

MiniGUI 技 术 白 皮 书

版本 2.0 (修订号 3), 适用 MiniGUI 版本 2.0.3/1.6.9
版权所有 ? 2002~2006 , 北京飞漫软件技术有限公司
最新改动日期 : 2006/06/19

声明 : 飞漫软件赋予所有人复制、发布本文档的权利 ,
但需保证本声明及文档的完整性 ;
除此之外的其它权利均予保留。

目 录

1 MiniGUI 简介	1
1.1 什么是 MiniGUI.....	1
1.2 MiniGUI 的起源和发展	1
1.3 MiniGUI 的应用领域	2
2 MiniGUI 的优势	4
2.1 MiniGUI 的技术特点	4
2.2 MiniGUI 的技术优势	5
3 运行 MiniGUI 的系统需求	8
3.1 MiniGUI 所支持的操作系统	8
3.2 MiniGUI 所支持的硬件平台	8
3.3 MiniGUI 对系统资源的占用情况	8
4 MiniGUI 的软件架构	9
4.1 基于 MiniGUI 的嵌入式系统结构	9
4.2 MiniGUI 运行模式	9
4.2.1 MiniGUI-Processes 运行模式	10
4.2.2 各操作系统上可运行的 MiniGUI 运行模式	11
4.3 窗口系统	12
4.4 通讯机制	12
4.5 字体	13
4.6 输入设备的支持	13
4.7 输入法	13
5 开发环境	14
6 程序样例和控件	15
6.1 Hello World 示例程序	15
6.2 静态框	16
6.3 按钮	16
6.4 列表框	16
6.5 编辑框	17
6.6 组合框	17
6.7 菜单按钮	17
6.8 进度条	18
6.9 滑块	18
6.10 工具栏	18
6.11 属性表	19
6.12 滚动型控件	19
6.13 树型控件	19
6.14 列表型控件	20
6.15 月历控件	20
6.16 动画控件	21
6.17 网格控件	21
6.18 图标型控件	22
7 MiniGUI 窗口和控件的显示风格	23

8	国际化	25
9	飞漫软件围绕 MiniGUI 的软件产品	26
9.1	MiniGUI 增值版	26
9.2	MiniGUI 组件产品	26
9.2.1	mGp.....	26
9.2.2	mGi.....	27
9.2.3	mG3d.....	28
9.3	飞漫嵌入式浏览器： mSpider.....	28
10	MiniGUI 相关资源	30
10.1	开源版本及开发包	30
10.2	飞漫软件发布的其他开源软件	30
10.3	MiniGUI 演示程序	30
11	MiniGUI GPL 版本的授权策略	31
11.1	如果您 100% 遵循 GPL ，则无需获得商业授权	31
11.2	如果您从不复制、修改和发布 MiniGUI ，则无需获得商业授权	31
11.3	其他情况均需获得商业授权	31
12	联系我们	33

1 MiniGUI 简介

1.1 什么是 MiniGUI

由北京飞漫软件技术有限公司开发的 MiniGUI (<http://www.minigui.com>), 是面向实时嵌入式系统的轻量级图形用户界面支持系统。自 1999 年初遵循 GPL 条款¹发布第一个版本以来, MiniGUI 已广泛应用于手持信息终端、机顶盒、工业控制系统及工业仪表、便携式多媒体播放机、查询终端等产品和领域。目前, MiniGUI 已成为跨操作系统的图形用户界面支持系统, 可在 Linux/uClinux、VxWorks、eCos、uC/OS-II、pSOS、ThreadX、Nucleus、OSE 等操作系统以及 Win32 平台上运行; 已验证的硬件平台包括 Intel x86、ARM (ARM7/ARM9/StrongARM/xScale)、PowerPC、MIPS、M68K (DragonBall/ColdFire) 等等。最新的 MiniGUI 2.0 更为基于嵌入式 Linux 的高端嵌入式设备提供了完整的多进程支持, 从而将 MiniGUI 从中端市场带到了高端市场。

我们将 MiniGUI 定义为“针对嵌入式设备的、跨操作系统的图形界面支持系统”, 属于一种“嵌入式图形中间件”软件产品。目前, MiniGUI 已得到了国内最大的民营通信设备制造商、中国最大的电视机生产商、TD-SCDMA 终端方案供应商和全球最大的处理器生产厂商的认可及使用, 而在诸如工业仪表、医疗仪器、军工等行业, 更有众多行业领先厂商选择 MiniGUI 开发他们的嵌入式产品。与此同时, MiniGUI 也得到了海外嵌入式设备开发者的认可, 并远销到韩国、日本、台湾、马来西亚、北美、欧洲等地区; 2005 年 8 月, 韩国 KSP 公司成为飞漫软件全线产品的第一个海外代理。MiniGUI 业已成为嵌入式图形中间件领域的工业事实标准。

飞漫软件除了遵循 GPL 条款发布 MiniGUI 的某些版本 (MiniGUI v1.3.3 以及 MiniGUI-STR V1.6.2) 之外, 还为商业用户提供 MiniGUI 增值版产品以及其他关键应用软件产品。本白皮书将详细介绍 MiniGUI V2.0.3/1.6.9 版本的技术特点以及应用领域。

1.2 MiniGUI 的起源和发展

MiniGUI 的开发起始于 1998 年底, 到现在已历经近八年的时间。最初, MiniGUI 仅仅为了能够在 Linux 上显示中文而开发, 但目前, MiniGUI 已发展为一个跨操作系统的嵌入式图形支持系统。从诞生开始, MiniGUI 就不断得到实际项目和产品的应用, 同时新的项目也提出了更多技术需求, 于是 MiniGUI 就一步步发展成为跨操作系统的嵌入式图形中间件产品。

1998 年 12 月, 飞漫软件创始人魏永明开始开发 MiniGUI 并在一个数控系统中得到应用。2000 年 3 月, 联想 HappyLinux 1.0 发行版采用 MiniGUI 开发其安装程序。这时, MiniGUI 已形成了一个较为完整的嵌入式图形用户界面支持系统。2000 年 4 月到 2002 年 9 月, MiniGUI 作为中国为数不多的几个自由软件项目之一, 继续以自由软件的形式进行开发和维护。

2002 年 9 月, MiniGUI 的主要开发者成立了北京飞漫软件技术有限公司, 尝试自由软件的商业化运作模式, 并于 2003 年 5 月发布了 MiniGUI 1.2.6 版本; 于 2003 年 9 月发布了 MiniGUI 1.3.0 版本。

2003 年 10 月, MiniGUI 完成了到 uClinux 和 eCos 操作系统的移植。至此, MiniGUI 成为一个跨平台的嵌入式图形用户界面支持系统。

¹ GPL 是自由软件基金会为自由软件定义的授权条款, 详情请见 <http://www.gnu.org>。

2004 年 8 月，中国最大的民营通信设备制造商之一（华为）选择 MiniGUI 作为平台产品，用于机顶盒、手持终端等产品领域；2005 年 1 月，TD-SCDMA 标准的主要制定者（大唐移动）采用 MiniGUI 作为商用 TD-SCDMA 手机的 MMI 方案²；2005 年 8 月，全球最大的处理器生产厂商之一（Intel）采用 MiniGUI 开发家庭数字多媒体网关产品。

目前，MiniGUI 已发展到 2.0.3/1.6.9 版本³，支持 Linux/uClinux、VxWorks、eCos、uC/OS-II、pSOS、ThreadX、Nucleus、OSE 等操作系统，也可以在 Win32 平台上运行。飞漫软件基于自由软件的商业化模式也获得了初步的成功。

1.3 MiniGUI 的应用领域

从最初的数控系统到目前流行的智能手持终端设备，MiniGUI 已经应用于大量产品。MiniGUI 最主要的应用领域大致可分为三类：

中高端手机、WiFi 手机、PDA 类产品。图 1.1 是采用 eCos 操作系统，基于 MiniGUI 开发的高端智能手机的应用程序界面。

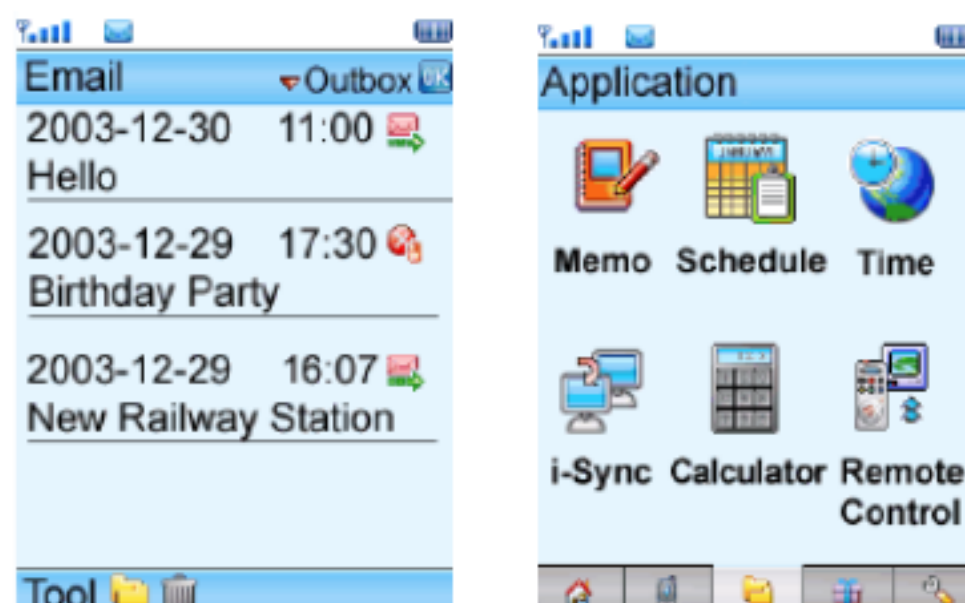


图 1.1 MiniGUI 典型应用：智能手持设备

数字媒体及机顶盒类产品。图 1.2 是基于 MiniGUI 开发的机顶盒浏览器产品以及由飞漫软件开发的法律政务查询终端产品。



图 1.2 MiniGUI 典型应用：数字媒体和机顶盒

² 其中也包括飞漫软件的另一个产品 mGarden（原 Fhas）。

³ MiniGUI 2.0.3 主要用于支持具有多进程特性的操作系统，如 Linux 和 VxWorks 6；1.6.9 版本主要用来支持基于线程或者任务的传统实时嵌入式操作系统。

工业仪表及控制系统。图 1.3 是基于 Linux 和 MiniGUI 操作系统开发的数控系统、工业仪表及医疗仪器的界面。



图 1.3 MiniGUI 典型应用：工业仪表及控制系统

2 MiniGUI 的优势

2.1 MiniGUI 的技术特点

MiniGUI 为实时嵌入式操作系统提供了非常完善的图形 (Graphics) 及用户界面 (UI , User Interface) 支持。MiniGUI 本身的可移植性设计, 使得不论在哪个硬件平台、哪种操作系统上运行, MiniGUI 均能为上层应用程序提供一致的应用程序编程接口 (API)。

首先, MiniGUI 是一个根据嵌入式系统应用特点量身定做的完整的图形支持系统。作为操作系统和应用程序之间的中间件, MiniGUI 将底层操作系统及硬件平台差别隐藏了起来, 并对上层应用程序提供了一致的功能特性, 这些功能特性主要包括:

- 1) 跨操作系统支持, 具体包括普通嵌入式 Linux/uClinux 、 VxWorks 、 eCos 、 uC/OS-II 、 pSOS 、 ThreadX 、 Nucleus 、 OSE 等, 同时还提供 Win32 平台上的 SDK 开发包, 方便嵌入式应用程序的开发和调试。
- 2) 多运行模式支持, 为了适应不同的操作系统运行环境, MiniGUI 可配置成三种运行模式: MiniGUI-Threads 、 MiniGUI-Processes⁴ 及 MiniGUI-Standalone 。
- 3) 内建资源支持。我们可以将 MiniGUI 所使用的资源, 诸如位图、图标和字体等编译到函数库中, 该特性可提高 MiniGUI 的初始化速度, 并且非常适合 eCos/uCOS-II/ThreadX 等无文件系统支持的实时嵌入式操作系统。
- 4) 完备的多窗口机制和消息传递机制。
- 5) 提供常用的控件类, 包括静态文本框、按钮、单行和多行编辑框、列表框、组合框、菜单按钮、进度条、滑块、属性页、工具栏、树型控件、月历控件、旋钮控件、酷工具栏、网格控件、动画控件等。
- 6) 对话框和消息框支持。
- 7) 其它 GUI 元素, 包括菜单、加速键、插入符、定时器等。
- 8) 界面皮肤支持。用户可通过皮肤支持获得华丽的图形界面。
- 9) 支持低端显示设备 (比如单色 LCD) 和高端显示设备 (8 位色及以上显示设备) 。通过 MiniGUI 的图形抽象层及图形引擎技术, 还可以支持特殊的显示设备, 比如 YUV 显示设备。
- 10) 提供有增强 GDI 函数, 包括光栅操作、复杂区域处理、椭圆、圆弧、多边形以及区域填充等函数。在提供有兼容于 C99 规范的数学库平台上, 还提供有高级二维绘图函数, 可设置线宽、线型以及填充模式等。通过 MiniGUI 的图形抽象层及图形引擎技术, 我们也可以让上述高级 GDI 接口在低端显示屏上实现。
- 11) Windows 的资源文件支持, 如位图、图标、光标等。
- 12) 各种流行图像文件的支持, 包括 Windows BMP 、 GIF 、 JPEG 、 PNG 等 (JPEG 及 PNG 的支持通过 libjpeg 及 libpng 函数库提供) 。
- 13) 多字符集和多字体支持, 目前支持 ISO8859-1 ~ ISO8859-15 、 GB2312 、 GBK 、 GB18030 、 BIG5 、 EUC-JP 、 Shift-JIS 、 EUC-KR 、 UNICODE (UTF-8 编码) 等字符集, 支持等宽点阵字体、变宽点阵字体、 Qt/Embedded 使用的嵌入式字体 QPF 、 TrueType 以及 Adobe Type1 等矢量字体 (TrueType 及 Type1 的支持分别通过 freetype 1.3 及 libt1 函数库提供) 。同时, 对于点阵字体 MiniGUI 也可以进行自动放大处理, 并可针对电视等特殊显示设备实现防锯齿功能。

⁴ 该运行模式目前在 MiniGUI V2.0.3 版本中提供, 主要针对 Linux 以及 VxWorks 6 操作系统; 在 Linux 操作系统上运行的 MiniGUI V1.6.9 版本, 提供一个简化的多进程运行模式, 称为 “ MiniGUI-Lite ”。

- 14) 多种键盘布局的支持。 MiniGUI 除支持常见的美式 PC 键盘布局之外,还支持法语、德语等西欧语种的键盘布局。
- 15) 输入法支持,用于提供各种可能的输入形式;内建有适合 PC 平台的汉字 (GB2312) 输入法支持,包括内码、全拼、智能拼音、五笔及自然码等。
- 16) 针对嵌入式系统的特殊支持,包括一般性的 I/O 流操作,字节序相关函数等。
- 17) 副屏支持。当系统中有多个视频设备时,可将一个作为 MiniGUI 的主屏,实现完整的多窗口系统;而其它设备作为副屏,在其上通过 MiniGUI 的图形接口来实现文字渲染、图形显示等功能。

其次,在 MiniGUI 近八年的发展过程中,有许多值得一提的技术创新点。正是由于这些技术上的创新,才使得 MiniGUI 更加适合实时嵌入式系统,而且也使得 MiniGUI 的灵活性非常好,可以应用在包括手持设备、机顶盒、游戏终端等等在内的各种高端或低端的嵌入式系统当中。 这些技术创新包括:

- 1) 图形和输入抽象层。 图形和输入抽象层对顶层 API 基本没有影响,但大大方便了 MiniGUI 自身以及应用程序的移植、调试等工作。 MiniGUI 现在已经被证明能够在基于 i386、ARM (包括 StrongARM、xScale 等)、MIPS、PowerPC,以及低端的 DragonBall、ColdFire 等 CPU 的嵌入式系统上流畅运行。 利用图形和输入抽象层,我们还可以实现基于软件结构的引擎。 比如模拟现实输入的 random 输入引擎,通过它,我们可以完成应用程序的自动测试。再比如,我们可以通过 Shadow 图形引擎支持 YUV 输出设备、提供对无法直接访问显示帧缓冲区的图形芯片的支持、提供对低于 8 位色显示屏的支持、实现屏幕旋转功能等等。
- 2) 多字体和多字符集支持。这部分通过设备上下文 (DC) 的逻辑字体 (LOGFONT) 实现,不管是字体类型还是字符集,都可以非常方便地进行扩充。利用 DrawText 等函数时,可通过指定字体而获得对各种字符集的支持, 比如 GB2312、GBK、BIG5、EUCKR、ShiftJIS、UTF-8 等等。这样,在单个 MiniGUI 应用程序中,同时显示多个不同语种的文字非常容易。 MiniGUI 的这种字符集支持不同于通过 UNICODE 内码实现的传统多字符集支持,这种实现占用资源少,更加适合于嵌入式系统。
- 3) 针对不同操作系统特点的运行模式。和 Linux 这样的类 UNIX 操作系统相比,一般意义上的传统嵌入式操作系统具有一些特殊性。举例而言,诸如 uClinux、uC/OS-II、eCos、VxWorks 等操作系统,通常运行在没有 MMU (内存管理单元,用于提供虚拟内存支持)的 CPU 上;这时,往往就没有进程的概念,而只有线程或者任务的概念,这样, GUI 系统的运行环境也就大相径庭。因此,为了适合不同的操作系统环境,我们可将 MiniGUI 配置成 MiniGUI-Threads、MiniGUI-Processes 及 MiniGUI-Standalone 三种运行模式。

最后,在 MiniGUI 2.0.x 的版本中,我们不仅通过 MiniGUI-Processes 运行模式提供了对 Linux 等操作系统的多进程环境的完整支持,而且在 MiniGUI 2.0.x 中保留了层的概念,这使得在 MiniGUI 下也可以实现类似 X Window 的工作区功能,而所占的系统资源却大大低于 X Window,完全适合应用在高端嵌入式设备上。

MiniGUI 新版本在图形方面的增强和提高,特别是 2.0.x 版本中对多进程的完整支持,将大大扩展它的应用领域,从而能够对嵌入式操作系统上的多媒体应用、游戏开发等提供更为强大的支持。

2.2 MiniGUI 的技术优势

和其它针对嵌入式产品的图形系统相比, MiniGUI 在对系统的需求上具有如下几大优势:

- 1) 可伸缩性强

MiniGUI 丰富的功能和可配置特性,使得它既可运行于 CPU 主频只有 60MHz 的低端产品中,

亦可运行于高端嵌入式设备中，并使用 MiniGUI 的高级控件风格及皮肤界面等技术，创建华丽的用户界面。

MiniGUI 的跨操作系统特性，使得 MiniGUI 可运行在最简单的嵌入式操作系统之上，如 uC/OS-II，也可以运行在具有现代操作系统特性的嵌入式操作系统之上，如 Linux，而且 MiniGUI 为嵌入式 Linux 系统提供了完整的多窗口图形环境。

这些特性，使得 MiniGUI 具有非常强的可伸缩性。可伸缩性是 MiniGUI 从设计之初就考虑且不断完善而来的。这个特性使得 MiniGUI 可应用于简单的行业终端，也可应用于复杂的消费类电子产品。

2) 轻型、占用资源少

MiniGUI 是一个定位于轻量级的嵌入式图形库，对系统资源的需求完全考虑到了嵌入式设备的硬件情况，如 MiniGUI 库所占的空间最小可以裁剪到 500K 左右，对目前的嵌入式设备来说，满足这一条件是绰绰有余的。

此外，测试结果表明，MiniGUI 能够在 CPU 主频为 30 MHz，仅有 4M RAM 的系统上正常运行（使用 uClinux 操作系统），这是其它针对嵌入式产品的图形系统所无法达到的。

3) 高性能、高可靠性

MiniGUI 良好的体系结构及优化的图形接口，可确保最快的图形绘制速度。在设计之初，我们就充分考虑到了实时嵌入式系统的特点，针对多窗口环境下的图形绘制开展了大量的研究及开发，优化了 MiniGUI 的图形绘制性能及资源占用。MiniGUI 在大量实际系统中的应用，尤其在工业控制系统的应用，证明 MiniGUI 具有非常好的性能。

从 1999 年 MiniGUI 的第一个版本发布以来，就有许多产品和项目使用 MiniGUI，MiniGUI 也不断从这些产品或者项目当中获得发展动力和新的技术需求，逐渐提高了自身的可靠性和健壮性。

有关 MiniGUI 的最新成功案例，您可以访问：

<http://www.minigui.com/project/cindex.shtml>

4) 可配置性

为满足嵌入式系统各种各样的需求，必须要求 GUI 系统是可配置的。和 Linux 内核类似，MiniGUI 也实现了大量的编译配置选项，通过这些选项可指定 MiniGUI 库中包括哪些功能而同时不包括哪些功能。大体说来，我们可以在如下几个方面对 MiniGUI 进行定制配置：

指定 MiniGUI 要运行的操作系统。

指定生成基于线程的 MiniGUI-Threads 运行模式还是基于进程的 MiniGUI-Processes 运行模式，或者只是最简单的 MiniGUI-Standalone 运行模式。

指定需要支持的 GAL 引擎和 IAL 引擎，以及引擎相关选项。

指定需要支持的字体类型。

指定需要支持的字符集。

指定需要支持的图像文件格式。

指定需要支持的控件类。

指定控件的整体风格，是经典（CLASSIC）风格、平面（FLAT）风格还是时尚（FASHION）风格。

这些配置选项大大增强了 MiniGUI 的灵活性，对用户来讲，可针对具体的应用需求量体裁衣，开

发最适合产品需求的应用软件。

总之，将现代窗口和图形技术带入到嵌入式设备的 MiniGUI，是一个非常适合于实时嵌入式设备的高效、可靠、可定制、小巧灵活的图形用户界面支持系统，其主要优点可以总结如下：

- 支持多种嵌入式操作系统，具备优秀的可移植性
- 可伸缩的系统架构，易于扩展
- 功能丰富，可灵活剪裁
- 小体积高性能间的最佳平衡
- 广泛的应用领域

3 运行 MiniGUI 的系统需求

3.1 MiniGUI 所支持的操作系统

理论上，MiniGUI 可以运行在任意一个支持多任务的嵌入式操作系统上；目前已经过验证的操作系统包括 Linux/uClinux、VxWorks、eCos、uC/OS-II、pSOS、ThreadX、Nucleus 和 OSE。MiniGUI 也可以在 Win32 平台上运行。同时，在不同操作系统上的 MiniGUI，提供完全兼容的 API 接口。

3.2 MiniGUI 所支持的硬件平台

理论上讲，MiniGUI 的运行和具体的硬件平台无关；只要某个硬件平台上运行有 MiniGUI 所支持的某个操作系统，MiniGUI 就能在这个平台上运行。在业界使用的众多硬件平台中，其中已验证可运行 MiniGUI 的硬件平台包括 Intel x86、ARM (ARM7/ARM9/StrongARM/xScale/MX1)、PowerPC、MIPS、M68K (DragonBall/ColdFire)、富士通 FRV 等。

3.3 MiniGUI 对系统资源的占用情况

MiniGUI 本身的占用空间非常小，以嵌入式 Linux 操作系统为例，MiniGUI 的典型存储空间占用情况如下：

Linux 内核：300KB ~ 500KB (由系统需求决定) ；

文件系统：500KB ~ 2MB (由系统需求决定) ；

MiniGUI 支持库：500KB ~ 900KB (由编译选项确定) ；

MiniGUI 字体、位图等资源：典型 400KB (由应用程序需求确定，最低可在 200KB 以内) ；

应用程序：100KB ~ 2MB (由具体的应用需求确定) 。

总体的系统占有空间应该在 2MB 到 4MB 左右。在某些系统上，尤其是在传统嵌入式操作系统中，功能完备的 MiniGUI 系统本身所占用的空间可进一步缩小到 1MB 以内。

有关 MiniGUI 2.0.3/1.6.9 版本资源占用情况的具体数据，可参阅《MiniGUI Data Sheet》V2.0-3。

4 MiniGUI 的软件架构

4.1 基于 MiniGUI 的嵌入式系统结构

为什么 MiniGUI 能够在如此众多的嵌入式操作系统上运行？这是因为 MiniGUI 具有良好的软件架构，通过抽象层将 MiniGUI 上层和底层操作系统隔离开来。如图 4.1 所示，基于 MiniGUI 的应用程序一般通过 ANSI C 库、操作系统和驱动程序接口以及 MiniGUI 自身提供的 API 来实现自己的功能；MiniGUI 中的“可移植层”可将特定操作系统及底层硬件的细节隐藏起来，而上层应用程序则无需关心底层的硬件平台输出和输入设备。

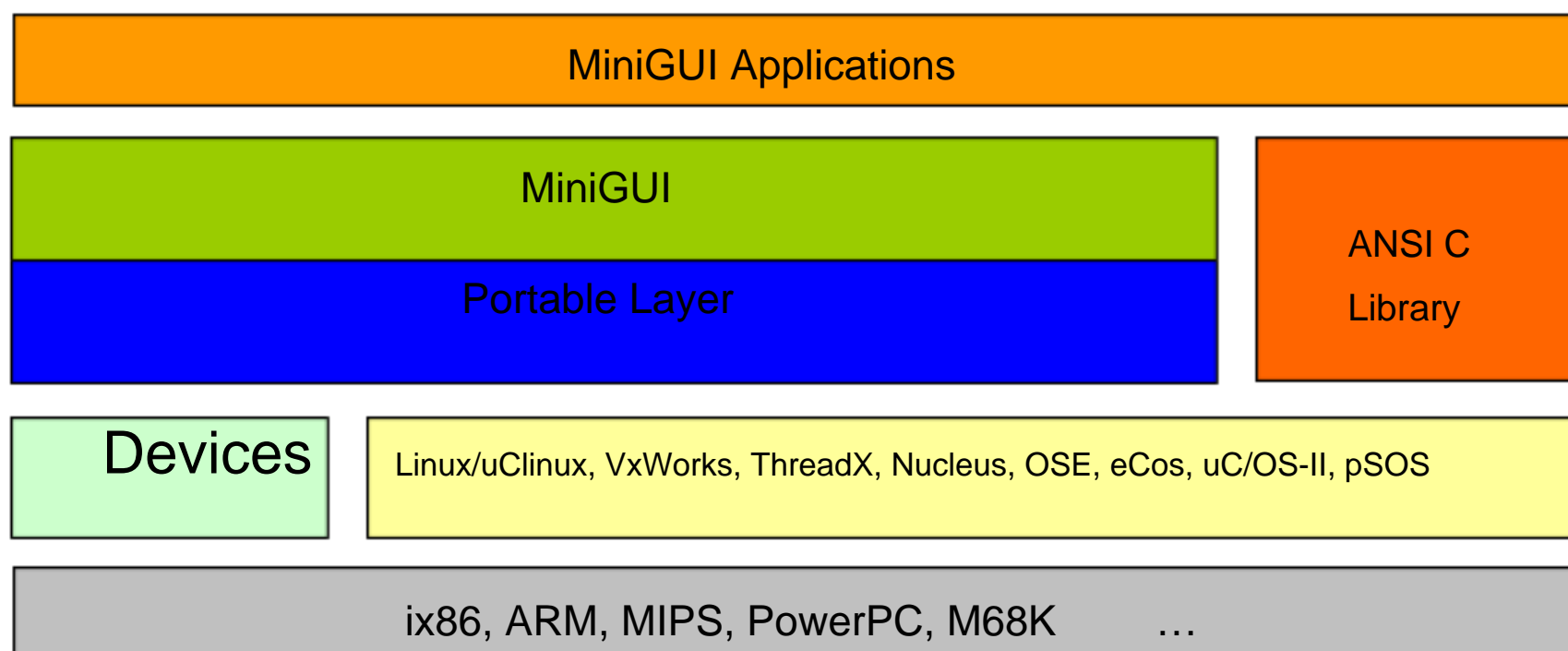


图 4.1 MiniGUI 和嵌入式操作系统的关系

另外，MiniGUI 特有的运行模式概念，也为跨操作系统的支持提供了便利。

4.2 MiniGUI 运行模式

如前所述，和 Linux 这样的类 UNIX 操作系统相比，一般意义上的传统嵌入式操作系统具有一些特殊性。举例而言，诸如 uClinux、uC/OS-II、eCos 等操作系统，通常运行在没有 MMU（内存管理单元，用于提供虚拟内存支持）的 CPU 上；这时，往往就没有进程的概念，而只有线程或者任务的概念，这样，MiniGUI 的运行环境也就大相径庭。因此，为了适合不同的操作系统环境，我们可将 MiniGUI 配置成三种不同的运行模式：

MiniGUI-Threads。运行在 MiniGUI-Threads 上的程序可以在不同的线程中建立多个窗口，但所有的窗口在一个进程或者地址空间中运行。这种运行模式主要用来支持大多数传统意义上的嵌入式操作系统，比如 VxWorks、ThreadX、Nucleus、OSE、pSOS、uC/OS-II、eCos 等等。当然，在 Linux 和 uClinux 上，MiniGUI 也能以 MiniGUI-Threads 的模式运行。

MiniGUI-Processes⁵。和 MiniGUI-Threads 相反，MiniGUI-Processes 上的每个程序是单独的进程，每个进程也可以建立多个窗口，并且实现了多进程窗口系统。MiniGUI-Processes 适

⁵ 在 Linux 操作系统上运行的 MiniGUI 1.6.x 版本中，该运行模式称为“MiniGUI-Lite”。MiniGUI-Lite 为多进程环境的 Linux 操作系统提供了折中解决方案，但没有解决进程间的窗口层叠问题。而 MiniGUI 2.0.x 实现的 MiniGUI-Processes 模式为 Linux 等多进程操作系统提供了完整的图形界面解决方案。

合于具有完整 UNIX 特性的嵌入式操作系统，比如嵌入式 Linux 、 VxWorks 6 等⁶。该运行模式在 MiniGUI V2.0.3 中提供，有关该模式的详细介绍将在下面阐述。

MiniGUI-Standalone 。这种运行模式下， MiniGUI 可以以独立任务的方式运行，既不需要多线程也不需要多进程的支持，这种运行模式适合功能单一的应用场合。比如在一些使用 uClinux 的嵌入式产品中，因为各种原因而缺少线程支持，这时，就可以使用 MiniGUI-Standalone 来开发应用软件。

一般而言， MiniGUI-Standalone 模式的适应面最广，可以支持几乎所有的操作系统⁷； MiniGUI-Threads 模式的适用面次之，可运行在支持多任务的实时嵌入式操作系统，或者具备完整 UNIX 特性的普通操作系统； MiniGUI-Processes 模式的适用面较小，它仅适合于具备完整 UNIX 特性的嵌入式操作系统，比如 Linux 和 VxWorks 6。

但不论采用哪种运行模式， MiniGUI 为上层应用软件提供了最大程度上的一致性；只有少数几个涉及初始化的接口在不同运行模式上有所不同。

4.2.1 MiniGUI-Processes 运行模式

MiniGUI-Processes 运行模式是 MiniGUI V2.0.x 在 MiniGUI-Lite 运行模式基础上为具有多进程支持的嵌入式操作系统提供的。 MiniGUI V1.6.x 及以前版本为具有多进程特性的 Linux 操作系统提供 MiniGUI-Lite 运行模式，使之在高效的客户 / 服务器架构之上运行多个客户进程，从而充分利用进程地址空间保护这样的高级特性，有了这样的特性，可大大提高基于 MiniGUI 的嵌入式系统的灵活性、稳定性以及可扩展性。比如，我们可以在 MiniGUI-Lite 上运行多个 MiniGUI 客户进程，而单个进程的异常退出，不会影响其他的 MiniGUI 客户进程。而且在这种架构之上，我们可以非常方便地集成第三方应用程序。其实，这也是许多嵌入式设备开发商采用 Linux 操作系统的重要理由。

但是， MiniGUI-Lite 运行模式虽然提供了多进程支持，但无法同时管理来自不同进程间的窗口，因此， MiniGUI-Lite 用层的概念将不同的进程之间的窗口区分开来。这种实现方法虽然可适用于大多数屏幕分辨率较小的嵌入式设备，但仍然给应用程序的开发带来了不便。

MiniGUI V2.0 则彻底解决了上述问题。 MiniGUI V2.0 在 MiniGUI-Lite 运行模式基础上，实现了完整的多进程环境中的窗口系统，来自不同进程的窗口可以在同一桌面上协调存在。图 4.2 给出了 MiniGUI V1.6.x 的 MiniGUI-Lite 运行模式及 MiniGUI 2.0.x 的 MiniGUI-Processes 运行模式在运行相同应用程序情况下的界面效果。

⁶ MiniGUI V2.0.x for VxWorks 6 将于 2006 年底发布。

⁷ 目前只用来提供对 Linux/uClinux 操作系统的支持。

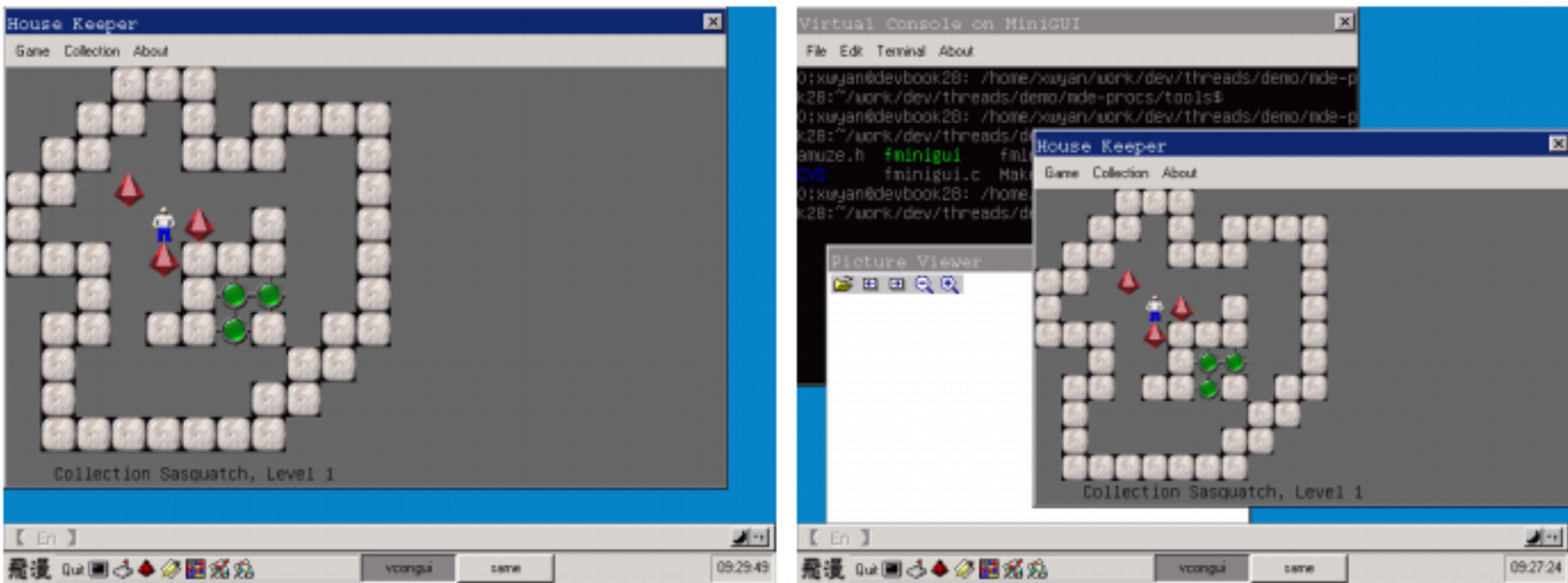


图 4.2 MiniGUI V1.6.x 的 MiniGUI-Lite 运行模式及 MiniGUI 2.0.x 的 MiniGUI-Processes 运行模式

图 4.2 中，第一个屏幕运行了 `vcngui` 和推箱子游戏这两个客户进程。可以看到，我们在运行推箱子程序之后，就看不到 `vcngui` 程序了；第二个屏幕运行 `vcngui`、`picview` 和推箱子游戏这三个客户进程，但我们可在桌面上看到所有的客户进程窗口。

相比 MiniGUI-Lite，MiniGUI-Processes 运行模式具有明显的优势。这使得 MiniGUI 不仅可适用于传统的嵌入式操作系统（MiniGUI-Threads），还可适用于具有多进程特性的嵌入式操作系统，比如 Linux 和 VxWorks 6 操作系统。另外，MiniGUI-Processes 也保留了 MiniGUI-Lite 的层概念，用户可以将来自不同客户进程的窗口放到不同的层中，从而实现类似 X Window 那样的工作区。有了 MiniGUI-Processes 运行模式，MiniGUI 的应用领域将进一步扩大，不仅可用于高端的嵌入式设备，还可能用于桌面环境。

4.2.2 各操作系统上可运行的 MiniGUI 运行模式

表 4.1 给出了 MiniGUI V2.0.x 和 V1.6.x 在各操作系统上可支持的运行模式。

表 4.1 MiniGUI 在操作系统上的运行模式

操作系统	MiniGUI 版本	所支持的运行模式
Linux	MiniGUI V2.0.x	MiniGUI-Processes MiniGUI-Threads MiniGUI-Standalone
Linux	MiniGUI V1.6.x	MiniGUI-Lite MiniGUI-Threads MiniGUI-Standalone
uClinux	MiniGUI V1.6.x	MiniGUI-Threads MiniGUI-Standalone
VxWorks 6.x	MiniGUI V2.0.x	MiniGUI-Processes MiniGUI-Threads
VxWorks 6.x	MiniGUI V1.6.x	MiniGUI-Threads
VxWorks 5.x	MiniGUI V1.6.x	MiniGUI-Threads
ThreadX	MiniGUI V1.6.x	MiniGUI-Threads
Nucleus	MiniGUI V1.6.x	MiniGUI-Threads
OSE	MiniGUI V1.6.x	MiniGUI-Threads
eCos	MiniGUI V1.6.x	MiniGUI-Threads
uC/OS-II	MiniGUI V1.6.x	MiniGUI-Threads
pSOS	MiniGUI V1.6.x	MiniGUI-Threads

4.3 窗口系统

在 MiniGUI 中窗口组织为层次体系结构的形式。根窗口作为所有窗口的祖先，除了根窗口以外的所有窗口都有父窗口，每一个窗口都可能有子窗口、兄弟窗口、祖先窗口和子孙窗口等。在同一级的窗口可以重叠，但是某个时刻只能有一个窗口输出到重叠区域。

MiniGUI 中有三种窗口类型：主窗口、对话框和控件窗口（子窗口）。主窗口通常包括一些子窗口，这些子窗口通常是控件窗口，也可以是自定义窗口类。应用程序还会创建其它类型的窗口，例如对话框和消息框。对话框本质上就是主窗口，应用程序一般通过对话框提示用户进行输入操作。

4.4 通讯机制

MiniGUI 下的通讯是一种类似于 Win32 的消息机制，对于运行在线程模式的 MiniGUI 版本，线程间的消息传递模型如下图所示，其中的 Desktop 线程充当一个微服务器，所有的消息在 Event 线程获取出来以后就会投递给 Desktop 线程，然后再分发到目的应用程序主窗口上面，如图 4.3 所示。

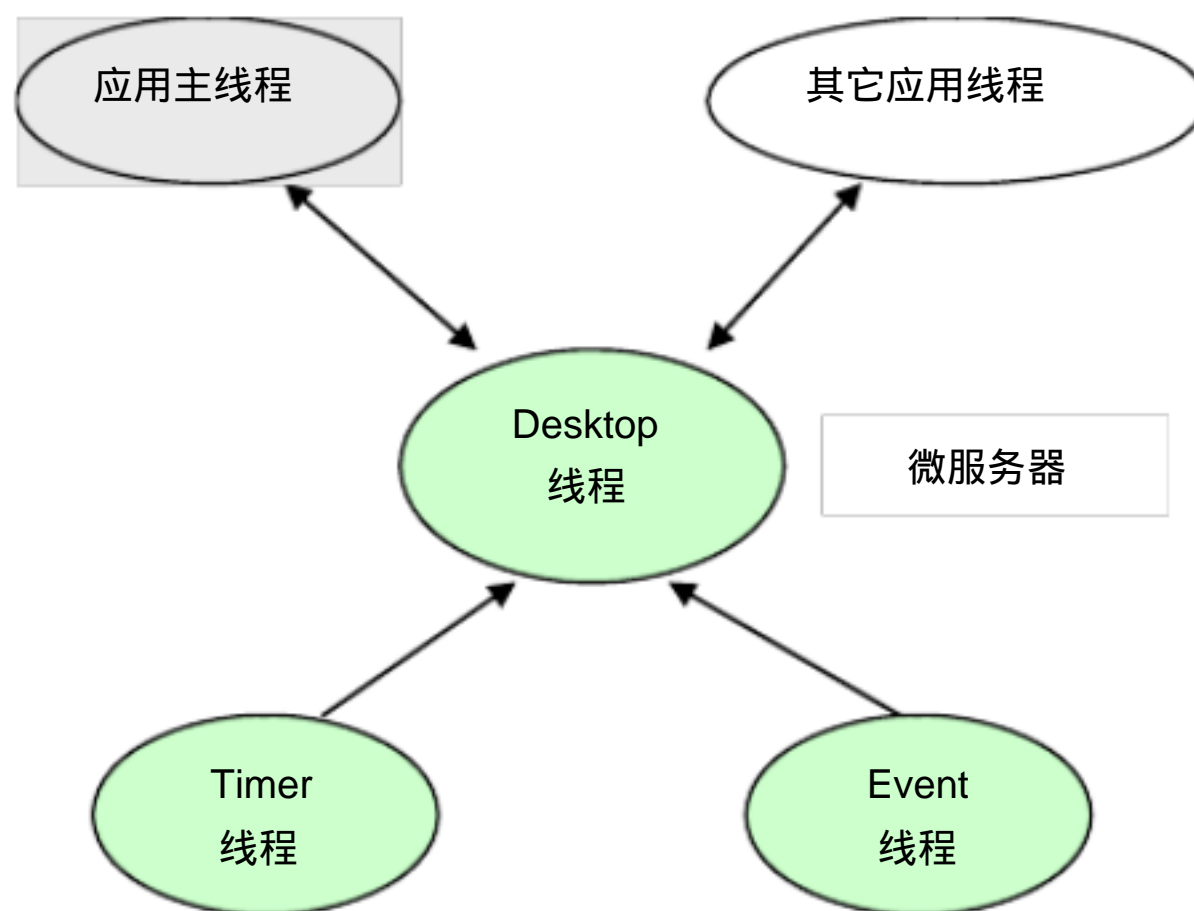


图 4.3 MiniGUI-Threads 运行模式的消息通讯机制

对于运行在进程版的 MiniGUI 来说，应用程序的消息传递则通过套接字来进行，相应的处理模型如图 4.4 所示。

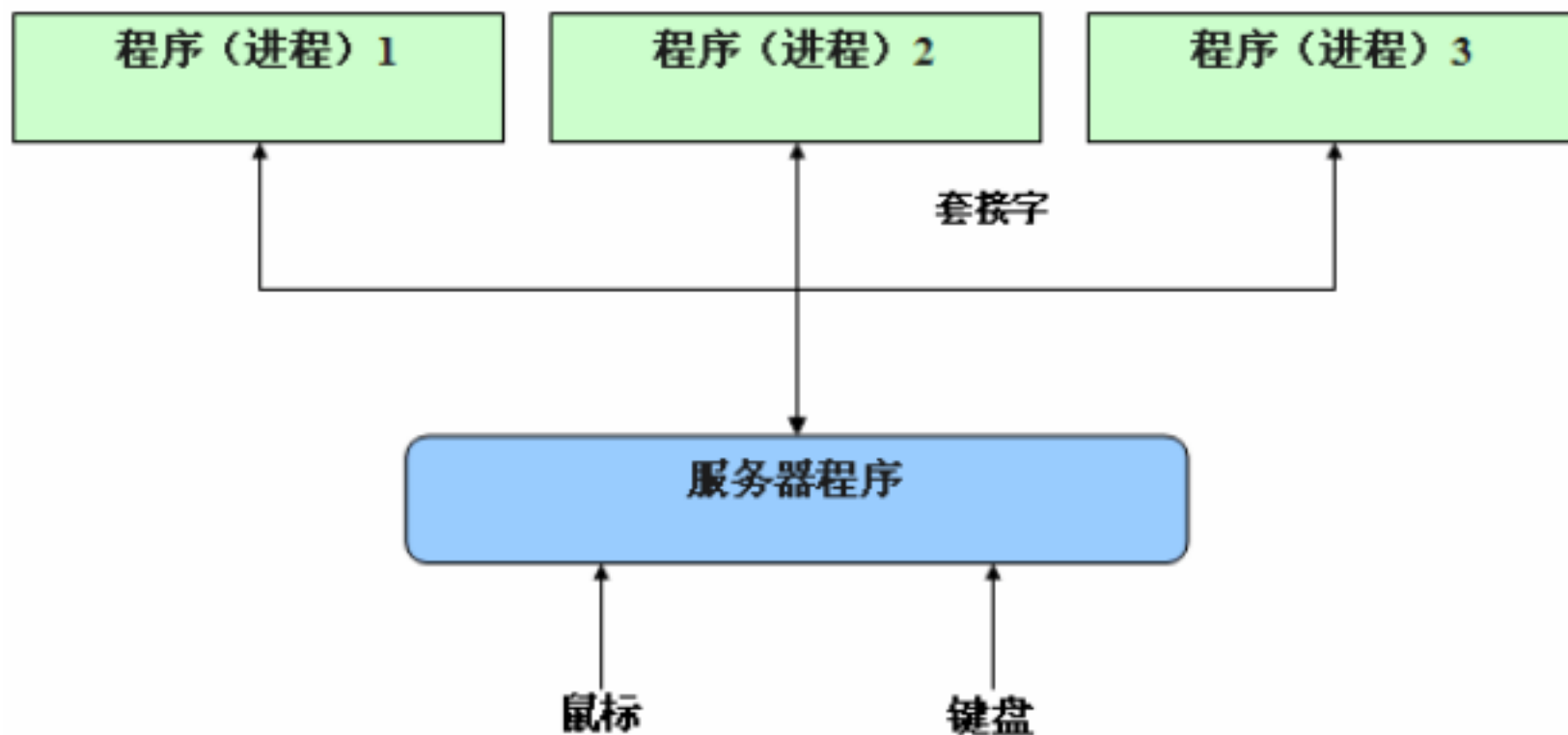


图 4.4 MiniGUI-Processes 运行模式的消息通讯机制

4.5 字体

MiniGUI 提供了对点阵字体及矢量字体的支持，到目前为止，MiniGUI 已经实现了对 RBF 和 VBF 字体（这是 MiniGUI 定义的两点阵字体格式），以及 TrueType 和 Adobe Type1 字体等的支持。同时 MiniGUI 还提供了对 QPF（Qt Pre-rendered Fonts）字体的支持。同时，MiniGUI 可以对点阵字体进行自动放大处理，并可针对电视等特殊显示设备实现防锯齿功能。

目前 MiniGUI 支持 ISO8859-1 ~ ISO8859-15、GB2312、GBK、GB18030、BIG5、EUC-JP、Shift-JIS、EUC-KR、UNICODE（UTF-8 编码）等字符集。

4.6 输入设备的支持

MiniGUI 支持各种通用的鼠标设备，对触摸屏的支持也非常出色，并针对触摸屏的校正为用户提供提供了校正接口。

MiniGUI 支持最多含 255 个键的各种键盘。此外，MiniGUI 还提供了对多种键盘布局的支持，除支持常见的美式 PC 键盘布局之外，还支持法语、德语等西欧语种的键盘布局。

除了常见的输入设备类型，如 PC 键盘、鼠标、触摸屏之外，MiniGUI 的输入抽象层机制也允许支持遥控器、小键盘、按键等特殊的输入设备。

4.7 输入法

输入法是 MiniGUI 为支持中文、韩文、日文等多字节字符集而引入的机制，和 Windows 系统下的输入法类似，输入法通常以顶层窗口的形式出现，并截获系统中的按键信息，经过适当的处理，将翻译之后的字符发送到当前活动窗口。MiniGUI 提供了多种输入法的支持，其中包括内码、全拼、智能拼音、五笔及自然码等输入法。

此外，在 MiniGUI 独立的输入法组件 mGi 中，还为用户提供了软键盘输入法及手写输入法的框架。

5 开发环境

基于 MiniGUI 的开发可以在 Linux 或 Windows 操作系统下进行。由于 MiniGUI 完全用 C 来编写，具有非常好的移植性，也使得 MiniGUI 应用程序的交叉编译工作十分方便。

为嵌入式设备编写的应用程序可以在任何安装在针对该设备的交叉编译工具链的平台上进行编译。最常见的方式是在 Linux 环境下安装 gcc 的交叉编译器，对应用程序进行编译。对于某些嵌入式系统（如 VxWorks，uC/OS-II），则一般在 Windows 下安装相应的编译环境（如 Tornado、ADS 等），对应用程序进行编译。

如果 MiniGUI 应用程序在 Linux 环境下开发，它可以有两种运行方式。一种是直接在内核支持的 FrameBuffer 控制台下运行，一种则是在一个模拟 FrameBuffer 的 X11 应用程序（qvfb）下运行并完成调试。

如果 MiniGUI 应用程序在 Windows 下开发，则可以使用 Visual Studio 集成开发环境进行开发及编译，并在模拟 FrameBuffer 的 Windows 应用程序（wvfb）下运行应用程序并调试（如图 5.1 所示）。



图 5.1 在 wvfb 模拟器上运行 MiniGUI 应用程序

直接在模拟器或控制台下运行调试 MiniGUI 应用程序，大大方便了嵌入式程序的开发，避免了用户重复刷写嵌入式设备的工作。同时也使得用户可以在开发主机上使用标准的调试器对应用程序进行调试。

6 程序样例和控件

6.1 Hello World 示例程序

下面是一个完整的 MiniGUI 应用程序，该程序在屏幕上创建一个大小为 240x180 像素的应用程序窗口，并在窗口客户区的中部显示 “ Hello world! ”，如图 6.1 所示。

```
#include <stdio.h>

#include <minigui/common.h>
#include <minigui/minigui.h>
#include <minigui/gdi.h>
#include <minigui/window.h>

static int HelloWinProc(HWND hWnd, int message, WPARAM wParam, LPARAM lParam)
{
    HDC hdc;
    switch (message) {
        case MSG_PAINT:
            hdc = BeginPaint (hWnd);
            TextOut (hdc, 60, 60, "Hello world!");
            EndPaint (hWnd, hdc);
            return 0;

        case MSG_CLOSE:
            DestroyMainWindow (hWnd);
            PostQuitMessage (hWnd);
            return 0;
    }
    return DefaultMainWinProc(hWnd, message, wParam, lParam);
}

int MiniGUIMain (int argc, const char* argv[])
{
    MSG Msg;
    HWND hMainWnd;
    MAINWINCREATE CreateInfo;

#ifdef _MGRM_PROCESSES
    JoinLayer(NAME_DEF_LAYER, "helloworld", 0, 0);
#endif

    CreateInfo.dwStyle = WS_VISIBLE | WS_BORDER | WS_CAPTION;
    CreateInfo.dwExStyle = WS_EX_NONE;
    CreateInfo.spCaption = "HelloWorld";
    CreateInfo.hMenu = 0;
    CreateInfo.hCursor = GetSystemCursor(0);
    CreateInfo.hIcon = 0;
    CreateInfo.MainWindowProc = HelloWinProc;
    CreateInfo.lx = 0;
    CreateInfo.ty = 0;
    CreateInfo.rx = 240;
    CreateInfo.by = 180;
    CreateInfo.iBkColor = COLOR_lightwhite;
    CreateInfo.dwAddData = 0;
    CreateInfo.hHosting = HWND_DESKTOP;

    hMainWnd = CreateMainWindow (&CreateInfo);

    if (hMainWnd == HWND_INVALID)
        return -1;

    ShowWindow(hMainWnd, SW_SHOWNORMAL);

    while (GetMessage(&Msg, hMainWnd)) {
        TranslateMessage(&Msg);
        DispatchMessage(&Msg);
    }

    MainWindowThreadCleanup (hMainWnd);

    return 0;
}
```



图 6.1 hello world 程序的输出

应用程序开发人员可以直接调用 MiniGUI 的窗口以及图形接口编写自己的应用程序，也可以利用 MiniGUI 内建的各种控件 (control/widget) 来快速开发自己的应用程序。MiniGUI 提供了各种丰富的控件，如按钮，工具栏等。同时还为开发者提供了自定义控件的接口，并能方便地对已有控件进行扩展。

6.2 静态框

静态框用来在窗口的特定位置显示文字、数字等信息，还可以用来显示一些静态的图片信息，比如公司徽标、产品商标等等。如图 6.2 所示。

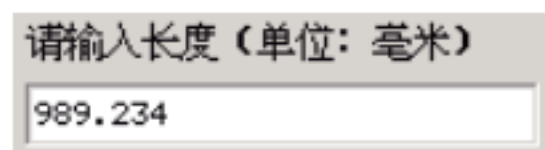


图 6.2 MiniGUI 的静态框控件 (用于其他控件的标签)

6.3 按钮

按钮是除静态框之外使用最为频繁的一种控件。按钮通常用来为用户提供开关选择。MiniGUI 的按钮可划分为普通按钮、复选框和单选钮等几种类型。如图 6.3 所示。

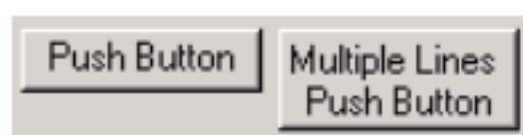


图 6.3 按钮控件

6.4 列表框

列表框通常为用户提供一系列的可选项，这些可选项显示在可滚动的子窗口中，用户可通过键盘或鼠标操作来选中某一项或者多个项。如图 6.4 所示。

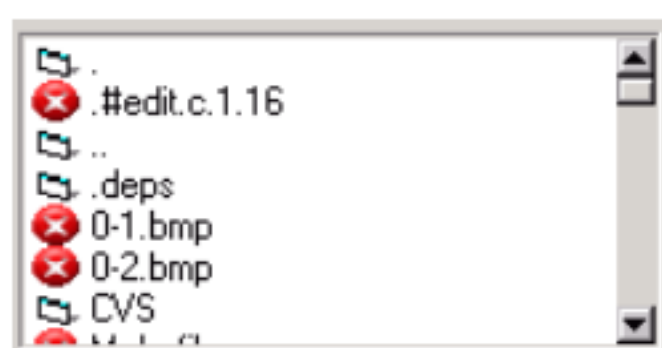


图 6.4 列表框控件

6.5 编辑框

编辑框为应用程序提供了接收用户输入和编辑文字的重要途径。MiniGUI 中提供了三种类型的编辑框，分别对应于三种控件类，它们是：简单编辑框（`EDIT`）（不推荐使用）、单行编辑框（`SLEDIT`）和多行编辑框（`TEXTEDIT`）。如图 6.5 所示。



图 6.5 编辑框控件

6.6 组合框

通常的组合框就是编辑框和列表框的组合。用户可以直接在编辑框中键入文本，也可以从列表框列出的可选项中选择一个已有的条目。MiniGUI 中的组合框可以划分为三种类型：简单组合框、下拉式组合框、旋钮组合框和旋钮数字框。如图 6.6 和图 6.7 所示。



图 6.6 下拉式组合框



图 6.7 旋钮组合框

6.7 菜单按钮

从外观上看，菜单按钮类似于一个普通按钮，不同的是在按钮矩形区域的右侧有一个向下的箭头。当用户点击该控件时，就会弹出一个菜单，而用户使用鼠标点击菜单中某一条目时，按钮的内容就变为该条目的内容。如图 6.8 所示。

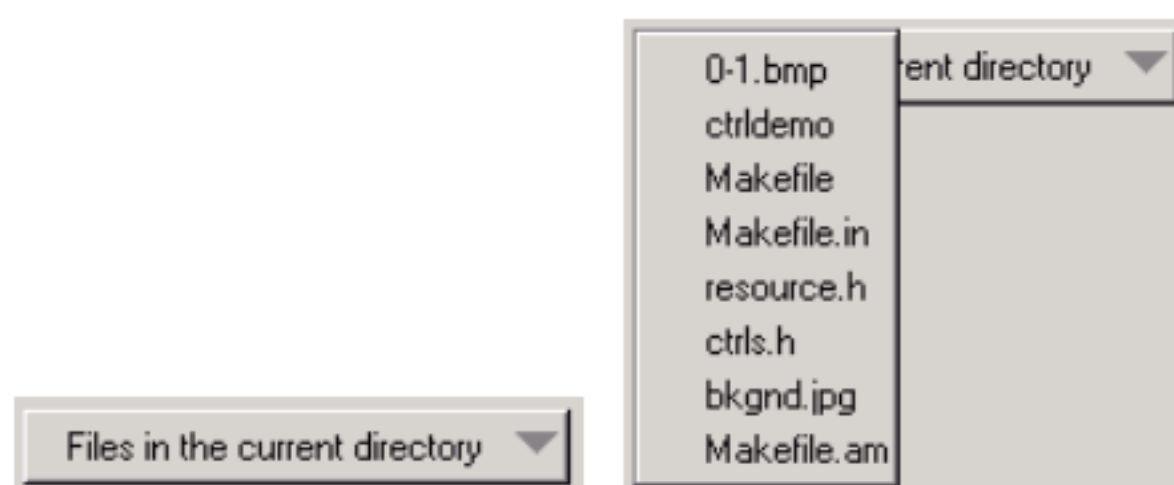


图 6.8 菜单按钮（左图为平常状态，右图为弹出菜单后的效果）

6.8 进度条

进度条通常用来向用户提示某项任务的完成进度，经常用于文件复制、软件安装等程序中。MiniGUI 的进度条可分水平进度条及竖直进度条两种风格。如图 6.9 和图 6.10 所示。

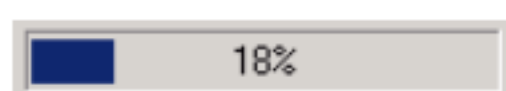


图 6.9 水平进度条



图 6.10 竖直进度条

6.9 滑块

滑块通常用于调节亮度、音量等的场合。在对某一范围的量值进行调节的场合，就可以使用滑块控件。如图 6.11 所示。

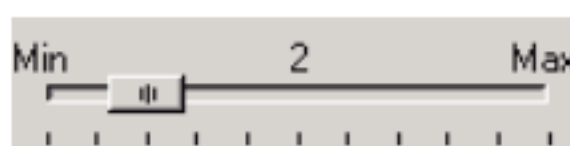


图 6.11 滑块控件

6.10 工具栏

MiniGUI 提供了三种不同的预定义工具栏控件类，分别是 TOOLBAR（不推荐使用）、NEWTOLBAR（推荐使用）以及 MiniGUIExt 库中的 COOLBAR 控件类。NEWTOLBAR 控件的显示效果如图 6.12 所示。

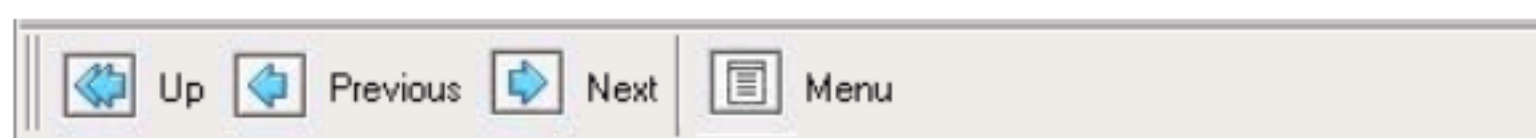


图 6.12 工具栏控件

6.11 属性表

属性表最常见的用途就是将本该属于不同对话框的交互内容分门别类地放在同一对话框中，一方面节省了对话框空间，另一方面也使得交互界面更加容易使用。如图 6.13 所示。

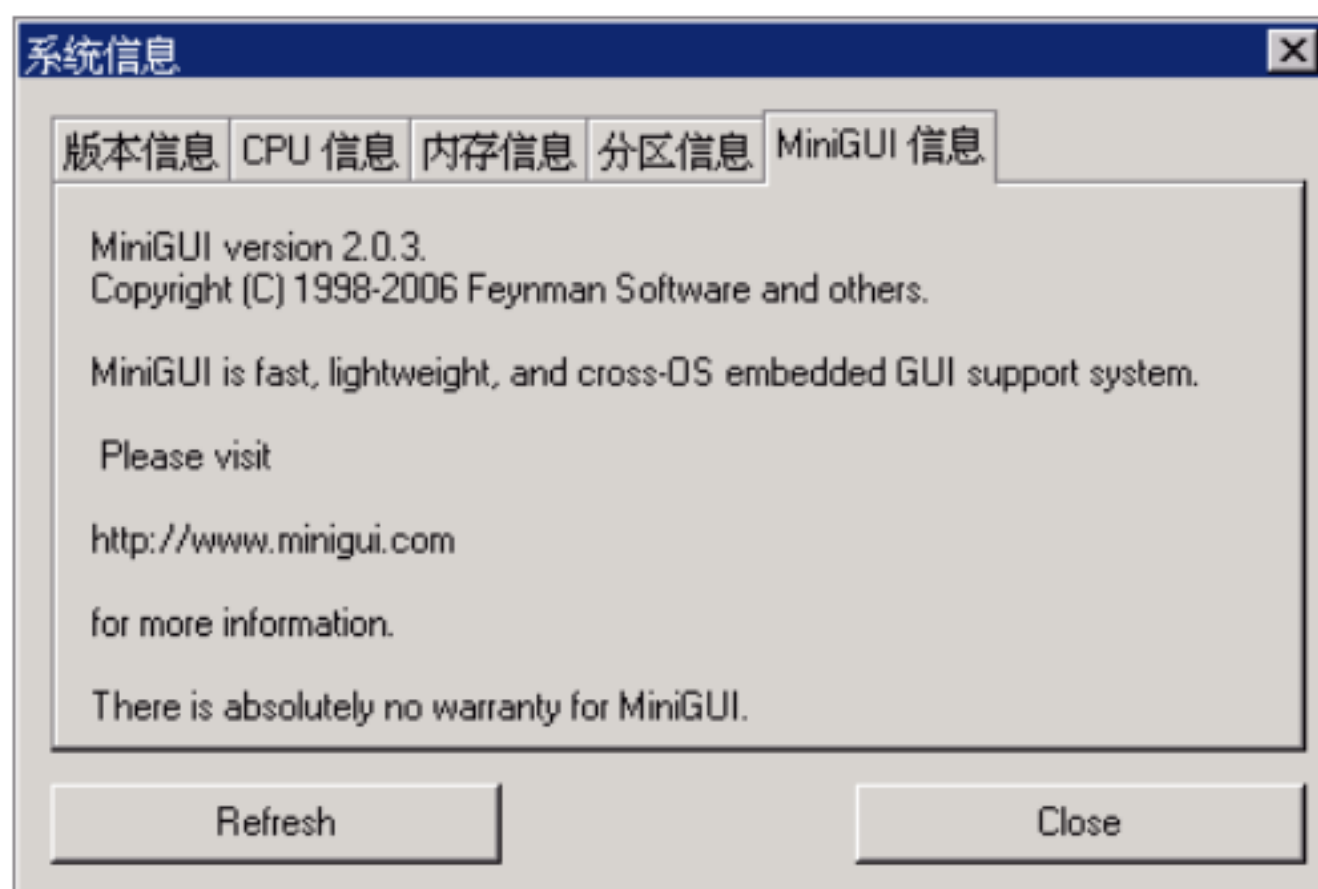


图 6.13 属性页控件

6.12 滚动型控件

滚动型 (ScrollView) 控件的主要用途是显示和处理列表项。滚动型控件中，列表项的绘制是完全由应用程序自己确定的，也就是说，滚动型控件是一个可定制性很强的控件。如图 6.14 所示。

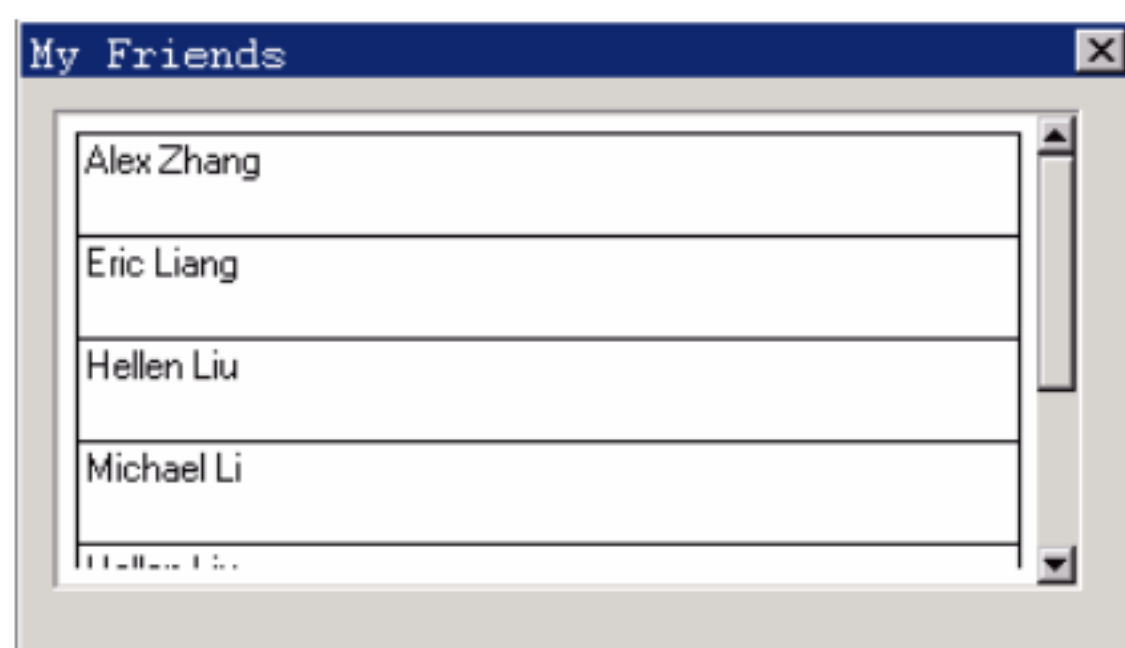


图 6.14 滚动型控件

6.13 树型控件

树型控件 (TreeView) 以树型的方式显示一系列的分层次的项，每个项 (子项) 可以包括一个或多个子项。每项或子项包括文字标题和可选的图标，用户可以通过点击该项来展开或折叠该项中的子项。树型控件比较适合用来表示具有从属关系的对象，例如，文件、目录结构，或者某一机构的组织情况。如图 6.15 所示。



图 6.15 树型控件

6.14 列表型控件

列表型控件 (ListView) 以列表的方式显示一系列的数据项 (列表项) ，每个列表项的内容可以由一个或多个子项构成，不同列表项的相同类型子项以列的方式组织，列表型控件的表头 (header) 内容通常反映了列表项不同子项的意义。外观上，列表型控件就是一个包括表头部分和列表项部分的矩形框。可以通过拖动表头来调整列表型控件中各个子项的列宽，列表中显示不下的内容可以通过滚动条来滚动显示。如图 6.16 所示。



图 6.16 列表型控件的使用

6.15 月历控件

月历控件 (MonthCalendar) 提供一个类似日历的用户界面，使用户可以方便的选择和设置日期。应用程序可以通过向月历控件发送消息来获取和设置日期。如图 6.17 所示。



图 6.17 月历控件

6.16 动画控件

动画控件 (Animation) 是一个可以显示 (GIF) 动画的控件。如图 6.18 所示。

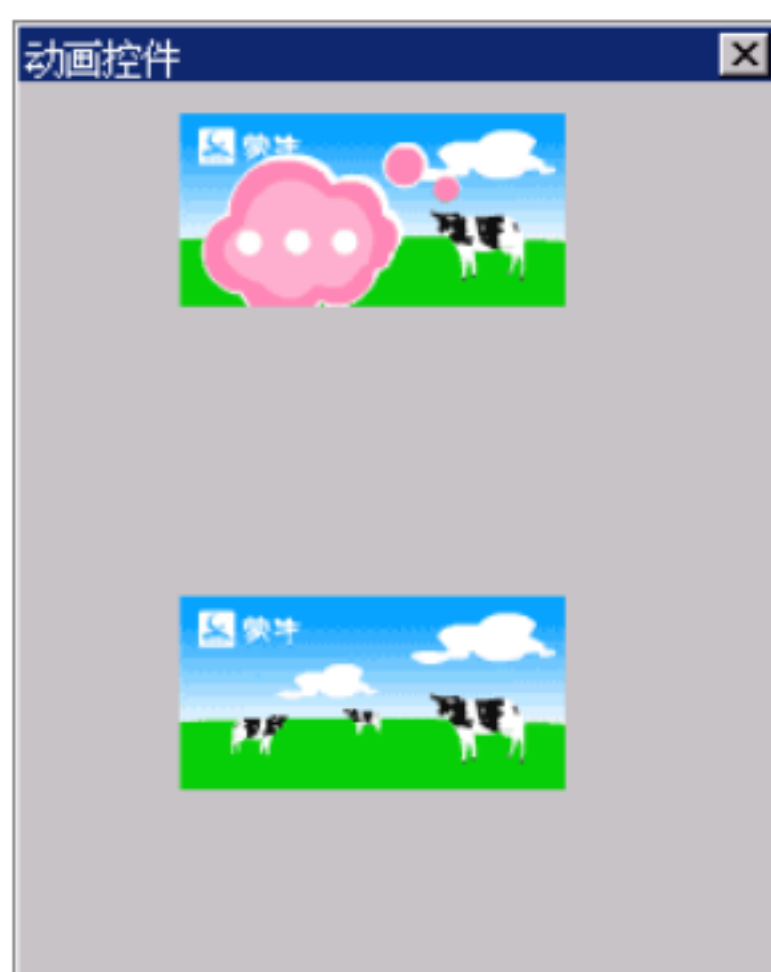


图 6.18 动画控件

6.17 网格控件

网格控件 (GridView) 以表格的方式显示一系列的数据项 (单元格) ，每个单元格的内容相互独立，网格控件的表头 (header) (包括一列的头和一行的头) 内容通常反映了表格一行或者一列的意义。外观上，网格控件就是一个包括表头部分的单元格部分的矩形框。可以通过拖动表头来调整网格控件中行的高度或者列的宽度。如图 6.19 所示。

网格控件				
	语文	数学	英语	总分
小明	50.00000	50.00000	50.00000	150.0000
小强	50.00000	50.00000	50.00000	150.0000
小亮	50.00000	50.00000	50.00000	150.0000
小力	50.00000	50.00000	50.00000	150.0000
平均分	50.00000	50.00000	50.00000	150.0000

图 6.19 网格控件

6.18 图标型控件

图标型控件（IconView）提供一个以图标加标签文字的方式供用户浏览条目的界面。这些图标项显示在可滚动的子窗口中，用户可通过键盘或鼠标操作来选中某一项或者多个项，选中的图标项通常高亮显示。图标型控件的典型用法是作为桌面图标的容器和目录下文件的显示。如图 6.20 所示。



图 6.20 图标型控件

7 MiniGUI 窗口和控件的显示风格

目前 MiniGUI 为用户提供了三种窗口和控件的整体显示风格：CLASSIC、FLAT 和 FASHION。这三种风格适用于不同的显示设备，用户可以在配置 MiniGUI 时指定相应的选项来将 MiniGUI 编译成特定的一种显示风格。

其中 CLASSIC 是一种类似于 Windows 的窗口显示风格，如图 7.1 所示。

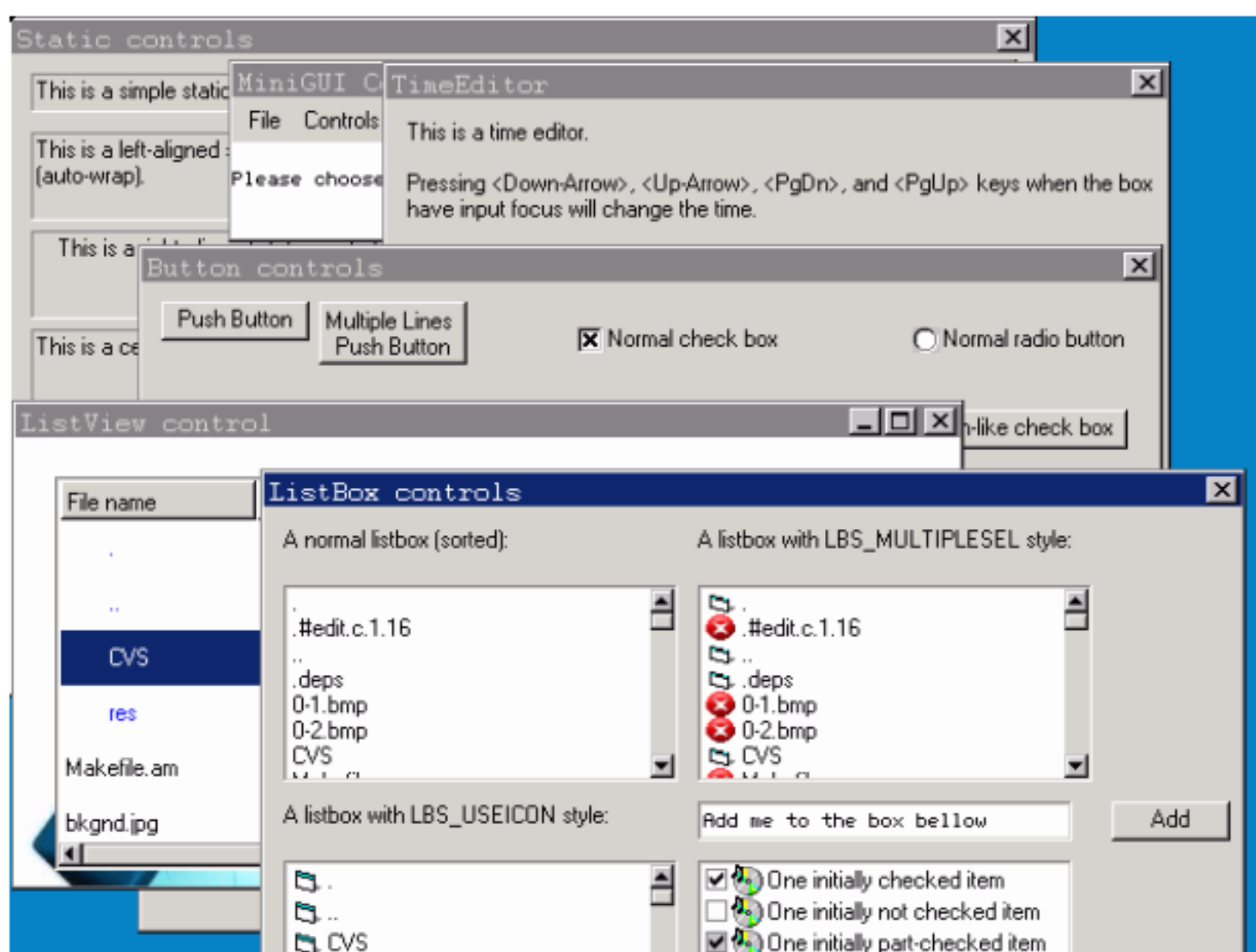


图 7.1 MiniGUI 的默认显示风格 (CLASSIC 风格)

FLAT 是一种适用于灰度屏的窗口显示风格，见图 7.2。

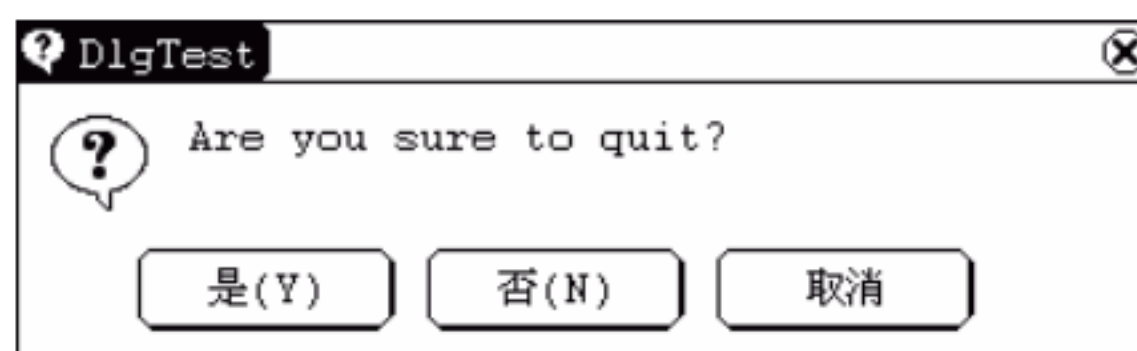


图 7.2 MiniGUI 的 FLAT 显示风格

FASHION 是适用于手持终端设备显示的华丽显示风格，如图 7.3 所示。

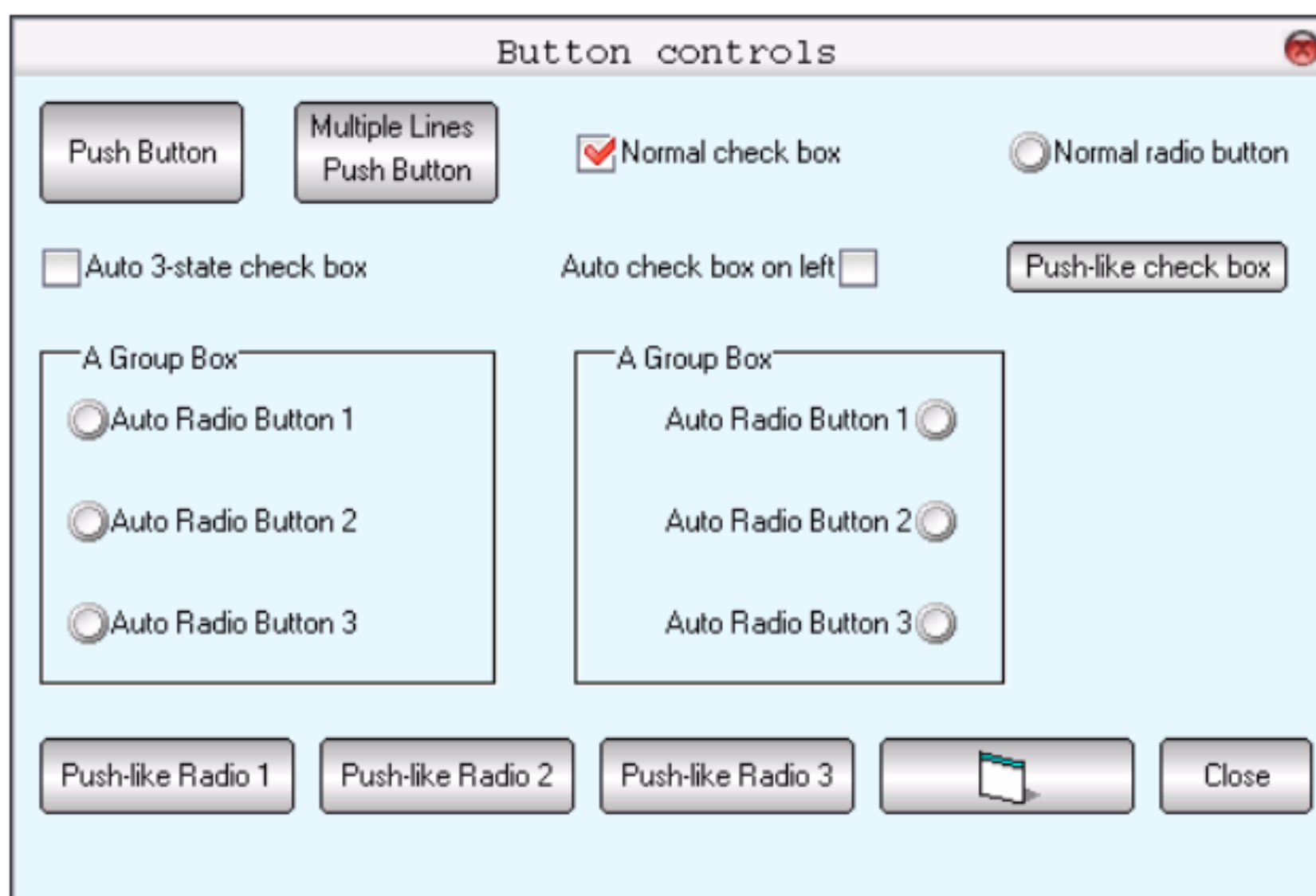


图 7.3 MiniGUI 的 FASHION 显示风格

8 国际化

MiniGUI 可以支持 ISO8859-1 ~ ISO8859-15 、GB2312 、GBK 、GB18030 、BIG5 、EUCKR 、EUCJP 、Shift-JIS 、UNICODE 等字符集，而且通过非 Unicode 内码的实现，对系统资源要求低，更加适用于嵌入式系统。

MiniGUI 完全支持 Unicode 这一国际化标准的字符集，这也就使得开发人员可以非常方便地将英文、中文、韩文、日文或者其它任何 Unicode 支持的语言应用到他们的程序里面。

在 MiniGUI 应用程序开发中，如果只需要显示某个特定字符集的文字（比如中文简体 GB2312 ），则只需配置 MiniGUI 使用 GB2312 字体，这样可节省存储空间的占用；当应用程序需要同时显示不同字符集的文字（比如日文或者韩文），则可以配置 MiniGUI 使用 Unicode 编码的字体（TrueType 字体或者 QPF 字体），而且在这种情况下，除了正确指定逻辑字体的字符集之外，应用程序不需要做更多的修改。MiniGUI 可以自动完成从某个特定字符集（如 GB2312 、GBK ）到 UNICODE 的转换，并选择正确的字体来渲染文本。这种实现方式和传统的 UNICODE 内码方式不同。传统的 UNICODE 内码方式需要将非 UNICODE 编码的文本转换为 UNICODE 编码之后才能正确显示。

9 飞漫软件围绕 MiniGUI 的软件产品

9.1 MiniGUI 增值版

MiniGUI 增值版 (MiniGUI Value-Added Release , 简称 MiniGUI-VAR) 是飞漫软件为嵌入式设备开发商提供的 MiniGUI 增值产品。如果您要基于某种嵌入式操作系统开发商用产品, 并且希望采用已成为工业事实标准的 MiniGUI 作为嵌入式图形界面支持系统, 那么 MiniGUI 增值版无疑是您最好的选择。

MiniGUI 增值版产品目前有两个版本: MiniGUI-VAR V1.6.x 及 MiniGUI-VAR V2.0.x 。MiniGUI-VAR V1.6.x 基于 MiniGUI V1.6.x 版本, 提供了针对 Linux/uClinux 、 VxWorks 、 pSOS 、 eCos 、 ThreadX 、 uC/OS-II 、 Nucleus 和 OSE 等操作系统的支持。用户还可以选择 MiniGUI SDK for Windows 可选组件, 在 Win32 平台上利用 Visual Studio 开发环境开发 MiniGUI 应用程序。

MiniGUI-VAR V2.0.x 基于 MiniGUI V2.0.x , 主要提供对 Linux 等具有多进程特性的嵌入式操作系统的支持。飞漫软件正在开发针对 VxWorks 6 的 MiniGUI-VAR V2.0.x 产品, 预计将于 2006 年第四季度发布。

关于 MiniGUI 增值版的产品信息, 可访问如下网页:

<http://www.minigui.com/product/cindex.html>

9.2 MiniGUI 组件产品

为了适应各种嵌入式设备的不同需求, 飞漫软件围绕 MiniGUI 开发了许多组件产品。用户使用这些组件产品, 可以扩展 MiniGUI 的功能, 并可以和已有的 MiniGUI 应用程序良好集成。图 9.1 给出了这些组件产品和 MiniGUI 以及应用程序之间的关系。

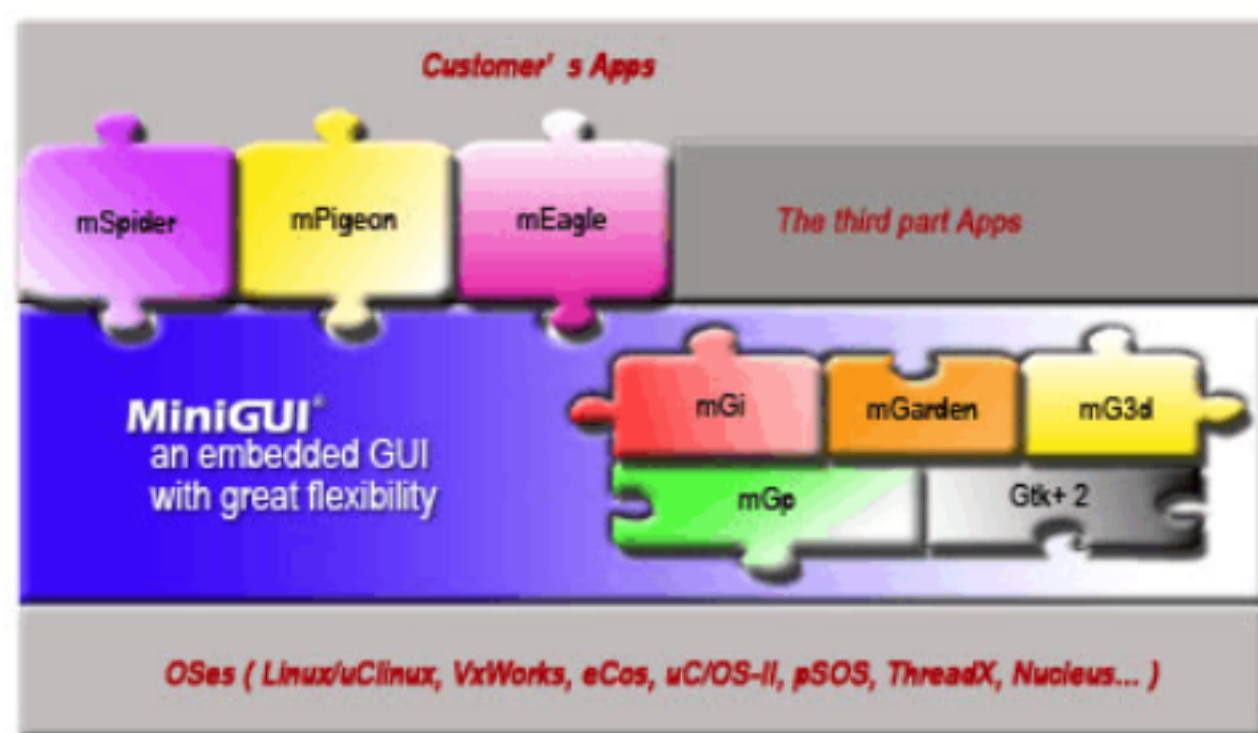


图 9.1 飞漫软件围绕 MiniGUI 的产品线

9.2.1 mGp

mGp 是飞漫软件针对 MiniGUI 应用程序的一个打印组件, 该组件使用户的 MiniGUI 程序具有打

印输出功能，可以将 MiniGUI 程序中的位图或文字输出到打印机去。mGp 现已提供对爱普生和惠普等多种打印机的支持。图 9.2 是 mGp 组件的打印设置对话框。

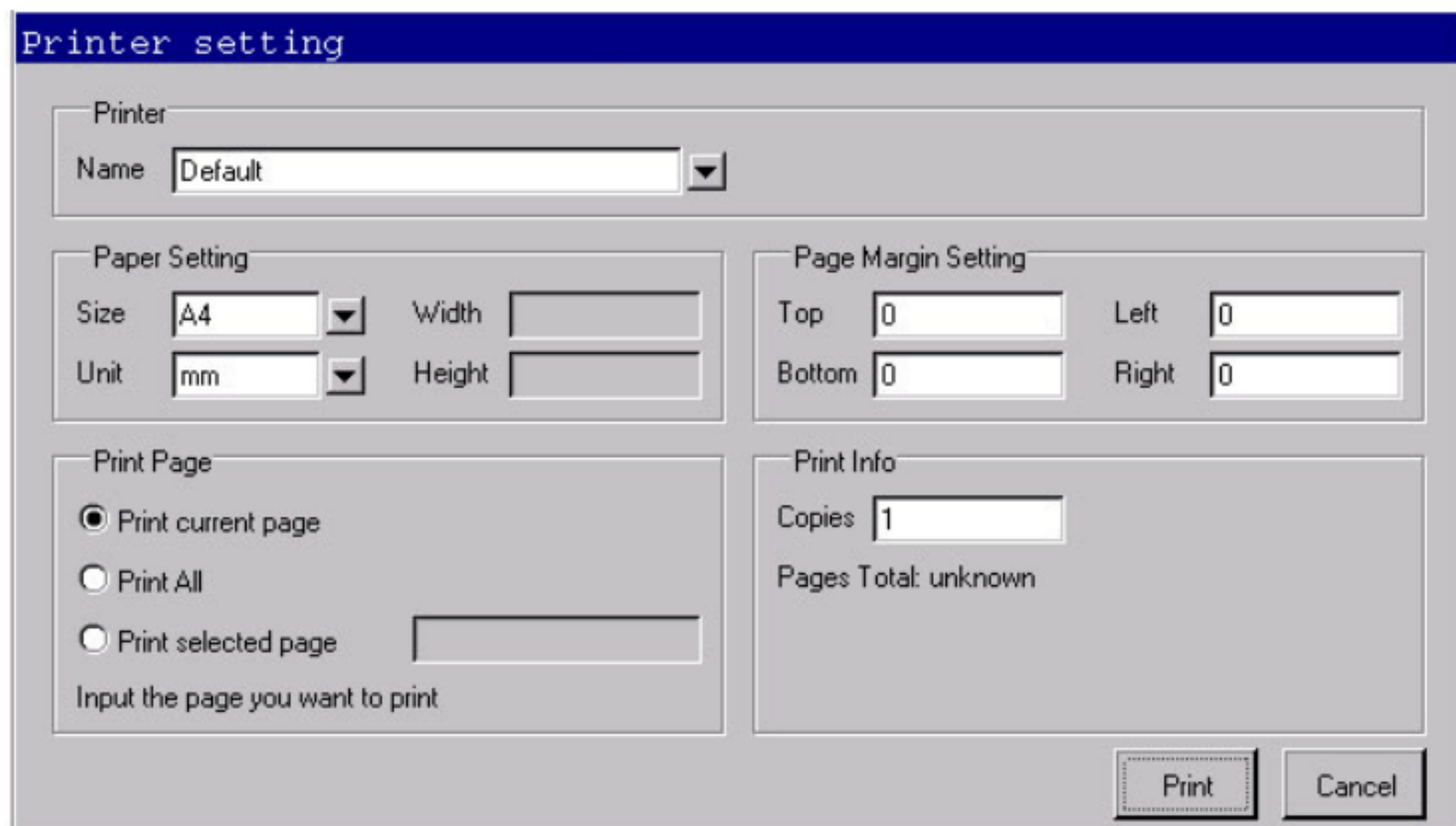


图 9.2 飞漫软件 mGp 打印组件的设置窗口

9.2.2 mGi

mGi 是飞漫软件提供的一个输入法组件，该组件目前提供了软键盘输入法和手写输入法框架，并提供给用户管理输入法的容器，通过这个容器，用户还可以添加自定义的输入法。此外，对于软键盘输入法，用户可以自定义显示的键盘位图，并可添加不同的输入翻译方式（自带中文全拼输入法）。图 9.3 和图 9.4 是利用 mGi 组件开发的软键盘及手写输入法界面。



图 9.3 mGi 输入法组件中的软键盘输入法

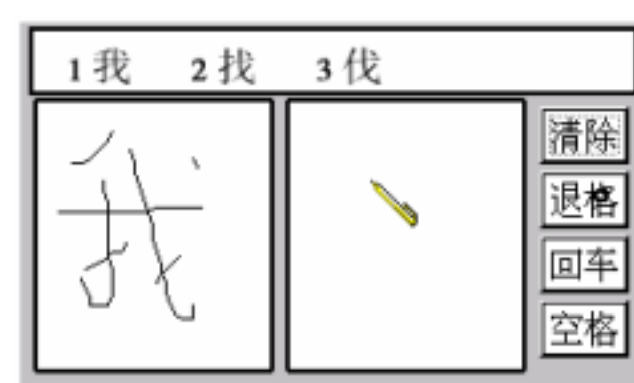


图 9.4 mGi 输入法组件中的手写输入法窗口

9.2.3 mG3d

mG3d 是一个为 MiniGUI 的应用程序提供 3D 接口的组件，通过这些接口，用户可以给自己的应用程序添加三维图像，文字渲染、场景渲染等效果。图 9.5 是 mG3d 组件的运行效果图。

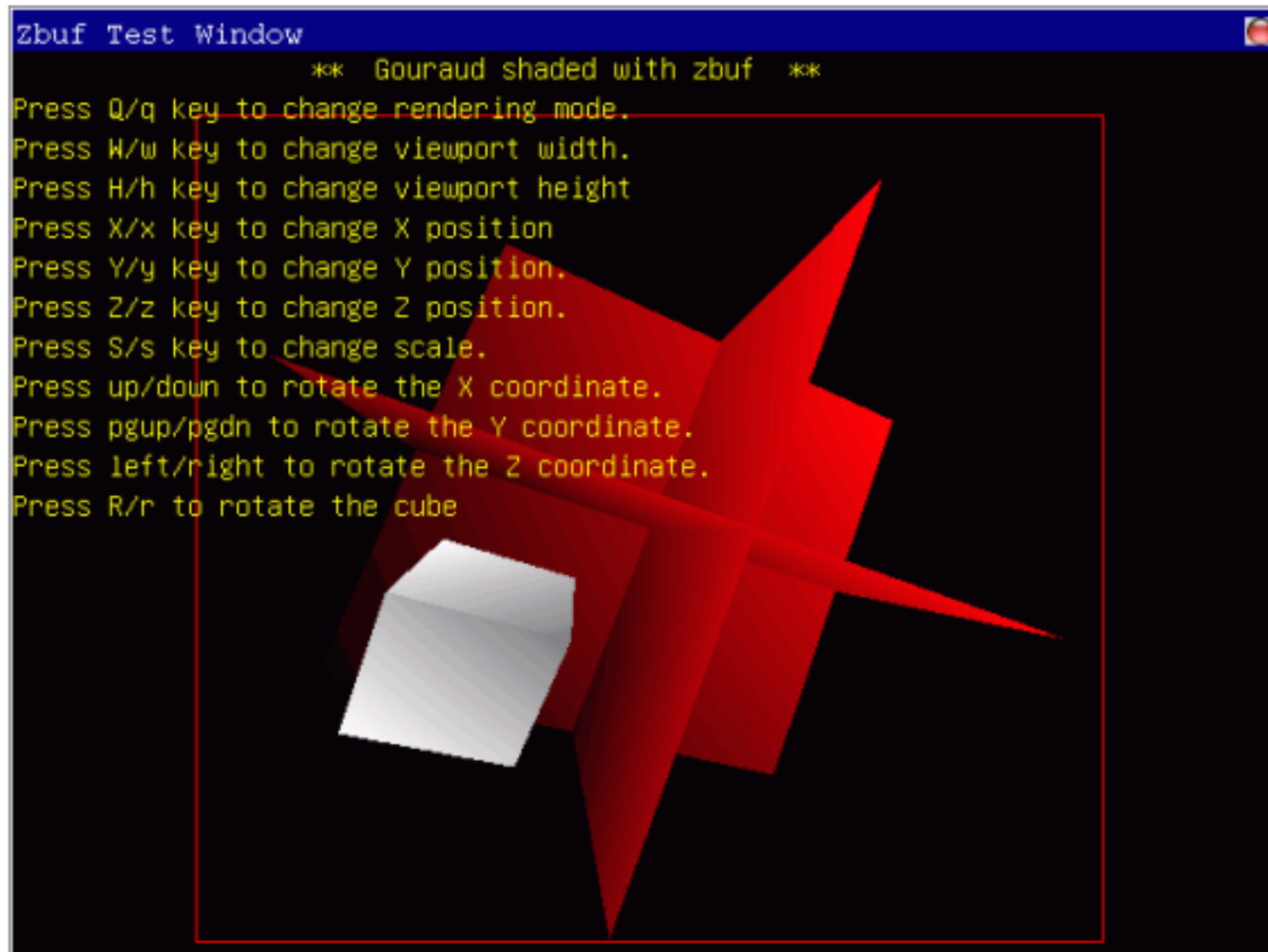


图 9.5 mG3d 组件的运行效果图

9.3 飞漫嵌入式浏览器：mSpider

mSpider 是飞漫软件基于 MiniGUI 开发的轻量级嵌入式浏览器。作为一个适用于嵌入式设备的浏览器，mSpider 在嵌入式网络设备中具有相当重要的意义，不仅可以浏览网页、电子购物，还可作为嵌入式系统的用户控制界面。mSpider 目前已经移植到了很多开发板上，包括 EM85xx、EM86xx、TI DM270、AU1200 等。

图 9.6 是 mSpider 的运行效果图。

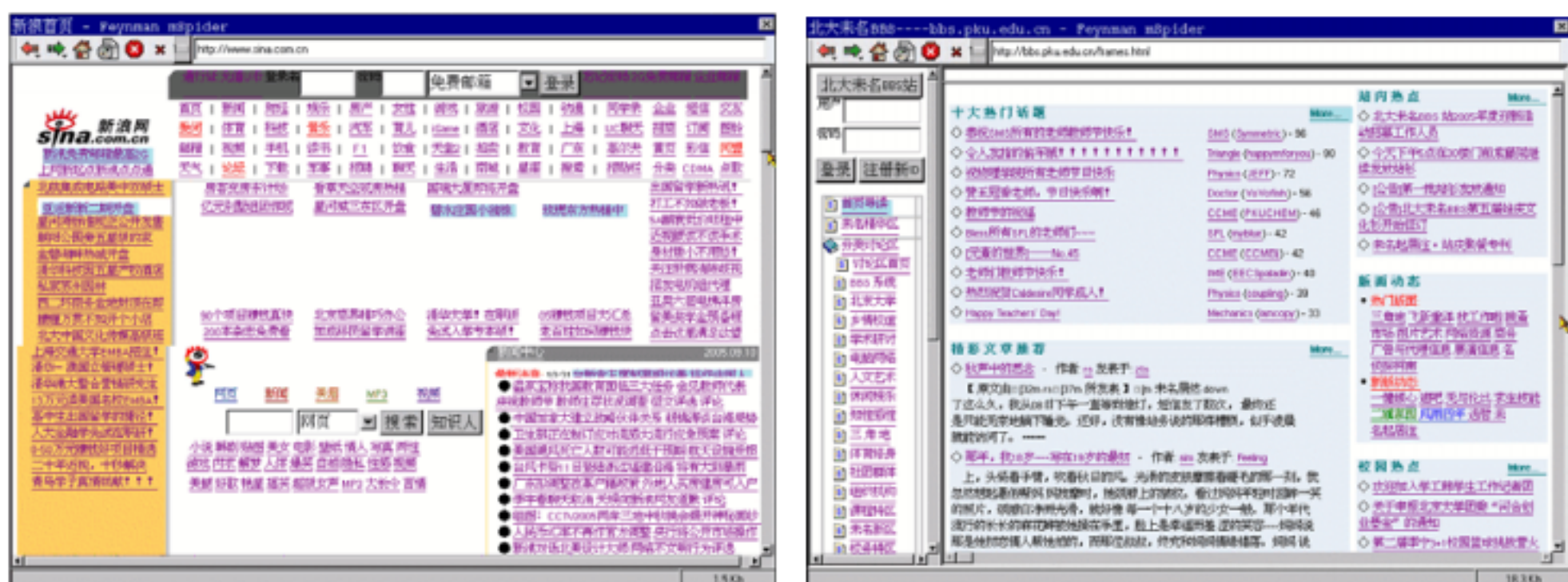


图 9.6 mSpider 的运行效果图（访问新浪和北大未名 BBS）

mSpider 的主要技术特性和优点：

各种网页规范和标准的兼容。

mSpider 提供对大部分 HTML 4.01 标记及属性、Cookie、Frame/iFrame、SSL 证书认证、常见 Javascript 1.5 对象的支持。

体积小、浏览速度快。

mSpider 是一个体积非常小的浏览器。最终生成的可执行程序大概为 2MB，运行时的系统内存只要 4MB 就可以了。

多种图片格式的支持。

目前 mSpider 支持网页通用的多种图片格式，如 PNG、JPEG、BMP 以及 GIF 动画；良好的文本渲染能力。

由于 mSpider 是基于 MiniGUI 库开发的，充分利用了 MiniGUI 良好的多语种、多字体支持特性，因此可以提供非常强大的文本渲染能力。

多语种的支持。

基于 MiniGUI 对多字符集的强大支持，mSpider 可以提供从中、日、韩文字到大多数单字节字符集的支持。

跨操作系统。

mSpider 继承了 MiniGUI 跨操作系统的特性，具有卓越的可移植性，目前已经成功移植到了 Linux、uClinux、VxWorks 等操作系统上。

界面可定制。

mSpider V2.0 提供了界面的定制功能。用户可以自行开发浏览器的人机交互界面。

10 MiniGUI 相关资源

10.1 开源版本及开发包

目前，MiniGUI 的开源版本包括 MiniGUI V1.3.3 及 MiniGUI-STR V1.6.2，您可访问如下网页下载相关源代码包：

<http://www.minigui.com/download/cindex.shtml>

如果您希望体验 MiniGUI V2.0.x 的完整特性，也可访问如下网页获得对应的开发包。将开发包安装到运行 Linux 的 PC 机上，您就可以开发 MiniGUI V2.0.x 应用程序了。MiniGUI 1.6.x 版本的开发包也可从上述网页下载获得。

您还可以方便地获得 MiniGUI 的完整 API 参考手册：

http://www.minigui.com/api_ref/index.html

如果您在使用 MiniGUI 进行开发的过程中遇到困难，请访问我们的技术论坛，提问、解答或者发表您的高见：

<http://www.minigui.org/cgi-bin/lb5000/leoboard.cgi>

10.2 飞漫软件发布的其他开源软件

您可以从飞漫软件网站获得由飞漫软件发布的其他开源软件，所以软件均可以在 MiniGUI 上运行：

mGIS 1.0：由飞漫软件基于 MiniGUI 开发的地理信息系统（GPL）。

eDillo 0.4.0：基于 Dillo 0.8.4 开发的嵌入式浏览器（GPL）。

Monqueror 0.6：基于 Konqueror 开发的嵌入式浏览器（GPL）。

MGXine 0.9.12：基于 Xine 开发的媒体播放器（GPL/LGPL）。

FlashPlayer 0.2：Flash 播放器（GPL）。

上述开源软件的下载地址是：

<http://www.minigui.com/download/cmgothor.shtml>

10.3 MiniGUI 演示程序

您可以从飞漫软件网站下载得到 MiniGUI 的演示程序，下载地址是：

<http://www.minigui.com/download/cmgedemo.shtml>

其中包括有 Linux PC 平台示例程序、Windows 平台示例程序、VxWorks 平台示例程序以及各种流行开发板上的示例程序。关于演示程序的使用，请参阅该网页中提供的 README 文件。

11 MiniGUI GPL 版本的授权策略

飞漫遵循 GPL 条款发布 MiniGUI 的某些版本，因此，任何与 MiniGUI 动态或静态链接的程序，均适用 GPL 条款。如果您基于 MiniGUI 的程序因为各种原因而无法或者不愿遵循 GPL 条款，则应向 MiniGUI 的版权拥有人（飞漫软件）购买商业授权。

11.1 如果您 100% 遵循 GPL，则无需获得商业授权

如果您使用 MiniGUI 的应用程序以 GPL 发布，则无需获得我们的商业授权。我们非常欢迎任何人在遵循 GPL 条款的基础上复制、修改和发布 MiniGUI。在这种情况下，您无需获得飞漫软件任何形式的（包括口头或书面）使用授权，因为 GPL 条款本身已充分确保您的权益。但需要注意的是，飞漫软件不对这种形式下的使用提供任何形式的担保和技术支持。

11.2 如果您从不复制、修改和发布 MiniGUI，则无需获得商业授权

只要您从不复制、修改和发布 MiniGUI，则您可以在您的应用程序中使用 MiniGUI，而无需获得商业授权。譬如，您在完成一篇学位论文，并在您的程序中使用了 MiniGUI，您的程序仅仅用来说明您论文中试验的可行性或者结果，而且您没有修改 MiniGUI，并且该程序不以任何方式被复制和发布，则您无需获得商业授权。飞漫软件也不对这种形式下的使用提供任何担保。

但需要特别指出：

修改。我们欢迎您对 MiniGUI 进行任意的修改。如果你发布该修改版本，则您对 MiniGUI 所做的任何修改、所有的接口代码以及直接和间接地与接口相关联的代码将遵循 GPL 许可证。复制。我们允许您复制 MiniGUI 二进制代码和 / 或源代码，但如果您发生此行为，所有的副本应遵循 GPL 许可证。

11.3 其他情况均需获得商业授权

如果您使用 MiniGUI 的应用程序但不希望以 GPL 条款发布，并打算在内部或外部发布使用 MiniGUI 的应用程序或者函数库，则您必须首先获得飞漫软件的商业授权。

特别是：

您的非 GPL 应用程序连接了 MiniGUI，不管静态还是动态连接，您需要为每一个 MiniGUI 函数库副本购买商业授权。

如果您在自己的单位使用 MiniGUI 函数库，但又不希望将其置于 GPL 许可证之下，则需要购买商业授权。

当然，更多的人购买 MiniGUI 的商业授权，其目的非常简单，他们希望获得来自飞漫软件的技术支持和软件质量担保。

有关 MiniGUI 授权策略的详细信息，请访问

<http://www.minigui.com/product/licensing.shtml>

商业授权的方式、价格及购买方法，请访问

<http://www.minigui.com/product/cindex.shtml>

对 MiniGUI 商业应用软件开发商，我们提供 MiniGUI 增值版产品，该产品适用于产品开发阶段，以支持 MiniGUI 之上的非 GPL 应用软件开发。

12 联系我们

如果您对 MiniGUI 及飞漫软件感兴趣，可参考如下站点获得详细信息：

<http://www.minigui.com>

您可以访问下面的网页获得飞漫软件的完整软件产品信息：

<http://www.minigui.com/product/cindex.html>

有关飞漫软件产品技术咨询、产品购买等问题，请致信：

consult@minigui.com
sales@minigui.com