持, 可以根据不同的系统选择适当的选项。

配置完成后, 执行 `make` 命令开始编译。编译完成后, 将生成的库以及可执行文件打包下载到目标板上。如果库文件没有放在目标系统的默认搜索库的路径中, 那么还需将此目录加到 LD-LIBRARY-PATH 环境变量中, 然后即可以运行 nano-X 服务器及各应用程序。

3.3 MiniGUI 的移植

下载 MiniGUI 的库文件源代码包 `libminigui-*.tar.gz` 和资源文件包 `minigui-res-*.tar.gz` 以及演示程序包 `mde-*.tar.gz` 后, 将它们分别解压到各自的目录下。首先介绍库文件的编译, 它的编译配置主要通过执行源程序根目录下 `configure` 这个文件, 并给它传递相关的参数来实现。比如: `CC=arm-linux-gcc ./configure --host=i386-linux --target=arm-linux`。其中, CC 参数表示所使用的编译器名, host 表示编译环境所在的主机类型, target 表示编译生成后目标系统的主机类型, 当前默认编译生成的是 MiniGUI Thread 版本, 如果要生成 MiniGUI Lite 版本需再加上参数 `--enable-lite`, 更多参数可以通过执行 `./configure -help` 来查询。在执行 `configure` 之前, 根据硬件特性可以修改根目录下的 `etc` 目录下的 `Minigui-3d.cfg` 和 `Minigui-flat.cfg` 两个文件中的选项。比如, 如果目标系统的 framebuffer 大小是 320 X 240, 每像素占 12bits, 则将上述两个文件中 `[fbcon]` 组中的 `defaultmode` 的值就应设置为 320 X 240-12bpp。配置完成 (下转第 238 页)

6 结束语

使用 Simulink 进行仿真需要对硬件进行访问, 可以通过编写 WDM 驱动程序, 并通过 S 函数扩展 Simulink 模块实现。这是一种可以很好地解决问题的方法。该方法可以扩展 Matlab 的半实物仿真能力。

参考文献:

- [1] 尤晋元, 等. Windows 操作系统原理[M]. 北京: 机械工业出版社, 2001.
- [2] 杨涤, 等. 系统实时仿真开发环境与应用[M]. 北京: 清华大学出版社, 2002.

(上接第 226 页)

以后执行 make 命令开始编译, 编译结束后执行 make install 命令, 在 /usr/local/lib 下面会生成交叉编译出来的动态库和静态库文件。其次, 资源文件的安装比较简单, 解压以后在其根目录下有一个名叫 config.linux 的文件, 修改其中的 TOPDIR 一项, 这样执行 make install 安装命令以后, 所有资源文件会被安装到 TOPDIR 所定义的目录下的 usr/local/lib/minigui 下。最后演示程序包的编译与库文件编译类似, 也是执行其根目录下的 configure 文件并传递相关参数, 然后执行 make 命令来编译。若编译库文件时选择 `--enable-lite` 参数, 由于 MiniGUI-Lite 采用的是微客户端/服务器架构, 要运行应用程序必须先启动服务器程序 mginit, 然后才能运行其它的应用程序; 若编译库文件时没有添 `--enable-lite` 参数, 则 mginit 目录根本不会编译进去, 演示程序可以直接执行。

3.4 QT 的移植

QT 的编译步骤比较复杂。Qtopia 的根目录下有 README.html 文件介绍了 PC 机上编译的步骤及参数。因此, 本文主要描述在编译成以 ARM 为目标平台时有所区别的几步及参数。下载 Qt/Embedded、Qt/X11 和 Qtopia 三个包并分别解压。这三者的编译配置都是通过执行相应根目录下的 configure 文件并传递相关参数来实现, 另外在 Qtopia 目录下 src/qt/qconfig-qpe.h 文件中要定义一些相关的宏, 这样在编译时编译器会根据所定义的宏选择需要的代码来进行编译。比如, 目标系统所使用的触摸屏实现方式与 IPAQ 类似, 若要使 QT 将所支持的代码编译进来, 则需在 qt-config-qpe.h 中定义宏 QT-QWS-IPAQ。首先, 我们在执行 configure 对 Qt/Embedded 进行配置时, 传递了三个参数 `-xplatform linux-arm-g++-qconfig-qpe` 和 `-no-qvfb`。其中第一个参数表示, 编译后二进制文件执行时所在的目标平台类型; 第二个参数告诉编译器配置头文件是在 src/tools/ 目

[作者简介]



王万里(1977.10-), 男(汉族), 广东广州人, 哈尔滨工程大学计算机科学与技术学院硕士研究生, 主要研究方向: 数据库与知识库, 系统仿真;
张文杰(1968.6-), 男(汉族), 天津人, 北京仿真中心研究员, 主要研究方向: 系统仿真;

王燕(1962.6-), 女(汉族), 黑龙江哈尔滨人, 副教授, 硕士生导师, 主要研究方向: 数据挖掘, 数据库与知识库。

录下叫做 qconfig-qpe.h, 通常这个文件就是 Qtopia 目录下 src/qt/qconfig-qpe.h 的一个拷贝; 第三个参数表示在编译时不需要编译 qvfb 工具, 当目标系统不支持 Framebuffer 时, 在执行启动 Qtopia 之前先通过执行 qvfb 来产生一个虚拟的 Framebuffer, 这时再启动 Qtopia 相应的内容就会显示在 qvfb 上。在对 Qtopia 进行配置时, 只传递了一个参数: `-platform am-linux-g++`, 表示目标平台是 ARM 平台。当然在上述配置过程中还可以使用其它一些参数, 这些参数可以通过传递 `-help` 参数给 configure 来查询。在配置并编译完之后, 将 Qtopia 目录下的运行时相关目录全部打包并下载到目标系统上, 设置了相应的环境变量(比如 QPEDIR、QTDIR 和 LD-LIBRARY-PATH 等)后, 就可以执行。当然目标系统上的特殊硬件的支持需要通过修改 Qt/Embedded 下的对应源程序来完成。

参考文献:

- [1] Greg Haer. Microwindows Architecture[M]. Century Software Inc, 2000.
- [2] 詹瑾瑜, 熊光泽, 孙明. 一种嵌入式 GUI 软件结构实现方案[J]. 电子科技大学学报, 2003, 32(1): 89-93.
- [3] 魏永明. MiniGUI 技术白皮书[M]. 北京: 飞漫软件技术有限公司, 2002.
- [4] Xitem(中国)软件技术有限公司. Qt 程序设计[M]. 北京: 清华大学出版社, 2002.



[作者简介]

纪竞舟(1980.6-), 男(汉族), 福建福鼎人, 研究生, 主要研究方向: 嵌入式系统;
付宇卓(1968.10-), 男(汉族), 黑龙江省哈尔滨市人, 博士, 上海交通大学芯片与系统研究中心副主任, 主要研究方向: 芯片与系统设计。