



MiniGUI 用户手册

版本 2.0 修订号 3

适用于 MiniGUI Ver 2.0.3/1.6.9

北京飞漫软件技术有限公司

2006 年 6 月

版权声明

《MiniGUI 用户手册》 版本 2.0 修订号 3，适用于 MiniGUI 版本 2.0.3/1.6.9。

版权所有 (C) 2003~2006，北京飞漫软件技术有限公司，保留所有权利。

无论您以何种方式获得该手册的全部或部分文字或图片资料，无论是普通印刷品还是电子文档，北京飞漫软件技术有限公司仅仅授权您阅读的权利，任何形式的格式转换、再次发布、传播以及复制其内容的全部或部分，或将其中的文字和图片未经书面许可而用于商业目的，均被视为侵权行为，并可能导致严重的民事或刑事处罚。

目 录

版权声明	1
1 MiniGUI增值版介绍	1
1.1 MiniGUI增值版简介	1
1.2 MiniGUI 增值版产品文档	2
1.3 MiniGUI 源代码以及示例程序	3
1.4 产品光盘中的其他资源	4
1.5 可选组件产品	4
1.6 关于本手册	5
2 MiniGUI 的配置、编译和安装	7
2.1 编译时配置选项的生成	7
2.1.1 在 GNU 开发环境中使用 <code>configure</code> 脚本进行配置	8
2.1.2 非 GNU 环境下的配置	14
2.2 MiniGUI的编译时配置选项详解	15
2.2.1 操作系统相关的选项和宏	15
2.2.2 目标板相关的选项和宏	16
2.2.3 运行模式相关的选项和宏	17
2.2.4 图形引擎相关的选项和宏	18
2.2.5 输入引擎相关的选项和宏	19
2.2.6 键盘布局的相关选项和宏	22
2.2.7 系统全局配置选项和宏	23
2.2.8 输入法相关的选项和宏	24
2.2.9 字符集和字体相关的选项和宏	24
2.2.10 图像文件格式相关的选项和宏	28
2.2.11 外观风格相关的选项和宏	28
2.2.12 控件相关的选项和宏	29
2.2.13 其他选项和宏	30
2.3 最小配置选项	31
2.3.1 使用 GNU <code>configure</code> 脚本	31
2.3.2 对应的 <code>mgconfig.h</code>	33
2.4 MiniGUI 的编译和安装	43
2.4.1 GNU 开发环境中的编译和安装	43
2.4.2 在非 GNU 开发环境中使用 <code>cygwin</code> 工具编译和安装 MiniGUI	44

3 MiniGUI 的运行时配置选项	47
3.1 配置文件	47
3.1.1 system段	48
3.1.2 fbcon段	48
3.1.3 qvfb段	49
3.1.4 rawbitmapfonts、varbitmapfonts、qpf、truetypefonts和type1fonts段	49
3.1.5 systemfont段	51
3.1.6 mouse段	52
3.1.7 event段	53
3.1.8 cursorinfo段	53
3.1.9 iconinfo段	54
3.1.10 bitmapinfo段	54
3.1.11 bgpicture段	55
3.1.12 mainwinmetrics段	56
3.1.13 windowelementcolors段	57
3.1.14 imeinfo段	59
3.1.15 默认配置文件	60
3.2 内建式配置选项	64
3.2.1 ETCSETCTCION结构	64
3.2.2 ETC_S结构	65
3.2.3 mgetc.c文件清单	66
3.3 配置示例	68
3.3.1 只支持 ISO8859-1 字符显示的运行时配置	68
3.3.2 指定不同的图形引擎和输入引擎	69
4 在 Windows 平台上开发 MiniGUI 应用程序	71
附录 A 常见问题及解答	73
A.1 GPL 版本问题	73
A.2 应用问题	73
A.3 移植性问题	74
A.4 编译问题	74
A.5 输入引擎	75
A.8 运行时问题	76
A.9 常见错误信息	77
附录 B 索引	79

1 MiniGUI 增值版介绍

1.1 MiniGUI 增值版简介

MiniGUI (<http://www.minigui.com>) 是根据嵌入式系统应用特点量身定做的图形支持系统。它源自一个由魏永明主持和开发的自由软件项目, 现由北京飞漫软件技术有限公司维护并开展后续开发。MiniGUI 项目的最初目标是为基于 Linux 的实时嵌入式系统提供一个轻量级的图形用户界面。该项目自 1998 年底开始到现在, 历经近八年的开发和应用过程, 已非常成熟和稳定, 并且在许多实际产品或项目中得到了广泛应用。MiniGUI 能够支持包含 Linux 在内的多种操作系统, 例如 uClinux、VxWorks、eCos、uC/OS-II、pSOS、ThreadX、Nucleus、OSE 等, 也可以在 Win32 平台上运行。MiniGUI 接口清晰, 设计严谨, 利用 MiniGUI 进行开发和调试也十分简单、高效。

MiniGUI 为应用程序定义了一组轻量级的窗口和图形设备接口。利用这些接口, 每个应用程序可以建立多个窗口并在这些窗口中创建各种控件。MiniGUI 还提供了丰富的图形功能, 帮助用户显示各种格式的位图并在窗口中输出各种文本或绘制复杂图形。

MiniGUI 增值版是飞漫软件为专有或商用产品开发商提供的 MiniGUI 商业产品。除 MiniGUI 增值版之外, 飞漫软件还遵循 GPL 发布 MiniGUI 的两个版本: MiniGUI V1.3.3 和 MiniGUI-STR V1.6.2。您可以在如下网页下载得到这两个 GPL 版本以及相关的示例代码和文档:

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

MiniGUI 增值版 (MiniGUI-VAR) 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 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, 如 VxWorks、ThreadX、Nucleus、OSE、eCos、uC/OS-II 等, 主要提供对 MiniGUI-Threads 运行模式的支持; MiniGUI-VAR V2.0.x 主要用于支持具有多进程特性的操作系统, 如 Linux 和 VxWorks 6, 提供对 MiniGUI-Processes 运行模式的支持。

MiniGUI 增值版按所支持的操作系统不同划分为不同的产品, 具体产品名称及版本见表 1.1。表 1.1 还给出了各产品所支持的 MiniGUI 运行模式。

表 1.1 MiniGUI 增值版产品及所支持的运行模式

产品名称及版本	支持的 MiniGUI 运行模式
MiniGUI-VAR for Linux/uClinux V2.0.x	MiniGUI-Processes MiniGUI-Threads MiniGUI-Standalone
MiniGUI-VAR for Linux/uClinux V1.6.x	MiniGUI-Lite MiniGUI-Threads

	MiniGUI-Standalone
MiniGUI-VAR for VxWorks 6 V2.0.x	MiniGUI-Processes MiniGUI-Threads
MiniGUI-VAR for VxWorks V1.6.x	MiniGUI-Threads
MiniGUI-VAR for ThreadX V1.6.x	MiniGUI-Threads
MiniGUI-VAR for Nucleus V1.6.x	MiniGUI-Threads
MiniGUI-VAR for OSE V1.6.x	MiniGUI-Threads
MiniGUI-VAR for pSOS V1.6.x	MiniGUI-Threads
MiniGUI-VAR for eCos V1.6.x	MiniGUI-Threads
MiniGUI-VAR for uC/OS-II V1.6.x	MiniGUI-Threads

除了运行模式上的不同之外，MiniGUI-VAR V2.0.x 和 MiniGUI-VAR V1.6.x 在其他技术特性及接口上是一致的。

有关 MiniGUI 运行模式以及 MiniGUI 技术特性的详细描述，请参阅《MiniGUI 技术白皮书 for V2.0.3/1.6.9》以及《Datasheet for MiniGUI V2.0.3/1.6.9》等文档。

1.2 MiniGUI 增值版产品文档

除本手册之外，MiniGUI 增值版产品中还包含有如下印刷的文档资料：

- 《MiniGUI 编程指南》 V2.0-3。描述利用 MiniGUI V2.0.3/V1.6.9 开发应用程序时的相关概念、主要的 API 接口及示例程序等。

产品光盘的 minigui/docs/ 目录中包含有本手册以及《MiniGUI 编程指南》 V2.0-3 的 PDF 格式电子版。除此之外，该目录中还包含有如下文档的电子版本（PDF 格式）：

- 《MiniGUI API Reference Manual》 V2.0.3。对 MiniGUI V2.0.3 接口（MiniGUI-Processes 运行模式）的详细描述¹。
- 《MiniGUI API Reference Manual》 V1.6.9。对 MiniGUI V1.6.9 接口（MiniGUI-Threads 运行模式）的详细描述²。
- 针对某特定操作系统的《MiniGUI 移植指南》。详细讲述将 MiniGUI 移植到某特定操作系统上的具体步骤以及应该注意的事项。
- 《MiniGUI 技术白皮书 for V2.0.3/1.6.9 以及《Datasheet for MiniGUI V2.0.3/1.6.9》。

产品光盘根目录中的 README 文件详细描述了上述文档对应的文件名称。产品光盘的根目录中还包含有 ReleaseNotes.pdf 文件，该文件详细描述了新版本中的新增功能、增强或者优化等等；请特别注意可能引起兼容性问题的增强或缺陷修正。

¹该文档以电子版形式提供：HTML 格式及 Windows 预编译帮助文档格式，仅提供英文版。

²该文档以电子版形式提供：HTML 格式及 Windows 预编译帮助文档格式，仅提供英文版。

1.3 MiniGUI 源代码以及示例程序

您所购买的 MiniGUI 增值版产品光盘中包含有对应版本和操作系统的 MiniGUI 源代码。

对 MiniGUI-VAR V2.0.3, 相关代码包保存在产品光盘的 minigui/2.0.x 目录中, 相关文件如下所列:

- libminigui-2.0.3-<os>.tar.gz: 针对 <os> (如 linux) 操作系统的 MiniGUI V2.0.3 函数库源代码。MiniGUI由三个函数库组成: 它们分别是libminigui、libmgext 以及 libvcongui。libminigui 是提供窗口管理和图形接口的核心函数库, 也提供了大量的标准控件; libmgext 是 libminigui 的一个扩展库, 提供了一些高级控件以及 “文件打开”、“颜色选择” 对话框等; libvcongui 则为 Linux 操作系统提供了一个应用程序可用的虚拟控制台窗口, 从而可以方便地在 MiniGUI 环境中运行字符界面的应用程序, libmgext和libvcongui库已经包含在这个源代码包中。
- minigui-res-2.0.3.tar.gz: MiniGUI 所使用的资源, 包括基本字体、图标、位图和鼠标光标。
- mg-samples-2.0.3.tar.gz: 《MiniGUI 编程指南》的配套示例程序。
- mde-2.0.3.tar.gz: MiniGUI 的综合演示程序包, 其中包含有一些较为复杂的示例程序。

对 MiniGUI-VAR V1.6.9, 相关代码包保存在产品光盘的 minigui/1.6.x 目录中, 相关文件如下所列:

- libminigui-1.6.9-<os>.tar.gz: 针对 <os> (如 vxworks) 操作系统的 MiniGUI V1.6.9 函数库源代码。MiniGUI由三个函数库组成: 它们分别是libminigui、libmgext 以及 libvcongui。libminigui 是提供窗口管理和图形接口的核心函数库, 也提供了大量的标准控件; libmgext 是 libminigui 的一个扩展库, 提供了一些高级控件以及 “文件打开”、“颜色选择” 对话框等; libvcongui 则为 Linux 操作系统提供了一个应用程序可用的虚拟控制台窗口, 从而可以方便地在 MiniGUI 环境中运行字符界面的应用程序, libmgext和libvcongui库已经包含在这个源代码包中。
- minigui-res-1.6.9.tar.gz: MiniGUI 所使用的资源, 包括基本字体、图标、位图和鼠标光标。
- mg-samples-1.6.9.tar.gz: 《MiniGUI 编程指南》的配套示例程序。
- mde-1.6.9.tar.gz: MiniGUI 的综合演示程序包, 其中包含有一些较为复杂的示例程序。

【注意】 为了便于您将 **MiniGUI** 进行交叉编译并运行在最终的目标版上, 飞漫软件在 **MiniGUI** 增值版产品中提供了针对某特定操作系统的 **MiniGUI** 完整源代码。飞漫软件仅仅允许您为了支持特定的硬件平台而增加或修改输入引擎、图形引擎以及相关的编译工具文件; 飞漫软件不允许您修改 **MiniGUI** 的其他源代码。

用程序添加三维图像，文字渲染、场景渲染等效果。mG3d 可以和所有的 MiniGUI 增值版产品配合使用。

除上述三种组件产品外，飞漫软件还提供 MiniGUI 的 Win32 SDK 软件包。利用该软件包，可以在 Win32 上运行 MiniGUI 以及相关应用程序，利用 Visual Studio 等集成开发环境，可以非常方便地编写和调试 MiniGUI 应用程序。但需要注意以下几点：

- MiniGUI SDK for Win32 只提供对 MiniGUI-Threads 运行模式的支持；它基于 MiniGUI V1.6.9 开发。
- 利用 MiniGUI SDK for Win32 开发 MiniGUI 应用程序时，请不要调用 Windows 的特有函数接口，因为这些函数接口可能在最终的操作系统上并不存在。

有关飞漫软件产品的完整介绍及购买信息，请访问如下网页：

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

1.6 关于本手册

本手册主要描述 MiniGUI 的编译时配置选项以及运行时配置选项。把 MiniGUI 及基于 MiniGUI 的应用程序交叉编译并在目标板上运行的方法，随开发环境以及目标板的不同而不同；相关的信息，请参阅针对特定操作系统的《MiniGUI 移植指南》。

2 MiniGUI 的配置、编译和安装

嵌入式系统往往是一种定制设备，它们对图形系统的需求也各不相同；有些系统只要求一些图形功能，而有些系统要求完备的图形、窗口以及控件的支持。因此，嵌入式图形系统必须是可定制的。MiniGUI 实现了大量的编译时配置选项，通过这些选项可指定 MiniGUI 库中包括哪些功能或者不包括哪些功能。大体说来，我们可以在编译时，在如下几个方面对 MiniGUI 进行配置：

- 指定 MiniGUI 要运行的操作系统或者目标板。
- 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。
- 指定需要支持的图形引擎和输入引擎，以及引擎相关选项。
- 指定需要支持的字体类型、内嵌的字体种类。
- 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。

本章将详细介绍 MiniGUI 的编译时配置选项，以使用户能够自行定制出最适合自己的嵌入式系统的 MiniGUI。本章还将简单介绍 MiniGUI 的编译和安装；因为 MiniGUI 的编译和安装和具体的开发环境相关，因此，详细的编译和安装过程，请参阅针对特定操作系统的《MiniGUI 移植指南》文档。

2.1 编译时配置选项的生成

在 MiniGUI 源代码最顶层目录中有一个名为 mgconfig.h 的头文件，该文件中定义有大量的 C 语言宏。通过控制这些宏的开启或关闭，我们就可以对 MiniGUI 进行定制。一般来讲，我们可以自行修改这个文件来达到定制 MiniGUI 的目的。在修改这个文件之后，一定要重新编译 MiniGUI，然后将头文件和函数库安装到系统中。如果您的 MiniGUI 应用程序是静态链接的，则还需要重新编译应用程序。需要注意的是，一定要将修改后的 mgconfig.h 文件复制到编译器能够找到的 MiniGUI 头文件目录中，并覆盖旧的文件。

文件 mgconfig.h 一般具有如下的形式：

```
...  
  
/* Define if compile for VxWorks operating system */  
#define __VXWORKS__ 1  
  
/* Define if include IAL engine for ABSSIG */  
/* #undef _ABSSIG_IAL */
```

```
/* Define if include IAL engine for Arca Tpanel Ads7846 */
/* #undef ADS7846 IAL */

/* Define if include IAL engine for ADS */
/* #undef _ADS_IAL */

/* Define if include advanced 2D graphics APIs */
#define ADV_2DAPI 1

/* Define if support Arabic charset */
/* #undef ARABIC_SUPPORT */

/* Define if include IAL engine for ARM3000 */
/* #undef ARM3000 IAL */

/* Define if include the automatic IAL engine */
/* #undef AUTO_IAL */

/* Define if support BIG5 charset */
#define BIG5_SUPPORT 1

/* Define if include clipboard support */
#define CLIPBOARD_SUPPORT 1

...
```

上面给出的是 `mgconfig.h` 的一个片断。该文件定义了 `__VXWORKS__` 宏，这个宏将打开 MiniGUI 源代码中针对 VxWorks 的支持代码；该文件中定义了 `_CLIPBOARD_SUPPORT` 宏，这个宏将打开 MiniGUI 中的剪切板功能；该文件没有定义 `_AUTO_IAL` 宏，则编译后的 MiniGUI 中将不包含对 Auto 输入引擎的支持。

注意，`mgconfig.h` 中还包含其他一些宏的定义，比如 MiniGUI 版本号等。请保持这些宏的定义不变，不要自行修改这些宏的定义。

手工修改 `mgconfig.h` 的做法非常繁琐，而且容易出错。如果您使用 GNU 的开发环境，则可以使用 `configure` 脚本来进行相关的配置。下面的小节将介绍如何使用 `configure` 脚本在 GNU 开发环境中自动生成 `mgconfig.h` 文件。

2.1.1 在 GNU 开发环境中使用 `configure` 脚本进行配置

我们知道，利用 `makefile` 可以非常方便地维护程序包。通过 `makefile`，我们可以编译、清除或者安装软件包中的函数库、可执行文件、头文件等等。尽管通过 `makefile` 文件可以组织一个应用软件工程项目，但往往手工编写一个 `makefile` 文件并不是一件轻松的事情，并且在需要维护大型的源代码目录树时，`makefile` 文件的维护工作就会大大增加。为此，自由软件基金会的 GNU 项目为众多基于 C 语言的软件项目开发了 `Autoconf/Automake` 工具。利用这个工具，我们可以自动生成 `makefile` 文件，并且能够检查系统的配置信息，从而帮助提高应用软件的可移植性。

MiniGUI（MiniGUI 库本身及示例程序包）就是通过 GNU 的 `Automake/Autoconf` 脚本组织的。因此，如果您使用 GNU 兼容的开发环境，比如 Linux 平台或者 Windows 平台上的 `cygwin` 环境等等，

就可以使用 MiniGUI 的 Automake/Autoconf 配置脚本来配置 MiniGUI。使用 MiniGUI 的 Automake/Autoconf 配置脚本，并不需要安装 Automake/Autoconf 工具本身，而只要运行 MiniGUI 源代码包中的 configure 脚本即可完成配置。运行 configure 脚本，除了会生成 makefile 文件之外，还可以根据传递给 configure 脚本的各种选项来生成 mgconfig.h 文件。之后，我们只要运行 make 和 make install 命令，就可以编译 MiniGUI 并将 MiniGUI 的头文件和函数库安装到指定的目录中。

【提示】MiniGUI 的 configure 脚本只能在 GNU 兼容的主机开发环境中使用。GNU 兼容的主机开发环境通常有：Linux 系统、Windows 上运行的 cygwin 环境等，可应用于 MiniGUI 增值版产品的 Linux、uClinux、eCos 等版本。

MiniGUI 的 configure 脚本中包含有大量的配置选项，基本上每个配置选项都对应于 mgconfig.h 中的一个宏或者若干个宏。如果在运行 configure 时打开某个选项，则会定义对应的宏；否则就不会定义这个宏。运行下面的命令

```
user$ ./configure --help
```

可以获得完整的可配置选项清单。比如，假定您使用的是 Debian Linux 3.1 系统作为开发主机，下面是在 MiniGUI 的源代码目录下运行上面的命令输出的结果（注意该命令在其它 Linux 发行版上的输出可能会有所不同）：

```
`configure' configures this package to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE. See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help                display this help and exit
  --help=short              display options specific to this package
  --help=recursive          display the short help of all the included packages
  -V, --version             display version information and exit
  -q, --quiet, --silent    do not print `checking...' messages
  --cache-file=FILE        cache test results in FILE [disabled]
  -C, --config-cache        alias for `--cache-file=config.cache'
  -n, --no-create           do not create output files
  --srcdir=DIR              find the sources in DIR [configure dir or `..']

Installation directories:
  --prefix=PREFIX           install architecture-independent files in PREFIX
                          [/usr/local]
  --exec-prefix=EPREFIX     install architecture-dependent files in EPREFIX
                          [PREFIX]

By default, `make install' will install all the files in
`/usr/local/bin', `/usr/local/lib' etc. You can specify
an installation prefix other than `/usr/local' using `--prefix',
for instance `--prefix=$HOME'.

For better control, use the options below.

Fine tuning of the installation directories:
  --bindir=DIR              user executables [EPREFIX/bin]
  --sbindir=DIR            system admin executables [EPREFIX/sbin]
  --libexecdir=DIR         program executables [EPREFIX/libexec]
```

```
--datadir=DIR          read-only architecture-independent data [PREFIX/share]
--sysconfdir=DIR       read-only single-machine data [PREFIX/etc]
--sharedstatedir=DIR   modifiable architecture-independent data [PREFIX/com]
--localstatedir=DIR    modifiable single-machine data [PREFIX/var]
--libdir=DIR           object code libraries [EPREFIX/lib]
--includedir=DIR       C header files [PREFIX/include]
--oldincludedir=DIR    C header files for non-gcc [/usr/include]
--infodir=DIR          info documentation [PREFIX/info]
--mandir=DIR           man documentation [PREFIX/man]

Program names:
--program-prefix=PREFIX      prepend PREFIX to installed program names
--program-suffix=SUFFIX      append SUFFIX to installed program names
--program-transform-name=PROGRAM run sed PROGRAM on installed program names

System types:
--build=BUILD      configure for building on BUILD [guessed]
--host=HOST        cross-compile to build programs to run on HOST [BUILD]
--target=TARGET    configure for building compilers for TARGET [HOST]

Optional Features:
--disable-FEATURE      do not include FEATURE (same as --enable-FEATURE=no)
--enable-FEATURE[=ARG] include FEATURE [ARG=yes]
--enable-shared=PKGS   build shared libraries default=yes
--enable-static=PKGS   build static libraries default=yes
--enable-fast-install=PKGS optimize for fast installation default=yes
--disable-libtool-lock avoid locking (might break parallel builds)
--disable-dependency-tracking Speeds up one-time builds
--enable-dependency-tracking Do not reject slow dependency extractors
--enable-ownmalloc      use own implementation of malloc functions <default=no>
--enable-ownstdio       use own implementation of stdio functions <default=no>
--enable-ownpthread     use own implementation of pthread functions <default=no>
--enable-procs          build MiniGUI-Processes version <default=no>
--enable-standalone     build MiniGUI-Standalone version <default=no>
--enable-incoreres      use incore resource instead file IO to initialize MiniGUI <default=no>
>
--enable-miniguientry   use minigui entry function in MiniGUI <default=no>
--enable-fixedmath      include fixed math routines <default=yes>
--enable-debug          build with debugging messages <default=no>
--enable-tracemsg       trace messages of MiniGUI <default=no>
--enable-msgstr         include symbol name of message <default=no>
--enable-micemoveable   user can move window by using mouse <default=yes>
--enable-dblclk         mouse button can do double click <default=yes>
--enable-cursor         include cursor support <default=yes>
--enable-clipboard      include clipboard support <default=yes>
--enable-coortrans cw   support clockwise rotation of screen in the native FB GAL engine <default=no>
--enable-coortrans ccw support counterclockwise rotation of screen in the native FB GAL engine <default=no>
--enable-ep7211ial      build the IAL engine for EP7211-based board <default=no>
--enable-adsial         build the IAL engine for ADS Graphics Client board <default=no>
--enable-ipaqial        build the IAL engine for iPAQ H3600 <default=no>
--enable-l7200ial       build the IAL engine for L7200 <default=no>
--enable-arm3000ial      build the IAL engine for ARM3000 <default=no>
--enable-dm270ial       build the IAL engine for DM270 <default=no>
--enable-evmv10ial      build the IAL engine for xScale EVMV 1.0 <default=no>
--enable-fxrm9200ial     build the IAL engine for FXRM9200 <default=no>
--enable-abssigial      build the IAL engine for ABSSIG <default=no>
--enable-embest2410ial   build the IAL engine for EMBEST ARM2410 <default=no>
--enable-embest44b0ial   build the IAL engine for EMBEST ARM44B0 <default=no>
--enable-fft7202ial     build the IAL engine for FFT7202 <default=no>
--enable-palm2ial       build the IAL engine for OKWAP Palm2 <default=no>
--enable-mpc823ial      build the IAL engine for mpc823 <default=no>
--enable-ucb1x00ial     build the IAL engine for ucb1x00 <default=no>
--enable-px255bial      build the IAL engine for px255b <default=no>
--enable-vr4181ial      build the IAL engine for NEC VR4181 debug board <default=no>
--enable-helioial       build the IAL engine for Helio Touch Panel <default=no>
--enable-tfstbial       build the IAL engine for Tongfang STB <default=no>
--enable-t800ial        build the IAL engine for MT T800 <default=no>
--enable-mc68x328ial    build the IAL engine for uClinux touch screen palm/mc68ez328 <default=no>
--enable-smdk2410ial    build the IAL engine for SMDK2410 touch screen <default=no>
--enable-hh2410r3ial     build the IAL engine for HHARM2410R3 touch screen <default=no>
--enable-hh2440ial      build the IAL engine for HHARM2440 touch screen <default=no>
--enable-skyeyeep7312ial build the IAL engine for touch screen of SkyEye EP7312 simulation <default=no>
```

```

fault=no>
--enable-dmgstb1al    build the IAL engine for Intel DMG STB remote controller <default=no>
--enable-fip1al       build the IAL engine for EM85xx Front Panel and Remote Controller <de
fault=no>
--enable-svpxx1al     build the IAL engine for WinBond SVPXX <Default=no>
--enable-ads78461al   build the IAL engine for Arca Tpanel Ads7846 <Default=no>
--enable-dummy1al     build the Dummy IAL engine <default=yes>
--enable-em861al      build the Em86 IAL engine <default=no>
--enable-em851al      build the Em85 IAL engine <default=no>
--enable-auto1al      build the Automatic IAL engine <default=no>
--enable-random1al    build the Random IAL engine <default=no>
--enable-hhkbd1al     build the IAL engine for HHCF5249 PS/2 keyboard and IrDA <default=no>
--enable-comm1al      build the COMM IAL engine <default=no>
--enable-qvfb1al      build the QVFB IAL engine <default=yes>
--enable-wvfb1al      build the WVFB IAL engine <default=no>
--enable-utp1al       build the IAL engine for UTPMC <default=no>
--enable-native1al    build the native (Linux console) IAL engine <default=yes>
--enable-dfb1al       build the IAL engine for DirectFB <default=no>
--enable-nativeps2    build the native engine subdriver for PS2 mouse <default=yes>
--enable-nativeimps2  build the native engine subdriver for IntelligentMouse (IMPS/2) mouse
<default=yes>
--enable-native1ms    build the native engine subdriver for old MS serial mouse <default=ye
s>
--enable-native1ms3   build the native engine subdriver for MS3 mouse <default=yes>
--enable-native1gpm   build the native engine subdriver for GPM daemon <default=yes>
--enable-text1mode    Linux system have console (text mode) on FrameBuffer <default=yes>
--enable-rbf1support  include raw bitmap font support <default=yes>
--enable-rbf116       include incore RBF font of ISO8859-1 8x16 fixed font (incore resource
) <default=no>
--enable-rbf124       include incore RBF font of ISO8859-1 12x24 fixed font (incore resourc
e) <default=no>
--enable-rbf1gb12     include incore RBF font of GB2312 12x12 fixed/song font (incore resou
rce) <default=yes>
--enable-rbf1gb16     include incore RBF font of GB2312 16x16 fixed/song font (incore resou
rce) <default=no>
--enable-rbf1gb24     include incore RBF font of GB2312 24x24 fixed/song font (incore resou
rce) <default=no>
--enable-rbf1gb16hei  include incore RBF font of GB2312 16x16 fixed/hei font (incore resourc
e) <default=no>
--enable-rbf1big12    include incore RBF font of BIG5 12x12 fixed/ming font (incore resourc
e) <default=no>
--enable-rbf1big16    include incore RBF font of BIG5 16x16 fixed/ming font (incore resourc
e) <default=no>
--enable-rbf1big24    include incore RBF font of BIG5 24x24 fixed/ming font (incore resourc
e) <default=no>
--enable-rbf1kj12     include incore RBF font of SHIFT-JIS Kanji fonts 12 (incore resource)
<default=no>
--enable-rbf1kj14     include incore RBF font of SHIFT-JIS Kanji fonts 14 (incore resource)
<default=no>
--enable-rbf1kj16     include incore RBF font of SHIFT-JIS Kanji fonts 16 (incore resource)
<default=no>
--enable-vbf1support  include var bitmap font support <default=yes>
--enable-font1sserif  include incore font sansserif <default=yes>
--enable-font1courier include incore font courier <default=yes>
--enable-font1symbol  include incore font symbol <default=no>
--enable-font1vgas    include incore font VGAs <default=no>
--enable-font1helv    include incore font Helvetica <default=no>
--enable-gp1fsupport  build support for Qt Prerendered Font (QPF) <default=yes>
--enable-ttf1support  build support for TrueType font <default=no>
--enable-ttf1cache    include ttf cache support <default=no>
--enable-type11support build support for Adobe Type1 font <default=yes>
--enable-latin21support include East European (Latin 2, ISO-8859-2) charset support <default=
no>
--enable-latin31support include South European (Latin 3, ISO-8859-3) charset support <default
=no>
--enable-latin41support include North European (Latin 4, ISO-8859-4) charset support <default
=no>
--enable-cyrillic1support include Cyrillic (ISO-8859-5) charset support <default=no>
--enable-arabic1support include Arabic (ISO-8859-6) charset support <default=no>
--enable-greek1support include Greek (ISO-8859-7) charset support <default=no>
--enable-hebrew1support include Hebrew (ISO-8859-8) charset support <default=no>
--enable-latin51support include Turkish (Latin 5, ISO-8859-9) charset support <default=no>
--enable-latin61support include Nordic, Latin 6, ISO-8859-10) charset support <default=no>
--enable-thai1support include Thai (ISO-8859-11) charset support <default=no>
--enable-latin71support include Latin 7 (ISO-8859-13) charset support <default=no>

```

12

```

--enable-videoutpmc      include NEWGAL engine for UTPMC <default=no>
--enable-videodfb        include NEWGAL engine for DirectFB <default=no>
--enable-adv2dapi        include advanced 2D graphics APIs <default=yes>

Optional Packages:
--with-PACKAGE[=ARG]     use PACKAGE [ARG=yes]

--without-PACKAGE        do not use PACKAGE (same as --with-PACKAGE=no)
--with-gnu-ld            assume the C compiler uses GNU ld default=no
--with-pic              try to use only PIC/non-PIC objects default=use both
--with-osname=linux/uclinux/ecos/ucos2/swlinux/vxworks/win32/darwin/threadx/cygwin/nucleus/ose
--with-targetname=unknown/stb810/vfanvil/vxi386/qvfb/fbcon/mx21
--with-style=classic/flat/fashion
--with-ttfcachesize=64/128/256/512/1024
--with-mttfcachenum=10/20/40

Some influential environment variables:
CC          C compiler command
CFLAGS      C compiler flags
LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in a
            nonstandard directory <lib dir>
CPPFLAGS    C/C++ preprocessor flags, e.g. -I<include dir> if you have
            headers in a nonstandard directory <include dir>
CPP         C preprocessor

Use these variables to override the choices made by `configure' or to help
it to find libraries and programs with nonstandard names/locations.

```

上面这些参数是已经在 `configure` 脚本中设置好的命令行参数, 这些参数可以控制在编译 MiniGUI 时包含支持哪些功能的代码。例如, 运行:

```
user$ ./configure --with-style=fashion --enable-procs --enable-standalone
```

就可以将 MiniGUI 配置成具有 **Fashion** 风格的外观, 且具有 **MiniGUI-Standalone** 运行模式。如果运行:

```
user$ ./configure --with-style=flat --enable-procs
```

则会将 MiniGUI 配置成 **MiniGUI-Processes** 模式, 使用 **Flat** 风格。不带任何参数执行 `./configure` 命令将按照默认编译配置选项生成 **Makefile**。注意在每个编译配置选项的说明中都给出了默认设置: `<default=yes>` (表示该编译配置选项默认为开启状态) 或者 `<default=no>` (表示该编译配置选项默认为关闭状态)。

除了 MiniGUI 定义的配置选项之外, `configure` 脚本还带有一些重要的通用编译配置选项。

1) prefix 选项

该编译配置选项用于指定 MiniGUI 函数库的安装路径。默认的安装路径是 `/usr/local`。如果运行:

```
user$ ./configure --prefix=/home/test
```

那么在执行 `make install` 命令之后, 函数库、头文件以及参考手册页将被安装在 `/home/test/lib`、`/home/test/include`、`/home/test/man` 目录下。

2) 交叉编译选项

--build、--host 以及 --target 等编译配置选项对应用程序的交叉编译非常重要。例如，如果您使用 arm-linux 交叉编译工具链，则可以如下这样指定 --build 等选项，从而让 configure 脚本生成的 makefile 文件可用来完成针对 arm-linux 的交叉编译：

```
user$ CC=arm-linux-gcc ./configure --prefix=/usr/local/arm/2.95.3/arm-linux/ \
--build=i386-linux \
--host=arm-unknown-linux \
--target=arm-unknown-linux
```

上面这条命令中的 --prefix 选项设定了安装 MiniGUI 配置文件、函数库及头文件的目录前缀，在执行 make install 命令时，将把 MiniGUI 配置文件、库文件和头文件分别安装到如下位置：

- /usr/local/arm/2.95.3/arm-linux/etc/
- /usr/local/arm/2.95.3/arm-linux/lib/
- /usr/local/arm/2.95.3/arm-linux/include/

3) --enable-static 和 --enable-shared

这两个配置选项指定是否生成函数库的静态库和动态库版本。如果不需要生成静态库，则可以使用 --disable-static 配置选项，这样将缩短编译函数库的时间。

利用 configure 脚本所产生的 makefile 文件有几个预先设定的目标可供使用，这里只对其中几个简述如下：

- make all 产生设定的目标。只敲入 make 命令也可以，此时会开始编译源代码，然后连接并产生可执行文件或者函数库。
- make clean 清除之前所编译的目标文件 (*.o)。
- make install 将函数库、头文件等安装到指定的路径中。

2.1.2 非 GNU 环境下的配置

在 MiniGUI 所支持的大部分传统嵌入式操作系统上，用户通常只能使用在 Windows 平台上运行的集成开发环境，比如 Tornado、ADS 等等。因为这些环境提供的开发工具链不是 GNU 兼容的，因此，我们无法使用 2.1.1 中描述的 configure 脚本来自动生成 makefile 以及 mgconfig.h 文件。这种情况下，需要我们自行修改 mgconfig.h 文件来完成对 MiniGUI 的编译时配置。所幸的是，飞漫软件已经为大多数操作系统准备好了可直接使用的 mgconfig.h 文件（保存在 MiniGUI 源代码的 build/ 目录下）；而且也准备好了相应的开发环境工程文件。您可以直接在这些工程环境基础上手工修改 mgconfig.h 文件，并编译 MiniGUI 库。具体信息，可参阅针对特定操作系统的《MiniGUI 移植指南》文档。

2.2 MiniGUI 的编译时配置选项详解

本节将详细给出 MiniGUI 定义的所有编译时配置选项。MiniGUI 有较多的编译配置选项，用户可以根据自己实际需求组合使用这些编译配置选项来生成最满足需求的 MiniGUI 函数库。

在 GNU 开发环境下，大部分的 MiniGUI 配置选项基本上都是基于 `--disable-FEATURE` 和 `--enable-FEATURE` 实现的；但 MiniGUI 配置脚本还提供了 `--with-` 配置选项，这种配置选项可用来从多个配置选项中选择其中一个来指定。比如，MiniGUI 的窗口及控件的外观风格，就是通过 `--with-style` 配置选项指定的。但不论用哪种方式指定配置选项，最终这些配置选项将变成 `mgconfig.h` 中的宏。

下面的小节将分类给出 MiniGUI 的配置选项，具体形式是 `configure` 脚本的选项名称以及对应的 `mgconfig.h` 文件中的宏名称。

2.2.1 操作系统相关的选项和宏

MiniGUI 支持多种操作系统，利用 `--with-osname` 选项，可以在运行 `configure` 脚本时指定要运行 MiniGUI 的操作系统，默认为 `linux`。如果要想 MiniGUI 在 `uClinux` 上运行，可以运行下面的命令：

```
user$ ./configure --with-osname=uclinux
```

指定某个操作系统名称，将在 `mgconfig.h` 中定义相应的宏。对某些操作系统，还将开启其他一些宏。表 2.1 给出了操作系统有关的选项和宏。

表 2.1 操作系统有关的选项及宏

configure 脚本选项	宏	其他应开启的宏	备注
<code>--with-osname=linux</code>	<code>__LINUX__</code>		默认值，用于 Linux 操作系统
<code>--with-osname=uclinux</code>	<code>__uClinux__</code>		用于 uClinux 操作系统
<code>--with-osname=ecos</code>	<code>__ECOS__</code>	<code>__NOUNIX__</code>	用于 eCos 操作系统
<code>--with-osname=ucos2</code>	<code>__UCOSII__</code>	<code>__NOUNIX__</code> <code>_INCORE_RES</code> <code>_USE_OWN_MALLO</code> <code>C</code> <code>_USE_OWN_STDIO</code> <code>_USE_OWN_PTHRE</code> <code>AD</code>	用于 uC/OS-II 操作系统
<code>--with-osname=swlin</code> <code>nux</code>	<code>__WINBOND_SWLINUX__</code> <code>-</code>		用于 SWLinux 操作系统，uClinux 操作系统的变种

--with-osname=vxworks	__VXWORKS__	__NOUNIX__ __USE_OWN_STDIO__ __USE_OWN_PTHREAD ³ __	用于 VxWorks 操作系统
--with-osname=cygwin	__CYGWIN__	__NOUNIX__	用于 cygwin 环境
--with-osname=threadx	WIN32	__NOUNIX__	用于 Win32 平台
--with-osname=darwin	__DARWIN__	__NOUNIX__	用于 MacOSX 操作系统
--with-osname=threadx	__THREADX__	__NOUNIX__ __INCORE_RES__ __USE_OWN_MALLOC__ __USE_OWN_STDIO__ __USE_OWN_PTHREAD__	用于 ThreadX 操作系统
--with-osname=nucleus	__NUCLEUS__	__NOUNIX__ __INCORE_RES__ __USE_OWN_MALLOC__ __USE_OWN_STDIO__ __USE_OWN_PTHREAD__	用于 Nucleus 操作系统
--with-osname=ose	__OSE__	__NOUNIX__ __INCORE_RES__ __USE_OWN_PTHREAD__	用于 OSE 操作系统

因为 MiniGUI 增值版是按照操作系统划分的，因此，针对某个操作系统的 MiniGUI 增值版产品是无法运行在其他操作系统上的。在手工修改配置时，为了您的增值版产品能够在对应的操作系统上运行，请确保正确定义上面的宏。

2.2.2 目标板相关的选项和宏

MiniGUI 中某些代码和具体的目标板相关；如果要在这些目标板上正确运行 MiniGUI，需要指定这些开发板的名称。在运行 **configure** 脚本时，通过 **--with-targetname** 选项，可以指定具体的目标板名称，默认为 **unknown**。目标板相关的选项通常在 MiniGUI 使用 **Shadow** 图形引擎或者 **CommLCD** 图形引擎时，用来指定这两个图形引擎的子驱动程序，也就是说，在使用这两个引擎时，通过目标板的名称来确定包含哪个子驱动程序。表 2.2 给出了目标板相关的选项和宏。

表 2.2 目标板相关的选项及宏

configure 脚本选项	宏	备注
--with-targetname=stb810	__TARGET_STB810__	用于运行 Linux 的 Philips STB810 开发板
--with-targetname=vfanvil	__TARGET_VFANVIL__	用于运行 ThreadX 的 VisualFone 开发板

³ 针对 VxWorks 5.5 以下版本。

--with-targetname=vxi386	__TARGET_VXI386__	用于使用 VxWorks 的 i386 目标机
--with-targetname=qvfb	__TARGET_QVFB__	包含 Linux 下 Shadow 引擎的 qvfb 子驱动
--with-targetname=vvfb	__TARGET_WVFB__	包含 Windows 下 Shadow 引擎的 vvfb 子驱动
--with-targetname=fbcon	__TARGET_FBCON__	包含 Linux 下 Shadow 引擎的 fbcon 子驱动
--with-targetname=mx21	__TARGET_MX21__	用于运行 OSE 的 MX21 开发板
--with-targetname=unknown	__TARGET_UNKNOWN__	未知开发板；默认值

2.2.3 运行模式相关的选项和宏

我们可以将 MiniGUI 配置成三种运行模式⁴之一：多进程的 MiniGUI-Processes 运行模式，多线程模式的 MiniGUI-Threads 运行模式，以及非多进程也非多线程的 MiniGUI-Standalone 运行模式。MiniGUI 的默认配置选项是 MiniGUI-Threads 运行模式。表 2.3 给出了运行模式相关的配置选项和宏。

表 2.3 运行模式相关的配置选项和宏

configure 脚本选项	宏	备注	默认
不指定	_MGRM_THREADS	MiniGUI-Threads 运行模式	
procs	_MGRM_PROCESSES _LITE_VERSION	MiniGUI-Processes 运行模式，仅用于 Linux/uClinux 操作系统	关闭
standalone	_MGRM_STANDALONE _LITE_VERSION _STAND_ALONE	MiniGUI-Standalone 运行模式，仅用于 Linux/uClinux 操作系统	关闭

MiniGUI-VAR for Linux/uClinux V1.6.9 产品不提供 MiniGUI-Processes 运行模式，而提供 MiniGUI-Lite 运行模式。表 2.4 给出了 MiniGUI-VAR for Linux/uClinux V1.6.9 产品的运行模式相关配置选项和宏。

表 2.4 MiniGUI-VAR for Linux/uClinux V1.6.9 产品的运行模式相关配置选项和宏

configure 脚本选项	宏	备注	默认
不指定		MiniGUI-Threads 运行模式	
lite	_LITE_VERSION	MiniGUI-Lite 运行模式，仅用于 Linux/uClinux 操作系统	关闭
standalone	_LITE_VERSION _STAND_ALONE	MiniGUI-Standalone 运行模式，仅用于 Linux/uClinux 操作系统。需要同时打开 lite 选项。	关闭

⁴ 有关运行模式以及各增值版产品支持的运行模式，请参阅《MiniGUI 技术白皮书》V2.0-3。

2.2.4 图形引擎相关的选项和宏

MiniGUI 支持众多的图形引擎。常用图形引擎主要包括 Dummy 图形引擎、Qt Virtual FrameBuffer 引擎、Linux FrameBuffer 控制台图形引擎、COMMLCD 图形引擎、Shadow 图形引擎、Windows Virtual FrameBuffer 图形引擎等。通过配置选项或者宏，我们可以将某个图形引擎包含到 MiniGUI 中。但如果要指定 MiniGUI 使用某个图形引擎，则需要指定特定的运行时配置选项。比如，要指定 MiniGUI 使用 dummy 图形引擎，可以指定 [system] 段中的运行时配置选项 gal_engine=dummy，该选项等号右侧就是图形引擎的具体名称。注意引擎名称是大小写敏感的。关于如何修改运行时配置选项的信息，请参阅本手册第 3 章“MiniGUI 的运行时配置选项”。表 2.5 给出了图形引擎相关的选项、宏以及名称。

表 2.5 图形引擎相关选项和宏

configure 脚本选项	宏	引擎名称	备注	默认
Videodummy	_NEWGAL_ENGINE_DUMMY	dummy	所有操作系统	开启
videofbcon	_NEWGAL_ENGINE_FBCON	fbcon	Linux/uClinux	开启
videoqxfb	_NEWGAL_ENGINE_QVFB	qxfb	Linux	开启
videowvfb	_NEWGAL_ENGINE_WVFB	wvfb	Win32	关闭
videocommlcd	_NEWGAL_ENGINE_COMMLCD	commlcd	所有操作系统	关闭
videoshadow	_NEWGAL_ENGINE_SHADOW	shadow	所有操作系统，MiniGUI-Threads、MiniGUI-Standalone 运行模式	关闭
videoem85xxosd	_NEWGAL_ENGINE_EM85XXOSD	em85xxosd	EM85xx 开发板的 OSD 图形引擎；uClinux	关闭
videosvpxxosd	_NEWGAL_ENGINE_SVPXXOSD	svpxxosd	uClinux	关闭
videobf533	_NEWGAL_ENGINE_BF533	bf533	uClinux	关闭
videomb93493	_NEWGAL_ENGINE_MB93493	mb93493	uClinux	关闭
videoutpmc	_NEWGAL_ENGINE_UTPMC	utpmc	uClinux	关闭
videodfb	_NEWGAL_ENGINE_DFB	dfb	将 MiniGUI 运行在 DirectFB 之上；Linux	关闭

Dummy 图形引擎（“哑”图形引擎），不作任何实际输出的图形引擎。因此，在针对您的开发板的图形引擎尚不可用时，可以使用该图形引擎来运行 MiniGUI。

Qxfb 图形引擎用于 Linux 操作系统。利用 qxfb，我们可以在 X Window 的窗口中运行 MiniGUI 程序，这样可大大方便应用程序的调试。和 qxfb 图形引擎类似，使用 MiniGUI SDK for Win32 在 Win32 平台上运行 MiniGUI 时，实际运行在 Windows Virtual FrameBuffer 程序上，使用的是 wvfb 图形引擎。

MiniGUI 中还存在一个特殊的图形引擎 Shadow 图形引擎，利用 Shadow 图形引擎，可以实现对低于 8 位色的显示设备的支持，也可以实现屏幕旋转功能。Shadow 图形引擎使用了子驱动程序的概念，通过目标板的名称来确定包含哪个子驱动程序。同一时刻，只能包含一个子驱动程序，由目标板

配置选项确定（见 2.2.2 小节）。注意，Shadow 图形引擎默认是关闭的，而且目前只适用于 MiniGUI-Threads 模式和 MiniGUI-Standalone 模式。

MiniGUI 中针对 Shadow 图形引擎的子驱动程序有（相关代码在 MiniGUI 源代码的 src/newgal/shadow 目录中）：

- unknown：采用默认的子驱动程序，其功能类似 dummy 图形引擎。用户可修改该子驱动程序，用来实现对底层图形设备的操作和访问。
- vfanvil：针对 VisualFone Anvil 板子的子驱动程序，用于 ThreadX 操作系统。
- qvfb：针对 Linux QVFB 各种显示模式的子驱动程序，同时可支持低于 8 位色的显示模式以及屏幕旋转。
- fbcon：针对 Linux console FrameBuffer 的子驱动程序，同时可支持低于 8 位色的显示模式以及屏幕旋转。
- wvfb：针对 Windows Virtual FrameBuffer(wvfb) 的子驱动程序，同时可支持低于 8 位色的显示模式以及屏幕旋转。

在 Shadow 引擎上，我们可以实现屏幕旋转功能。表 2.6 给出了屏幕旋转功能的相关选项和宏。

表 2.6 屏幕旋转功能相关的选项和宏

configure 脚本选项	宏		宏的定义值	注释	默认值
coortrans_cw	_COOR_TRAN_S	_ROT_DIR_CW	1	顺时针旋转屏幕	关闭
coortrans_ccw		_ROT_DIR_CW	0	逆时针旋转屏幕	关闭

CommLCD 图形引擎是传统嵌入式操作系统上运行 MiniGUI 时使用得最多的图形引擎。CommLCD 也采用类似 Shadow 图形引擎那样的子驱动程序结构。目前，针对 CommLCD 图形引擎的子驱动程序有：

- vxi386：针对 VxWorks i386 目标板的子驱动程序。
- unknown：如果是 eCos 操作系统，则采用 eCos 的标准接口实现了子驱动程序。否则，需要由用户自己定义子驱动程序。在 MiniGUI 源代码的 include/mgdrv-ucosii.c 中，包含了一个针对 uC/OS-II 操作系统的默认实现，相当于 dummy 图形引擎。您可以修改这个文件来支持自己的 LCD 控制器。

2.2.5 输入引擎相关的选项和宏

MiniGUI 为众多开发板提供有直接可用的输入引擎。常用输入引擎主要包括 Dummy 输入引擎、Qt Virtual FrameBuffer 引擎、Linux FrameBuffer 控制台输入引擎、COMM 输入引擎、Random 输入引擎、Windows Virtual FrameBuffer 图形引擎等。通过配置选项或者宏，我们可以将某个输入引擎包

含到 MiniGUI 中。但如果要指定 MiniGUI 使用某个输入引擎，则需要指定特定的运行时配置选项。比如，要指定 MiniGUI 使用 dummy 输入引擎，可以指定 [system] 段中的运行时配置选项 ial_engine=dummy，该选项等号右侧就是输入引擎的具体名称。注意，输入引擎的名称是大小写敏感的。关于如何修改运行时配置选项的信息，请参阅本手册第 3 章“MiniGUI 的运行时配置选项”。表 2.7 给出了输入引擎相关的选项和宏。

表 2.7 输入引擎相关的选项和宏

configure 脚本选项	宏	引擎名称	注释	默认值
dummyial	_DUMMY_IAL	dummy	Dummy 输入引擎；所有操作系统	开启
autoial	_AUTO_IAL	auto	Automatic 输入引擎；所有操作系统	关闭
nativeial	_NATIVE_IAL_ENGINE	console	Linux 控制台输入引擎；Linux	开启
qvfbial	_QVFB_IAL	qvfb	QVFB 输入引擎；Linux，配合 QVFB 图形引擎	开启
randomial	_RANDOM_IAL	random	Random 输入引擎；所有操作系统	关闭
wvfbial	_WVFB_IAL	wvfb	WVFB 输入引擎；Win32，配合 WVFB 图形引擎	关闭
commial	_COMM_IAL	comm.	COMM 输入引擎；所有操作系统	关闭
dfbial	_DFB_IAL	dfb	基于 DirectFB 的输入引擎；Linux，配合 DFB 图形引擎	关闭
ep7211ial	_EP7211_IAL	EP7211	为 EP7211 板构造输入引擎	关闭
adsial	_ADS_IAL	ADS	为 ADS Graphics Client 板构造输入引擎	关闭
ipaqial	_IPAQ_IAL	ipaq	为 iPAQ H3600 构造输入引擎	关闭
l7200ial	_L7200_IAL	l7200	为 L7200 构造输入引擎	关闭
arm3000ial	_ARM3000_IAL	arm3000	为 ARM3000 构造输入引擎	关闭
dm270ial	_DM270_IAL	dm270	为 DM270 构造输入引擎	关闭
evmv10ial	_EVMV10_IAL	evmv10	为 xScale EVMV 1.0 构造输入引擎	关闭
fxrm9200ial	_FXRM9200_IAL	fxrm9200	为 FXRM9200 构造输入引擎	关闭
abssigial	_ABSSIG_IAL	abssig	为 ABSSIG 构造输入引擎	关闭
embest2410ial	_EMBEST2410_IAL	embest2410	为 EMBEST ARM2410 构造输入引擎	关闭
embest44b0ial	_EMBEST44B0_IAL	embest44b0	为 EMBEST ARM44B0 构造输入引擎	关闭
fft7202ial	_FFT7202_IAL	fft7200	为 FFT7202 构造输入引擎	关闭
palm2ial	_PALMII_IAL	palm2	为 OKWAP Palm2 构造输入引擎	关闭
mpc823ial	_MPC823_IAL	MPC823	为 mpc823 构造输入引擎	关闭

ucb1x00ial	_UCB1X00_IAL	UCB1X00	为 ucb1x00 构造输入引擎	关闭
px255bial	_PX255B_IAL	PX255B	为 px255b 构造输入引擎	关闭
vr4181ial	_VR4181_IAL	VR4181	为 NEC VR4181 调试板构造输入引擎	关闭
helioial	_HELIO_IAL	Helio	为 Helio Touch Panel 构造输入引擎	关闭
tfstbial	_TFSTB_IAL	TF-STB	为 Tongfang STB 构造输入引擎	关闭
t800ial	_T800_IAL	T800	为 MT T800 构造输入引擎	关闭
mc68x328ial	_MC68X328_IAL	MC68X328	为 palm/mc68ez328 uClinux 触摸屏构造输入引擎	关闭
smdk2410ial	_SMDK2410_IAL	SMDK2410	为 SMDK2410 触摸屏构造输入引擎	关闭
hh2410r3ial	_HH2410R3_IAL	hh2410r3	为 HHARM2410R3 触摸屏构造输入引擎	关闭
hh2440ial	_HH2440_IAL	hh2440	为 HHARM2440 触摸屏构造输入引擎	关闭
skyeleep7312ial	_SKYEYE_EP7312_IAL	SkyEyeEP7312	为 SkyEye EP7312 模拟器触摸屏构造输入引擎	关闭
dmgstbial	_DMGSTB_IAL	dmg-stb	为 Intel DMG STB 摇控器构造输入引擎	关闭
fipial	_FIP_IAL	fip	为 EM85xx 前面板和摇控器构造输入引擎	关闭
svpxxial	_SVPXX_IAL	svpxx	为 WinBond SVPXX 构造输入引擎	关闭
ads7846ial	_ADS7846_IAL	ads7846	为 Arca Tpanel Ads7846 构造输入引擎	关闭
em86ial	_EM86_IAL	em86	为 Em86XX 开发板构造输入引擎	关闭
em85ial	_EM85_IAL	em85	为 Em85XX 开发板构造输入引擎	关闭
hhkdbirial	_HH5249KBDIR_IAL	hh5249kbidir	为 HHCF5249 PS/2 键盘和 IrDA 构造输入引擎	关闭
utpmcial	_UTPMC_IAL	utpmc	为 UTPMC 构造输入引擎	关闭

Dummy 输入引擎（哑输入引擎），不和任何实际的输入设备关联，因此也不会产生任何输入。在还没有编写好针对特定开发板的输入引擎时，您可以使用该输入引擎来运行 MiniGUI。注意，当 MiniGUI 根据运行时配置选项中的设定找不到匹配的输入引擎时，会自动选择使用 Dummy 输入引擎。

类似 Dummy 输入引擎，MiniGUI 还提供了其他不和实际输入设备关联的软件输入引擎，比如 Auto 输入引擎和 Random 输入引擎。Auto 引擎可按事先设定自动循环产生输入事件；而 Random 输入引擎则产生随机输入事件。这两个引擎可以用来完成 MiniGUI 及其应用程序的自动测试。

Console 输入引擎是针对 Linux 操作系统的 PC 控制台编写的。该输入引擎可以支持标准 PC 键盘以及多种鼠标协议。在使用 console 输入引擎时，您需要通过运行时配置选项 [system] 段中的 mtype

和 `mdev` 来指定鼠标协议类型以及鼠标的设备。

Console 输入引擎支持的鼠标协议以及相关的选项和宏，可见表 2.8。注意，尽管 MiniGUI 支持智能鼠标，但 MiniGUI 不支持鼠标中键以及滚轮的输入事件。

表 2.8 鼠标协议相关的选项和宏

configure 脚本选项	宏	注释	默认值
<code>nativeps2</code>	<code>_PS2_SUPPORT</code>	支持 PS2 鼠标协议	开启
<code>nativeimps2</code>	<code>_IMPS2_SUPPORT</code>	支持智能鼠标 (IMPS/2) 协议	开启
<code>nativevms</code>	<code>_MS_SUPPORT</code>	支持旧的 MS 串口鼠标	开启
<code>nativevms3</code>	<code>_MS3_SUPPORT</code>	支持 MS3 鼠标协议	开启
<code>nativegpm</code>	<code>_GPM_SUPPORT</code>	支持 GPM 守护进程	开启

除上述选项之外，MiniGUI 还为应用程序提供了鼠标或者触摸屏的校正接口。如果要使用该接口，需要打开触摸屏校正的选项。表 2.9 给出了触摸屏校正相关的选项和宏。

表 2.9 鼠标及触摸屏校正相关的选项和宏

configure 脚本选项	宏	注释	默认值
<code>mousecalibrate</code>	<code>_MISC_MOUSECALIBRATE</code>	支持触摸屏校正	开启

2.2.6 键盘布局的相关选项和宏

MiniGUI 的键盘布局用来控制 `TranslateMessage` 函数的行为。不同的键盘布局会将相同的按键（以扫描码区分）翻译为不同的字符，该翻译过程是通过查询扫描码映射表实现的。目前，MiniGUI 中包含有对西欧国家常用的键盘布局的支持，默认的是标准的美式 101/102 键盘。在程序中，如果要指定使用不同的键盘布局，应该使用键盘布局的名称来调用 `SetKeyboardLayout` 函数。具体用法，请参阅《MiniGUI 编程指南》V2.0-3。表 2.10 给出了键盘布局的相关选项、宏和名称。

表 2.10 键盘布局的相关选项和宏

配置选项	宏	键盘布局名称	注释	默认值
<code>Kbdfrpc</code>	<code>_KBD_LAYOUT_FRPC</code>	<code>frpc</code>	包含对 French PC (non-US102 keys) 键盘布局的支持	关闭
<code>Kbdfr</code>	<code>_KBD_LAYOUT_FR</code>	<code>fr</code>	包含对支持法语键盘布局的支持	关闭
<code>Kbdde</code>	<code>_KBD_LAYOUT_DE</code>	<code>de</code>	包含对支持德语键盘布局的支持	关闭
<code>kbddelatin1</code>	<code>_KBD_LAYOUT_DELATIN1</code>	<code>delatin1</code>	包含对支持德语 Latin1 键盘布局的支持	关闭
<code>Kbdit</code>	<code>_KBD_LAYOUT_IT</code>	<code>it</code>	包含对支持意大利语键	关闭

			盘布局的支持	
Kbdes	_KBD_LAYOUT_ES	es	包含对支持西班牙语键盘布局的支持	关闭
kbdescp850	_KBD_LAYOUT_ESCP850	escp850	包含对支持西班牙语 CP850 键盘布局的支持	关闭

2.2.7 系统全局配置选项和宏

MiniGUI 的系统全局配置选项及其对应的宏可见表 2.11。

表 2.11 系统全局配置选项和宏

configure 脚本选项	宏	注释	默认值
incoreres	_INCORE_RES	使用 MiniGUI 的内建资源	关闭
miniguientry	_USE_MINIGUIENTRY	使用 MiniGUI 的 minigui_entry 函数	关闭
debug	_DEBUG	包含调试信息	关闭
tracemsg	_TRACE_MSG	追踪 MiniGUI 的消息	关闭
msgstr	_MSG_STRING	包含消息的字符串名	关闭
micemoveable	_MOVE_WINDOW_BY_MOUSE	可以使用鼠标移动窗口	开启
dblclk	_DOUBLE_CLICK	可以支持鼠标双击	开启
cursor	_CURSOR_SUPPORT	包含对鼠标光标的支持	开启
clipboard	_CLIPBOARD_SUPPORT	包含对剪贴板的支持	开启
textmode	_HAVE_TEXT_MODE	Linux 系统具有基于 FrameBuffer 的控制台	开启
savebitmap	_SAVE_BITMAP	支持 SaveBitmap 的相关函数	开启
aboutdlg	_MISC_ABOUTDLG	包含 About 对话框	开启
savescreen	_MISC_SAVESCREEN	支持屏幕截图	开启
libvcongui	_LIB_VCONGUI	支持 Linux 的虚拟控制台	开启
fixedmath	_FIXED_MATH	使用定点数学函数	开启
adv2dapi	_ADV_2DAPI	支持高级 2D 图形 API	开启

下面介绍其中的一些重要配置选项。

incoreres 选项用来控制是否将 MiniGUI 所需要的字体、位图、光标、图标等内建到函数库中。该选项对传统嵌入式操作系统非常有用。因为在大多数情况下，传统嵌入式操作系统没有文件系统的支持，有了内建资源支持，就可以将上述资源内建到函数库中，从而无需文件系统即可运行 MiniGUI。注意，当使用内建资源时，MiniGUI 的运行时配置选项也将直接编译到函数库中。

miniguientry 选项用来控制 MiniGUIMain 函数的实现方式。默认情况下（即关闭该选项时），MiniGUIMain 函数将展开成 main 函数，这样，应用程序就不必自行定义 main 函数。当开启 miniguientry 选项时，将把 MiniGUIMain 函数展开成 minigui_entry 函数。在某些传统嵌入式操作系统上，这样做可以方便调试以及系统的集成。

`fixedmath` 选项用来控制是否在 MiniGUI 函数库中包含定点数学函数，比如 `fixcos` 等。
`clipboard` 选项控制是否在 MiniGUI 函数库中包含对剪切板的支持；如果关闭，则编辑框不能实现剪切和复制功能。`adv2api` 选项用来控制是否在 MiniGUI 函数库中包含高级二维图形函数接口。

`debug`、`tracemsg`、`msgstr` 等选项用于 MiniGUI 的调试，不建议用户使用。

MiniGUI 对鼠标光标的支持默认是开启的。当目标系统没有鼠标、触摸屏等定点设备，则不需要显示鼠标光标，这时，就可以关闭 `cursor` 配置选项从而取消对鼠标光标的支持。

在 MiniGUI 中使用鼠标来移动窗口默认是开启的而在多数的嵌入式系统上，通常不使用层叠式的多窗口用户界面，也不需要移动窗口，这时，就可以关闭 `micemoveable` 配置选项。

2.2.8 输入法相关的选项和宏

MiniGUI 中内建有 GB2312 输入法与 GB2312 智能拼音输入法，这两个输入法的配置选项默认是开启的。表 2.12 给出了输入法配置的相关选项和宏。

表 2.12 输入法配置选项和宏

configure 配置选项	宏	注释	默认值
<code>lme gb2312</code>	<code>_IME_GB2312</code>	包含对 GB2312 输入法的支持	开启
<code>lme gb2312py</code>	<code>_IME_GB2312_PINYIN</code>	包含对 GB2312 智能拼音输入法的支持	开启

2.2.9 字符集和字体相关的选项和宏

MiniGUI 对字体的支持十分丰富。MiniGUI 支持 RBF 字体和 VBF 字体（这是 MiniGUI 定义的两点阵字体格式）、QPF 字体、TrueType 字体以及 Adobe Type1 字体等。由于 MiniGUI 支持多种字体，相关的配置选项也就相对较多，配置也很灵活。

和字体类型一样，MiniGUI 也为各种字符集提供了良好的支持。对特定字符集的支持也是可以灵活配置的。表 2.13 给出了字符集和字体相关的选项和宏。

表 2.13 字符集及字体相关配置选项和宏

configure 配置选项	宏	注释	默认值
<code>latin2support</code>	<code>_LATIN2_SUPPORT</code>	包含对 East European (Latin2, ISO-8859-2) 字符集的支持	关闭
<code>latin3support</code>	<code>_LATIN3_SUPPORT</code>	包含对 South European (Latin3, ISO-8859-3) 字符集的支持	关闭
<code>latin4support</code>	<code>_LATIN4_SUPPORT</code>	包含对 North European (Latin4, ISO-8859-4) 字符集的支持	关闭

		ISO-8859-4)字符集的支持	
cyrillicsupport	_CYRILLIC_SUPPORT	包含对 Cyrillic (ISO-8859-5) 字符集的支持	关闭
arabicsupport	_ARABIC_SUPPORT	包含对 Arabic (ISO-8859-6) 字符集的支持	关闭
greekssupport	_GREEK_SUPPORT	包含对 Greek (ISO-8859-7) 字符集的支持	关闭
hebrewsupport	_HEBREW_SUPPORT	包含对 Hebrew (ISO-8859-8) 字符集的支持	关闭
latin5support	_LATIN5_SUPPORT	包含对 Turkish (Latin5, ISO-8859-9) 字符集的支持	关闭
latin6support	_LATIN6_SUPPORT	包含对 Nordic (Latin6, ISO-8859-10) 字符集的支持	关闭
thaisupport	_THAI_SUPPORT	包含对 Thai (ISO-8859-11) 字符集的支持	关闭
latin7support	_LATIN7_SUPPORT	包含对 Latin7 (ISO-8859-13) 字符集的支持	关闭
latin8support	_LATIN8_SUPPORT	包含对 Latin8 (ISO-8859-14) 字符集的支持	关闭
latin9support	_LATIN9_SUPPORT	包含对 Latin 9 (ISO-8859-15, West Extended) 字符集的支持	关闭
latin10support	_LATIN10_SUPPORT	包含对 Latin 10 (ISO-8859-16, Romanian) 字符集的支持	关闭
gbssupport	_GB_SUPPORT	包含对 EUC 编码的 GB2312 字符集的支持	开启
gbksupport	_GBK_SUPPORT	包含对 GBK 字符集的支持	开启
gb18030support	_GB18030_SUPPORT	包含对 GB18030-0 字符集的支持	关闭
big5support	_BIG5_SUPPORT	包含对 BIG5 字符集的支持	开启
euckrsupport	_EUCKR_SUPPORT	包含对 EUC 编码的 KSC5636 和 KSC5601 字符集的支持	关闭
eucjpsupport	_EUCJP_SUPPORT	包含对 EUC 编码的 JISX0201 和 JISX0208 字符集的支持	关闭
shiftjissupport	_SHIFTJIS_SUPPORT	包含对 Shift-JIS 编码的 JISX0201 和 JISX0208 字符集的支持	关闭
unicodesupport	_UNICODE_SUPPORT	包含对其他字符集到 UNICODE 的转换码表以及对 UTF-8 编码的支持	开启
rbfsupport	_RBF_SUPPORT	包含对 RBF 字体格式的支持	开启
rbf16	_INCORERBF_LATIN1_16	包含 RBF ISO8859-1 8x16 fixed 内建字体	关闭
rbf24	_INCORERBF_LATIN1_24	包含 RBF ISO8859-1 12x24 fixed 内建字体	关闭
rbfgb12	_INCORERBF_GB12	包含 RBF GB2312 12x12 fixed/song 内建字体	开启
rbfgb16	_INCORERBF_GB16	包含 RBF GB2312 16x16 fixed/song 内建字体	关闭
rbfgb24	_INCORERBF_GB24	包含 RBF GB2312 24x24 fixed/song 内建字体	关闭
rbfgb16hei	_INCORERBF_GB16_H	包含 RBF GB2312 EI16x16	关闭

		fixed/hei 内建字体	
rbfbig12	_INCORERBF_BIG5_12	包含 RBF BIG512x12 fixed/ming 内建字体	关闭
rbfbig16	_INCORERBF_BIG5_16	包含 RBF BIG5 16x16 fixed/ming 内建字体	关闭
rbfbig24	_INCORERBF_BIG5_24	包含 RBF BIG5 24x24 fixed/ming 内建字体	关闭
rbfkj12	_INCORERBF_KJ12	包含 RBF SHIFT-JIS Kanji 12 内建字体	关闭
rbfkj14	_INCORERBF_KJ14	包含 RBF SHIFT-JIS Kanji 14 内建字体	关闭
rbfkj16	_INCORERBF_KJ16	包含 RBF SHIFT-JIS Kanji 16 内建字体	关闭
vbfsupport	_VBF_SUPPORT	包含对 VBF 字体的支持	开启
fontsserif	_INCOREFONT_SANSSERIF	包含 sansserif 内建 VBF 字体	开启
fontcourier	_INCOREFONT_COURIER	包含 courier 内建 VBF 字体	开启
fontsymbol	_INCOREFONT_SYMBOL	包含 symbol 内建 VBF 字体	关闭
fontvgas	_INCOREFONT_VGAS	包含 VGA 内建 VBF 字体	关闭
fonthelv	_INCOREFONT_HELV	包含 Helvetica 内建 VBF 字体	关闭
qpfsupport	_QPF_SUPPORT	包含对 Qt Prerendered Font(QPF) 字体的支持	开启
ttfsupport	_TTF_SUPPORT	包含对 TrueType 字体的支持	关闭
ttfcache	_TTF_CACHE_SUPPORT	包含对 TrueType 缓存的支持	关闭
type1support	_TYPE1_SUPPORT	支持 Adobe Type1 字体	开启

latin2support、latin3support、cyrillicsupport、arabicsupport、greekssupport、hebrewsupport、latin5support、latin6support、thaisupport、latin7support、latin8support、latin9support、latin10support 等配置选项，用来控制对 ISO8859-2 到 ISO8859-16 字符集的支持，这些字符集均是单字节字符集。MiniGUI 内建包括有对 ASCII 字符集和 ISO8859-1 (Latin1) 字符集的支持，这两种字符集没有对应的配置选项。

gbssupport、gbkssupport、gb18030support、big5support、euckrssupport、eucjpsupport、shiftjissupport、unicodesupport 等编译配置选项，分别用来控制对 GB2312, GBK, GB18030, BIG5, EUCKR、EUCJP、SHIFTJIS、UNICODE 等多字节字符集/编码系统的支持。

rbfsupport 配置选项控制是否包含对 Raw Bitmap Font (RBF) 字体的支持，默认是开启的。因为 RBF 是默认的字体格式，因此，不建议用户关闭对该字体类型的支持。

rbfgbl12、rbfgbl16、rbfgbl24 和 rbfgbl16hei 等配置选项，可控制是否在 MiniGUI 库中内建 GB2312 的 12、16、24 及 16 黑体点阵字体。这些编译配置选项默认是关闭的，并且这些编译配置选项仅在 incoreres 配置选项被开启时才有效。

`rbfbig12`、`rbfbig16` 及 `rbfbig24` 等配置选项⁵，可控制是否在 MiniGUI 库中内建 BIG5 的 12、16 及 24 点阵字体。这些配置选项默认是关闭的，而且仅在 `incoreres` 配置选项被开启时才有效。

`rbfkj12`、`rbfkj14` 以及 `rbfkj16` 等配置选项⁶，可控制是否在 MiniGUI 库中内建日文 12、14 及 16 点阵字体。这些配置选项默认是关闭的，而且仅在 `incoreres` 选项被开启时才有效。

`vbfsupport` 配置选项控制是否包含对 Variable Bitmap Font (VBF) 字体的支持，默认是开启的。如果关闭该编译配置选项不仅可关闭 MiniGUI 对 VBF 字体的支持，也可关闭在 MiniGUI 库中包含内建的 VBF 字体的支持。MiniGUI 启动时，也将忽略运行时配置选项的 `[varbitmapfonts]` 段。

`fontsserif` 配置选项以及 `fontcourier`、`fontsymbol`、`fonthelv`、`fontvgas` 编译配置选项可控制是否在 MiniGUI 库中内建 SanSerif、Courier、Symbol、Helvetica 以及若干 VGA 字体。注意这些字体是 VBF 格式的，且不受 `incoreres` 选项的影响。

`qpfsupport` 配置选项控制是否在 MiniGUI 库中包含对 Qt/Embedded Prerendered Font (QPF) 字体的支持。因为 QPF 字体使用 UNICODE 编码，因此，允许对 QPF 字体的支持，将自动开启 MiniGUI 的 UNICODE 支持。如果同时开启有 `incoreres` 选项，则 MiniGUI 会在函数库中内建一些 QPF 字体。

`typesupport` 配置选项控制是否在 MiniGUI 库中包含对 Adobe Type1 字体的支持。MiniGUI 使用 `libt1` 库来提供对 Type1 字体的渲染。如果系统中没有安装 `t1` 库，则配置脚本会自动关闭该编译配置选项。

`ttfsupport` 配置选项用来控制是否在 MiniGUI 库中包含对 TrueType 字体的支持。MiniGUI 使用 FreeType 版本 1.3.0 来提供对 TrueType 字体的渲染。如果您的系统中没有安装 FreeType 1.3.0，配置脚本将自动关闭该选项。需要注意的是，FreeType 2 的接口和 FreeType 1 不兼容，所以，MiniGUI 不能在只有 FreeType 2 的系统上正常编译。

`ttfcache` 配置选项控制是否支持对 TrueType 字体的缓存功能，默认为关闭。当要开启 `ttfcache` 时，必须首先开启 `ttfsupport` 选项。

`--with-mttfcachenum` 配置选项用来指定在开启 TrueType 字体缓存时要使用的缓存块数量，默认值为 10。

`--with-ttfcachesize` 编译配置选项用来指定在开启 TrueType 缓存时所要使用的每各缓存块的大小，默认值为 64KB。

⁵ 标准增值版中不包含内建的 BIG5 字体。

⁶ 标准增值版中不包含内建的日文字体。

表 2.14 和表 2.15 给出了 TrueType 字体缓存功能的相关参数以及配置选项和宏。

表 2.14 TrueType 字体缓存相关的配置选项和宏

configure 配置选项	宏	宏的定义值	备注
--with-mttfcachenum=10	_MAX_TTF_CACHE	10	默认值
--with-mttfcachenum=20		20	
--with-mttfcachenum=40		40	

表 2.15 TrueType 字体缓存相关的配置选项和宏

configure 配置选项	宏	宏的定义值	备注
--with-ttfcachesize=64	_TTF_CACHE_SIZE	64	默认值
--with-ttfcachesize=128		128	
--with-ttfcachesize=256		256	
--with-ttfcachesize=512		512	
--with-ttfcachesize=1024		1024	

2.2.10 图像文件格式相关的选项和宏

MiniGUI 可以支持多种图像文件格式，具体包括 Windows BMP、GIF、JPEG、PNG、PCX、LBM/PBM、TGA 等。其中，对 Windows BMP 的支持是内建的，所以没有对应的配置选项；GIF、JPEG、PNG 文件的配置选项默认是开启的；而 PCX、LBM/PBM、TGA 的支持默认是关闭的。还需要注意，对 JPEG 和 PNG 图片格式的支持，需要系统中安装有对应版本的 libjpeg 和 libpng 库，这两个函数库的源代码，包含在 MiniGUI 增值版产品光盘的 deplibs/ 目录下。

表 2.16 给出了图像文件格式相关的配置选项和宏。

表 2.16 图像文件格式相关的配置选项和宏

configure 配置选项	宏	注释	默认值
gifsupport	_GIF_FILE_SUPPORT	包含对 GIF 文件的支持	开启
jpgsupport	_JPG_FILE_SUPPORT	包含对 JPG 文件的支持	开启
pngsupport	_PNG_FILE_SUPPORT	包含对 PNG 文件的支持	开启
pcxsupport	_PCX_FILE_SUPPORT	包含对 PCX 文件的支持	关闭
lbmsupport	_LBM_FILE_SUPPORT	包含对 LBM/PBM 文件的支持	关闭
tgasupport	_TGA_FILE_SUPPORT	包含对 TGA 文件的支持	关闭

2.2.11 外观风格相关的选项和宏

MiniGUI 定义了三种窗口和控件的外观风格：Classic、Flat 和 Fashion。表 2.17 给出了控制外观风格的配置选项以及对应的宏。

表 2.17 外观风格相关的配置选项和宏

configure 配置选项	宏	注释	备注
--with-style=classic	_PC3D_WINDOW_STYLE	经典风格	默认值
--with-style=flat	_GRAY_SCREEN _FLAT_WINDOW_STYLE	适于工控领域，灰度屏显示	
--with-style=fashion	_PHONE_WINDOW_STYLE	流行风格	
tinyscreen	_TINY_SCREEN	目标板是小尺寸屏幕	关闭
fixedbarlen	_USE_FIXED_SB_BARLEN	滚动条具有确定的长度	关闭

表 2.17 还给出了两个选项：tinyscreen 和 fixedbarlen。前者用来告诉 MiniGUI，目标板的屏幕分辨率较低，这样，MiniGUI 在显示系统对话框时，将尽量紧凑显示其中的内容。后者告诉 MiniGUI，窗口或者控件的滚动条具有固定的长度，不随可滚动范围的大小而变化。

2.2.12 控件相关的选项和宏

MiniGUI 的控件可以分为基本控件和扩展库中的控件。MiniGUI 提供了针对所有控件的配置选项。大部分控件的配置选项默认都是开启的，但您可以将某些控件排除掉，从而减少 MiniGUI 核心库和/或扩展库的尺寸。

除了控件之外，extskin 配置选项还可以用来控制是否在 MiniGUI 扩展库中包含对皮肤界面的支持，该配置选项默认是开启的。

表 2.18 给出了所有控件相关的配置选项和宏。

表 2.18 控件相关的配置选项和宏

configure 配置选项	宏	注释	默认值
ctrlstatic	_CTRL_STATIC	包含 STATIC 控件	开启
ctrlbutton	_CTRL_BUTTON	包含 BUTTON 控件	开启
ctrlsimit	_CTRL_SIMEDIT	包含 Simple EDIT 控件	开启
ctrlsedit	_CTRL_SLEDIT	包含 Single-Line EDIT 控件	开启
ctrllistbox	_CTRL_LISTBOX	包含 LISTBOX 控件	开启
ctrlpgbar	_CTRL_PROGRESSBAR	包含 PROGRESSBAR 控件	开启
ctrltoolbar	_CTRL_TOOLBAR	包含 TOOLBAR 控件	开启
ctrlnewtoolbar	_CTRL_NEWTOOLBAR	包含 NEWTOOLBAR 控件	开启
ctrlmenubtn	_CTRL_MENUBUTTON	包含 MENUBUTTON 控件	开启
ctrltrackbar	_CTRL_TRACKBAR	包含 TRACKBAR 控件	开启
ctrlcombobox	_CTRL_COMBOBOX	包含 COMBOBOX 控件	开启
ctrlpropsheet	_CTRL_PROPSHEET	包含 PROPSHEET 控件	开启
ctrlscrollview	_CTRL_SCROLLVIEW	包含 SCROLLVIEW 和 SCROLLWINDOW 控件	开启

ctrltextedit	_CTRL_TEXTEDIT	包含基于 SCROLLVIEW 的 TEXTEDIT 控件	开启
extctrlmonthcal	_EXT_CTRL_MONTHCAL	包含 MiniGUIExt 库中的 MONTHCALENDAR 控件	开启
extctrltreeview	_EXT_CTRL_TREEVIEW	包含 MiniGUIExt 库中的 TREEVIEW 控件	开启
extctrlspinbox	_EXT_CTRL_SPINBOX	包含 MiniGUIExt 库中的 SPINBOX 控件	开启
extctrlcoolbar	_EXT_CTRL_COOLBAR	包含 MiniGUIExt 库中的 COOLBAR 控件	开启
extctrllistview	_EXT_CTRL_LISTVIEW	包含 MiniGUIExt 库中的 LISTVIEW 控件	开启
extctrliconview	_EXT_CTRL_ICONVIEW	包含 MiniGUIExt 库中的 ICONVIEW 控件	开启
extctrlgridview	_EXT_CTRL_GRIDVIEW	包含 MiniGUIExt 库中的 GRIDVIEW	关闭
extctrlanimation	_EXT_CTRL_ANIMATION	包含 MiniGUIExt 库中的 ANIMATION 控件并提供对 GIF89a 文件的支持	开启
extskin	_EXT_SKIN	包含 MiniGUIExt 库中对皮肤的支持	开启

2.2.13 其他选项和宏

为适应各种嵌入式操作系统的运行环境，MiniGUI 自身实现了一些标准 C 函数库的函数族，其包括 malloc 函数族(malloc、calloc、free 等函数)，stdio 格式化输入输出函数族(sprintf、printf 等)，以及 POSIX 线程库的接口(pthread_create、sem_post 等)。这些函数族的编译配置选项默认是关闭的，而且仅在某些基于线程和任务的传统嵌入式操作系统上可用。具体应该在哪些操作系统上开启这些选项，请参阅 2.2.1 小节中的描述。表 2.19 给出了 MiniGUI 自实现 C 库接口的配置选项和相应的宏。

表 2.19 自实现 C 库接口相关的配置选项和宏

configure 配置选项	宏	注释	默认值
ownmalloc	_USE_OWN_MALLOC	使用 MiniGUI 实现的 malloc 函数族	关闭
ownstdio	_USE_OWN_STDIO	使用 MiniGUI 实现的 stdio 格式化输入输出函数族	关闭
ownpthread	_USE_OWN_PTHREAD	使用 MiniGUI 实现的 pthread 函数族	关闭

另外，在非 GNU 开发环境中，使用自定义 makefile 编译 MiniGUI 函数库时，还必须显式定义表 2.20 中的两个宏：__MINIGUI_LIB__ 和 __MGEXT_LIB__。

表 2.20 其他编译宏

宏	注释	备注
<code>__MINIGUI_LIB__</code>	编译 MiniGUI 库的宏	使用非 GNU 的 makefile 时，必须定义这两个宏
<code>__MGEXT_LIB__</code>	编译 MiniGUI 扩展库的宏	

2.3 最小配置选项

本节给出一个将 MiniGUI 配置成最小功能集合的范例。

2.3.1 使用 GNU configure 脚本

在 MiniGUI 源代码 build/ 目录下包含了一个 buildlib-min 脚本。该脚本内容如下：

```
#!/bin/sh

./configure \
  --disable-micemoveable \
  --disable-dblclk \
  --disable-cursor \
  --disable-mousecalibrate \
  --disable-clipboard \
  --disable-vbfsupport \
  --disable-rbfgb12 \
  --disable-fontsserif \
  --disable-fontcourier \
  --disable-qpfsupport \
  --disable-ttfsupport \
  --disable-latin9support \
  --disable-gbsupport \
  --disable-gbksupport \
  --disable-big5support \
  --disable-unicodesupport \
  --disable-savebitmap \
  --disable-jpgsupport \
  --disable-pngsupport \
  --disable-gifsupport \
  --disable-imegb2312 \
  --disable-imegb2312py \
  --disable-aboutdlg \
  --disable-savescreen \
  --disable-adv2dapi \
  --disable-videoqxfb \
  --disable-qvfbial \
  --disable-extctrlmonthcal \
  --disable-extctrltreeview \
  --disable-extctrlspinbox \
  --disable-extctrlcoolbar \
  --disable-extctrltreeview \
  --disable-extctrliconview \
  --disable-extctrlgridview \
  --disable-extctrlanimation \
  --disable-extskin \
  --disable-libvcongui \
  --disable-nativeeps2 \
  --disable-nativeimps2 \
  --disable-nativems \
  --disable-nativems3 \
  --disable-nativegpm \
  --with-style=flat
```

利用这个脚本可将 MiniGUI 配置成只支持 ISO8859-1 字符集的较小函数库。由这个脚本配置的

MiniGUI 函数库功能如下：

- 将 MiniGUI 编译为 MiniGUI-Threads。
- 不可以使用鼠标移动窗口。
- 不支持鼠标双击。
- 不包含对光标的支持。
- 不支持触摸屏校正。
- 不包含对剪贴板的支持。
- 不包含对 var bitmap 字体的支持。
- 不包含 RBF GB2312 12x12 fixed/song 内建字体。
- 不包含 sansserif 内建字体。
- 不包含 courier 内建字体。
- 不包含对 Qt Prerendered Font(QPF)字体的支持。
- 不包含对 TrueType 字体的支持。
- 不包含对 Latin 9(ISO-8859-15, West Extended)字符集的支持。
- 不包含对 EUC 编码的 GB2312 字符集的支持。
- 不包含对 GBK 字符集的支持。
- 不包含对 BIG5 字符集的支持。
- 不包含对 UNICODE(ISO-10646-1 和 UTF-8 编码)的支持。
- 不支持 SaveBitmap 的相关函数。
- 不包含对 JPG 文件的支持。
- 不包含对 PNG 文件的支持。
- 不包含对 GIF 文件的支持。
- 不包含对 GB2312 输入法的支持。
- 不包含对 GB2312 智能拼音输入法的支持。
- 不包含 About 对话框。
- 不支持屏幕截图。
- 不支持高级 2D 图形 API。
- 不支持 Qt Virtual FrameBuffer NEWGAL 引擎。
- 不构造 QVFB IAL 引擎。
- 不包含 MiniGUIExt 库中的 MONTHCALENDAR 控件。
- 不包含 MiniGUIExt 库中的 TREEVIEW 控件。
- 不包含 MiniGUIExt 库中的 SPINBOX 控件。
- 不包含 MiniGUIExt 库中的 COOLBAR 控件。
- 不包含 MiniGUIExt 库中的 LISTVIEW 控件。

- 不包含 MiniGUIExt 库中的 ICONVIEW 控件。
- 不包含 MiniGUIExt 库中的 GRIDVIEW 控件。
- 不包含 MiniGUIExt 库中对皮肤的支持。
- 不包含 MiniGUIExt 库中的 ANIMATION 控件并且不提供对 GIF87a/GIF89a 文件的支持。
- 不支持 Linux 的虚拟控制台。
- 不构造 PS2 鼠标本地引擎子驱动程序。
- 不构造 IntelligentMouse (IMPS/2) 鼠标本地引擎子驱动程序。
- 不构造旧的 MS 串行鼠标本地引擎子驱动程序。
- 不构造 MS3 鼠标本地引擎子驱动程序。
- 不构造 GPM 守护进程本地引擎子驱动程序。
- 外观风格为 Flat。

在上述配置基础上, 用户还可以根据需求取消一些功能。例如, 应用程序不需要使用 menu button 控件的话, 可以在上述脚本中加入 --disable-ctrlmenubtn 选项, 这样编译出来的 MiniGUI 函数库就不含有菜单按钮控件, MiniGUI 函数库也就会变得更小。

2.3.2 对应的 mgconfig.h

上述配置脚本生成的 mgconfig.h 文件如下:

```
/* mgconfig.h.  Generated by configure.  */
/* mgconfig.h.in.  Generated from configure.in by autoheader.  */

/* Define to one of `_getb67', `GETB67', `getb67' for Cray-2 and Cray-YMP
systems. This function is required for `alloca.c' support on those systems.
*/
/* #undef CRAY_STACKSEG_END */

/* Define to 1 if using `alloca.c'.  */
/* #undef C_ALLOCA */

/* Define to 1 if you have `alloca', as a function or macro.  */
#define HAVE_ALLOCA 1

/* Define to 1 if you have <alloca.h> and it should be used (not on Ultrix).
*/
#define HAVE_ALLOCA_H 1

/* Define to 1 if you have the <dirent.h> header file, and it defines `DIR'.
*/
#define HAVE_DIRENT_H 1

/* Define to 1 if you have the <dlfcn.h> header file.  */
#define HAVE_DLFCN_H 1

/* Define to 1 if you don't have `vprintf' but do have `_doprnt.'  */
/* #undef HAVE_DOPRNT */

/* Define to 1 if you have the `getpagesize' function.  */
#define HAVE_GETPAGESIZE 1

/* Define to 1 if you have the `getpt' function.  */
#define HAVE_GETPT 1

/* Define to 1 if you have the <inttypes.h> header file.  */
```

```

#define HAVE INTTYPES H 1

/* Define to 1 if you have the <locale.h> header file. */
#define HAVE LOCALE H 1

/* Define to 1 if you have the <math.h> header file. */
#define HAVE MATH H 1

/* Define to 1 if you have the <memory.h> header file. */
#define HAVE_MEMORY_H 1

/* Define to 1 if you have a working `mmap' system call. */
#define HAVE MMAP 1

/* Define to 1 if you have the <ndir.h> header file, and it defines `DIR'. */
/* #undef HAVE_NDIR_H */

/* Define to 1 if you have the `setenv' function. */
#define HAVE SETENV 1

/* Define to 1 if you have the `setlocale' function. */
#define HAVE_SETLOCALE 1

/* Define to 1 if you have the <stdint.h> header file. */
#define HAVE STDINT H 1

/* Define to 1 if you have the <stdlib.h> header file. */
#define HAVE_STDLIB_H 1

/* Define to 1 if you have the `strcasecmp' function. */
#define HAVE STRCASECMP 1

/* Define to 1 if you have the `strdup' function. */
#define HAVE_STRDUP 1

/* Define to 1 if you have the `strerror' function. */
#define HAVE STRERROR 1

/* Define to 1 if you have the <strings.h> header file. */
#define HAVE_STRINGS_H 1

/* Define to 1 if you have the <string.h> header file. */
#define HAVE STRING H 1

/* Define to 1 if you have the <sys/dir.h> header file, and it defines `DIR'.
   */
/* #undef HAVE_SYS_DIR_H */

/* Define to 1 if you have the <sys/ndir.h> header file, and it defines `DIR'.
   */
/* #undef HAVE_SYS_NDIR_H */

/* Define to 1 if you have the <sys/stat.h> header file. */
#define HAVE_SYS_STAT H 1

/* Define to 1 if you have the <sys/time.h> header file. */
#define HAVE_SYS_TIME H 1

/* Define to 1 if you have the <sys/types.h> header file. */
#define HAVE_SYS_TYPES H 1

/* Define to 1 if you have the <termio.h> header file. */
#define HAVE TERMIO H 1

/* Define to 1 if you have the `tmpfile' function. */
#define HAVE TMPFILE 1

/* Define to 1 if you have the <unistd.h> header file. */
#define HAVE UNISTD H 1

/* Define to 1 if you have the `vprintf' function. */
#define HAVE_VPRINTF 1

/* Binary age of MiniGUI */
#define MINIGUI_BINARY_AGE 0

```

```

/* Interface age of MiniGUI */
#define MINIGUI_INTERFACE_AGE 0

/* Major version of MiniGUI */
#define MINIGUI_MAJOR_VERSION 2

/* Micro version of MiniGUI */
#define MINIGUI_MICRO_VERSION 3

/* Minor version of MiniGUI */
#define MINIGUI_MINOR_VERSION 0

/* Define to the address where bug reports for this package should be sent. */
#define PACKAGE_BUGREPORT ""

/* Define to the full name of this package. */
#define PACKAGE_NAME ""

/* Define to the full name and version of this package. */
#define PACKAGE_STRING ""

/* Define to the one symbol short name of this package. */
#define PACKAGE_TARNAME ""

/* Define to the version of this package. */
#define PACKAGE_VERSION ""

/* If using the C implementation of alloca, define if you know the
   direction of stack growth for your system; otherwise it will be
   automatically deduced at run-time.
       STACK_DIRECTION > 0 => grows toward higher addresses
       STACK_DIRECTION < 0 => grows toward lower addresses
       STACK_DIRECTION = 0 => direction of growth unknown */
/* #undef STACK_DIRECTION */

/* Define to 1 if you have the ANSI C header files. */
#define STDC_HEADERS 1

/* Define to 1 if you can safely include both <sys/time.h> and <time.h>. */
#define TIME_WITH_SYS_TIME 1

/* Define to 1 if your <sys/time.h> declares `struct tm'. */
/* #undef TM_IN_SYS_TIME */

/* Define if compile for Win32 platform */
/* #undef WIN32 */

/* Define if include IAL engine for ABSSIG */
/* #undef _ABSSIG_IAL */

/* Define if include IAL engine for Arca Tpanel Ads7846 */
/* #undef ADS7846_IAL */

/* Define if include IAL engine for ADS */
/* #undef _ADS_IAL */

/* Define if include advanced 2D graphics APIs */
/* #undef ADV_2DAPI */

/* Define if support Arabic charset */
/* #undef _ARABIC_SUPPORT */

/* Define if include IAL engine for ARM3000 */
/* #undef ARM3000_IAL */

/* Define if include the automatic IAL engine */
/* #undef _AUTO_IAL */

/* Define if support BIG5 charset */
/* #undef BIG5_SUPPORT */

/* Define if include clipboard support */
/* #undef CLIPBOARD_SUPPORT */

```

```
/* Define if include the common IAL engine */
/* #undef  COMM_IAL */

/* Define if use coordinate transformation */
/* #undef  _COOR_TRANS */

/* Define if include BUTTON control */
#define  CTRL_BUTTON 1

/* Define if include COMBOBOX control */
#define  _CTRL_COMBOBOX 1

/* Define if include LISTBOX control */
#define  CTRL_LISTBOX 1

/* Define if include MENUBUTTON control */
#define  _CTRL_MENUBUTTON 1

/* Define if include NEWTOOLBAR control */
#define  CTRL_NEWTOOLBAR 1

/* Define if include PROGRESSBAR control */
#define  _CTRL_PROGRESSBAR 1

/* Define if include PROPSHEET control */
#define  CTRL_PROPSHEET 1

/* Define if include SCROLLVIEW control */
#define  CTRL_SCROLLVIEW 1

/* Define if include SIMEDIT control */
#define  CTRL_SIMEDIT 1

/* Define if include SLEDIT control */
#define  CTRL_SLEDIT 1

/* Define if include STATIC control */
#define  CTRL_STATIC 1

/* Define if include TEXTEDIT control */
#define  CTRL_TEXTEDIT 1

/* Define if include TOOLBAR control */
#define  _CTRL_TOOLBAR 1

/* Define if include TRACKBAR control */
#define  CTRL_TRACKBAR 1

/* Define if include cursor support */
/* #undef  _CURSOR_SUPPORT */

/* Define if support Cyrillic charset */
/* #undef  CYRILLIC_SUPPORT */

/* Define if build with debugging messages */
/* #undef  _DEBUG */

/* Define if include IAL engine for DirectFB */
/* #undef  DFB_IAL */

/* Define if include IAL engine for DM270 */
/* #undef  _DM270_IAL */

/* Define if include IAL engine for Intel DMG STB Remote Controller */
/* #undef  DMGSTB_IAL */

/* Define if mouse button can do double click */
/* #undef  _DOUBLE_CLICK */

/* Define if include the dummy IAL engine */
#define  DUMMY_IAL 1

/* Define if include the em85 IAL engine */
/* #undef  EM85_IAL */
```

```

/* Define if include the em86 IAL engine */
/* #undef EM86 IAL */

/* Define if include IAL engine for EMBEST ARM2410 */
/* #undef _EMBEST2410_IAL */

/* Define if include IAL engine for EMBEST ARM44B0 */
/* #undef EMBEST44B0 IAL */

/* Define if include IAL engine for EP7211 */
/* #undef _EP7211_IAL */

/* Define if support EUCJP charset */
/* #undef EUCJP SUPPORT */

/* Define if support EUCKR charset */
/* #undef _EUCKR_SUPPORT */

/* Define if include IAL engine for xScale EVMV 1.0 */
/* #undef EVMV10 IAL */

/* Define if include ANIMATION control */
/* #undef _EXT_CTRL_ANIMATION */

/* Define if include COOLBAR control */
/* #undef EXT_CTRL_COOLBAR */

/* Define if include GRIDVIEW control */
/* #undef EXT_CTRL_GRIDVIEW */

/* Define if include ICONVIEW control */
/* #undef EXT_CTRL_ICONVIEW */

/* Define if include LISTVIEW control */
/* #undef EXT_CTRL_LISTVIEW */

/* Define if include MONTHCALENDAR control */
/* #undef EXT_CTRL_MONTHCAL */

/* Define if include SPINBOX control */
/* #undef EXT_CTRL_SPINBOX */

/* Define if include TREEVIEW control */
/* #undef _EXT_CTRL_TREEVIEW */

/* Define if include skin support */
/* #undef EXT_SKIN */

/* Define if include IAL engine for FFT7202 */
/* #undef _FFT7202_IAL */

/* Define if include IAL engine for EM85xx Front Panel and Remote Controller */
/* #undef FIP IAL */

/* Define if include fixed math routines */
#define FIXED_MATH 1

/* Define if window style is flat */
#define FLAT_WINDOW_STYLE 1

/* Define if include IAL engine for FXRM9200 */
/* #undef FXRM9200 IAL */

/* Define if support GB18030 charset */
/* #undef GB18030 SUPPORT */

/* Define if support GBK charset */
/* #undef GBK SUPPORT */

/* Define if support GB2312 charset */
/* #undef _GB_SUPPORT */

/* Define if support GIF bmp file format */
/* #undef _GIF_FILE_SUPPORT */

```

```
/* Define if include GPM mouse subdriver */
/* #undef GPM SUPPORT */

/* Define if is gray screen */
#define _GRAY_SCREEN 1

/* Define if support Greek charset */
/* #undef GREEK SUPPORT */

/* Define if have math library */
#define HAVE_MATH_LIB 1

/* Define if your Linux have text mode */
#define HAVE_TEXT_MODE 1

/* Define if support Hebrew charset */
/* #undef HEBREW SUPPORT */

/* Define if include IAL engine for Helio Touch Panel */
/* #undef HELIO IAL */

/* Define if include IAL engine for HHARM2410R3 touch screen */
/* #undef HH2410R3 IAL */

/* Define if include IAL engine for HHARM2440 touch screen */
/* #undef _HH2440_IAL */

/* Define if include IAL engine for HHARM44B0 touch screen */
/* #undef HH44B0 IAL */

/* Define if include the IAL engine for HHCF5249 PS/2 keyboard and IrDA */
/* #undef _HH5249KBDIR_IAL */

/* Define if include GB2312 IME */
/* #undef IME GB2312 */

/* Define if include GB2312 Intelligent Pinyin IME module */
/* #undef _IME_GB2312_PINYIN */

/* Define if include IMPS2 mouse subdriver */
/* #undef IMPS2 SUPPORT */

/* Define if include in-core font: Courier */
/* #undef _INCOREFONT_COURIER */

/* Define if include in-core font: Helvetica */
/* #undef INCOREFONT HELV */

/* Define if include in-core font: SansSerif */
/* #undef _INCOREFONT_SANSERIF */

/* Define if include in-core font: Symbol */
/* #undef INCOREFONT SYMBOL */

/* Define if include in-core font: VGAS */
/* #undef INCOREFONT VGAS */

/* Define if include in-core font: 12x12 fixed/ming RBF for BIG5 */
/* #undef INCORERBF BIG5 12 */

/* Define if include in-core font: 16x16 fixed/ming RBF for BIG5 */
/* #undef INCORERBF BIG5 16 */

/* Define if include in-core font: 24x24 fixed/ming RBF for BIG5 */
/* #undef INCORERBF BIG5 24 */

/* Define if include in-core font: 12x12 fixed/song RBF for GB2312 */
/* #undef INCORERBF GB12 */

/* Define if include in-core font: 16x16 fixed/song RBF for GB2312 */
/* #undef _INCORERBF_GB16 */

/* Define if include in-core font: 16x16 fixed/hei RBF for GB2312 */
/* #undef _INCORERBF_GB16_HEI */
```

```

/* Define if include in-core font: 24x24 fixed/song RBF for GB2312 */
/* #undef INCORERBF GB24 */

/* Define if include in-core font: 12-Pixel Kanji RBFs for SHIFT-JIS */
/* #undef _INCORERBF_KJ12 */

/* Define if include in-core font: 14-Pixel Kanji RBFs for SHIFT-JIS */
/* #undef INCORERBF KJ14 */

/* Define if include in-core font: 16-Pixel Kanji RBFs for SHIFT-JIS */
/* #undef INCORERBF KJ16 */

/* Define if include in-core font: 8x16 fixed RBF for ISO8859-1 */
/* #undef INCORERBF LATIN1 16 */

/* Define if include in-core font: 12x24 fixed RBF for ISO8859-1 */
/* #undef INCORERBF LATIN1 24 */

/* Define if build MiniGUI for no file I/O system */
/* #undef INCORE RES */

/* Define if include IAL engine for iPAQ */
/* #undef IPAQ IAL */

/* Define if support JPEG bmp file format */
/* #undef _JPG_FILE_SUPPORT */

/* Define if use the German keyboard layout */
/* #undef KBD_LAYOUT DE */

/* Define if use the German-Latin1 keyboard layout */
/* #undef _KBD_LAYOUT_DELATIN1 */

/* Define if use the Spanish keyboard layout */
/* #undef KBD_LAYOUT ES */

/* Define if use the Spanish CP850 keyboard layout */
/* #undef _KBD_LAYOUT_ESCP850 */

/* Define if use the French keyboard layout */
/* #undef KBD_LAYOUT FR */

/* Define if use the French PC keyboard layout */
/* #undef _KBD_LAYOUT_FRPC */

/* Define if use the Italian keyboard layout */
/* #undef KBD_LAYOUT IT */

/* Define if include IAL engine for L7200 */
/* #undef _L7200_IAL */

/* Define if support Latin 10 charset */
/* #undef LATIN10 SUPPORT */

/* Define if support Latin 2 charset */
/* #undef LATIN2 SUPPORT */

/* Define if support Latin 3 charset */
/* #undef LATIN3 SUPPORT */

/* Define if support Latin 4 charset */
/* #undef LATIN4 SUPPORT */

/* Define if support Latin 5 charset */
/* #undef LATIN5 SUPPORT */

/* Define if support Latin 6 charset */
/* #undef LATIN6 SUPPORT */

/* Define if support Latin 7 charset */
/* #undef _LATIN7_SUPPORT */

/* Define if support Latin 8 charset */
/* #undef _LATIN8_SUPPORT */

```

```
/* Define if support Latin 9 charset */
/* #undef LATIN9_SUPPORT */

/* Define if support LBM bmp file format */
/* #undef _LBM_FILE_SUPPORT */

/* Define if include vconguit support */
/* #undef LIB_VCONGUI */

/* Define if build MiniGUI-Processes */
/* #undef LITE_VERSION */

/* Define if compile max ttf cahce number for 10 (default value) */
/* #undef MAX_TTF_CACHE */

/* Define if include IAL engine for uClinux touch screen palm/mc68ez328 */
/* #undef MC68X328_IAL */

/* Define if build MiniGUI-Processes */
/* #undef MGRM_PROCESSES */

/* Define if build MiniGUI-Standalone */
/* #undef MGRM_STANDALONE */

/* Define if build MiniGUI-Threads */
#define _MGRM_THREADS 1

/* Define if include About MiniGUI Dialog Box */
/* #undef MISC_ABOUTDLG */

/* Define if include code for mouse calibration */
/* #undef _MISC_MOUSECALIBRATE */

/* Define if include code for screenshots */
/* #undef MISC_SAVESCREEN */

/* Define if we can move window by mouse */
/* #undef _MOVE_WINDOW_BY_MOUSE */

/* Define if include IAL engine for MPC823 */
/* #undef MPC823_IAL */

/* Define if include MS3 mouse subdriver */
/* #undef _MS3_SUPPORT */

/* Define if include messages' string names */
/* #undef MSG_STRING */

/* Define if include MS mouse subdriver */
/* #undef _MS_SUPPORT */

/* Define if include native (Linux console) IAL engine */
#define NATIVE_IAL_ENGINE 1

/* Define if include NEWGAL engine for BF533 OSD via SPI */
/* #undef NEWGAL_ENGINE_BF533 */

/* Define if include NEWGAL engine for Common LCD */
/* #undef NEWGAL_ENGINE_COMMLCD */

/* Define if include NEWGAL engine for DirectFB */
/* #undef NEWGAL_ENGINE_DFB */

/* Define if include dummy NEWGAL engine */
#define NEWGAL_ENGINE_DUMMY 1

/* Define if include NEWGAL engine for EM85xx OSD */
/* #undef NEWGAL_ENGINE_EM85XXOSD */

/* Define if include FrameBuffer console NEWGAL engine */
#define _NEWGAL_ENGINE_FBCON 1

/* Define if include NEWGAL engine for mb93493 YUV FrameBuffer driver */
/* #undef _NEWGAL_ENGINE_MB93493 */
```

```

/* Define if include Qt Virtual FrameBuffer NEWGAL engine */
/* #undef NEWGAL_ENGINE_QVFB */

/* Define if include Shadow NEWGAL engine */
/* #undef _NEWGAL_ENGINE_SHADOW */

/* Define if include NEWGAL engine for SVPXX OSD */
/* #undef NEWGAL_ENGINE_SVPXXOSD */

/* Define if include NEWGAL engine for UTPMC */
/* #undef NEWGAL_ENGINE_UTPMC */

/* Define if include windows Virtual FrameBuffer NEWGAL engine */
/* #undef NEWGAL_ENGINE_WVFB */

/* Define if include IAL engine for OKWAP Palm2 */
/* #undef PALMII_IAL */

/* Define if window style is classic */
/* #undef PC3D_WINDOW_STYLE */

/* Define if support PCX bmp file format */
/* #undef PCX_FILE_SUPPORT */

/* Define if window style is fashion */
/* #undef _PHONE_WINDOW_STYLE */

/* Define if support PNG bmp file format */
/* #undef PNG_FILE_SUPPORT */

/* Define if include PS2 mouse subdriver */
/* #undef _PS2_SUPPORT */

/* Define if include IAL engine for PX255B */
/* #undef PX255B_IAL */

/* Define if support QPF font */
/* #undef _QPF_SUPPORT */

/* Define if include the QVFB IAL engine */
/* #undef QVFB_IAL */

/* Define if include the random IAL engine */
/* #undef _RANDOM_IAL */

/* Define if support raw bitmap fonts */
#define RBF_SUPPORT 1

/* Define to 0 if use clockwise rotation of screen, otherwise to 1 */
/* #undef _ROT_DIR_CW */

/* Define if include SaveBitmap function */
/* #undef SAVE_BITMAP */

/* Define if support SHIFTJIS charset */
/* #undef SHIFTJIS_SUPPORT */

/* Define if include IAL engine for touch screen of SkyEye EP7312 simulation */
/* #undef _SKYEYE_EP7312_IAL */

/* Define if include IAL engine for SMDK2410 touch screen */
/* #undef SMDK2410_IAL */

/* Define if build MiniGUI-Standalone */
/* #undef _STAND_ALONE */

/* Define if include IAL engine for WinBond SPVXX */
/* #undef SVPXX_IAL */

/* Define if include IAL engine for MT T800 */
/* #undef _T800_IAL */

/* Define if include IAL engine for Tongfang STB */

```

```
/* #undef TFSTB IAL */

/* Define if support TGA bmp file format */
/* #undef TGA FILE SUPPORT */

/* Define if support Thai charset */
/* #undef THAI SUPPORT */

/* Define if include IAL engine for THOR */
/* #undef _THOR_IAL */

/* Define if the unit of timer is 10ms */
#define TIMER UNIT 10MS 1

/* Define if it is tiny screen */
/* #undef _TINY_SCREEN */

/* Define if trace message dispatching of MiniGUI */
/* #undef TRACE MSG */

/* Define if compile max ttf cahce size for 64k in KB */
/* #undef _TTF_CACHE_SIZE */

/* Define if include ttf cache */
/* #undef TTF CACHE SUPPORT */

/* Define if support TrueType font based on FreeType 1.3 */
/* #undef _TTF_SUPPORT */

/* Define if support Adobe Type1 fonts */
/* #undef TYPE1 SUPPORT */

/* Define if include IAL engine for UCB1X00 */
/* #undef _UCB1X00_IAL */

/* Define if support UNICODE */
/* #undef UNICODE SUPPORT */

/* Define if use fixed scrollbar bar length */
/* #undef _USE_FIXED_SB_BARLEN */

/* Define if use minigui entry function in MiniGUI */
/* #undef USE MINIGUIENTRY */

/* Define if use new GAL interfaces */
#define USE NEWGAL 1

/* Define if use own implementation of malloc functions */
/* #undef USE OWN MALLOC */

/* Define if use own implementation of pthread functions */
/* #undef USE OWN PTHREAD */

/* Define if use own implementation of stdio functions */
/* #undef _USE_OWN_STDIO */

/* Define if include the UTPMC IAL engine */
/* #undef UTPMC IAL */

/* Define if support var bitmap fonts */
/* #undef _VBF_SUPPORT */

/* Define if include IAL engine for NEC VR4181 */
/* #undef VR4181 IAL */

/* Define if include the WVFB IAL engine */
/* #undef _WVFB_IAL */

/* Define if compile for Cygwin platform */
/* #undef CYGWIN */

/* Define if compile for OpenDarwin */
/* #undef __DARWIN__ */

/* Define if compile for eCos */
```

```

/* #undef ECOS */

/* Define if compile for Linux */
#define LINUX 1

/* Define if compile for non-UNIX like OS */
/* #undef NOUNIX */

/* Define if compile for Nucleus */
/* #undef __NUCLEUS__ */

/* Define for Linux FrameBuffer console (used by Shadow NEGAL engine) */
/* #undef TARGET FBCON */

/* Define for Linux QVFB (used by Shadow NEGAL engine) */
/* #undef __TARGET_QVFB__ */

/* Define for Philips STB810 target */
/* #undef TARGET STB810 */

/* Define for unknown target */
#define __TARGET_UNKNOWN__ 1

/* Define for VirtualFone ANVIL target */
/* #undef TARGET VFANVIL */

/* Define for VxWorks on i386 */
/* #undef __TARGET_VXI386__ */

/* Define for the Windows QVFB (used by Shadow NEGAL engine) */
/* #undef TARGET WVFB */

/* Define if compile for ThreadX */
/* #undef __THREADX__ */

/* Define if compile for uC/OS-II */
/* #undef UCOSII */

/* Define if compile for VxWorks */
/* #undef __VXWORKS__ */

/* Define if compile for Winbond SWLinux */
/* #undef WINBOND SWLINUX */

/* Define if compile for uClinux */
/* #undef uClinux */

/* Define to empty if `const' does not conform to ANSI C. */
/* #undef const */

/* Define as `__inline' if that's what the C compiler calls it, or to nothing
   if it is not supported. */
/* #undef inline */

/* Define to `unsigned' if <sys/types.h> does not define. */
/* #undef size_t */

```

2.4 MiniGUI 的编译和安装

2.4.1 GNU 开发环境中的编译和安装

如果您在 GNU 开发环境中使用 `configure` 脚本配置 MiniGUI，则可以直接使用 `make` 工具编译并安装 MiniGUI。

例如，假定您使用的是 MiniGUI-VAR for Linux/uClinux 产品，则在运行 Linux 的 PC 机上，您可

以在 MiniGUI 源代码所在目录中用如下几个命令将 MiniGUI 配置、编译并安装到系统中：

```
user$ ./configure
user$ make
user$ su -c 'make install'
```

您还可以通过 `configure` 脚本的选项，指定进行交叉编译、安装的路径等等。具体的方法，请参阅针对特定操作系统的《MiniGUI 移植指南》文档。

2.4.2 在非 GNU 开发环境中使用 cygwin 工具编译和安装 MiniGUI

在非 GNU 开发环境中（通常就是 Windows 平台），我们可以将 MiniGUI 源代码组织成适合某特定集成开发环境（比如 Tornado、ADS）的工程，然后编译 MiniGUI 甚至 MiniGUI 应用程序。

但是，利用 Windows 平台上的 cygwin 开发环境，我们可以更加方便地编译和安装 MiniGUI。从理论上讲，这种方法可以适用于任何一个运行在 Windows 平台上的开发环境，因此，我们将在本节对这个方法进行一些一般性的描述。

Cygwin 是一个开源软件项目，其目的就是在 Windows 平台上建立类似于 Linux 的使用环境。在 Windows 上安装了 cygwin 之后，我们可以运行很多 Linux 平台上的应用程序，比如 BASH 脚本、VIM 编辑器、PERL 脚本解释器、GNU 的 `make` 工具、`gcc` 编译器等等，而且在 cygwin 环境中，我们还可以直接调用其他的 Windows 应用程序。这样，只要针对 MiniGUI 编写出符合 GNU 规则的 `makefile` 文件，我们就可以利用 cygwin 提供的 `make` 工具来调用对应的编译器、链接器程序来编译生成 MiniGUI 函数库。

许多操作系统的开发环境包含有 cygwin 环境，比如 OSE。如果您使用的开发环境没有提供 cygwin，则可以到 <http://www.cygwin.com> 上下载并安装。注意在安装时，一定要确保安装 `make` 编译器、BASH Shell 脚本等软件包。

在 MiniGUI 源代码中，为了和 GNU `configure` 工具生成的 `makefile` 文件相区别，针对 cygwin 环境的 `makefile` 文件具有 `.ng` 后缀（这里的 `ng` 表示 non-GNU）。另外，我们还提供有一个独立的编译规则文件（命名为 `rules.make`），保存在 MiniGUI 源代码顶层目录中。通常，我们只需要根据真实的开发环境修改 `rules.make` 文件，然后就可以利用 cygwin 的 `make` 工具编译 MiniGUI 了：

```
cygwin$ /usr/bin/make -f makefile.ng
```

注意，`makefile.ng` 也提供有 `clean` 和 `install` 两个目标。运行

```
cygwin$ /usr/bin/make -f makefile.ng install
```

可将 MiniGUI 的头文件、函数库安装到 `rules.make` 文件指定的路径中。而运行

```
cygwin$ /usr/bin/make -f makefile.ng clean
```

可清除编译生成的文件，以便重新编译。

注意，在使用 **cygwin** 环境时，如果修改了 **mgconfig.h** 文件或者其他文件的内容，请首先运行上面的命令清除所有的编译生成文件，然后再重新编译 **MiniGUI**。

实际上，在利用 **cygwin** 环境和 **makefile.ng** 文件编译 **MiniGUI** 时，主要的工作在于提供正确的 **rules.make** 文件。我们已经为不同的操作系统开发环境准备了若干备用的 **rules.make** 文件，具体保存在 **MiniGUI** 源代码的 **build/** 路径下，具有 **rules-<platform>.<os>** 这样的命名规则⁷。比如，我们要为 **PC** 平台的 **VxWorks** 操作系统编译 **MiniGUI**，则可以用 **build/rules-pc.vxworks**⁸ 文件覆盖顶层的 **rules.make** 文件，然后利用上面的命令调用 **cygwin** 的 **make** 命令编译 **MiniGUI**。该 **rules.make** 文件的内容如下：

```
# rules for pc-vxworks

AS=
CC=ccpentium
CXX=c++pentium
CPP=ccpentium
AR=arpentium
RANLIB=ranlibpentium
MAKE=/usr/bin/make

ARFLAGS=crus
COFLAG=-c

OBJ=o
LIBA=a

PREFIX=c:/cross
#vxworks6.0
#TARGET_DIR=C:/WindRiver/vxworks-6.0/target

#vxworks5.5.1
TARGET_DIR=C:/Tornado2.2x86/target

INCS+=-I${TARGET_DIR}/h

CFLAGS+=-g -mcpu=pentium -march=pentium -Wall -DTOOL_FAMILY=gnu -DTOOL=gnu -D_WRS_KERNEL -DCPU=
PENTIUM
```

在根据自己的开发环境编译 **rules.make** 文件时，一定要正确定义表 2.21 给出的变量。

表 2.21 **cygwin** 的 **makefile.ng** 文件需要的变量

变量名称	用途	备注
CC	指定 C 编译器	
CPP	指定 C 预编译器	
AR	指定归档工具，即用来生成静态库的工具	

⁷ 其中也包含针对特定平台及操作系统的默认 **mgconfig.h** 文件，具有 **config-<os>-<platform>.h** 这样的命名规则。

⁸ 注意，该文件只存在于针对 **VxWorks** 操作系统的 **MiniGUI** 增值版产品中。

RANLIB	指定静态库索引工具	
MAKE	指定 make 工具	在 cygwin 环境中，通常为 /usr/bin/make
ARFLAGS	调用归档工具生成静态库时的选项	
COFLAG	调用 C 编译器而不进行链接时的选项	
OBJ	目标文件的后缀名	
LIBA	静态库文件的后缀名	
PREFIX	安装路径的前缀	
INCS	指定头文件的搜索路径选项	
CFLAGS	C 编译器选项	

注意，**cygwin** 的 **makefile.ng** 工程文件，将把 **MiniGUI** 的头文件安装到 **\$PREFIX/include/minigui** 目录下，而函数库（如 **libminigui.a**、**libmgext.a**）则会被安装到 **\$PREFIX/lib/** 目录下。如上面的 **rules.make** 文件，将把 **MiniGUI** 头文件安装到 **c:/cross/include/minigui** 下，而将 **MiniGUI** 函数库安装到 **c:/cross/lib** 目录下。

您可以参照表 2.21 的说明以及上述示例 **rules.make** 文件，根据真实的开发环境，编写自己的 **rules.make** 文件。

另外需要说明的是，因为 **makefile.ng** 使用的是 **GNU** 兼容的 **makefile** 格式，因此，实际上我们也可以在 **Linux** 下使用 **makefile.ng** 来编译 **MiniGUI**。这种情况往往发生在使用针对 **uClinux** 的交叉编译工具链时。如果在 **Linux** 环境下，只需要调用 **make** 命令就可以了：

```
user$ make -f makefile.ng
```

3 MiniGUI 的运行时配置选项

本章将描述 MiniGUI 的运行时配置选项。运行时配置选项影响 MiniGUI 的一些运行行为，比如要使用的图形引擎或者输入引擎；要装载的设备字体；要装载的位图、光标资源等等。我们已经了解到，MiniGUI 的运行时配置一般是通过配置文件（MiniGUI.cfg）指定的，但在使用内建资源方式配置并编译 MiniGUI 后，运行时配置选项将直接编译到函数库中。

在 GNU 开发环境下，使用默认配置安装 MiniGUI 之后，将把 MiniGUI 源代码树中的 etc/MiniGUI-classic.cfg 文件安装到系统 /usr/local/etc/ 目录，并更名为 MiniGUI.cfg。在 MiniGUI 应用程序启动时，MiniGUI 优先查找当前目录下的 MiniGUI.cfg 文件、用户主目录下的 .MiniGUI.cfg 文件，其次是 /usr/local/etc/MiniGUI.cfg，最后是 /etc/MiniGUI.cfg 文件。如果用户没有在当前目录及自己的主目录下建立对应的 MiniGUI.cfg 文件，则通常情况下，/usr/local/etc/MiniGUI.cfg 文件就是 MiniGUI 所使用的默认运行时配置文件。

针对 MiniGUI 不同的编译配置选项，共有四种不同的配置文件：MiniGUI-classic.cfg 对应 --with-style=classic，MiniGUI-fashion.cfg 对应 --with-style=fashion，MiniGUI-flat.cfg 对应 --with-style=flat 和 MiniGUI-min.cfg 对应 build/buildlib-min 配置脚本。有关 MiniGUI 编译配置选项的详细内容，请参考本手册第 2 章。

在使用内建式资源（incore resources）时，MiniGUI 不需要 MiniGUI.cfg 配置文件，相应的配置选项在 src/sysres/mgetc.c 文件中指定。

本章将首先描述使用配置文件时的运行时配置选项，然后描述如何在内建资源方式下指定运行时配置选项。

3.1 配置文件

本节将以 MiniGUI-classic.cfg 内容为例进行说明。

配置文件采用了非常简洁的格式，所以修改起来也很容易。其格式如下：

```
[section-name1]
key-name1=key-value1
key-name2=key-value2

[section-name2]
key-name3=key-value3
key-name4=key-value4
```

配置文件中由注释（#）、段（section）、键（key）和键值（Key Value）四部分组成，注释是以“#”为第一个字符开始的行，段是由[section-name]形式给出，包含若干键和键值对组合，用key=key_value的形式指定键和键值。下面按段进行介绍配置文件内容。

3.1.1 system 段

该段指定了 MiniGUI 运行时使用的输入输出引擎以及鼠标设备和协议类型。指定的输入输出引擎必须是编译配置 MiniGUI 库时多个引擎中的一个。

该段定义如下键：

- gal_engine: 指定使用的图形引擎
- defaultmode: 指定图形引擎的显示模式，格式为：宽 x 高-显示位数(bpp)
- ial_engine: 指定使用的输入引擎
- mdev: 指定鼠标设备文件
- mtype: 指定鼠标协议类型

MiniGUI.cfg 文件 system 段内容如下：

```
[system]
# GAL engine and default options
gal engine=qvfb
defaultmode=800x600-16bpp

# IAL engine
ial engine=qvfb

mdev=/dev/input/mice
mtype=IMPS2
```

自 1.6.8 版本，MiniGUI 提供通过环境变量修改图形及输入引擎的方式。比如，您的 MiniGUI 已经包含了 fbcon 和 qvfb 两种图形引擎以及对应的 console 和 qvfb 输入引擎，并在 MiniGUI.cfg 或者内建资源中指定了使用 qvfb 图形及输入引擎，但在运行时，您可以通过如下方式指定 fbcon 和 console 引擎，而不需要修改 MiniGUI.cfg 文件或者内建资源配置文件：

```
$ export gal_engine=fbcon
$ export ial_engine=console
$ export mdev=/dev/input/mice
$ export mtype=ps2
$ export defaultmode=1024x768-16bpp
```

3.1.2 fbcon 段

该段是在[system]段 gal_engine 的键值指定为 fbcon 时使用，表示运行 fbcon 时使用的显示模式。若该段未定义或键值为空，则使用[system]段定义的键值。

该段只包含一个 **defaultmode** 键，具体含义和[system]段相同：

- **defaultmode**: 指定图形引擎的显示模式，格式为：宽 x 高-显示位数(bpp)

```
[fbcon]
defaultmode=1024x768-16bpp
```

3.1.3 qvfb 段

该段是在[system]段 **gal_engine** 的键值指定为 **qvfb** 时使用，表示运行 **qvfb** 时使用了 Xwindow 的哪个 **display** 及显示模式。

该段定义如下键，具体含义如下：

- **defaultmode**: 指定图形引擎的显示模式，格式为：宽 x 高-显示位数(bpp)
- **display**: 运行 **qvfb** 时使用 X Window 的哪个 **display**，一般取 0。

MiniGUI.cfg 文件 **qvfb** 段内容如下：

```
[qvfb]
defaultmode=640x480-16bpp
display=0
```

3.1.4 rawbitmapfonts、varbitmapfonts、qpf、truetypefonts 和 type1fonts 段

这些段用来指定要装载的设备字体⁹信息。定义了要装载的设备字体个数，每个设备字体名称及对应的设备字体文件。

MiniGUI 使用的设备字体名称格式如下：

```
<type>-<facename>-<style>-<width>-<height>-<charset1[,charset2,...]>
```

设备字体名称各部分的含义如下：

- **type** 是设备字体类型，对 RBF、VBF、QPF、TrueType、Adobe Type1 设备字体，分别取 **rbf**、**vbf**、**qpf**、**ttf** 和 **t1f**。
- **facename** 指设备字体的样式名，比如 **Courier**、**Times** 等等。
- **style** 是由六个字母组成的字符串，用来指定字体风格，比如是否加粗、是否斜体、是否含有下划线或者删除线等等。通常这六个字母取 “**rrncnn**”。
- **width** 表示字体的宽度。对变宽字体来讲，用来指定最大宽度。对可缩放的矢量字体来讲，设置为 0。

⁹有关设备字体的详细内容，参考《MiniGUI 编程指南》。

- height 表示字体的高度。对可缩放的矢量字体来讲，设置为 0。
- charset1、charset2 等，表示该设备字体所支持的字符集。

每个段均包含 font_number、name<NR> 和 fontfile<NR>等键：

- font_number: 装载的设备字体个数
- name<NR>: 编号为<NR>的设备字体名称
- fontfile<NR>: 编号为<NR>的设备字体文件

如果不想装载某个类型的设备字体，则可以通过设置对应段的 font_number 为 0 来跳过对这种设备字体的装载。

MiniGUI.cfg 文件这些段内容如下：

```
[rawbitmapfonts]
font number=4
name0=rbf-fixed-rrncnn-8-16-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/8x16-iso8859-1.bin
name1=rbf-fixed-rrncnn-16-16-GB2312-0
fontfile1=/usr/local/lib/minigui/res/font/song-16-gb2312.bin
name2=rbf-fixed-rrncnn-6-12-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/6x12-iso8859-1.bin
name3=rbf-fixed-rrncnn-12-12-GB2312-0
fontfile3=/usr/local/lib/minigui/res/font/song-12-gb2312.bin

[varbitmapfonts]
font number=6
name0=vbf-Courier-rrncnn-8-13-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/Courier-rr-8-13.vbf
name1=vbf-Helvetica-rrncnn-11-12-ISO8859-1
fontfile1=/usr/local/lib/minigui/res/font/Helvetica-rr-11-12.vbf
name2=vbf-Times-rrncnn-10-12-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/Times-rr-10-12.vbf
name3=vbf-Courier-rrncnn-10-15-ISO8859-1
fontfile3=/usr/local/lib/minigui/res/font/Courier-rr-10-15.vbf
name4=vbf-Helvetica-rrncnn-15-16-ISO8859-1
fontfile4=/usr/local/lib/minigui/res/font/Helvetica-rr-15-16.vbf
name5=vbf-Times-rrncnn-13-15-ISO8859-1
fontfile5=/usr/local/lib/minigui/res/font/Times-rr-13-15.vbf

[qpf]
font number=3
name0=qpf-unifont-rrncnn-16-16-ISO8859-1,ISO8859-15,GB2312-0,GBK,BIG5
fontfile0=/usr/local/lib/minigui/res/font/unifont 160 50.qpf
name1=qpf-times-rrncnn-5-10-ISO8859-1,ISO8859-15
fontfile1=/usr/local/lib/minigui/res/font/smoothtimes_100_50.qpf
name2=qpf-helvetica-rrncnn-5-10-ISO8859-1,ISO8859-15
fontfile2=/usr/local/lib/minigui/res/font/helvetica 100 50.qpf
name3=qpf-micro-rrncnn-4-4-ISO8859-1,ISO8859-15
fontfile3=/usr/local/lib/minigui/res/font/micro 40 50.qpf

[truetypefonts]
font number=3
name0=ttf-arial-rrncnn-0-0-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/arial.ttf
name1=ttf-times-rrncnn-0-0-ISO8859-1
fontfile1=/usr/local/lib/minigui/res/font/times.ttf
name2=ttf-pinball-rrncnn-0-0-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/pinball.ttf

[type1fonts]
font number=0
name0=type1-Charter-rrncnn-0-0-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/bchr.pfb
name1=type1-Charter-rincnn-0-0-ISO8859-1
```

```
fontfile1=/usr/local/lib/minigui/res/font/bchri.pfb
name2=type1-Charter-brncnn-0-0-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/bchb.pfb
name3=type1-Charter-bincnn-0-0-ISO8859-1
fontfile3=/usr/local/lib/minigui/res/font/bchbi.pfb
name4=type1-Courier-rrncnn-0-0-ISO8859-1
fontfile4=/usr/local/lib/minigui/res/font/dcr10.pfb
name5=type1-Courier-rincnn-0-0-ISO8859-1
fontfile5=/usr/local/lib/minigui/res/font/dcti10.pfb
name6=type1-Courier-brncnn-0-0-ISO8859-1
fontfile6=/usr/local/lib/minigui/res/font/dcbx10.pfb
name7=type1-Courier-bincnn-0-0-ISO8859-1
fontfile7=/usr/local/lib/minigui/res/font/dcbxti10.pfb
name8=type1-eufm10-rrncnn-0-0-ISO8859-1
fontfile8=/usr/local/lib/minigui/res/font/eufm10.pfb
```

3.1.5 systemfont 段

该段定义了 MiniGUI 的系统字体和字体个数。并定义了系统默认使用的字体，用于 MiniGUI 的标题、菜单、控件的显示。

系统字体是 MiniGUI 装载了由 rawbitmapfonts、varbitmapfonts、qpf、truetypefonts、t1fonts 等段定义的设备字体之后，根据上述字体名称调用 CreateLogFontByName 函数建立的逻辑字体¹⁰。

逻辑字体名称的格式如下：

```
<type>-<facename>-<style>-<width>-<height>-<charset1>
```

逻辑字体名称各部分的含义如下：

- type 是所选用的设备字体类型，如果不想指定，则取 *。
- facename 指字体样式名，比如 Courier、Times 等等。
- style 是由六个字母组成的字符串，用来指定逻辑字体风格，比如是否加粗、是否斜体、是否对字体做镜像处理、是否自动放大、是否含有下划线或者删除线等等。
- width 指定要创建的逻辑字体的宽度。一般不用指定，取 *。
- height 指定要创建的逻辑字体的高度。
- charset 指定要创建的逻辑字体的字符集。

MiniGUI 的许多窗口元素尺寸是根据系统字体的大小定义的，详情请参阅下面对 mainwinmetric 段的说明。

此外，MiniGUI V2.0.3/1.6.9 还提供了字体的自动放大功能，以便将点阵字体适当放大来匹配逻辑字体所要求的大小。使用该功能的方法非常简单，请在指定逻辑字体风格时，在第四个字符处使用大写的“S”字母。注意，如果您要使用矢量字体类型，比如 TrueType 字体，则不需要指定该风格，因为矢量字体可以根据逻辑字体的期望大小生成对应的字体字模。

¹⁰ 有关逻辑字体及 CreateLogFontByName 函数的具体解释，请参考《MiniGUI 编程指南》。

该段定义了如下键：

- **font_number**: 指定了要创建的逻辑字体个数。
- **font<NR>**: 表示编号为 <NR> 的逻辑字体名称。
- **default**: 系统（单字节字符集）默认字体，键值为上述逻辑字体编号。
- **wchar_def**: 多字节字符集使用的字体，键值为上述逻辑字体编号。
- **fixed**: 等宽字符集使用的字体，键值为上述逻辑字体编号。
- **caption**: 窗口标题使用的字体，键值为上述逻辑字体编号。
- **menu**: 菜单使用的字体，键值为上述逻辑字体编号。
- **control**: 控件的字体，键值为上述逻辑字体编号。

可以修改要创建的系统字体个数，但至少要创建一种单字节字符集（比如 ISO8859-1）的字体。

MiniGUI 根据 **default**、**wchar_def** 系统字体来定义系统的默认字符集，这影响 **GetSysCharset**、**GetSysCharWidth**、**GetSysCCharWidth** 和 **GetSysHeight** 函数的返回值。一般来讲，**default** 和 **wchar_default** 必须是等宽点阵字体，即 RBF 字体，并且多字节字符集字体的宽度必须是单字节字符集宽带的两倍。

MiniGUI.cfg 文件 **systemfont** 段内容如下：

```
# The first system font must be a logical font using RBF device font.
[systemfont]
font number=6
font0=rbf-fixed-rrncnn-8-16-ISO8859-1
font1=*-fixed-rrncnn-*-16-GB2312
font2=*-Courier-rrncnn-*-16-GB2312
font3=*-SansSerif-rrncnn-*-16-GB2312
font4=*-Times-rrncnn-*-16-GB2312
font5=*-Helvetica-rrncnn-*-16-GB2312

default=0
wchar_def=1
fixed=1
caption=2
menu=3
control=3
```

3.1.6 mouse 段

该段定义了鼠标双击间隔时间。用于系统的内部事件处理，一般无须作任何改动。该段定义了如下键：

- **dblclicktime**: 指定了双击间隔时间，单位毫秒

MiniGUI.cfg 文件 **mouse** 段内容如下：

```
[mouse]
dblclicktime=300
```

3.1.7 event 段

该段定义了底层事件的超时时间和自动重复事件时间。用于系统的内部事件处理，一般无须作任何改动。该段定义了如下键：

- **timeoutusec**: 事件超时时间，单位微秒
- **repeatusec**: 重复事件时间，单位微秒

MiniGUI.cfg 文件 mouse 段内容如下：

```
timeoutusec=300000  
repeatusec=50000
```

3.1.8 cursorinfo 段

该段指定了 MiniGUI 要装载的鼠标光标相关信息。

如果在编译配置 MiniGUI 时使用了 `--disable-cursor` 选项，则 MiniGUI 会忽略 `cursorinfo` 段。该段定义如下键：

- **cursorpath**: 指定光标所在路径
- **cursornumber**: 指定装载的光标个数，可通过减少个数并删除对应光标减少 MiniGUI 的存储空间占用量
- **cursor<NR>**: 指定编号为<NR>的光标

MiniGUI.cfg 文件 `cursorinfo` 段内容如下：

```
[cursorinfo]  
# Edit following line to specify cursor files path  
cursorpath=/usr/local/lib/minigui/res/cursor/  
cursornumber=23  
cursor0=d_arrow.cur  
cursor1=d_beam.cur  
cursor2=d_pencil.cur  
cursor3=d_cross.cur  
cursor4=d_move.cur  
cursor5=d_sizenwse.cur  
cursor6=d_sizenesw.cur  
cursor7=d_sizewe.cur  
cursor8=d_sizens.cur  
cursor9=d_uparrow.cur  
cursor10=d_none.cur  
cursor11=d_help.cur  
cursor12=d_busy.cur  
cursor13=d_wait.cur  
cursor14=g_rarrow.cur  
cursor15=g_col.cur  
cursor16=g_row.cur  
cursor17=g_drag.cur  
cursor18=g_nodrop.cur  
cursor19=h_point.cur  
cursor20=h_select.cur  
cursor21=ho_split.cur  
cursor22=ve_split.cur
```

3.1.9 iconinfo 段

该段定义了 MiniGUI 要装载的图标相关信息。该段定义如下键：

- **iconpath**: 指定图标所在路径
- **iconnumber**: 指定装载的图标个数，最大值为 5，可通过减少个数并删除对应图标减少 MiniGUI 的存储空间占用量
- **icon<NR>**: 指定编号为<NR>的图标
- **fold**、**unfold**: 这两个键指定 TreeView 控件默认使用的图标
- **dir**、**file**: 这两个键指定 OpenFileDialogBox 控件默认使用的图标

MiniGUI.cfg 文件 iconinfo 段内容如下：

```
[iconinfo]
# Edit following line to specify icon files path
iconpath=/usr/local/lib/minigui/res/icon/
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form.ico
icon1=failed.ico
icon2=mg_help.ico
icon3=warning.ico
icon4=exclamatory.ico

# default icons for TREEVIEW control
fold=fold.ico
unfold=unfold.ico

# default icons for new OpenFileDialogBox
dir=folder.ico
file=textfile.ico
```

3.1.10 bitmapinfo 段

该段定义了 MiniGUI 要装载的位图相关信息。该段定义如下键：

- **bitmappath**: 指定位图文件所在路径
- **bitmapnumber**: 指定装载的位图文件个数，最大值为 7。可通过减少个数并删除对应位图文件以减少 MiniGUI 的存储空间占用量
- **bitmap<NR>**: 指定编号为<NR>的位图文件
- **caption**: 指定窗口标题栏图片，只在 fashion 风格下有效
- **checkmark**: 指定 ListBox 控件中复选框位图文件
- **downarrow**: 指定 ComboBox 控件中下拉按钮位图文件
- **updownarrow**: 指定 ComboBox 控件中上下箭头位图文件
- **leftrightarrow**: 指定 ComboBox 控件中左右箭头位图文件
- **spinbox_vert**: 指定 SpinBox 控件中竖直方向箭头位图文件
- **spinbox_horz**: 指定 SpinBox 控件中水平方向箭头位图文件
- **IMEctrlbtn**: 指定输入法窗口使用的位图文件，编译配置 MiniGUI 时使用了

--disable-imegb2312 选项，可以删除该键定义的位图

- **logo**: 指定关于对话框中使用的位图文件，编译配置 MiniGUI 时使用了 --disable-aboutdlg 选项，则可以删除该键定义的位图

MiniGUI.cfg 文件 bitmapinfo 段内容如下：

```
[bitmapinfo]
# Edit following line to specify bitmap files path
bitmappath=/usr/local/lib/minigui/res/bmp/
# Note that max number defined in source code is 7
bitmapnumber=2
bitmap0=capbtns.bmp
bitmap1=arrows.bmp
# use large bitmap if your default font is 16 pixel height.
# bitmap1=arrows16.bmp
bitmap2=none
bitmap3=none
bitmap4=none
bitmap5=none

# bitmap used by captionbar of main window (only Fashion style)
caption=

# bitmap used by BUTTON control
button=button.bmp

# bitmap used by LISTBOX control
checkmark=checkmark.bmp

# bitmap used by COMBOBOX control
downarrow=downarrow.bmp
updownarrow=updownarrow.bmp
leftrightharrow=leftrightharrow.bmp

# bitmap used by SPINBOX control
spinbox_vert=spinbox-vert.bmp
spinbox_horz=spinbox-horz.bmp

# bitmap used by listview control
lvfold=lvfold.bmp
lvunfold=lvunfold.bmp

# bitmap used by IME window
IMEctrlbtn=shurufa.bmp

# bitmap used by About dialog box
logo=MiniGUI256.bmp
# logo=MiniGUI16.bmp
```

3.1.11 bgpicture 段

该段定义了 MiniGUI 桌面背景图片和背景图片显示位置。该段定义如下键：

- **file**: 指定背景图片全路径名，不指定背景时指定为 none
- **position**: 指定背景图片显示位置，可取 center、upleft、downleft、upright、downright、upcenter、downcenter、vcenterleft、vcenterright、none 之一，分别表示正中、左上、左下、右上、右下、中上、中下、左中、右中、不显示。

MiniGUI.cfg 文件 bgpicture 段内容如下：

```
# background picture, use your favirate photo
file=none
position=center
# position=upleft
# position=downleft
# position=upright
# position=downright
# position=upcenter
# position=downcenter
# position=vcenterleft
# position=vcenterright
# position=none
```

3.1.12 mainwinmetrics 段

该段定义了默认的窗口元素的尺寸，一般不需要进行修改。

我们可以使用系统字体的高度来设定窗口元素的尺寸。比如在指定标题栏的高度时，用 `captiony=+4` 这样的形式指定标题栏高度是系统字体高度加 4 个像素。该段定义如下键：

- `minwidth`: 窗口最小宽度
- `minheight`: 窗口最小高度
- `border`: 窗口边框大小
- `thickframe`: 窗口厚边框大小
- `thinframe`: 窗口薄边框大小
- `captiony`: 标题栏高度
- `iconx`: 图标 x 坐标位置
- `icony`: 图标 y 坐标位置
- `menubary`: 菜单栏高度
- `menubaroffx`: 菜单栏水平偏移值
- `menubaroffy`: 菜单栏垂直偏移值
- `menuitemy`: 菜单项高度
- `intermenuitemx`: 两个水平菜单项的间距
- `intermenuitemy`: 两个垂直菜单项的间距
- `menuitemoffx`: 菜单项的水平偏移量
- `menutopmargin`: 菜单项上边距
- `menubottommargin`: 菜单项下边距
- `menuleftmargin`: 菜单项左边距
- `menurightmargin`: 菜单项右边距
- `menuitemminx`: 菜单项的最小宽度
- `menuseparatory`: 菜单水平分隔条间距
- `menuseparatorx`: 菜单垂直分隔条间距

- **sb_height**: 滚动条上按钮的高度
- **sb_width**: 滚动条上按钮的宽度
- **sb_interx**: 滚动条的水平间距
- **cxvscroll**: 垂直滚动条的宽度
- **cyvscroll**: 垂直滚动条的高度
- **cxhscroll**: 水平滚动条的宽度
- **cyhscroll**: 水平滚动条的高度
- **minbarlen**: 滚动条滑块的最小长度
- **defbarlen**: 滚动条滑块的默认长度

MiniGUI.cfg 文件 **mainwinmetrics** 段内容如下:

```
[mainwinmetrics]
minwidth=50
minheight=50
border=2
thickframe=2
thinframe=1
captiony=+4
iconx=16
icony=16
menubary=+0
menubaroffx=8
menubaroffy=5
menuitemy=+0
intermenuitemx=12
intermenuitemy=2
menuitemoffx=18
menutopmargin=4
menubottommargin=4
menuleftmargin=4
menurightmargin=4
menuitemminx=64
menuseparatory=4
menuseparatorx=4
sb_height=14
sb_width=16
sb_interx=2
cxvscroll=12
cyvscroll=12
cxhscroll=12
cyhscroll=12
minbarlen=9
defbarlen=18
```

3.1.13 windowelementcolors 段

该段定义了默认的窗口元素所使用的颜色，一般不需要进行修改。该段定义了如下键:

- **bkc_caption_normal**: 正常状态的标题栏背景色
- **fgc_caption_normal**: 正常状态的标题栏前景色
- **bkc_caption_activated**: 活动状态的标题栏背景色
- **fgc_caption_activated**: 活动状态的标题栏前景色
- **bkc_caption_disabled**: 禁止状态的标题栏背景色

- `fgc_caption_disabled`: 禁止状态的标题栏前景色
- `wec_frame_normal`: 正常状态的边框颜色
- `wec_frame_activated`: 活动状态的边框颜色
- `wec_frame_disabled`: 禁止状态的边框颜色
- `bkc_menubar_normal`: 正常状态的菜单栏背景色
- `fgc_menubar_normal`: 正常状态的菜单栏前景色
- `bkc_menubar_hilite`: 高亮状态的菜单栏背景色
- `fgc_menubar_hilite`: 高亮状态的菜单栏前景色
- `fgc_menubar_disabled`: 禁止状态的菜单栏前景色
- `bkc_menuitem_normal`: 正常状态的菜单项背景色
- `fgc_menuitem_normal`: 正常状态的菜单项前景色
- `bkc_menuitem_hilite`: 高亮状态的菜单项背景色
- `fgc_menuitem_hilite`: 高亮状态的菜单项前景色
- `fgc_menuitem_disabled`: 禁止状态的菜单项前景色
- `bkc_pppmenutitle`: 弹出菜单头背景色
- `fgc_pppmenutitle`: 弹出菜单头前景色
- `fgc_menuitem_frame`: 菜单项边框前景色
- `wec_3dbox_normal`: 三维框的正常颜色
- `wec_3dbox_reverse`: 三维框的反显颜色
- `wec_3dbox_light`: 三维框的高亮颜色
- `wec_3dbox_dark`: 三维框的灰暗颜色
- `wec_flat_border`: Flat 风格下的边框颜色
- `bkc_control_def`: 默认控件背景色
- `fgc_control_normal`: 正常状态的控件前景色
- `fgc_control_disabled`: 禁止状态的控件前景色
- `bkc_hilight_normal`: 控件选中部分的背景色
- `bkc_hilight_lostfocus`: 失去焦点后, 控件选中部分的背景色
- `fgc_hilight_normal`: 控件选中部分的前景色
- `fgc_hilight_disabled`: 禁止状态时控件选中部分的前景色
- `bkc_desktop`: 桌面的背景色
- `bkc_dialog`: 对话框的默认背景色
- `bkc_tip`: 提示框背景色

MiniGUI.cfg 文件 `windowelementcolors` 段内容如下:

```
bkc_caption_normal=0x00808080
```

```

fgc caption normal=0x00C8D0D4
bkc caption actived=0x006A240A
fgc caption actived=0x00FFFFFF
bkc caption disabled=0x00808080
fgc_caption_disabled=0x00C8D0D4

wec frame normal=0x00FFFFFF
wec frame actived=0x00FFFFFF
wec frame disabled=0x003704EA

bkc_menubar_normal=0x00CED3D6
fgc menubar normal=0x00000000
bkc menubar hilite=0x003704EA
fgc menubar hilite=0x00FFFFFF
fgc menubar disabled=0x00848284
bkc_menuitem_normal=0x00CED3D6
fgc_menuitem_normal=0x00000000
bkc_menuitem hilite=0x006B2408
fgc_menuitem hilite=0x00FFFFFF
fgc_menuitem disabled=0x00848284

bkc_pppmenutitle=0x00C0C0C0
fgc_pppmenutitle=0x006B2408
fgc_menuitem frame=0x00C66931

wec_3dbox_normal=0x00CED3D6
wec_3dbox_reverse=0x00000000
wec_3dbox_light=0x00FFFFFF
wec_3dbox_dark=0x00808080
wec_flat_border=0x00808080

bkc_control_def=0x00CED3D6
fgc_control_normal=0x00000000
fgc_control_disabled=0x00848284

bkc_hilight_normal=0x006B2408
bkc_hilight_lostfocus=0x00BDA69C
fgc_hilight_normal=0x00FFFFFF
fgc_hilight_disabled=0x00C0C0C0
bkc_desktop=0x00C08000
bkc_dialog=0x00CED3D6
bkc_tip=0x00E7FFFF

```

3.1.14 imeinfo 段

该段指定了 GB2312 输入法要装载的输入法个数及对应的模块。该段定义了如下键：

- **imetabpath**: 输入法模块所在路径
- **imenunder**: 装载输入模块个数
- **ime<NR>**: 编号为 NR 的输入法模块 pinyin、wubi、shuangpin、ziranma 等分别表示全拼、五笔、双拼、自然码等输入法模块。

如果在配置 MiniGUI 时打开了 GB2312 输入法 (--enable-imegb2312)，则会根据 imenunder 的值装载指定的输入法模块；如果 imenunder 的值为 0，则输入法模块仅提供内码输入法。

MiniGUI.cfg 文件 imeinfo 段内容如下：

```

imetabpath=/usr/local/lib/minigui/res/imetab/
imenunder=1
ime0=pinyin

```

3.1.15 默认配置文件

本节给出 MiniGUI 默认的运行时配置文件。

```
# MiniGUI Ver 2.0.x
# This configuration file is for classic window style.
#
# Copyright (C) 2002~2006 Feynman Software
# Copyright (C) 1998~2002 Wei Yongming.
#
# Web:    http://www.minigui.com
# Web:    http://www.minigui.org
#
# This configuration file must be installed in /etc,
# /usr/local/etc or your home directory. When you install it in your
# home directory, it should be named ".MiniGUI.cfg".
#
# The priority of above configuration files is ~/.MiniGUI.cfg,
# /usr/local/etc/MiniGUI.cfg, and then /etc/MiniGUI.cfg.
#
# If you change the install path of MiniGUI resource, you should
# modify this file to meet your configuration.
#
# NOTE:
# The format of this configuration file has changed since the last release.
# Please DONT forget to provide the latest MiniGUI.cfg file for your MiniGUI.
#

[system]
# GAL engine and default options
gal engine=qvfb
defaultmode=800x600-16bpp

# IAL engine
ial_engine=qvfb
mdev=/dev/input/mice
mtype=IMPS2

[fbcon]
defaultmode=1024x768-16bpp

[qvfb]
defaultmode=640x480-16bpp
display=0

# The first system font must be a logical font using RBF device font.
[systemfont]
font number=6
font0=rbf-fixed-rrncnn-8-16-ISO8859-1
font1=*-fixed-rrncnn-*-16-GB2312
font2=*-Courier-rrncnn-*-16-GB2312
font3=*-SansSerif-rrncnn-*-16-GB2312
font4=*-Times-rrncnn-*-16-GB2312
font5=*-Helvetica-rrncnn-*-16-GB2312

default=0
wchar_def=1
fixed=1
caption=2
menu=3
control=3

[rawbitmapfonts]
font_number=4
name0=rbf-fixed-rrncnn-8-16-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/8x16-iso8859-1.bin
name1=rbf-fixed-rrncnn-16-16-GB2312-0
fontfile1=/usr/local/lib/minigui/res/font/song-16-gb2312.bin
name2=rbf-fixed-rrncnn-6-12-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/6x12-iso8859-1.bin
name3=rbf-fixed-rrncnn-12-12-GB2312-0
fontfile3=/usr/local/lib/minigui/res/font/song-12-gb2312.bin
```

```

[varbitmapfonts]
font_number=6
name0=vbf-Courier-rrncnn-8-13-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/Courier-rr-8-13.vbf
name1=vbf-Helvetica-rrncnn-11-12-ISO8859-1
fontfile1=/usr/local/lib/minigui/res/font/Helvetica-rr-11-12.vbf
name2=vbf-Times-rrncnn-10-12-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/Times-rr-10-12.vbf
name3=vbf-Courier-rrncnn-10-15-ISO8859-1
fontfile3=/usr/local/lib/minigui/res/font/Courier-rr-10-15.vbf
name4=vbf-Helvetica-rrncnn-15-16-ISO8859-1
fontfile4=/usr/local/lib/minigui/res/font/Helvetica-rr-15-16.vbf
name5=vbf-Times-rrncnn-13-15-ISO8859-1
fontfile5=/usr/local/lib/minigui/res/font/Times-rr-13-15.vbf

[qpf]
font_number=3
name0=qpf-unifont-rrncnn-16-16-ISO8859-1,ISO8859-15,GB2312-0,GBK,BIG5
fontfile0=/usr/local/lib/minigui/res/font/unifont 160 50.qpf
name1=qpf-times-rrncnn-5-10-ISO8859-1,ISO8859-15
fontfile1=/usr/local/lib/minigui/res/font/smoothtimes 100 50.qpf
name2=qpf-helvetica-rrncnn-5-10-ISO8859-1,ISO8859-15
fontfile2=/usr/local/lib/minigui/res/font/helvetica_100_50.qpf
name3=qpf-micro-rrncnn-4-4-ISO8859-1,ISO8859-15
fontfile3=/usr/local/lib/minigui/res/font/micro 40 50.qpf

[truetypefonts]
font_number=3
name0=ttf-arial-rrncnn-0-0-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/arial.ttf
name1=ttf-times-rrncnn-0-0-ISO8859-1
fontfile1=/usr/local/lib/minigui/res/font/times.ttf
name2=ttf-pinball-rrncnn-0-0-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/pinball.ttf

[typelfonts]
font_number=0
name0=typel-Charter-rrncnn-0-0-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/bchr.pfb
name1=typel-Charter-rincnn-0-0-ISO8859-1
fontfile1=/usr/local/lib/minigui/res/font/bchri.pfb
name2=typel-Charter-brncnn-0-0-ISO8859-1
fontfile2=/usr/local/lib/minigui/res/font/bchb.pfb
name3=typel-Charter-bincnn-0-0-ISO8859-1
fontfile3=/usr/local/lib/minigui/res/font/bchbi.pfb
name4=typel-Courier-rrncnn-0-0-ISO8859-1
fontfile4=/usr/local/lib/minigui/res/font/dcr10.pfb
name5=typel-Courier-rincnn-0-0-ISO8859-1
fontfile5=/usr/local/lib/minigui/res/font/dcti10.pfb
name6=typel-Courier-brncnn-0-0-ISO8859-1
fontfile6=/usr/local/lib/minigui/res/font/dcbx10.pfb
name7=typel-Courier-bincnn-0-0-ISO8859-1
fontfile7=/usr/local/lib/minigui/res/font/dcbxti10.pfb
name8=typel-eufm10-rrncnn-0-0-ISO8859-1
fontfile8=/usr/local/lib/minigui/res/font/eufm10.pfb

[mouse]
dblclicktime=300

[event]
timeoutusec=300000
repeatusec=50000

[cursorinfo]
# Edit following line to specify cursor files path
cursorpath=/usr/local/lib/minigui/res/cursor/
cursornumber=23
cursor0=d_arrow.cur
cursor1=d_beam.cur
cursor2=d_pencil.cur
cursor3=d_cross.cur
cursor4=d_move.cur
cursor5=d_sizenwse.cur
cursor6=d_sizenesw.cur
cursor7=d_sizewe.cur

```

```

cursor8=d_sizens.cur
cursor9=d_uparrow.cur
cursor10=d_none.cur
cursor11=d_help.cur
cursor12=d_busy.cur
cursor13=d_wait.cur
cursor14=g_rarrow.cur
cursor15=g_col.cur
cursor16=g_row.cur
cursor17=g_drag.cur
cursor18=g_nodrop.cur
cursor19=h_point.cur
cursor20=h_select.cur
cursor21=ho_split.cur
cursor22=ve_split.cur

[iconinfo]
# Edit following line to specify icon files path
iconpath=/usr/local/lib/minigui/res/icon/
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form.ico
icon1=failed.ico
icon2=mg_help.ico
icon3=warning.ico
icon4=excalmatory.ico

# default icons for TREEVIEW control
fold=fold.ico
unfold=unfold.ico

# default icons for new OpenFileDialogBox
dir=folder.ico
file=textfile.ico

[bitmapinfo]
# Edit following line to specify bitmap files path
bitmappath=/usr/local/lib/minigui/res/bmp/
# Note that max number defined in source code is 7
bitmapnumber=2
bitmap0=capbtns.bmp
bitmap1=arrows.bmp
# use large bitmap if your default font is 16 pixel height.
# bitmap1=arrows16.bmp
bitmap2=none
bitmap3=none
bitmap4=none
bitmap5=none

# bitmap used by captionbar of main window (only Fashion style)
caption=

# bitmap used by BUTTON control
button=button.bmp

# bitmap used by LISTBOX control
checkmark=checkmark.bmp

# bitmap used by COMBOBOX control
downarrow=downarrow.bmp
updownarrow=updownarrow.bmp
leftrightharrow=leftrightharrow.bmp

# bitmap used by SPINBOX control
spinbox_vert=spinbox-vert.bmp
spinbox_horz=spinbox-horz.bmp

# bitmap used by listview control
lvfold=lvfold.bmp
lvunfold=lvunfold.bmp

# bitmap used by IME window
IMEctrlbtn=shurufa.bmp

```

```

# bitmap used by About dialog box
logo=MiniGUI256.bmp
# logo=MiniGUI16.bmp

[bgpicture]
# background picture, use your favorite photo
file=none
position=center
# position=upleft
# position=downleft
# position=upright
# position=downright
# position=upcenter
# position=downcenter
# position=vcenterleft
# position=vcenterright
# position=none

[mainwinmetrics]
minwidth=50
minheight=50
border=2
thickframe=2
thinframe=1
captiony=+4
iconx=16
icony=16
menubary=+0
menubaroffx=8
menubaroffy=5
menuitemy=+0
intermenuitemx=12
intermenuitemy=2
menuitemoffx=18
menutopmargin=4
menubottommargin=4
menuleftmargin=4
menurightmargin=4
menuitemminx=64
menuseparator=4
menuseparatorx=4
sb height=14
sb width=16
sb_interx=2
cxvscroll=12
cyvscroll=12
cxhscroll=12
cyhscroll=12
minbarlen=9
defbarlen=18

[windowelementcolors]
bkc caption normal=0x00808080
fgc caption normal=0x00C8D0D4
bkc_caption_activated=0x006A240A
fgc_caption_activated=0x00FFFFFF
bkc_caption_disabled=0x00808080
fgc_caption_disabled=0x00C8D0D4

wec frame normal=0x00FFFFFF
wec_frame_activated=0x00FFFFFF
wec_frame_disabled=0x003704EA

bkc menubar normal=0x00CED3D6
fgc menubar normal=0x00000000
bkc_menubar_hilite=0x003704EA
fgc_menubar_hilite=0x00FFFFFF
fgc_menubar_disabled=0x00848284
bkc_menuitem normal=0x00CED3D6
fgc_menuitem normal=0x00000000
bkc_menuitem_hilite=0x006B2408
fgc_menuitem_hilite=0x00FFFFFF
fgc_menuitem_disabled=0x00848284

bkc_pppmenutitle=0x00C0C0C0

```

```
fgc boomenutitle=0x006B2408
fgc menuitem frame=0x00C66931

wec 3dbox normal=0x00CED3D6
wec 3dbox_reverse=0x00000000
wec 3dbox_light=0x00FFFFFF
wec 3dbox_dark=0x00808080
wec flat border=0x00808080

bkc_control_def=0x00CED3D6
fgc_control_normal=0x00000000
fgc_control_disabled=0x00848284

bkc hilight normal=0x006B2408
bkc hilight_lostfocus=0x00BDA69C
fgc_hilight_normal=0x00FFFFFF
fgc_hilight_disabled=0x00C0C0C0
bkc_desktop=0x00C08000
bkc_dialog=0x00CED3D6
bkc_tip=0x00E7FFFF

[imeinfo]
imetabpath=/usr/local/lib/minigui/res/imetab/
imenumber=1
ime0=pinyin
```

3.2 内建式配置选项

在使用内建式资源（incore resources）时，MiniGUI 不需要 MiniGUI.cfg 配置文件，相应的配置选项在 src/sysres/mgetc.c 文件中指定。

和 MiniGUI.cfg 配置文件的结构类似，mgetc.c 中定义了一个 ETCSECTION 类型的结构数组 mgetc_sections 来对应配置文件内各个段信息；定义了一个 ETC_S 类型的结构变量 MGETC 来对应配置文件。

3.2.1 ETCSETCTION 结构

ETCSECTION 结构定义在 minigui.h 中，内容如下

```
/** Etc The current config section information */
typedef struct ETCSECTION
{
    /** Allocated number of keys */
    int key_nr_alloc;
    /** Key number in the section */
    int key_nr;
    /** Name of the section */
    char *name;
    /** Array of keys */
    char** keys;
    /** Array of values */
    char** values;
} ETCSECTION;
```

其中：key_nr_alloc 为其他配置选项的接口，内建时必须取 0，key_nr 指定段中键（key）的个数，name 指定该段的名称，keys 和 values 分别指定键（key）数组和值（value）数组。键数组和值数组

中元素的个数应和 **key_nr** 指定的键个数一致。

mgetc.c 文件中数组 **mgetc_sections** 定义如下：

```
static ETCSECTION mgetc_sections [] =
{
    {0, 4, "system",          SYSTEM_KEYS,    SYSTEM_VALUES},
    {0, 2, "qvfb",           QVFB_KEYS,     QVFB_VALUES},
    {0, 2, "wvfb",           FBCON_KEYS,    FBCON_VALUES},
    {0, 2, "shadow",         FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "fbcon",          FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "dummy",          FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "em85xxosd",      FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "svpxxosd",       FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "utpmc",          FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "bf533",          FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "mb93493",        FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "commlcd",        FBCON_KEYS,    FBCON_VALUES},
    {0, 1, "dfb",            FBCON_KEYS,    FBCON_VALUES},
    {0, TABLESIZE(SYSTEMFONT_KEYS), "systemfont", SYSTEMFONT_KEYS, SYSTEMFONT_VALUES},
    {0, 1, "cursorinfo",     CURSORINFO_KEYS, CURSORINFO_VALUES},
    {0, 1, "iconinfo",       ICONINFO_KEYS,  ICONINFO_VALUES},
    {0, 1, "bitmapinfo",     BITMAPINFO_KEYS, BITMAPINFO_VALUES},
    /* optional sections */
    {0, 2, "bgpicture",      BGPICTURE_KEYS, BGPICTURE_VALUES},
    /*
    {1, "mouse", MOUSE_KEYS, MOUSE_VALUES},
    */
    {0, 2, "event",          EVENT_KEYS,    EVENT_VALUES},
#ifdef IME_GB2312
    {0, 3, "imeinfo",        IMEINFO_KEYS, IMEINFO_VALUES},
#endif
#ifdef TTF_SUPPORT
    {0, 5, "truetypefonts",  TTFINFO_KEYS, TTFINFO_VALUES},
#endif
};
```

mgetc_sections 数组中定义的段基本上都是必须的（**fbcon** 和 **qvfb** 可只定义适用的一个），其它注释掉的段是可选的。这几个段的含义和 **MiniGUI.cfg** 配置文件中相应段的含义是一样的。用户一般情况下只需修改“**system**”段和“**fbcon**”段的键值数组（**SYSTEM_VALUES** 和 **FBCON_VALUES**），指定 MiniGUI 的 GAL 引擎、IAL 引擎和显示模式。键值数组 **SYSTEM_VALUES** 和 **FBCON_VALUES** 根据编译配置选项定义在对应的 **mgetc-xxx.c** 文件内。例如对应 PC 环境下的对应文件为 **mgetc-pc.c**。

“**systemfont**”段定义了系统使用的内建式字体，MiniGUI 2.0.x 目前支持 ISO8859-1、GB2312 字符集的 12 点阵 RBF 字体、BIG5 字符集的 12 点阵字体和 SHIFT-JIS 字符集 12 点阵字体、QPF 字体。在采用内建式资源时，TTF 字体及 Type1 字体不被支持。

3.2.2 ETC_S 结构

ETC_S 结构定义在 **minigui.h**，内容如下：

```
/** ETC_S The current config file information*/
typedef struct ETC_S
{
```

```
/** Allocated number of sections */
int sect_nr_alloc;
/** Number of sections */
int section_nr;
/** Pointer to section arrays */
PETCSECTION sections;
} ETC S;
```

其中: `sect_nr_alloc` 为其他配置选项的接口, 内建时必须取 0, `sect_nr` 指定段的个数, `sections` 指定为 `ETCSECTION` 类型的结构数组, 其元素个数应不小于第一项指定的值。

`mgetc.c` 文件中数组 `mgetc_sections` 定义如下:

```
ETC S MGETC = {0, TABLESIZE (mgetc_sections), mgetc_sections};
```

`MGETC` 结构中指定段的个数为 `TABLESIZE(mgetc_sections)`, 段数组为上述定义的数组 `mgetc_sections`。

3.2.3 mgetc.c 文件清单

```
/*
** $Id: mgetc.c,v 1.127 2006/06/14 02:26:59 xwyan Exp $
**
** mgetc.c: definitions for incore resource.
**
** Copyright (C) 2003 ~ 2006 Feynman Software.
**
** Create date: 2003/09/22
**
*/
#include <stdio.h>

#include "common.h"
#include "minigui.h"

#ifdef INCORE RES

#define ETC CONFIG EVENT
#ifdef DYNAMIC_LOAD
#include "mgetc-pc.c"
#include "mgetc-uptech.c"
#include "mgetc-palm2.c"
#include "mgetc-wifi.c"
#include "mgetc-bf533.c"
#include "mgetc-mb93493.c"
#include "mgetc-em86.c"

#include "mgetc-em85.c"
#include "mgetc-utpmc.c"
#include "mgetc-swlinux.c"
#include "mgetc-ucosii-arm3000.c"
#include "mgetc-ucosii-skyeye.c"
#include "mgetc-em85xx.c"
#include "mgetc-ipaq.c"
#include "mgetc-fft7202.c"
#include "mgetc-vxsim.c"
#include "mgetc-vxi386.c"
#include "mgetc-dm270.c"
#include "mgetc-xscale.c"
#include "mgetc-embest2410.c"
#include "mgetc-rm9200.c"
#include "mgetc-hh2410r3.c"
```

```

#include "mgetc-hh2440.c"
#include "mgetc-hh44b0.c"
#include "mgetc-ecos-ipaq-wifi.c"
#include "mgetc-cygwin.c"
#include "mgetc-win32.c"
#include "mgetc-threadx.c"
#include "mgetc-nucleus.c"
#include "mgetc-nucleus-mnt.c"
#include "mgetc-ecos-default.c"
#include "mgetc-psos-default.c"
#include "mgetc-ose-mx21.c"

static char *SYSTEM_KEYS[] = {"gal engine", "ial engine", "mdev", "mtype"};

static char *FBCON_KEYS[] = {"defaultmode"};

static char *QVFB_KEYS[] = {"defaultmode", "display"};
static char *QVFB_VALUES[] = {"640x480-16bpp", "0"};

static char *CURSORINFO_KEYS[] = {"cursornumber"};
static char *CURSORINFO_VALUES[] = {"23"};

static char *ICONINFO_KEYS[] = {"iconnumber"};
static char *ICONINFO_VALUES[] = {"5"};

static char *BITMAPINFO_KEYS[] = {"bitmapnumber"};
static char *BITMAPINFO_VALUES[] = {"6"};
static char *BGPICTURE_KEYS[] = {"position", "file"};
#if 0
/* for MGDesktop */
static char *BGPICTURE_VALUES[] = {"center", "/usr/local/lib/minigui/res/bmp/bk.jpg"};
#else
static char *BGPICTURE_VALUES[] = {"none", ""};
#endif

/*
static char *MOUSE_KEYS[] = {"dblclicktime"};
static char *MOUSE_VALUES[] = {"300"};
*/

#ifdef ETC_CONFIG_EVENT
static char *EVENT_KEYS[] = {"timeoutusec", "repeatusec"};
static char *EVENT_VALUES[] = {"300000", "50000"};
#endif

#ifdef IME_GB2312
static char* IMEINFO_KEYS[] = {"imetabpath", "imenumber", "ime0"};
static char* IMEINFO_VALUES[] = {"usr/local/lib/minigui/res/imetab/", "1", "pinyin"};
#endif

#ifdef TTF_SUPPORT
static char* TTFINFO_KEYS[] = {"font number", "name0", "fontfile0", "name1", "fontfile1"};
static char* TTFINFO_VALUES[] = {"2",
    "ttf-arial-rrncnn-0-0-ISO8859-1", "/usr/local/lib/minigui/res/font/arial.ttf",
    "ttf-times-rrncnn-0-0-ISO8859-1", "/usr/local/lib/minigui/res/font/times.ttf"
};
#endif

static ETCSECTION mgetc sections [] =
{
    {0, 4, "system",          SYSTEM_KEYS,    SYSTEM_VALUES},
    {0, 2, "qvfb",           QVFB_KEYS,    QVFB_VALUES},
    {0, 2, "wvfb",           FBCON_KEYS,   FBCON_VALUES},
    {0, 2, "shadow",         FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "fbcon",          FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "dummy",          FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "em85xxosd",      FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "svpxxosd",       FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "utpmc",          FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "bf533",          FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "mb93493",        FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "commlcd",        FBCON_KEYS,   FBCON_VALUES},
    {0, 1, "dfb",            FBCON_KEYS,   FBCON_VALUES},
    {0, TABLESIZE(SYSTEMFONT_KEYS), "systemfont", SYSTEMFONT_KEYS, SYSTEMFONT_VALUES},

```

```
{0, 1, "cursorinfo",  CURSORINFO KEYS, CURSORINFO VALUES},
{0, 1, "iconinfo",    ICONINFO KEYS,   ICONINFO VALUES},
{0, 1, "bitmapinfo",  BITMAPINFO KEYS, BITMAPINFO VALUES},
/* optional sections */
{0, 2, "bgpicture",   BGPICTURE_KEYS, BGPICTURE_VALUES},
/*
{1, "mouse", MOUSE KEYS, MOUSE VALUES},
*/
{0, 2, "event", EVENT KEYS, EVENT VALUES},
#ifdef _IME_GB2312
{0, 3, "imeinfo", IMEINFO_KEYS, IMEINFO_VALUES},
#endif
#ifdef TTF_SUPPORT
{0, 5, "truetypefonts", TTFINFO KEYS, TTFINFO VALUES},
#endif
};

ETC S MGETC = {0, TABLESIZE (mgetc sections), mgetc sections};
#endif /* !DYNAMIC LOAD */
#endif /* INCORE RES */
```

3.3 配置示例

对运行时配置文件的修改大多数情况下限定在不多的几个段内，主要是 **system** 段和字体相关的几个段。本节给出两个配置示例。

3.3.1 只支持 ISO8859-1 字符显示的运行时配置

1) 配置文件

```
# The first system font must be a logical font using RBF device font.
[systemfont]
font number=1
font0=rbf-fixed-rrncnn-8-16-ISO8859-1

default=0
wchar_def=0
fixed=0
caption=0
menu=0
control=0

[rawbitmapfonts]
font number=1
name0=rbf-fixed-rrncnn-8-16-ISO8859-1
fontfile0=/usr/local/lib/minigui/res/font/8x16-iso8859-1.bin

[varbitmapfonts]
font number=0

[qpf]
font number=0

[truetypefonts]
font number=0

[typelfonts]
font number=0
```

2) 内建式配置选项

```
static char *SYSTEMFONT_KEYS[] =
{"font number", "font0", "default", "wchar def", "fixed", "caption", "menu", "control"};

static char *SYSTEMFONT_VALUES[] =
{
    "1", "rbf-fixed-rrncnn-8-16-ISO8859-1", "0", "0", "0", "0", "0", "0"
};
```

3.3.2 指定不同的图形引擎和输入引擎

1) 配置文件

```
[system]
# GAL engine and default options
gal_engine=commlcd

# IAL engine
ial_engine=auto

mdev=/dev/ts
mtype=IMPS2
```

2) 内建式配置选项

```
static char *SYSTEM_KEYS[] = {"gal_engine", "ial_engine", "mdev", "mtype"};

static char *SYSTEM_VALUES[] = {"commlcd", "auto", "/dev/ts", "IMPS2"};
```


4 在 Windows 平台上开发 MiniGUI 应用程序

飞漫软件为习惯于在 Windows 平台上开发应用软件的开发人员提供了如下两种方法：

- 使用 MiniGUI for Win32 开发包开发 MiniGUI 应用程序。这是预先编译好的针对 Win32 平台的标准开发包，其中包含 wvfb 程序、MiniGUI 的两个函数库（libminigui 和 libmgext）及对应的头文件。
- 使用 MiniGUI SDK for Win32¹¹。这是 MiniGUI 增值版产品的一个可选组件，包含有完整源代码，可以方便用户根据自己的需求建立自己的 MiniGUI for Win32 开发包。

通过 MiniGUI for Win32 开发包，或者 MiniGUI SDK for Win32 组件产品，开发人员就可以在 Windows 环境下编译和调试 MiniGUI 应用程序了。

本章将简单介绍 MiniGUI for Win32 开发包的使用。用户可联系飞漫软件购买 MiniGUI SDK for Win32 组件产品。

MiniGUI for Win32 开发包保存在增值版产品光盘的 win32-dev/ 目录下，文件名为 minigui-dev-1.6.9-win32.tar.gz。您也可以访问飞漫软件网站，从如下地址下载 MiniGUI for Win32 开发包：

```
http://www.minigui.com/download/cindex.shtml
```

为了在 Windows 平台上开发 MiniGUI 应用程序，您需要在 Windows 上安装 MS Visual Studio 98 集成开发工具。将上述开发包解开到 Windows 平台上的任意目录下，然后根据 README 文件中的描述在 Visual Studio 中打开 helloworld 工程，如图 4.1 所示。

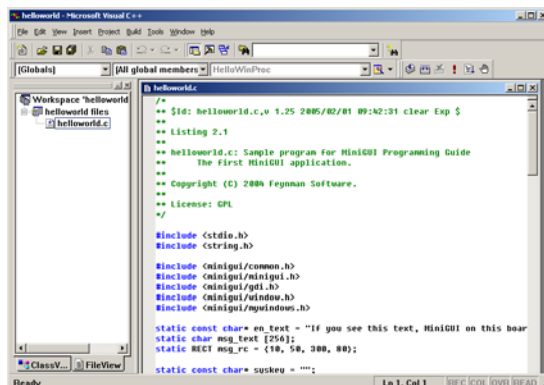


图 4.1 打开 MiniGUI 的 helloworld 工程

¹¹ MiniGUI SDK for Win32 只能编译为 MiniGUI-Threads 模式，且只能使用内嵌资源模式。因平台限制，不能用来装载 JPEG、PNG 格式的图片，同时无法提供对 TrueType 和 Type1 字体的支持（这些特性均需要第三方函数库的支持）。

编译后，需首先运行开发包中的 `wvfb` 程序，然后运行 `helloworld` 程序（注意需要将 `helloworld.exe` 文件复制到 `dll/` 目录中，以便程序能够在当前目录下找到 `minigui.dll` 等 DLL 文件），将在 `wvfb` 中看到运行效果，如图 4.2 所示。



图 4.2 在 Windows 平台上编译并运行 MiniGUI 应用程序

参考上面的 `helloworld` 工程，您可以在 VC 集成开发环境中创建自己的 MiniGUI 应用程序工程，从而可以充分利用 VC 开发环境提供的便利来开发和调试 MiniGUI 应用软件。但需要注意如下事项：

- MiniGUI for Win32 开发包是预编译的函数库，因此，其中的功能特性是固定，包括编译时选项和运行时选项。另外，该开发包只提供对 MiniGUI-Threads 运行模式的支持。
- 利用 MiniGUI for Win32 开发包开发 MiniGUI 应用程序时，请不要调用 Windows 的特有函数接口，因为这些函数接口可能在最终的操作系统上并不存在。

附录 A 常见问题及解答

A.1 GPL 版本问题

1. 我使用 MiniGUI 的 GPL 版本需要向飞漫软件技术有限公司支付授权费用吗？

答：MiniGUI 的用户或者爱好者可以通过飞漫软件的网站获得 MiniGUI GPL 版本的源代码，只要用户遵循 GPL 条款的精神使用 MiniGUI，就无需为使用 MiniGUI 而向飞漫软件支付授权费用，但这要求您基于 MiniGUI 的应用程序也必须遵循 GPL 条款发布；如果您的专有或商用软件产品使用了 MiniGUI，而又不想遵循 GPL 条款发布您的应用软件，则应该向 MiniGUI 的版权所有者北京飞漫软件技术有限公司支付商业授权费用。

2. 使用 MiniGUI GPL 版本时，哪些行为会侵犯飞漫软件的合法权益？

答：飞漫软件作为多个自由软件项目的版权所有者，以源代码方式发布这些自由软件，其目的是为了给用户了解软件内部机制，并根据需求进行适当定制的自由和方便。但是，由于大多数用户对 GPL 条款的不了解，有时会无意中出现违反 GPL 条款的情况，这对商业用户尤其不利。这些行为主要有：

- 盗用自由软件的部分或全部代码并使用在其他场合；更甚者，将 MiniGUI 等自由软件据为己有，改头换面，以私有软件形式出售。这种行为属于严重侵权行为，自由软件的版权拥有人会追究这种侵权行为，侵权人将面对严重的刑事和民事处罚。
- 对自由软件源代码进行了修改或者增强，并用于商业目的，但并未遵守 GPL 条款公开修改后的源代码。常见情况：修改 MiniGUI 某些控件的实现以满足其需求；修改 MiniGUI 的输入法实现以满足其需求等等。

按照 GPL 条款，采用 MiniGUI 开发图形用户界面应用程序也应该遵循 GPL 条款发布。如果您使用 MiniGUI 的应用程序既没有遵循 GPL 条款发布，也没有购买 MiniGUI 的商业授权，则这种行为属于典型的软件盗版行为。

A.2 应用问题

3. MiniGUI 可用于哪些产品领域？

答：在手持终端领域、数字控制领域、POS 机/彩票机、电力控制以及信息终端等领域，均有使用 MiniGUI 且成功上市的产品。有关某些成功案例或产品的介绍，您可以到 <http://www.minigui.com/project/cindex.shtml> 上查阅。

4. MiniGUI 的稳定性如何？

答：这个问题很难回答，因为有时引起系统不稳定的因素可能在应用程序，而不是底层函数库。不过，我们可以给您提供一些数字供参考：

- 在一个比较复杂的 MiniGUI 应用程序中，模拟用户的按键操作进行多个窗口之间的切换测试，连续测试 2 天共约 100,000 次按键不会出现任何问题。
- 架构在 MiniGUI 之上的许多工业控制系统（有的甚至要求达到毫秒级的实时性），已经应用于真正的工业场合，并稳定运行。

A.3 移植性问题

5. MiniGUI 支持哪些操作系统？

答：MiniGUI 可支持 Linux/uClinux、VxWorks、ThreadX、Nucleus、pSOS、OSE、eCos 以及 uC/OS-II 等操作系统。

6. MiniGUI 在哪些嵌入式 CPU 上成功运行过，最低的 CPU 主频大概是多少？

答：MiniGUI 在含有 MMU 单元的 ARM 系列 CPU（比如 StrongARM、xScale 等），PowerPC、MIPS 系列 CPU（比如 EP7312、VR4181 等）、无 MMU 单元的 m68k 系列 CPU 上都有过成功运行的实例。在这些 CPU 中，最低的 CPU 主频大约为 16 MHz，CPU 运算能力为 10 MIPS。

7. MiniGUI 能支持单色 LCD 显示屏吗？

答：可以。MiniGUI 能支持从单色、四色、十六色到 256 色、4096 色、65536 色等几乎所有颜色深度的显示屏。

8. MiniGUI 可在多大的显示屏上正常运行？

答：理论上讲，MiniGUI 的正常运行不受显示屏大小的限制。

A.4 编译问题

9. 我在 Linux 上打开了 MiniGUI 的 TrueType 字体支持选项，为什么编译时出现了许多错误？

答：这是因为您系统中的 TrueType 字体支持库 libtff 版本太高的原因造成的。MiniGUI 使用 LibTTF 1.3.1 版本。在 RedHat Linux 7.0 等某几个特定的发行版系统中，默认安装的是 LibTTF 2.0 版本的函数库，并且未进行版本差异处理，因此会由于版本差异而导致编译时出现大量问题。这时，您

可以安装 LibTTF 的 1.3.1 版本，或者使用 `--disable-ttfsupport` 选项禁止 TrueType 字体支持。

10. 为什么在 Linux 上编译 MiniGUI 函数库的时候，有时会出现

```
can not make hard link filename.o to filename.lo.
```

这样的错误？

答：符号链接和硬链接是 UNIX 文件系统中才具有的文件类型。如果您在非 UNIX 文件系统上编译由 Automake/Autoconf 脚本维护的函数库，则无法建立硬链接或符号链接。请检查您的文件系统，是否是 FAT32 之类的文件系统。

11. 在使用文件对话框时，无法成功链接生成可执行文件，提示

```
undefined reference to showopendialog
```

这是什么原因导致的？

答：文件对话框的接口包含在 MinGUI 扩展库中。如果要使用，则要包括两个头文件：`<minigui/mgext.h>` 和 `<minigui/filedlg.h>`，并且在编译的时候要链接 `-lmgext` 库。另外，在非 UNIX 类操作系统上，因为通常不提供文件系统的支持，因此，文件对话框的接口也是不包括的。

12. MiniGUI 中的 64 位数据类型是否可以不使用，我的系统中没有相应类型？

答：64 位数据是用于处理一些复杂图形的生成的，如椭圆生成器之类。如果您不需要，或者碰到问题的话，可以使用

```
--disable-fixedmath
```

配置选项将该部分关闭，这样相应的算法仍然可以工作，但是精确度会有所降低。

A.5 输入引擎

13. 在运行 Linux 的 PC 上，MiniGUI 到底支持哪些鼠标类型？

答：MiniGUI 在 PC Linux 上运行时，目前所支持的鼠标协议有 MS、MS3、PS2、智能 PS2 (IMPS2) 等常见的鼠标协议。目前的 MiniGUI 还不支持智能鼠标上的滚轮以及中键。

14. 我想在 Linux 上支持老式的串口鼠标，该怎么办？

答：MiniGUI 可通过 GPM 支持几乎所有的鼠标类型。配置过程如下：

- 1) 运行 `gpm -k` 命令杀掉正在运行的 `gpm`。

- 2) 运行 `mouse-test` 命令确认自己的鼠标设备和协议。
- 3) 运行 `gpm`，指定鼠标设备和协议：

```
# gpm -R -t <yourmousetype> -m <yourmousedevice>
```

- 4) 编辑 `MiniGUI.cfg` 文件，将 `mtype` 设置为 `gpm`；将 `mdev` 设置为 `/dev/gpmdata`：

```
[system]
...
mtype=gpm
mdev=/dev/gpmdata
```

然后启动 MiniGUI 就可以了。需要注意的是，用 `gpm` 设置鼠标格式的时候，可以使用 `-R` 参数，`gpm` 的 `-R` 参数可用来将原有鼠标协议转换成 GPM 所定义的鼠标协议，并出现在 `/dev/gpmdata` 文件上。

A.8 运行时问题

15. 在 Linux 上，如何才能关闭 MDE 程序启动之后的输入法条？

答：MDE 程序启动时的输入条可以用左边的 `<Ctrl>` 键切换；另外，也可以通过配置 `MiniGUI.cfg` 文件里的 `imenu` 选项：

```
[imeinfo]
imenu=2
```

将 `imenu` 设置为 `imenu=0`，启动时就不会出现输入法条了。

16. 在 Linux 上，怎么样把 MiniGUI 的运行画面保存为图片？

答：在运行 MiniGUI 程序时，按 `<PrntScrn>` 键就会在当前目录保存整个屏幕的 BMP 文件，文件名为 `0-<NO>.bmp`，其中 `<NO>` 是按键次数的编号；按 `<Ctrl+PrntScrn>` 键，可保存当前活动主窗口的 BMP 文件，文件名为 `<HWND>-<NO>.bmp`，其中 `<HWND>` 是活动主窗口的句柄，`<NO>` 是按键次数编号。

17. 在 Linux 上，我运行 MDE 的 `mginit` 程序时，为什么显示两个对话框之后程序就退出了？

答：这通常是因为您编译安装的 MiniGUI 不含 PNG 图像文件的支持功能导致的。在某些 Linux 发行版上（比如早期的 TurboLinux 版本），因为所含的 PNG 图像支持库（即 `libpng`）版本太早，在进行 MiniGUI 配置时自动禁止了对 PNG 图像的支持而导致的。这种情况下，MiniGUI 的 `LoadBitmapFromFile` 函数无法正确装载 PNG 图像文件，而 MDE 的 `mginit` 在启动时恰好要装载两个

PNG 图像文件，这样，由于无法装载图像文件，**mginit** 程序退出。

解决这个问题的办法有两个。第一，您可以从网上下载安装最新版的 **libpng** 库；第二，修改 **mginit.rc** 文件中 **[mginit]** 段的 **nr** 键值，使之小于 8。

另外一个可能导致这个错误的原因是，您没有在 **mginit** 程序所在的目录启动这个程序。请使用 **cd** 命令改变到 **mginit** 文件所在的目录，然后运行该程序。

18. 在 MiniGUI-Lite 或 MiniGUI-Processes 运行模式中，我如何从 MiniGUI 切换到其它控制台？

答：在 MiniGUI-Lite 或 MiniGUI-Processes 运行模式中，可以按 **<Right_Ctrl + Fx>** 组合键，从 MiniGUI 中切换到其它虚拟控制台。另外，您还可以通过 **<Ctrl+Alt+Backspace>** 组合键强制退出 MiniGUI。MiniGUI-Threads 运行模式目前还不提供类似的功能。

A.9 常见错误信息

19. 在 Linux 上，为什么我运行 MDE 包中的程序时，会告诉我

```
AttachSharedResource: No such file or directory
Error in step 6: Can not attach shared resource!
Initialize minigui failure when using /etc/MiniGUI.cfg as cfg file.
```

答：如果您的 MiniGUI 被配置为 MiniGUI-Lite 或者 MiniGUI-Processes，则应该首先运行 **mginit** 程序。MiniGUI-Lite 或者 MiniGUI-Processes 运行模式基于客户/服务器体系运行，在运行客户程序之前，必须启动一个名称为 **mginit** 的服务器程序。在 MDE 包中，首先运行 **mginit/** 目录下的 **mginit** 服务器程序，然后就可以运行其它目录下的演示程序了。

20. 在 Linux 上，我在运行 MiniGUI 时，遇到如下提示信息：

```
GAL ENGINE: Error when opening /dev/fb0: Permission denied. Please check your kernel config.
GAL: Init GAL engine failure.
Error in step 3: Can not initialize graphics engine!
Initialize minigui failure when using /usr/local/etc/MiniGUI.cfg as cfg file
```

为什么会出现这样的情况？

答：这是因为您的操作系统尚未激活 **FrameBuffer** 驱动程序，或者，您的 **/dev/fb0** 设备的访问许可不正确。

21. MiniGUI-Processes 或者 MiniGUI-Lite 运行模式下，我在运行 MDE 的 **mginit** 时，出现如下错误消息：

```
Error in step 2 : There is already an instance of minigui.  
Initialize minigui failure when using /usr/local/etc/MiniGUI.cfg as config file.
```

请问是怎么回事？

答：这种情况一般有两个原因。第一，您已经运行了一个 **mginit** 程序。第二，您上次运行 **mginit** 时没有正常退出系统。如果是第二个原因，您只需删除 **/var/tmp/** 目录下的 **minigui** 和 **mginit** 两个文件即可。如果还有错误，请重新启动计算机。

22. 出现错误信息：

```
NEWGAL: Does not find matched engine: fbcon.  
Error in step 3: Can not get graphics engine information!
```

是什么原因？

答：这是因为指定的图形引擎无法正确初始化导致的。请检查您的运行时配置选项。

23. 在 Linux 上，出现下面的显示错误信息：

```
vesafb does not support changing the video mode
```

是什么意思？

答：这句话是一条警告消息，可以忽略。它是针对 **VESA FrameBuffer** 驱动程序而言的。**VESA FrameBuffer** 驱动程序不支持运行过程中的显示模式切换，而只能通过内核的引导选项来确定显示模式。一旦确定，就无法改变，除非修改引导选项并重新启动系统。

24. 在 Linux 上，出现下面的显示错误信息：

```
NEWGAL: No video mode large enough for the resolution specified.  
NewGAL: Set video mode failure.
```

是什么意思？

答：这是因为您在 **MiniGUI.cfg** 中指定的显示分辨率比图形引擎能达到的显示分辨率大。您可以试着修改运行时配置选项指定一个较小的显示分辨率。

附录 B 索引

__CYGWIN__	15	_CTRL_NEWTOOLBAR	27
__DARWIN__	15	_CTRL_PROGRESSBAR	27
__ECOS__	14	_CTRL_PROPSHEET	27
__LINUX__	14	_CTRL_SCROLLVIEW	27
__MGEXT_LIB__	27	_CTRL_SIMEDIT	26
__MINIGUI_LIB__	27	_CTRL_SLEDIT	26
__NOUNIX__	14	_CTRL_STATIC	26
__NUCLEUS__	15	_CTRL_TEXTEDIT	27
__OSE__	15	_CTRL_TOOLBAR	27
__TARGET_FBCON__	15	_CTRL_TRACKBAR	27
__TARGET_QVFB__	15	_CURSOR_SUPPORT	21
__TARGET_STB810__	15	_CYRILLIC_SUPPORT	22
__TARGET_UNKNOWN__	15	_DFB_IAL	18
__TARGET_VFANVIL__	15	_DOUBLE_CLICK	21
__TARGET_VXI386__	15	_DUMMY_IAL	18
__TARGET_WVFB__	15	_EUCJP_SUPPORT	23
__THREADX__	15	_EUCKR_SUPPORT	23
__uClinux__	14	_EXT_CTRL_ANIMATION	27
__UCOSII__	14	_EXT_CTRL_COOLBAR	27
__VXWORKS__	14	_EXT_CTRL_GRIDVIEW	27
__WINBOND_SWLINUX__	14	_EXT_CTRL_ICONVIEW	27
_ADV_2DAPI	21	_EXT_CTRL_LISTVIEW	27
_ARABIC_SUPPORT	22	_EXT_CTRL_MONTHCAL	27
_AUTO_IAL	18	_EXT_CTRL_SPINBOX	27
_BIG5_SUPPORT	23	_EXT_CTRL_TREEVIEW	27
_CLIPBOARD_SUPPORT	21	_EXT_SKIN	27
_COMM_IAL	18	_FIXED_MATH	21
_COORD_TRANS	17	_FLAT_WINDOW_STYLE	26
_CTRL_BUTTON	26	_GB_SUPPORT	23
_CTRL_COMBOBOX	27	_GB18030_SUPPORT	23
_CTRL_LISTBOX	26	_GBK_SUPPORT	23
_CTRL_MENUBUTTON	27	_GIF_FILE_SUPPORT	26

_GPM_SUPPORT	20	_LATIN2_SUPPORT	22
_GRAY_SCREEN	26	_LATIN3_SUPPORT	22
_GREEK_SUPPORT	22	_LATIN4_SUPPORT	22
_HAVE_TEXT_MODE	21	_LATIN5_SUPPORT	22
_HEBREW_SUPPORT	22	_LATIN6_SUPPORT	22
_IME_GB2312	22	_LATIN7_SUPPORT	22
_IME_GB2312_PINYIN	22	_LATIN8_SUPPORT	23
_IMPS2_SUPPORT	20	_LATIN9_SUPPORT	23
_INCORE_RES	21	_LBM_FILE_SUPPORT	26
_INCOREFONT_COURIER	23	_LIB_VCONGUI	21
_INCOREFONT_HELV	23	_LITE_VERSION	16
_INCOREFONT_SANSSERIF	23	_MAX_TTF_CACHE	25
_INCOREFONT_SYMBOL	23	_MGRM_PROCESSES	16
_INCOREFONT_VGAS	23	_MGRM_STANDALONE	16
_INCORERBF_BIG5_12	23	_MGRM_THREADS	16
_INCORERBF_BIG5_16	23	_MISC_ABOUTDLG	21
_INCORERBF_BIG5_24	23	_MISC_MOUSECALIBRATE	20
_INCORERBF_GB12	23	_MISC_SAVESCREEN	21
_INCORERBF_GB16	23	_MOVE_WINDOW_BY_MOUSE	21
_INCORERBF_GB16_H	23	_MS_SUPPORT	20
_INCORERBF_GB24	23	_MS3_SUPPORT	20
_INCORERBF_KJ12	23	_NATIVE_IAL_ENGINE	18
_INCORERBF_KJ14	23	_NEWGAL_ENGINE_COMMLCD	16
_INCORERBF_KJ16	23	_NEWGAL_ENGINE_DFB	17
_INCORERBF_LATIN1_16	23	_NEWGAL_ENGINE_DUMMY	16
_INCORERBF_LATIN1_24	23	_NEWGAL_ENGINE_FBCON	16
_JPG_FILE_SUPPORT	26	_NEWGAL_ENGINE_QVFB	16
_KBD_LAYOUT_DE	20	_NEWGAL_ENGINE_SHADOW	16
_KBD_LAYOUT_DELATIN1	20	_NEWGAL_ENGINE_WVFB	16
_KBD_LAYOUT_ES	20	_PC3D_WINDOW_STYLE	26
_KBD_LAYOUT_ESCP850	20	_PCX_FILE_SUPPORT	26
_KBD_LAYOUT_FR	20	_PHONE_WINDOW_STYLE	26
_KBD_LAYOUT_FRPC	20	_PNG_FILE_SUPPORT	26
_KBD_LAYOUT_IT	20	_PS2_SUPPORT	20
_LATIN10_SUPPORT	23	_QPF_SUPPORT	23

_QVFB_IAL	18	coortrans_cw	17
_RANDOM_IAL	18	ctrlbutton	26
_RBF_SUPPORT	23	ctrlcombobox	27
_ROT_DIR_CW	17	ctrllistbox	26
_SAVE_BITMAP	21	ctrlmenubtn	27
_SHIFTJIS_SUPPORT	23	ctrlnewtoolbar	27
_STAND_ALONE	16	ctrlpgbar	27
_TGA_FILE_SUPPORT	26	ctrlpropsheet	27
_THAI_SUPPORT	22	ctrlscrollview	27
_TINY_SCREEN	26	ctrlsimit	26
_TTF_CACHE_SIZE	25	ctrlsledit	26
_TTF_CACHE_SUPPORT	23	ctrlstatic	26
_TTF_SUPPORT	23	ctrltextedit	27
_TYPE1_SUPPORT	23	ctrltoolbar	27
_UNICODE_SUPPORT	23	ctrltrackbar	27
_USE_FIXED_SB_BARLEN	26	cursor	21
_USE_MINIGUIENTRY	21	cyrillicsupport	22
_USE_OWN_MALLOC	27	dblclk	21
_USE_OWN_PTHREAD	27	debug	21
_USE_OWN_STDIO	27	dfbial	18
_VBF_SUPPORT	23	dm270ial	18
_WVFB_IAL	18	dmgstbial	19
aboutdlg	21	dummyial	18
abssigial	19	em85ial	19
ads7846ial	19	em86ial	19
adsial	18	embest2410ial	19
adv2dapi	21	embest44b0ial	19
arabicsupport	22	ep7211ial	18
arm3000ial	18	eucjpsupport	23
autoial	18	euckrsupport	23
big5support	23	evmv10ial	18
classic	26	extctrlanimation	27
clipboard	21	extctrlcoolbar	27
commial	18	extctrlgridview	27
coortrans_ccw	17	extctrliconview	27

extctrltreeview	27	kbdescp850.....	20
extctrlmonthcal	27	Kbdfr	20
extctrlspinbox.....	27	Kbdfrc	20
extctrltreeview.....	27	Kbdit.....	20
extskin	27	l7200ial	18
fashion	26	latin10support	23
fft7202ial	19	latin2support	22
fipial	19	latin3support	22
fixedbarlen	26	latin4support	22
fixedmath	21	latin5support	22
flat.....	26	latin6support	22
fontcourier.....	23	latin7support	22
fonthelv	23	latin8support	23
fontsserif	23	latin9support	23
fontsymbol	23	lbmsupport	26
fontvgas	23	libvcongui	21
fxrm9200ial	19	linux	14
gb18030support.....	23	mc68x328ial.....	19
gbksupport.....	23	micemoveable.....	21
gbsupport.....	23	miniguientry.....	21
gifsupport.....	26	mousecalibrate.....	20
greeksupport.....	22	mpc823ial.....	19
hebrewsupport.....	22	msgstr	21
helioial	19	nativegpm	20
hh2410r3ial.....	19	nativeial.....	18
hh2440ial	19	nativeimps2.....	20
hhkbidrial	19	nativems	20
Imegb2312.....	22	nativems3.....	20
Imegb2312py.....	22	nativeps2.....	20
incoreres.....	21	ownmalloc	27
ipaqial	18	ownpthead	27
jpgsupport.....	26	ownstdio.....	27
Kbdde	20	palm2ial.....	19
kbdde latin1	20	pcxsupport	26
Kbdes	20	pngsupport.....	26

px255bial	19	tracemsg	21
qpfsupport.....	23	tffcachecache	23
qvfbial	18	tffsupport	23
randomial	18	type1support	23
rbf16	23	ucb1x00ial	19
rbf24	23	unicodesupport	23
rbfbig12.....	23	unknown	15
rbfbig16.....	23	utpmcial	19
rbfbig24.....	23	vbfsupport	23
rbfgeb12	23	videobf533	17
rbfgeb16	23	videocommlcd	16
rbfgeb16hei.....	23	videodfb	17
rbfgeb24	23	videodummy.....	16
rbfkj12.....	23	videoem85xxosd	16
rbfkj14.....	23	videofbcon.....	16
rbfkj16.....	23	videomb93493	17
rbfsupport	23	videoqvfb	16
savebitmap	21	videoshadow	16
savescreen	21	videosvpxxosd	17
shiftjissupport.....	23	videoutpmc.....	17
skyeleep7312ial	19	videowvfb	16
smdk2410ial	19	vr4181ial	19
stb810.....	15	WIN32	15
svpxxial	19	--with-mttfcachenum	25
t800ial	19	--with-osname	14
textmode	21	--with-style.....	26
tfstbial	19	--with-targetname	15
tgasupport.....	26	--with-ttfcachesize	25
thaisupport.....	22	wvfbial	18
tinyscreen	26		