MiniGUI 技 术 白 皮 书

版本 3.0（修订号 2），适用 MiniGUI 版本 3.0

最新改动日期：2017/11/07

目　　录

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 已成为跨操作系统跨硬件平台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，已验证的硬件包括 ix86、ARM、PowerPC、MIPS、DragonBall、ColdFire 等等。MiniGUI V2.0 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。最新的 MiniGUI V3.0 则是继 2.0 之后的一个重要增强，增加了如外观渲染器技术、双向文本支持、透明控件、独立滚动条控件、UPF 字体和位图字体等新的特性，并新增两个组件 mGUtils、mGPlus、mGNCS 和 mGEff。

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商的认可，并远销到韩国、日本、台湾、马来西亚、北美、欧洲等地区；MiniGUI 业已成为嵌入式图形中间件领域的工业事实标准。值得一提的是，在中国自主开发的 3G 通讯标准 TD-SCDMA 中，约有 60% 获得入网许可证的 TD-SCDMA 手机使用 MiniGUI 作为其嵌入式图形平台，以支撑浏览器、可视电话等 3G 应用软件的运行；其中有海信 T68、中兴通讯 U85 等 TD-SCDMA 手机型号。

飞漫软件除了遵循 GPL 条款发布 MiniGUI 的某些版本之外，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 V3.0 版本的技术特点以及应用领域。

1.2 MiniGUI 的起源和发展

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 V1.2.6 版本；于 2003 年 9 月发布了 MiniGUI V1.3.0 版本。

[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。

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。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；2004 年 11 月，发布了针对嵌入式 GIS 应用的 MGIS 嵌入式地理信息系统软件。

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 3 月，MiniGUI 定制版产品(MiniGUI-CMR)和 MiniGUI 标准版产品(MiniGUI-STD)隆重面世；2005 年 7 月，基于 VxWorks Simulator 的 MiniGUI 演示开发环境顺利完成，该环境的成功搭建进一步加深了飞漫软件与美国风河间的合作；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；2005 年 9 月，正式发布全功能嵌入式浏览器 mSpider；2005 年 9 月，公司遵循 GNU GPL 授权条款，将自主开发的嵌入式浏览器 edillo 以自由软件的形式发布。

2006 年 1 月，正式发布 MiniGUI 的相关组件产品 mGp V1.0 、 mGi V1.0 以及 mG3d V1.0；2006 年 3 月，四川长虹电器股份有限公司采用 MiniGUI 和嵌入式浏览器 mSpider 进行 DTV 和 IPTV 产品的开发，进一步确定了飞漫软件产品在数字电视 、机顶盒的软件解决方案的领导地位；2006 年 5 月，公司成为美国风河系统公司全球合作伙伴，为 VxWorks 提供全面的图形解决方案。2006 年 6 月，飞漫软件与美国 AMD 达成战略合作伙伴关系，AMD 公司在其新发布的 Argon PMP 参考设计方案中采用了飞漫软件提供的 MiniGUI 和 Fhas 作为图形环境。

2006 年 6 月，飞漫软件把 MiniGUI 嵌入式图形系统与 Enea 的 OSE 实时操作系统（RTOS）集成到一起。MiniGUI 可以极大地简化运行 OSE 的手持终端、电信设备、医疗设备、机顶盒及工业控制系统上的体积小、性能高的图形用户界面的设计和开发；2006 年 8 月，国际领先的半导体供应商 Atmel 公司与飞漫软件合作，使用 MiniGUI 为其多款应用于 WiFi 领域的芯片开发 MMI 软件参考设计，进一步加强了飞漫软件在 WiFi 领域的全球领先地位；2006 年 11 月，台湾英华达电子技术有限公司选用 MiniGUI 开发出了支持 skype 功能的可视 IP 电话，奠定了飞漫软件在数字媒体领域的高端地位。2006 年 12 月，香港世界电信展上，大唐移动通信隆重推出采用 MiniGUI 和 Fhas 应用开发平台作为图形显示系统和终端应用开发平台的 TD-SCDMA 3G 手机终端软件标准平台 Arena。这标志着 MiniGUI 业已成为 TD-SCDMA 标准平台的手机事实标准。

2007 年 3 月，飞漫软件公司正式发布 mGDesktop V3.0 版本；2007 年 7 月，飞漫软件推出全功能高端嵌入式浏览器 mDolphin V1.0。

2008 年 2 月，飞漫软件发布 mGDesktop V4.0 版本；2008 年 6 月，飞漫软件发布 mDolphin V2.0。

2010 年 10 月，飞漫软件发布了 MiniGUI v3.0.12 以及 miniStudio V1.0.8。

2017 年 8 月，飞漫软件在 GitHub 上遵循 GPL 2.0/3.0 许可证发布了 MiniGUI V3.0 及其组件的最新源代码，同时遵循 Apache 2.0 许可证发布了 mDolphin V3.0。

随着物联网的兴起，2017 年 10 月，飞漫软件重启了 MiniGUI 的开发。

[2] 其中也包括飞漫软件的另一个产品 Fhas。

目前，MiniGUI V3.0 版本支持 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。

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：

功能手机、WiFi 手机、PDA、智能电话类产品

图 1.1 MiniGUI 典型应用：智能手持设备

图 1.1 左图是中国通信领域的领军企业大唐移动通信推出的 TD-SCDMA 3G 后机终端软件标准平台 Arena，Arena 采用 ThreadX 作为实时操作系统，采用 MiniGUI 和 Fhas 应用开发平台作为图形显示系统和终端应用开发平台；右图是台湾英华达电子技术有限公司选用 MiniGUI 开发出了支持 skype 功能的可视 IP 电话。

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

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。

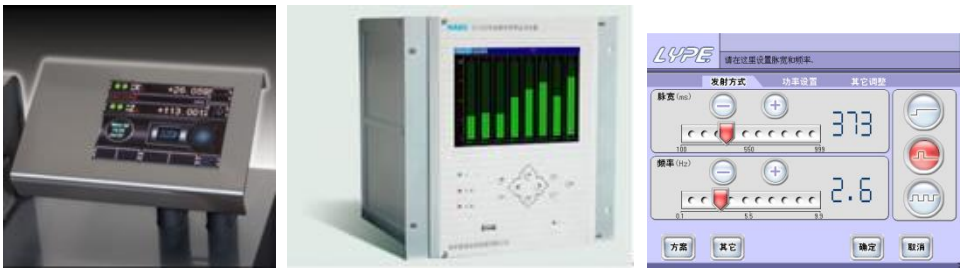


图 1.3 MiniGUI 典型应用：工业仪表及控制系统

2 MiniGUI 的优势

2.1 MiniGUI 的技术特点

MiniGUI 为嵌入式 Linux 系统提供了完整的图形系统支持，是全球针对嵌入式 Linux 仅有的两个商用嵌入式 GUI 系统之一。MiniGUI 为嵌入式 Linux 系统提供了完整的多进程支持；可以 MiniGUI-Processes、MiniGUI-Threads 或者 MiniGUI-Standalone 三种运行模式运行。

MiniGUI 的主要技术特性描述如下：

硬件适配性

- ✓ 可运行于各种含有 MMU（内存管理单元）的 32 位处理器架构之上，如 ix386、ARM、MIPS、PowerPC 等。
- ✓ 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。对显示设备分辨率无最大和最小限制。
- ✓ 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。
- ✓ 可支持各种输入设备，如 PC 键盘、PC 鼠标、小键盘（Keypad）、触摸屏、遥控器等等。
- ✓ 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。

资源消耗

- ✓ MiniGUI 的静态存储随配置选项的不同而不同，最少需占用 1MB 静态存储空间。
- ✓ MiniGUI 启动后，初始占用 1MB 动态存储空间。建议系统内存为 8MB 以上。

操作系统适配性

- ✓ 支持 Linux 操作系统（非 uClinux 操作系统），可以 MiniGUI-Processes、MiniGUI-Threads 或者 MiniGUI-Standalone 三种运行模式运行。

- ✓ 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，并且非常适合无文件系统支持的实时嵌入式操作系统。
- ✓ 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。

窗口子系统特性

- ✓ 完备的多窗口机制和消息传递机制。使用 MiniGUI-Threads 运行模式时，可在不同线程中创建主窗口，并支持线程间的消息传递；使用 MiniGUI-Processes 运行模式时，支持完整的多进程窗口系统。
- ✓ 对话框和消息框支持。
- ✓ 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。
- ✓ 其它 GUI 元素，包括菜单、加速键、插入符、定时器等。

图形子系统特性

- ✓ 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。
- ✓ 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）。
- ✓ Windows 的资源文件支持，如位图、图标、光标等。
- ✓ 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、UTF-16 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 矢量字体（对 TrueType 的支持通过 freetype 1.3 函数库提供）。
- ✓ 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。

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：

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。

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。

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，您可以访问飞漫公司网站的典型案例部分：

```
http://www.minigui.com/
```

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 要运行的硬件平台。
- ✓ 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。
- ✓ 指定控件和窗口的整体风格，可以通过指定不同的渲染器完成。

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。

2.3 MiniGUI 的新特性

MiniGUI V3.0 在以前版本的基础上新增了如下新特性：

主窗口双缓冲区（Double Buffering Main Window）

当 MiniGUI 3.0 的主窗口具有双缓冲区时，我们可以在自定义缓冲区中获得整个主窗口的渲染结果。在此基础上，我们可以利用高级 2D 图形接口或者 3D 图形接口获得主窗口的各种特殊显示效果，如推拉切换、翻页切换、卷曲效果等等。

外观渲染器（Look and Feel Renderer）支持

MiniGUI V3.0 改变了以往只支持三种控件风格的方式，引入了渲染器(Look and Feel)这一全新的模式。渲染器是定义如何绘制窗口元素的渲染器，是在 MiniGUI V2.0.X 的基础上继续完善的。窗口元素包括边框、标题栏、标题栏按钮、滚动条、选定项目、无效项目、高亮项目、突出项目、三维对象等；窗口元素的外观属性，包括窗口元素的颜色、尺寸、字体等信息；窗口元素渲染器是对窗口元素进行定制大小、颜色、图形、字体，便于用户设计个性化的外观显示风格。用户可以指定某个主窗口或某个控件使用特定的渲染器，也可定制非客户区渲染器、窗口元素的尺寸、颜色、字体、图标，同时增强资源管理功能，从而获得更加华丽的图形界面。MiniGUI 实现了几种默认整体显示风格：Classic 、Flat、Fashion 和 Skin。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。

- ✓ Classic: 这种风格的界面是标准的 Window 95 风格界面，也是最广泛使用的风格了，如图 2. 1 所示；

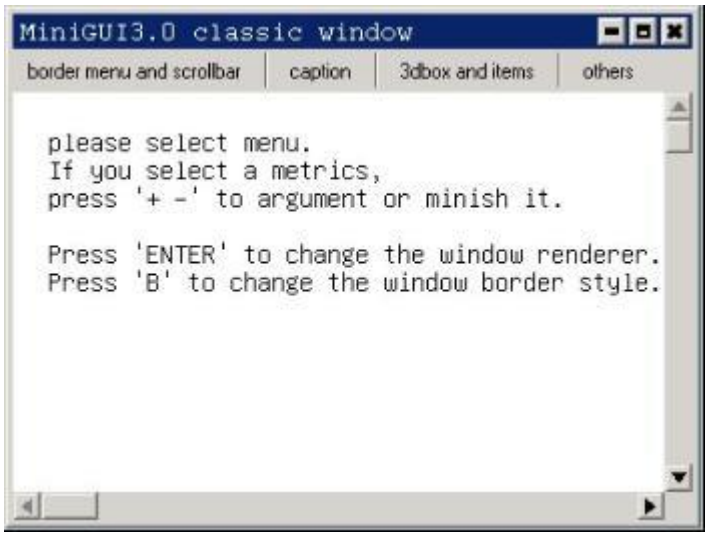


图 2.1 MiniGUI 的默认显示风格(Classic 渲染器显示风格)

- ✓ Fashion: 此种风格的界面，采用 MiniGUI 3.0 组件 mGPlus 提供的颜色渐变填充技术，因此，可获得非常炫丽的界面效果，如图 2.2 所示；

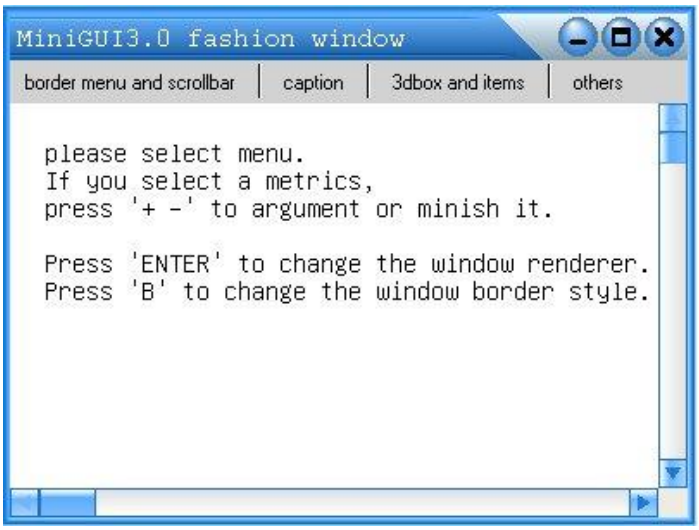


图 2.2 MiniGUI 的 Fashion 渲染器显示风格

- ✓ Flat: 此种风格的窗口界面，线条清晰，简洁，因此适用于单色或者灰度显示屏。因为绘制简单，因此该渲染器占用资源最少，运行速度最快，如图 2.3 所示；

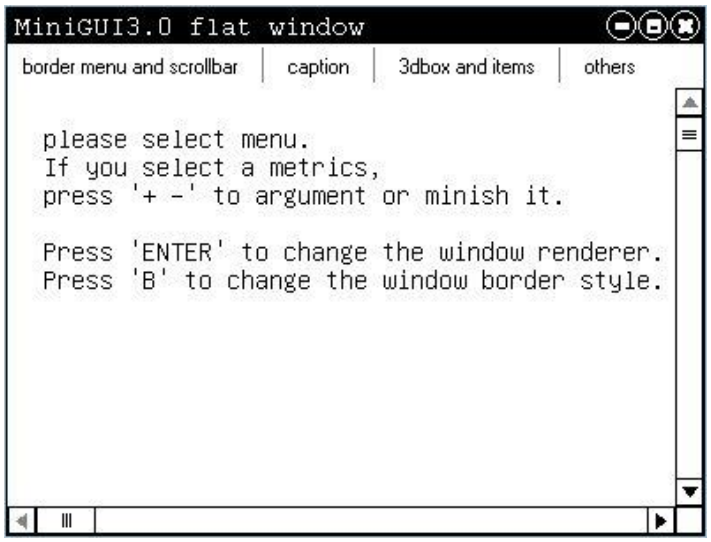


图 2.3 MiniGUI 的 Flat 渲染器显示风格

✓ Skin: 上面三种外观渲染器基本上都是由代码绘制出来的，具有小巧灵活的特点。但是，在嵌入式应用领域，设备的差别非常大。有些设备，已经具备了非常高的运算性能。在这种情况下，可以考虑使用皮肤外观渲染器来美化界面。皮肤外观渲染器需要一整套和界面相关的图片，因此需要占用一些存储资源。皮肤外观渲染器的最大的特点是允许用户定制界面，用户可以使用自己设计的图片替换系统原有的图片，展现在用户面前的就将是你自己设计的界面效果，如图 2.4 所示。

图 2.4 MiniGUI 的 Skin 渲染器显示风格

双向文本(BIDI Text)的显示与输入

我们知道，除了大家熟知的从左向右书写的文字（如英语、汉语等）之外，还有许多语言采用从右向左的书写习惯，如阿拉伯文和希伯来文等。为了支持这些语言，MiniGUI 3.0 中增加了对这

两种语言所属字符集的处理，并增加了阿拉伯和希伯来键盘布局的支持，从而实现了对双向文本的输入输出处理。阿拉伯文以及希伯来文的显示，如图 2.5 所示：

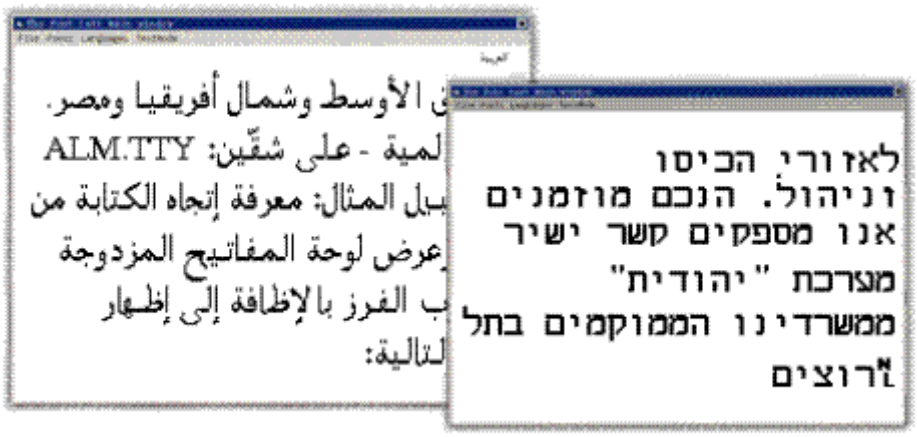


图 2.5 阿拉伯文及希伯来文的显示效果

不规则窗口

MiniGUI V3.0 实现了不规则窗口与控件，可满足用户对窗口外观各种不同的需求。不规则窗口通过一个 Region 数据结构来表示可见区域，或者通过 8 位 MYBITMAP 中的透明值形成不可见区域。非规则主窗口和非规则控件如图 2.6 所示。

图 2.6 非矩形窗口及控件

字体

在 MiniGUI 3.0 中，飞漫软件发明了一种新的 UNICODE 字体文件格式，称为“UPF”字体。这种字体的最大特点，是便于在多进程环境下使用，从而极大地节约了内存的使用。同时，飞漫

软件增强了 VBF 字体格式，将 VBF 字体升级到了 3.0，扩大了其能适用的字符集范围，以便支持阿拉伯文等语言文字的显示。

其他增强

MiniGUI 3.0 实现了桌面的可定制。通过桌面的外部编程接口，用户可以在桌面放置图标并响应桌面事件，实现类似 Windows 桌面的界面效果。除此之外，MiniGUI 3.0 还增强了透明控件的实现，使之效率更高，且不依赖于控件的内部实现代码。MiniGUI 3.0 还提供独立的滚动条控件，提供统一的虚拟帧缓冲区程序支持等等。

另外最新的 MiniGUI V3.0 新增加了两个新的组件：mGUtils 和 mGPlus，这两个组件将在第九章详细介绍。把字体、位图、图标、光标等资源进行统一管理，资源的内嵌和非内嵌方式并不影响模块的组成，由此抽象出系统资源管理模块。图 2.7 展示了可定制桌面以及透明控件和非透明控件的比较效果图。

图 2.7 可定制桌面以及透明空间和非透明控件比较效果图

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、OSE 以及 Win32 平台上。同时，在不同操作系统上的 MiniGUI，提供完全兼容的 API 接口。最新的 MiniGUI V3.0 支持的是 Linux/uClinux 操作系统。

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、PowerPC、MIPS、DragonBall、ColdFire 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 V3.0 版本资源占用情况的具体数据，可参阅《MiniGUI Data Sheet》 V3.0。

4 MiniGUI 的软件架构

4.1 MiniGUI 的软件架构

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 ISO 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 中的抽象层将特定操作系统及底层硬件的细节隐藏起来，因而上层应用程序无需关心底层的硬件平台输出和输入设备。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。

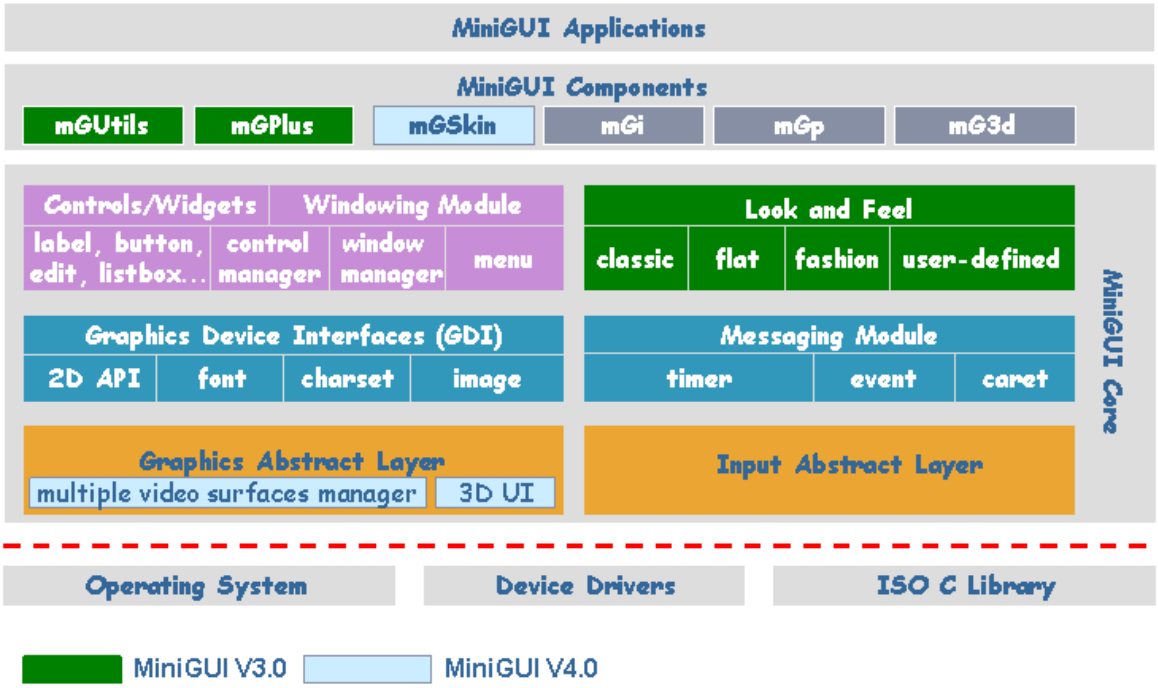


图 4.1 MiniGUI 的软件架构图

如上图所示，从底至上，MiniGUI 由如下几个模块组成：

- ✓ 图形抽象层（Graphics Abastract Layer，GAL）。图形抽象层将来自不同操作系统或设备的图形接口进行抽象，为 MiniGUI 上层提供统一的图形接口。在图形抽象层内，包含有针对 Linux FB 设备、eCos LCD 设备等的软件组成部分。这些软件组成部分通过调用底层设备的接口来实现具体的图形抽象层操作，如打开设备、设置分辨率及显示模式、关闭设备等。我们将这些用于适配图形抽象层接口的软件组成部分称为“引擎（engine）”，其概念和操作系统中的设备驱动程序类似。
- ✓ 输入抽象层（Input Abstract Layer，IAL）。和 GAL 类似，输入抽象层将 MiniGUI 涉及的所有输入设备，如键盘（keyboard）、小键盘（keypad）、鼠标（mouse）、触摸屏（touch screen）等抽象了出来，为上层提供一致的接口。要支持不同的键盘、触摸屏或者鼠标接口，则通过为 IAL 编写不同的输入引擎实现。MiniGUI 通过 IAL 及其输入引擎，提供对 Linux 控制台（键盘及鼠标）、触摸屏、遥控器、小键盘等输入设备的支持。
- ✓ 图形设备接口（Graphics Device Interfaces，GDI）。该模块基于图形抽象层为上层应用程序提供图形相关的接口，如绘制曲线、输出文本、填充矩形等等。图形设备接口中含包含其他比较独立的子模块，如字体字符集（font and charset）支持、图像（image）支持等。
- ✓ 消息处理模块（Messaging Module）。该模块在输入抽象层基础上，实现了 MiniGUI 的消息处理机制，为上层提供了完备的消息管理接口。众所周知，几乎所有的 GUI 系统本质上都是事件驱动的，系统自身的运行，以及 GUI 应用程序的运行，都依赖于消息处理模块。
- ✓ 多窗口处理模块（Windowing Module）和控件（Control 或 Widget）。基于图形设备接口和消息处理模块，MiniGUI 实现了多窗口处理模块。该模块为上层应用程序提供了创建主窗口和控件的基本接口，并负责维护控件类。控件类是用来实现控件代码重用的重要概念，利用控件类（control class），我们可以创建属于某个控件类的多个控件实例（instance），

从而让这些控件实例使用同一个控件类的代码，这样，我们就实现了类似 C++ 那样的类和实例概念，从而可以最大程度上重复利用已有代码，并提高软件的可维护性。MiniGUI 的控件模块实现了常见的 GUI 控件，如静态框、按钮、编辑框、列表框、下拉框等等。

- ✓ 外观支持（Look and Feel）。这个模块是 MiniGUI V3.0 提供给上层应用程序的接口，可用来定制 MiniGUI 窗口、控件的绘制。在 MiniGUI V3.0 之前的版本中，对主窗口和控件的定制能力，还没有被抽离出来形成独立的模块，但我们仍然可以通过配置选项让 MiniGUI 的主窗口、控件具有三种显示风格，分别是：类似 PC 的三维风格（PC3D）、平板风格（FLAT）、流行风格（FASHION）。在 MiniGUI 3.0 中，主窗口和控件的外观可完全由应用程序自行定制，在创建主窗口或者控件时，指定外观渲染器（renderer）的名称，就可以让主窗口或者控件具有各自不同的外观。

以上模块组成了 MiniGUI 的核心（core）；在 MiniGUI 接口之上，我们还为应用程序提供若干组件：mGi mGp mG3d mGUtils mGPlus，这些组件分别为应用程序提供某些特殊的功能特性。我们将在后面详细介绍这些组件。

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-Processes[3] 。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 适合于具有完整 UNIX 特性的嵌入式操作系统，比如嵌入式 Linux。该运行模式在 MiniGUI V2.0.x 中提供，在 MiniGUI V3.0 中得到进一步增强，有关该模式的详细介绍将在下面阐述。
- ✓ MiniGUI-Standalone[5]。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。

[3] 在 Linux 操作系统上运行的 MiniGUI V1.6.8 及早期版本中，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 V2.0.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。

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[4]；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。

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。

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.X 则彻底解决了上述问题。MiniGUI V2.0.X 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。MiniGUI V3.0 不但完全继承了 MiniGUI V2.0.X 的 MiniGUI-Processes 运行模式,而且用户可自定义桌面上的图标并响应桌面事件，以方便实现桌面的定制。图 4.2 给出了 MiniGUI V1.6.x 的 MiniGUI-Lite 运行模式及 MiniGUI V3.0 的 MiniGUI-Processes 运行模式在运行相同应用程序情况下的界面效果。

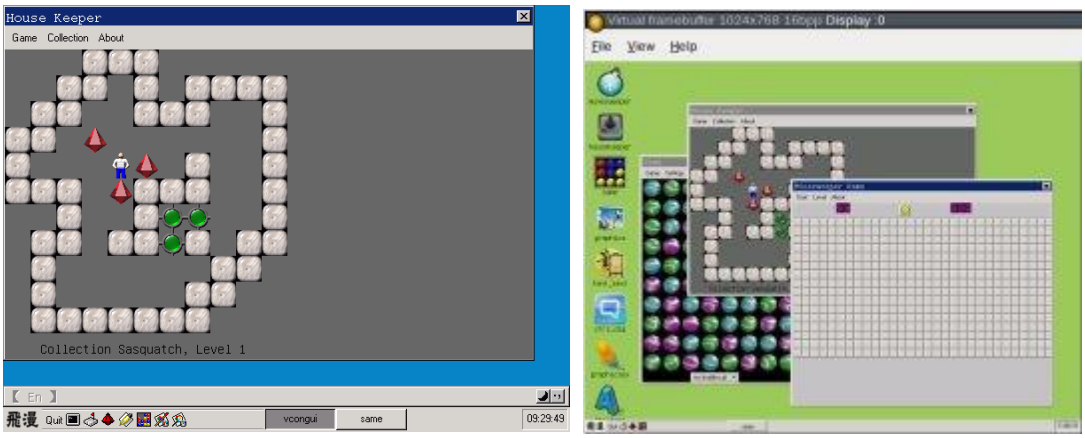


图 4.2 MiniGUI V1.6.x 的 MiniGUI-Lite 运行模式及 MiniGUI V3.0 的 MiniGUI-Processes 运行模式

[4] 目前提供对 Linux/uClinux 操作系统的支持。

图 4.2 中，第一个屏幕运行了 vcongui 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 vcongui 程序了；第二个屏幕运行 same、扫雷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 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。

各操作系统上可运行的 MiniGUI 运行模式

表 4.1 给出了 MiniGUI V3.0.x、 MiniGUI V2.0.x /1.6.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	所支持的运行模式
Linux	MiniGUI-Processes MiniGUI-Threads MiniGUI-Standalone
uClinux	MiniGUI-Threads MiniGUI-Standalone
VxWorks 6.x	MiniGUI-Threads MiniGUI-Standalone
VxWorks 5.x	MiniGUI-Threads MiniGUI-Standalone
ThreadX	MiniGUI-Threads MiniGUI-Standalone
Nucleus	MiniGUI-Threads MiniGUI-Standalone
OSE	MiniGUI-Threads MiniGUI-Standalone
eCos	MiniGUI-Threads MiniGUI-Standalone
uC/OS-II	MiniGUI-Threads MiniGUI-Standalone
pSOS	MiniGUI-Threads MiniGUI-Standalone

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 4.5 所示。

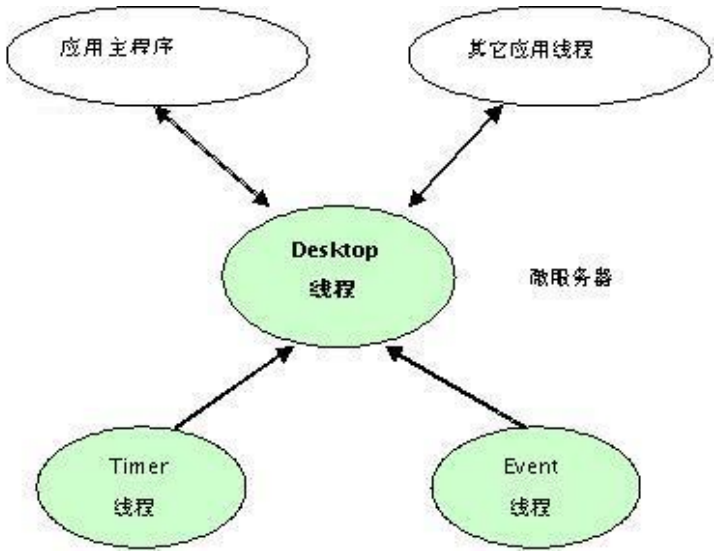


图 4.5 MiniGUI-Threads 运行模式的消息通讯机制

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 4.6 所示。

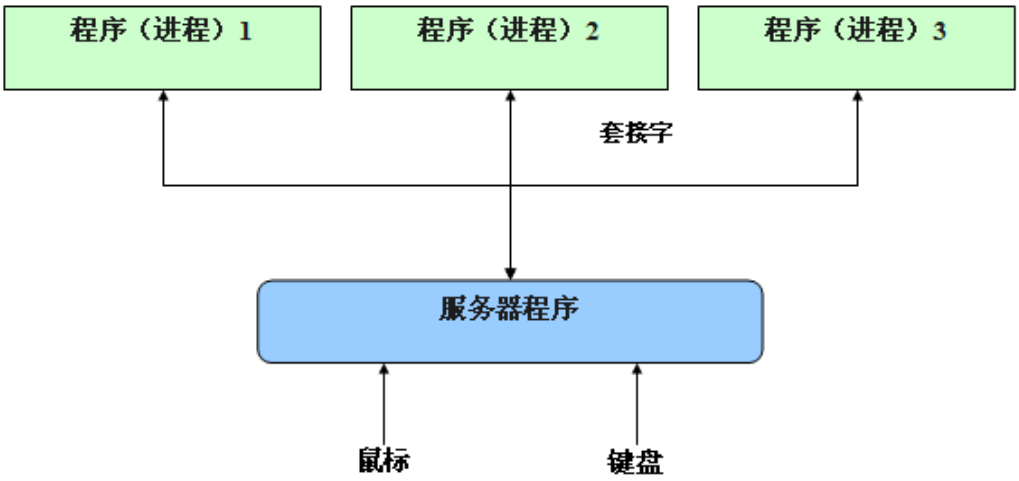


图 4.6 MiniGUI-Processes 运行模式的消息通讯机制

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、VBF 字体（这是 MiniGUI 定义的两种点阵字体格式）、TrueType、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。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。

MiniGUI V3.0 实现一种新的设备字体----位图字体，这种字体可以根据用户自己制作的位图来构建字型，可以与其它逻辑字体接口一样提供给应用程序及控件使用；VBF V3.0 字体是对 VBF V2.0 的升级版本，不再限制只能用于支持 Latin 体系的字符集，单个 VBF 字体可包含的字型数目不再限制于 255 以内；UPF 字体是 QPF 字体的优化版本，便于 MiniGUI 通过内存映射的方式使用，降低了 MiniGUI-Processes 模式下使用 QPF 字体时的内存占用。支持阿拉伯文，希伯莱文等双向文本的显示。

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，如支持常见的美式 PC 键盘，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 输入引擎

以前使用的图形引擎 qvfb、wvfb 给 MiniGUI 的开发带来了一些便利，但毕竟作为独立的应用程序存在，导致了一些不方便，如虚拟帧缓冲区程序需要单独启动，且需要用户自行设定显示模式；在 MiniGUI-Threads 和 MiniGUI-Standalone 模式下，只能启动一个 MiniGUI 应用程序。理论上，

应该是可以让不同的程序运行在不同的虚拟帧缓冲区程序中。为此最新 MiniGUI V3.0 解决了这个问题，统一了虚拟缓冲区图形引擎称为 xvfb，实现了 xvfb 将虚拟帧缓冲区中的数据刷新到窗口中的程序，使用户执行程序时更加简单易用。

5 开发环境

5.1 MiniGUI 目前的开发方式：

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。MiniGUI V3.0 推出的 xvfb 兼容 MiniGUI V2.0.x 以前的 qvfb，wvfb。

5.2 MiniGUI 集成开发环境 miniStudio:

miniStudio 是一个基于 Eclipse 开发的 MiniGUI 应用开发的集成开发环境（IDE）。开发人员可以利用 miniStudio 快速定制 MiniGUI，并设计 MiniGUI 应用程序的界面，还可进行资源的管理并调试 MiniGUI 应用程序。miniStudio 为用户提供以下图形用户界面应用程序的开发利器：

- ✓ MiniGUI 定制工具：基于 MiniGUI 3.0 板级支持包配置生成最终的 MiniGUI 函数库。
- ✓ 所见即所得的界面设计器：可以帮助用户快速的布置界面，并且所见即所得。
- ✓ 字体设计器：能够对现有的字体进行修改，增删。以使字体文件满足用户的需求。
- ✓ 位图管理器：提供位图管理功能。
- ✓ 文本管理器：统一进行文本管理。

当前，您可以从飞漫软件官网（http://www.fmsoft.cn）免费下载 miniStudio v1.0.8。

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!”。在此示例程序中，使用了渲染器来控制窗口的外观。如图 6.1.a, 6.1.b, 6.1.c 分别显示了 classic, flat, skin 三种渲染器所渲染的窗口外观。

```
#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;
    const char* old_renderer;
```

```
#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;

    old_renderer =SetDefaultWindowElementRenderer("classic");

    hMainWnd = CreateMainWindow (&CreateInfo);

    if (hMainWnd == HWND_INVALID)
        return -1;
    ShowWindow(hMainWnd, SW_SHOWNORMAL);

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

    return 0;
}
```

图 6.1.a classic 风格的窗口外观

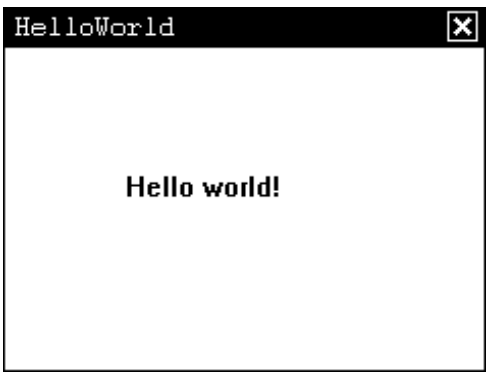


图 6.1.b flat 风格的窗口外观

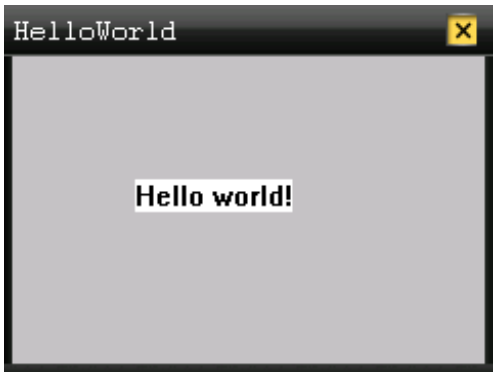


图 6.1.c skin 风格的窗口外观

要实现以上不同风格的窗口界面，只要调用函数 SetDefaultWindowElementRenderer，就可以实现了。具体对应的代码已经在上面的示例代码中用红颜色标识出来了。下面列出对应每种风格的渲染器函数的调用方法。

classic 渲染器的调用：

```
old_renderer =SetDefaultWindowElementRenderer("classic");
```

flat 渲染器的调用：

```
old_renderer =SetDefaultWindowElementRenderer("flat");
```

skin 渲染器的调用：

```
old_renderer =SetDefaultWindowElementRenderer("skin");
```

根据上面的方法，应用程序开发人员可以很方便的选择自己喜欢的窗口风格。同时，MiniGUI 还提供了各种丰富的控件，如按钮，工具栏等。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。下面继续介绍 MiniGUI 所提供的控件，为了页面的简洁，在显示控件的效果图时，都只显示 classic 风格下的控件外观效果。如果用户需要更多的控件效果，可以和本公司联系。

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。

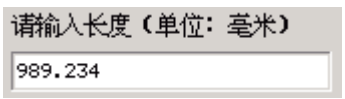


图 6.2 MiniGUI 的静态框控件（用于其他控件的标签）

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。

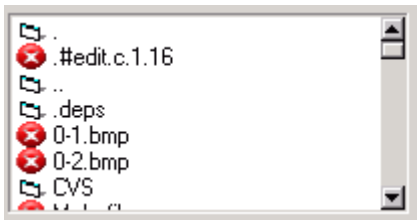
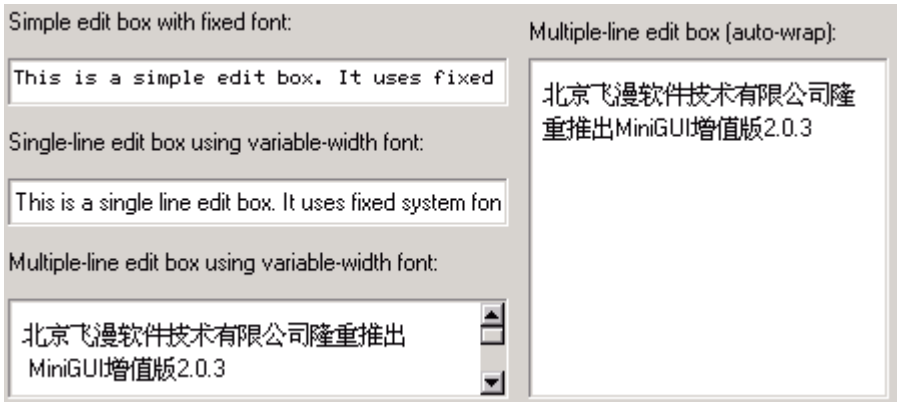


图 6.4 列表框控件

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）。最新版本 MiniGUI V3.0 当中新增了双向文本的编辑框(BIDISELEDIT)，主要为了完成双向文本的输入。如图 6.5 所示。



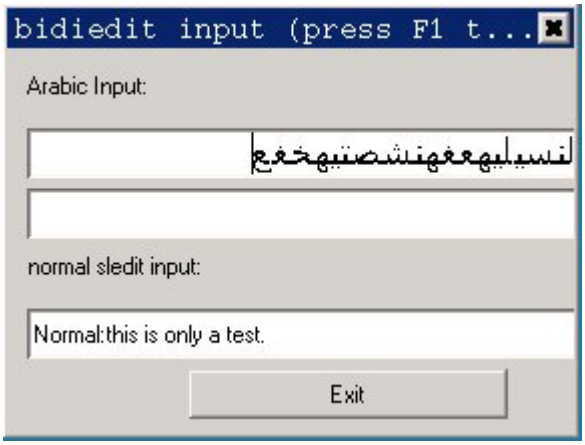


图 6.5　编辑框控件

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四种类型：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。

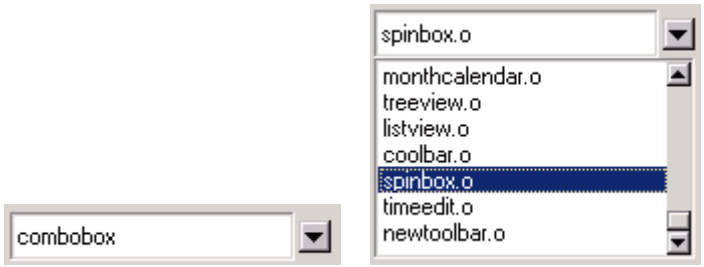


图 6.6　下拉式组合框

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。

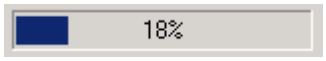


图 6.9　水平进度条

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。

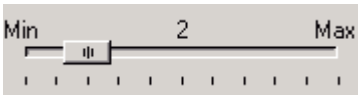


图 6.11　滑块控件

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）、NEWTOOLBAR（推荐使用）以及 MiniGUIExt 库中的 COOLBAR 控件类。NEWTOOLBAR 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。

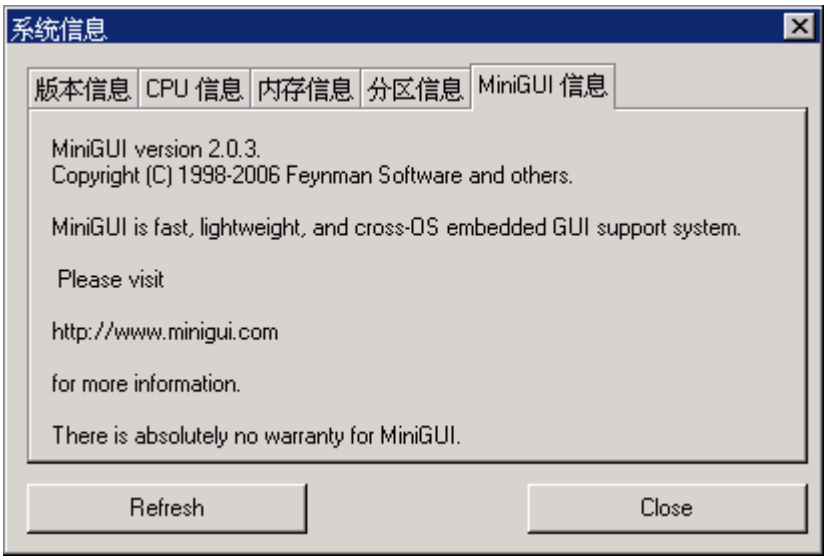
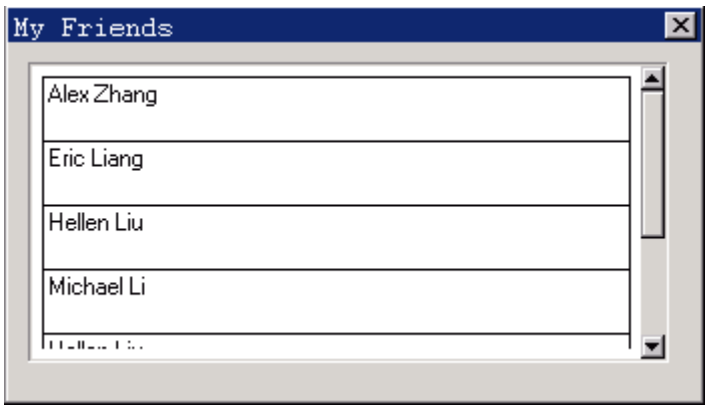


图 6.13 属性页控件

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。在最新的 MiniGUI V3.0 当中我们新增了滚动条控件(ScrollBar control)，使用不同的渲染器时，滚动条控件将会呈现不同的外观，使窗口更加美观、协调。如图 6.14 所示。



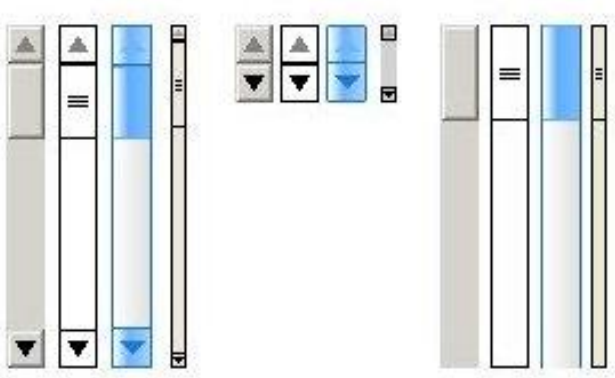


图 6.14　滚动型控件

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。

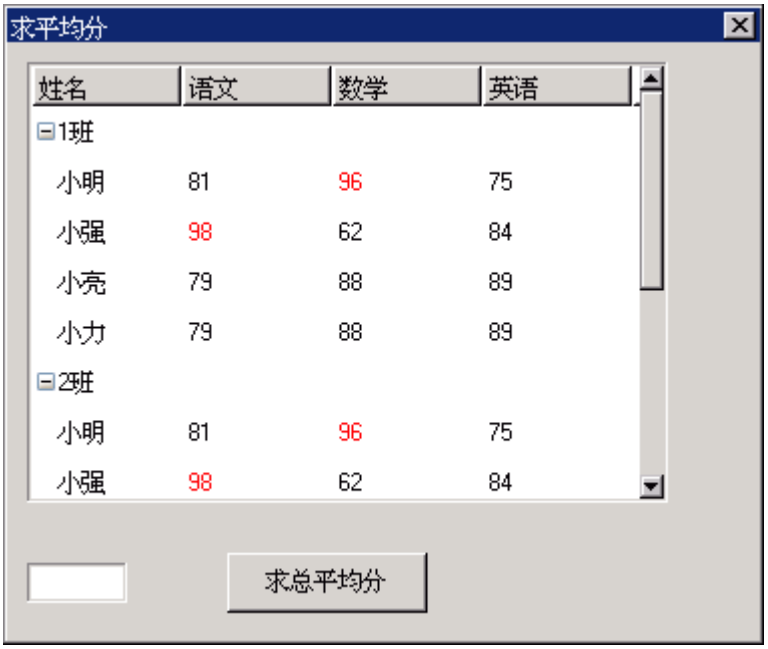


图 6.16 列表型控件的使用

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， 图上显示了一个 GIF 动画的两帧图像。

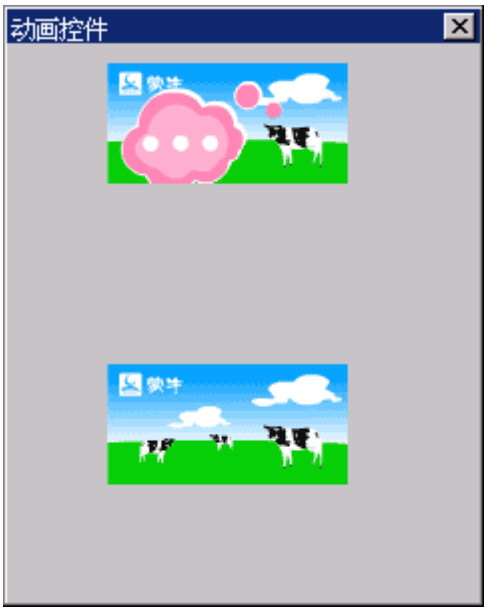


图 6.18 动画控件

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。

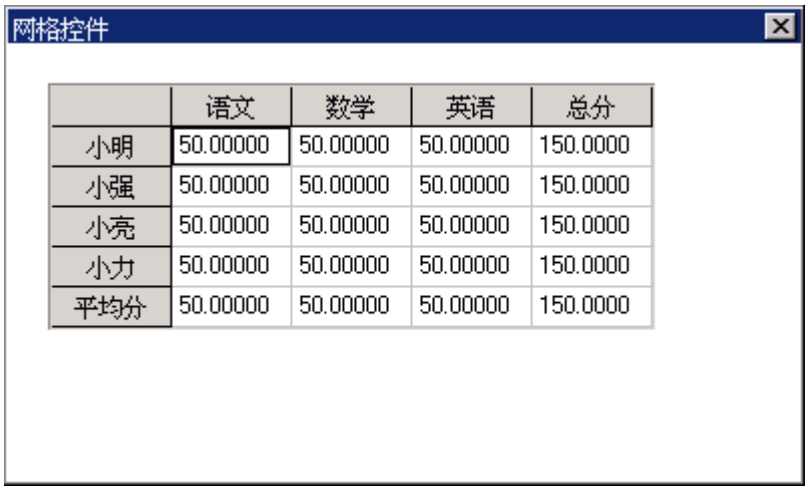


图 6.19 网格控件

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。

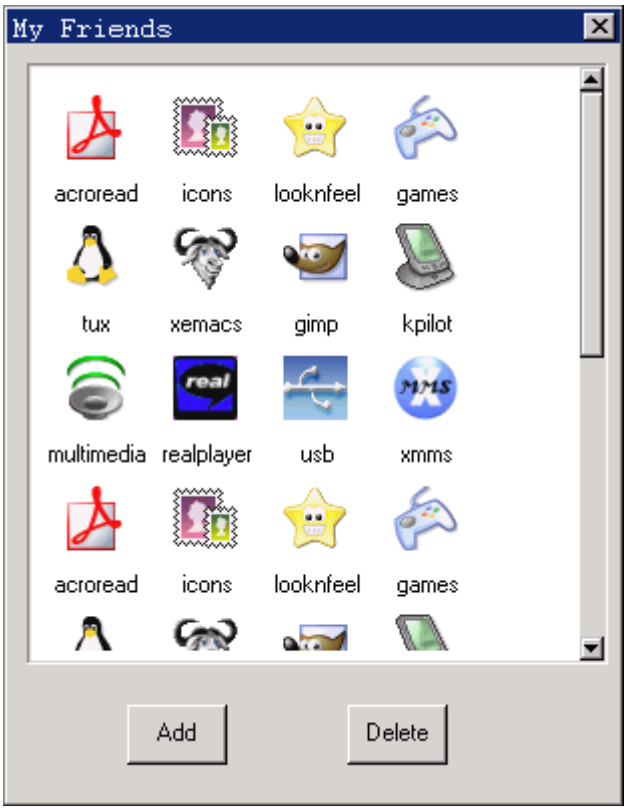


图 6.20 图标型控件

7 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。最新的 MiniGUI V3.0 还新增加了对阿拉伯文、希伯莱文等双向文本的显示支持。

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。

8 MiniGUI 组件

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。

8.1 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 8.1 是 mGp 组件的打印设置对话框。

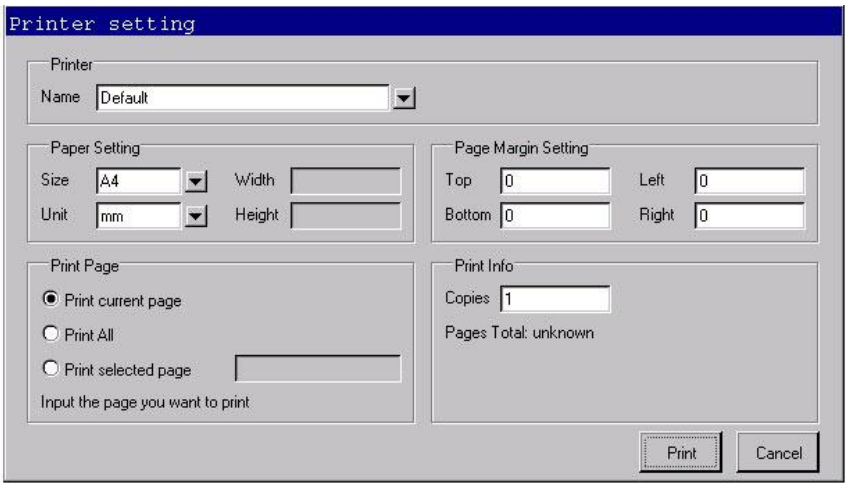


图 8.1 飞漫软件 mGp 打印组件的设置窗口

8.2 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 8.2 和图 8.3 是利用 mGi 组件开发的软键盘及手写输入法界面。

图 8.2 mGi 输入法组件中的软键盘输入法

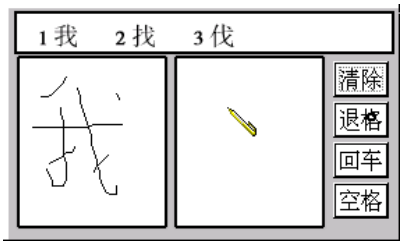


图 8.3 mGi 输入法组件中的手写输入法窗口

8.3 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 8.4 是 mG3d 组件的运行效果图。

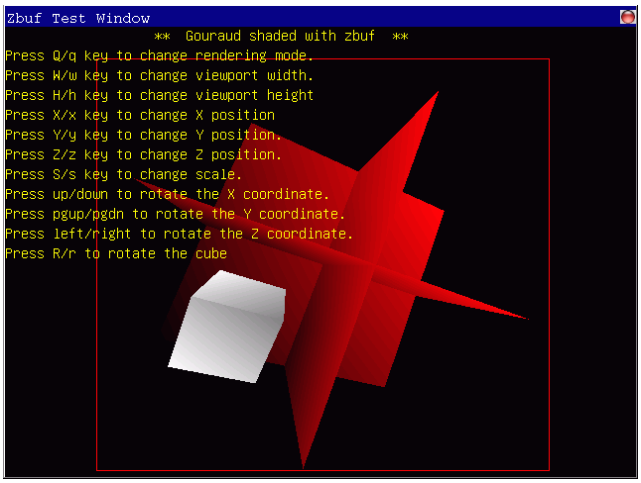


图 8.4 mG3d 组件的运行时效果图

8.4 mGUtils

mGUtils 组件为用户提供了好几个功能模板，有了这些模板，用户就不用为一些常用的功能，再去编写代码了。本组件提供的功能模板有：

- ✓ 普通文件对话框：此对话框模板具有文件打开，保存以及另存为功能。有两种外观模式：简洁模式和 PC 模式。用户只要设置好数据结构 FILEDLGDATA，然后调用相关的函数接口，就能执行相应的功能了。使用起来，简洁方便。
- ✓ 颜色设置对话框：此对话框模板在你需要为应用层客户提供调色板的时候，绝对派的上用场。颜色对话框也分为简洁模式和标准 PC 模式。用户只要设置好数据结构 COLORDLGDATA，然后调用颜色对话框函数接口，就可以获得一个漂亮的颜色配置界面了，获取之易，可谓信手拈来！
- ✓ 字体设置对话框：此对话框提供了一个和 Windows 的一模一样字体设置对话框。字体对话框在应用开发中的作用，每个开发者都很清楚。用户首先设置好 FONTDLGDATA 数据结构，然后调用字体设置函数接口，便可实现字体的设置了。
- ✓ 信息设置对话框：此对话框提供了一个显示特性信息的对话框模板。有了这个模板，用户就不用为了显示一些弹出信息而专门去写一个对话框了。直接设置好数据结构 INFODLGDATA，然后就可以调用模板函数接口显示信息了。图 8.5 为 mGUtils 组件中的颜色设置对话框和普通文件对话框运行效果图。

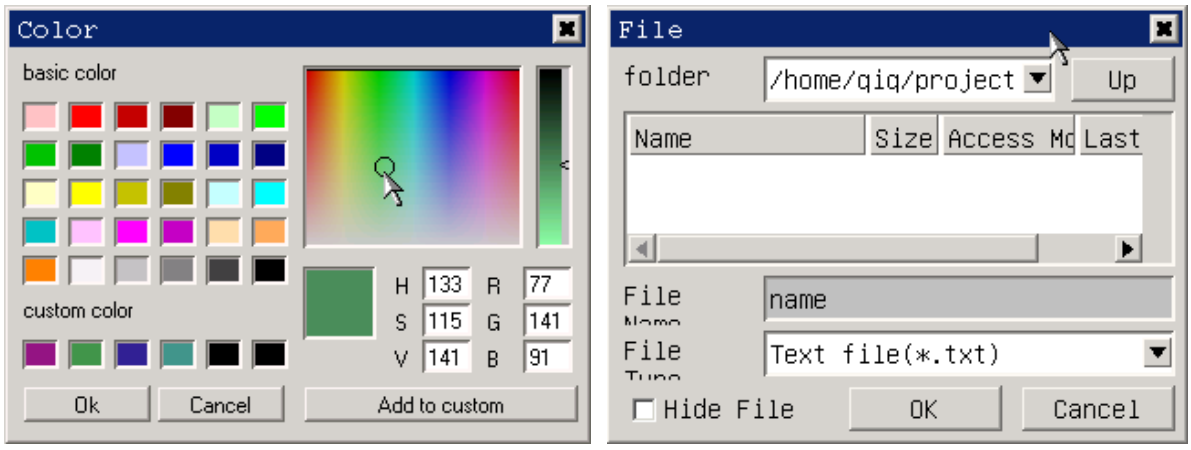


图 8.5　mGUtils 组件的颜色设置对话框和普通文件对话框的运行效果图

8.5　mGPlus

mGPlus 组件是对 MiniGUI 图形绘制接口的一个扩充和增强，主要提供对二维矢量图形和高级图形算法的支持，如路径、渐变填充和颜色组合等：

- ✓ 路径：路径是由一组有严格的顺序折线和曲线组成的。可以使用路径进行填充和剪切。通过提供对路径的支持，我们可以实现矢量图形的绘画，可以支持对矢量图形的无极缩放，旋转等功能，同时还能对矢量字体提供更好的支持。
- ✓ 渐变填充：渐变填充是指使用一个颜色线性渐变或者按某个路径渐变的画刷，在某个指定区域，或者路径区域内，或者图形进行填充。有了渐变填充，我们就可以实现更加漂亮的更有立体感的控件了，这后续会有详细介绍。目前 MiniGUI 实现的渐变方式有弧形渐变填充和线性渐变填充。
- ✓ 颜色组合：在当今非常重视产品包装的年代里，每一个应用开发者都希望能够开发出非常漂亮精致的用户界面，以获得用户的第一好感。颜色组合可谓是这方面的利器，它能够实现图片之间千变万化的组合，让你的界面获得意想不到的效果。MiniGUI 3.0 实现了十二种组合模式。

图 8.6、图 8.7 分别为路径、颜色组合和颜色渐变显示运行效果图。

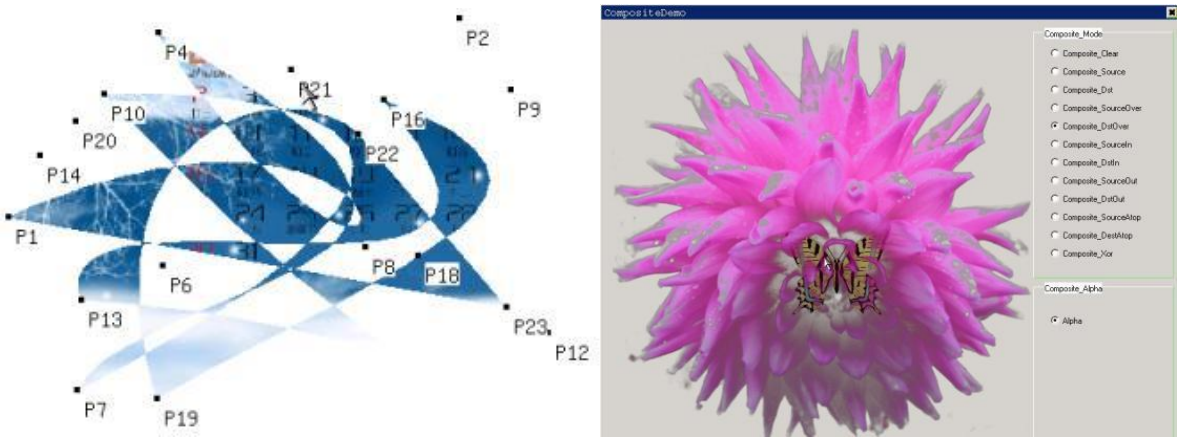


图 8.6　路径和颜色组合运行效果图

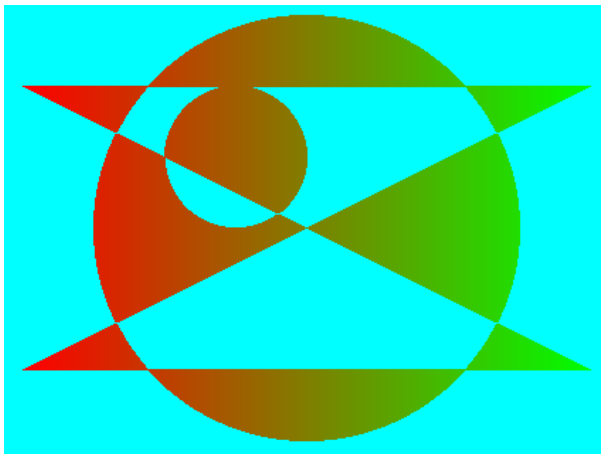


图 8.7 颜色渐变运行效果图

8.6 mGNCS

在 miniStudio 的开发中，为实现可视化图形界面的设计，飞漫软件在 MiniGUI 现有接口基础上，开发了一套新的控件集。miniStudio 引入的新控件集是在原 MiniGUI 控件集基础上发展而来的，为与 MiniGUI 固有控件集（Intrinsic Control Set）区别，称为“新控件集（New Control Set）”，并作为一个新的 MiniGUI 组件 mGNCS 发布。

新控件集的特点如下：

- ✓ 采用面向对象编程思想，重新整理了控件之间的继承关系，用 C 语言实现了类似 C++ 类概念，对外提供 C 语言接口。这点上，miniStudio 引入的新控件集和 Gtk+ 的控件集类似。
- ✓ 对控件的接口和风格做了规范，使所有控件都能用统一的接口访问。
- ✓ 在 MiniGUI 3.0 外观渲染器的基础上，进一步扩展了外观渲染器的概念，新控件集中的每个控件都具有自己的外观渲染器，并按照控件的继承规则定义每个控件专有的渲染器接口，使其能够随着控件的变化而变化。

通过开发新控件集，我们同时解决了 MiniGUI 固有控件集在如下几个方面的不足：

- ✓ 接口不统一，不规范；
- ✓ 通过消息操作控件，代码维护困难，易于出错；
- ✓ 固有控件的层次关系不明，重复代码较多，且不方便定制和扩展控件功能；
- ✓ 固有控件的绘制效率较低，在开发板上运行时闪烁现象比较严重。

新控件集主要配合 miniStudio 使用，也可以作为 MiniGUI 3.0 的一个组件而直接使用，且可以和固有控件集中的控件混合使用。

我们强烈建议新的 MiniGUI 应用开发使用 mGNCS，而不再使用 MiniGUI 内置控件。

8.7 mGEff

mGEff 为 MiniGUI 应用程序提供了一个动画框架。mGEff 提供了大量稳定而高效的特效器，为开发者快速实现翻转，放大，滚屏，卷页等常用动画提供了便利。另外，mGEff 可以与 MiniGUI 结合，以双缓冲为基础开发了针对主窗口动画的动画接口。

9 MiniGUI 相关资源

有关 MiniGUI 及其组件的源代码包、示例程序包，运行在 MiniGUI 之上的其他开源软件及其文档等，请访问：

```
http://www.minigui.com/
```

飞漫软件同时通过 GitHub 网站发布 MiniGUI 及其组件的源代码包：

```
https://github.com/VincentWei
```

有关 MiniGUI 的完整文档：

```
http://wiki.minigui.com
```

10 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，则应该向 MiniGUI 的版权拥有人（飞漫软件）购买商业授权。

10.1 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。

10.2 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。

10.3 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。

有关商业授权的方式、价格及购买方法，请访问

```
http://www.fmsoft.cn
```

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。

11 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
http://www.fmsoft.cn
```

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，请查看飞漫网站的技术及产品部分：

```
http://www.fmsoft.cn
```

有关飞漫软件产品技术咨询、产品购买等问题，请致信：

```
sales@minigui.com
```