# MiniGUI 用 户 手 册

版本 *3.0* 修订号 *5*
适用于 *MiniGUI Ver 3.0.x*

北京飞漫软件技术有限公司
2018 年 02 月

# 版权声明

《MiniGUI 用户手册》 版本 3.0 修订号5，适用于MiniGUI 版本 3.0.x。

版权所有 (C) 2003～2018，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介绍.....1
1.1 MiniGUI简介.....1
1.2 其他 MiniGUI 文档.....1
1.3 MiniGUI 源代码包以及示例程序包.....2
1.5 MiniGUI 组件.....2
1.6 miniStudio开发工具.....3
1.6 关于本手册.....3
2 MiniGUI 的配置、编译和安装.....4
2.1 编译时配置选项的生成.....4
2.1.1 在 GNU 开发环境中使用 configure 脚本进行配置.....5
2.1.2 非 GNU 环境下的配置.....8
2.2 MiniGUI的编译时配置选项详解.....8
2.2.1 操作系统相关的选项和宏.....9
2.2.2 目标板相关的选项和宏.....10
2.2.3 运行模式相关的选项和宏.....10
2.2.4 图形引擎相关的选项和宏.....11
2.2.5 输入引擎相关的选项和宏.....12
2.2.6 键盘布局的相关选项和宏.....14
2.2.7 系统全局配置选项和宏.....14
2.2.8 字符集和字体相关的选项和宏.....15
2.2.9 图像文件格式相关的选项和宏.....18
2.2.10 外观风格相关的选项和宏.....19
2.2.11 原生控件相关的选项和宏.....19
2.2.12 其他选项和宏.....20
2.3 最小配置选项.....21
2.3.1 使用 GNU configure 脚本.....21
2.3.2 对应的 mgconfig.h.....22
2.4 在 PC Linux 上编译和安装MiniGUI.....33
2.4.1 编译和安装依赖库.....33
2.4.2 编译和安装虚拟帧缓冲区程序.....35
2.4.3 编译和安装 MiniGUI.....35
2.4.4 安装 MiniGUI 资源包.....36
2.4.5 编译并运行MiniGUI示例程序.....36
2.5 在非 GNU 开发环境中使用 cygwin 工具编译和安装 MiniGUI.....36
2.6 使用 Ubuntu on Windows 配置和编译MiniGUI.....39
3 MiniGUI 的运行时配置选项.....40
3.1 配置文件.....40
3.1.1 system段.....41
3.1.2 fbcon段.....41
3.1.3 qvfb段.....42
3.1.4 pc_xvfb段.....42
3.1.5 rawbitmapfonts、varbitmapfonts、qpf、truetypefonts和type1fonts段.....42
3.1.6 systemfont段.....44
3.1.7 mouse段.....45
3.1.8 event段.....45

3.1.9 cursorinfo段 ..... 45
3.1.10 classic 段 ..... 46
3.1.11 默认配置文件 ..... 49
3.2 内建式配置选项 ..... 58
3.2.1 ETCSETCTION结构 ..... 58
3.2.2 ETC_S结构 ..... 60
3.2.3 mgetc.c文件清单 ..... 60
3.3 配置示例 ..... 76
3.3.1 只支持 ISO8859-1 字符显示的运行时配置 ..... 76
3.3.2 指定不同的图形引擎和输入引擎 ..... 77
4 在 Windows 平台上开发 MiniGUI 应用程序 ..... 78
附录 A 常见问题及解答 ..... 80
A.1 GPL 版本问题 ..... 80
A.2 应用问题 ..... 80
A.3 移植性问题 ..... 80
A.4 编译问题 ..... 81
A.5 输入引擎 ..... 82
A.6 运行时问题 ..... 82
A.7 常见错误信息 ..... 83

# 1 MiniGUI介绍

## 1.1 MiniGUI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[1]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，历经近二十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。

正式发布的MiniGUI 及其组件的源代码包、示例程序包等，可从 MiniGUI 官方网站下载[2]。

另外，MiniGUI 及其组件的完整源代码现已托管到 GitHub，其中包含开发中的（尚未正式发布的）MiniGUI 及其组件的源代码仓库[3]。

正式发布的MiniGUI源代码包按照所支持的操作系统划分，表 1.1 给出了各操作系统版本所支持的 MiniGUI 运行模式。

表 1.1 MiniGUI 各操作系统所支持的运行模式

| 产品名称及版本 | 支持的 MiniGUI 运行模式 |
| --- | --- |
| MiniGUI V3.0.x for Linux | MiniGUI-Processes<br>MiniGUI-Threads<br>MiniGUI-Standalone |
| MiniGUI V3.0.x for uCLinux | MiniGUI-Threads<br>MiniGUI-Standalone |
| MiniGUI V3.0.x for VxWorks | MiniGUI-Threads<br>MiniGUI-Standalone |
| MiniGUI V3.0.x for ThreadX | MiniGUI-Threads<br>MiniGUI-Standalone |
| MiniGUI V3.0.x for uC/OS-II | MiniGUI-Threads<br>MiniGUI-Standalone |

针对其他操作系统（eCOS、OSE、Nucleus、pSOS等）的版本，请访问GitHub仓库获得。

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》V3.0以及《Datasheet for MiniGUI》V3.0.x等文档。

## 1.2 其他 MiniGUI 文档

除本用户手册之外，还有如下文档可通过 MiniGUI 官方网站下载或访问[4]：

[1] 简称“飞漫软件（FMSoft）”，官网：http://www.fmsoft.cn

[2] http://www.minigui.com/zhcn/download/

[3] https://github.com/VincentWei

[4] http://www.minigui.com/zhcn/documentation/

- ✓ 《MiniGUI 技术白皮书》V3.0。MiniGUI 起源、发展历史及技术特点。
- ✓ 《Datasheet for MiniGUI》V3.0.x。MiniGUI 特性表。
- ✓ 《MiniGUI编程指南》 V3.0-5。描述利用 MiniGUI V3.0.x 开发应用程序时的相关概念、主要的 API 接口及示例程序等。
- ✓ 《MiniGUI API Reference Manual》 V3.0.x。对 MiniGUI V3.0.x 应用编程接口（MiniGUI-Processes运行模式）的详细描述[5]。

MiniGUI 开发者同时维护有 Wiki 网站[6]，用来维护以上文档的最新版本，敬请访问。

## 1.3 MiniGUI 源代码包以及示例程序包

在MiniGUI官方网站的下载区，罗列有如下 MiniGUI 源代码包及示例程序包：

- ✓ MiniGUI Core Lib：libminigui-3.0.x-<os>.tar.gz，针对 <os> （如 linux）操作系统的 MiniGUI V3.0.x 函数库源代码。
- ✓ MiniGUI Resources：minigui-res-3.0.x.tar.gz，MiniGUI 所使用的资源，包括基本字体、图标、位图和鼠标光标等。
- ✓ MiniGUI Samples：mg-samples-3.0.x.tar.gz，《MiniGUI 编程指南》的配套示例程序。

同时提供有如下源代码包：

- ✓ MiniGUI 组件。
- ✓ 工具和依赖库，虚拟缓冲区程序GVFB以及freetype、libjpeg、libpng、zlib 等函数库。

## 1.5 MiniGUI 组件

为了适应各种嵌入式设备的不同需求，飞漫软件围绕 MiniGUI 开发了许多组件产品。用户使用这些组件产品，可以扩展 MiniGUI 的功能，并可以和已有的 MiniGUI 应用程序良好集成。图 1.1 给出了这些组件产品和 MiniGUI 以及应用程序之间的关系。

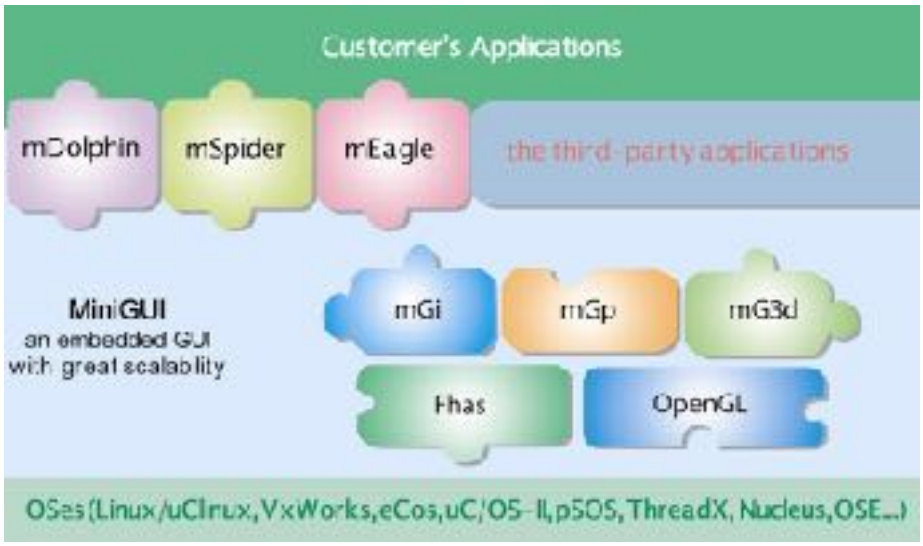


图 1.1 飞漫软件围绕 MiniGUI 的产品线

mGUtils 组件为用户提供了好几个功能模板，有了这些模板，用户就不用为一些常用的功能，再去编

[5]该文档以电子版本形式提供：HTML 格式及 Windows 预编译帮助文档格式，仅提供英文版。

[6] http://wiki.minigui.com/twiki/bin/view

写代码了。

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

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

在 miniStudio 的开发中，为实现可视化图形界面的设计，飞漫软件在 MiniGUI 现有接口基础上，开发了一套新的控件集。miniStudio 引入的新控件集是在原 MiniGUI 控件集基础上发展而来的，为与 MiniGUI 固有控件集（Intrinsic Control Set）区别，称为“新控件集（New Control Set）”，并作为一个新的 MiniGUI 组件mGNCS 发布。mGNCS 主要配合 miniStudio 使用，也可以作为 MiniGUI 3.0 的一个组件而直接使用，且可以和固有控件集中的控件混合使用。**我们强烈建议新的 *MiniGUI* 应用开发使用 *mGNCS*，而不再使用 *MiniGUI* 内置控件。**

mGi 是飞漫软件提供的一个输入法组件，该组件目前提供了软键盘输入法和手写输入法框架，并提供给用户管理输入法的容器，通过这个容器，用户还可以添加自定义的输入法。此外，对于软键盘输入法，用户可以自定义显示的键盘位图，并可添加不同的输入翻译方式（自带中文全拼输入法）。mGi 可以和所有的 MiniGUI 增值版产品配合使用。

mGp 是飞漫软件针对 MiniGUI 应用程序的一个打印组件，该组件使用户的 MiniGUI 程序具有打印输出功能，可以将 MiniGUI 程序中的位图或文字输出到打印机去。mGp 现已提供对爱普生和惠普等多种打印机的支持。mGp 目前只在 Linux 操作系统上运行，也就是说，只能配合 MiniGUI-VAR for Linux 增值版产品使用。

mG3d 是一个为 MiniGUI 的应用程序提供 3D 接口的组件，通过这些接口，用户可以给自己的应用程序添加简单的三维图像，文字渲染、场景渲染等效果。

## 1.6 miniStudio开发工具

miniStudio是一款面向MiniGUI的集成开发环境，为用户提供所见即所得的界面设计，自动生成和维护MiniGUI程序框架，基于Eclispe进行代码编辑、编译、运行、调试，加快MiniGUI应用程序的开发，降低使用MiniGUI的门槛。用户使用MiniGUI时可以更专注于业务相关的具体应用，大大降低MiniGUI相关应用的研发成本，提供更好的产品。

miniStudio 是飞漫软件开发的非开源商业软件产品，提供 Windows 及 Ubuntu Linux两个版本，您可以访问 MiniGUI 官方网站下载该产品获得评估或试用许可证。

## 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。

## 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 要运行的硬件平台。
- ✓ 指定 MiniGUI 要运行的操作系统。
- ✓ 指定生成基于线程的 MiniGUI-Threads 运行模式还是基于进程的 MiniGUI-Processes 运行模式，或者只是最简单的 MiniGUI-Standalone 运行模式。
- ✓ 指定需要支持的 GAL 引擎和 IAL 引擎，以及引擎相关选项。
- ✓ 指定需要支持的字体类型。
- ✓ 指定需要支持的字符集。
- ✓ 指定需要支持的图像文件格式。
- ✓ 指定需要支持的控件类。
- ✓ 指定控件和窗口的整体风格，可以通过指定不同的渲染器完成。

本章将详细介绍 MiniGUI 的编译时配置选项，以便用户能够自行定制出最适合自己嵌入式系统的 MiniGUI。本章还将简单介绍 MiniGUI 的编译和安装。

### 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 advanced 2D graphics APIs */
#define _MGHAVE_ADV_2DAPI 1

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

/* Define if include the 2440 IAL engine */
/* #undef _MGIAL_2440 */

/* Define if include the automatic IAL engine */
/* #undef _MGIAL_AUTO */
```

```
/* Define if support BIG5 charset */
#define _MGCHARSET_BIG5 1

/* Define if include clipboard support */
#define _MGHAVE_CLIPBOARD 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 _MGHAVE_CLIPBOARD 宏，这个宏将打开 MiniGUI 中的剪切板功能；该文件没有定义 _MGIAL_AUTO 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、大部分组建以及示例程序包）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 环境或者 Linux 兼容子系统（需 Windows 10 及以上，并安装 Ubuntu、SUSE 等发行版），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 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 Ubuntu Linux 16.04 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节选（注意该命令在其它Linux发行版

上的输出可能会有所不同）：

```
$ ./configure --help
`configure' configures libminigui 3.0.13 to adapt to many kinds of systems.

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

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-option-checking  ignore unrecognized --enable/--with options
  --disable-FEATURE       do not include FEATURE (same as --enable-
FEATURE=no)
  --enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
  --enable-silent-rules   less verbose build output (undo: "make V=1")
  --disable-silent-rules  verbose build output (undo: "make V=0")
  --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
  --enable-dependency-tracking
                          do not reject slow dependency extractors
  --disable-dependency-tracking
                          speeds up one-time build
  --disable-libtool-lock  avoid locking (might break parallel builds)
  --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-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-dblclk          mouse button can do double click <default=yes>
  --enable-cursor          include cursor support <default=yes>
  --enable-clipboard       include clipboard support <default=yes>
  --enable-ownstdio        use own implementation of stdio functions
<default=no>
  --enable-ownmalloc       use own implementation of malloc functions
<default=no>
  --enable-ownpthread      use own implementation of pthread functions
<default=no>
  --enable-adv2dapi        include advanced 2D graphics APIs <default=yes>
  --enable-minimalgdi      build a minimal GDI library only <default=no>
  --enable-productid       insert a productid into the library file
<default=no>
  --enable-splash          enable splash <default=yes>
  --enable-screensaver     enable screensaver <default=yes>
```

```
  --enable-flatlf          include flat Look and Feel renderer <default=yes>
  --enable-skinlf          include skin Look and Feel renderer <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-ttfsupport=ft1/ft2/none
...

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>
  LIBS        libraries to pass to the linker, e.g. -l<library>
  CPPFLAGS    (Objective) 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.

Report bugs to the package provider.
```

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 --enable-procs --with-ttfsupport=ft2
```

就可以将 MiniGUI 配置成 MiniGUI-Procs运行模式，且使用 FreeType 2 支持 TrueType 字体。如果运行：

```
user$ ./configure --disable-cursor --disable-screensaver
```

则会屏蔽鼠标光标以及默认屏幕保护程序。

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的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，可参阅下面的2.4.2节。

## 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 可通过 FreeType 1 或者 FreeType 2来支持 TrueType 字体，可通过 --with-ttfsupport 配置选项指定使用哪个库。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。如果要让 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 |
|---|---|---|---|
| --with-osname=linux | __LINUX__ | | 默认值，用于 Linux 操作系统 |
| --with-osname=uclinux | __uClinux__ | | 用于 uClinux 操作系统 |
| --with-osname=ecos | __ECOS__ | __NOUNIX__ | 用于 eCos 操作系统 |
| --with-osname=ucos2 | __UCOSII__ | __NOUNIX__<br>_INCORE_RES<br>_USE_OWN_MALLOC<br>_USE_OWN_STDIO<br>_USE_OWN_PTHREAD | 用于 uC/OS-II操作系统 |
| --with-osname=swlinux | __WINBOND_SWLINUX__ | | 用于 SWLinux 操作系统，uClinux 操作系统的变种 |
| --with-osname=vxworks | __VXWORKS__ | __NOUNIX__<br>_USE_OWN_STDIO<br>_USE_OWN_PTHREAD | 用于 VxWorks 操作系统 |
| --with-osname=cygwin | __CYGWIN__ | __NOUNIX__ | 用于 cygwin 环境 |
| --with-osname=win32 | WIN32 | __NOUNIX__ | 用于 Win32 平台 |
| --with-osname=darwin | __DARWIN__ | __NOUNIX__ | 用于 MacOSX 操作系统 |
| --with-osname=threadx | __THREADX__ | __NOUNIX__<br>_INCORE_RES<br>_USE_OWN_MALLOC<br>_USE_OWN_STDIO<br>_USE_OWN_PTHREAD | 用于 ThreadX 操作系统 |
| --with-osname=nucleus | __NUCLEUS__ | __NOUNIX__<br>_INCORE_RES<br>_USE_OWN_MALLOC<br>_USE_OWN_STDIO<br>_USE_OWN_PTHREAD | 用于 Nucleus 操作系统 |
| --with-osname=ose | __OSE__ | __NOUNIX__<br>_INCORE_RES<br>_USE_OWN_PTHREAD | 用于 OSE 操作系统 |
| --with-osname=psos | __PSOS__ | __NOUNIX__<br>_INCORE_RES<br>_USE_OWN_PTHREAD | 用于 pSOS 操作系统 |

因为 MiniGUI 正式发布版本是按照操作系统划分的，因此，针对某个操作系统的 MiniGUI软件包是无

法运行在其他操作系统上的。在手工修改配置时，为了您的 MiniGUI 版本能够在对应的操作系统上编译，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 `unkown`。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 |
| --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=wvfb | __TARGET_WVFB__ | 包含 Windows下 Shadow引擎的 wvfb 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=c33l05 | __TARGET_C33L05__ | 用于运行axLinux的c33l05开发板 |
| --with-targetname=bfin | __TARGET_BLACKFIN__ | 用于运行uCLinux的bfin开发板 |
| --with-targetname=vxppc | __TARGET_PPC__ | 用于使用VxWorks 的 powerpc目标机 |
| --with-targetname=monaco | __TARGET_MONACO__ | 用于运行ThreadX 的 monaco开发板 |
| --with-targetname=unkown | __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[7]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 | |

[7] 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》V2.0-4。

| | | | |
|---|---|---|---|
| procs | _MGRM_PROCESSES | MiniGUI-Processes 运行模式，仅用于 Linux操作系统 | 关闭 |
| standalone | _MGRM_STANDALONE | MiniGUI-Standalone 运行模式，可用于所有操作系统 | 关闭 |

### 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、Linux FrameBuffer控制台图形引擎、XVFB虚拟缓冲区图形引擎、COMMLCD 图形引擎、Shadow 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 给出了通用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 | _MGGAL_DUMMY | dummy | 支持所有操作系统；不做任何实际绘制的图形引擎，可用于测试。 | 开启 |
| videofbcon | _MGGAL_FBCON | fbcon | Linux/uClinux；使用 Linux FrameBuffer 控制台。 | 开启 |
| videopcxvfb | _MGGAL_PCXVFB | pc_xvfb | Linux/Win32；适合 PC 的虚拟缓冲区图形引擎，不依赖于具体的实现平台。 | 开启 |
| videortosxvfb | _MGGAL_RTOSXVFB | rtos_xvfb | 所有操作系统；适合于 RTOS 的虚拟缓冲区引擎。 | 关闭 |
| videoqvfb | _MGGAL_QVFB | qvfb | Linux；虚拟缓冲区图形引擎，使用 Qt 开发。 | 关闭 |
| videowvfb | _MGGAL_WVFB | wvfb | Win32；虚拟缓冲区图形引擎，使用 Win32 接口开发。 | 关闭 |
| videocommlcd | _MGGAL_COMMLCD | commlcd | 所有操作系统；适合多种RTOS系统上的 LCD 驱动。 | 关闭 |
| videoshadow | _MGGAL_SHADOW | shadow | 所有操作系统；影子图形引擎，通过异步更新机制，可实现屏幕旋转、像素格式转换等。 | 关闭 |
| videodfb | _MGGAL_DFB | dfb | Linux；基于DirectFB 之上。 | 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。

Qvfb 图形引擎用于 Linux 操作系统。利用 qvfb，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 qvfb 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。

需要注意的是，在 MiniGUI 3.0 中，原有的 QVFB（Qt Virtual Frame Buffer）、WVFB（Windows Virtual Frame Buffer）已被全新设计的 XVFB 通用虚拟缓冲区图形引擎取代。

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。在使用Shadow 图形引擎时，通过运行时配置选项，可指定实际使用的图形引擎，比如 fbcon，并指定屏幕旋转方向。注意，Shadow 图形引擎默认是关闭的。

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 采用子驱动程序结构，通过目标名称（由`--with-targetname`选项指定）来确定所包含的子驱动程序。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` 源代码树的 `rtos/` 目录中，包含了针对各个操作系统的`CommLCD`图形引擎实现，您可以修改这个文件来支持自己的 `LCD` 控制器。

除了上述通用图形引擎之外，MiniGUI 中还包含有其他一些针对特定硬件平台的图形引擎，具体请运行 `./configure --help` 命令查询。

### 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 | _MGIAL_DUMMY | dummy | Dummy 输入引擎；所有操作系统。 | 开启 |
| autoial | _MGIAL_AUTO | auto | 自动输入引擎；所有操作系统。 | 关闭 |
| randomial | _MGIAL_RANDOM | random | Random 输入引擎；所有操作系统。 | 关闭 |
| consoleial | _MGIAL_CONSOLE | console | Linux 控制台输入引擎；Linux。 | 开启 |
| qvfbial | _MGIAL_QVFB | qvfb | QVFB 输入引擎；Linux，配合 QVFB 图形引擎。 | 开启 |
| wvfbial | _MGIAL_WVFB | wvfb | WVFB 输入引擎；Win32，配合 WVFB 图形引擎。 | 关闭 |

| | | | | |
|---|---|---|---|---|
| commial | _MGIAL_COMM | comm | COMM 输入引擎；所有操作系统。 | 关闭 |
| dfbial | _MGIAL_DFB | dfb | 基于 DirectFB的输入引擎；Linux，配合 DFB 图形引擎。 | 关闭 |
| tslibial | _MGIAL_TSLIB | tslib | 基于 tslib 的输入引擎；Linux。 | 关闭 |
| qemuial | _MGIAL_QEMU | qemu | QEMU 输入引擎；Linux。 | 关闭 |
| customial | _MGIAL_CUSTOM | custom | 由 MiniGUI 应用程序负责定义具体实现的图形引擎；任何操作系统。 | 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 及其应用软件的自动测试。

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 目前尚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 |
|---|---|---|---|
| consoleps2 | _MGCONSOLE_PS2 | 支持 PS2鼠标协议 | 开启 |
| consoleimps2 | _MGCONSOLE_IMPS2 | 支持智能鼠标（IMPS/2）协议 | 开启 |
| consolems | _MGCONSOLE_MS | 支持旧的MS串口鼠标 | 开启 |
| consolems3 | _MGCONSOLE_MS3 | 支持 MS3鼠标协议 | 开启 |
| consolegpm | _MGCONSOLE_GPM | 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 |
|---|---|---|---|
| mousecalibrate | _MGHAVE_MOUSECALIBRATE | 支持触模屏校正 | 开启 |

除上述输入引擎之外，MiniGUI 还包含有针对特定硬件平台的输入引擎，具体可运行 ./configure

`--help` 命令查阅。
值得注意的是，如果在配置 MiniGUI 时打开了虚拟图形引擎，如 qvfb、gvfb、pc_xvfb 和 rtos_xvfb 等，则配置脚本会自动打开对应同名的输入引擎。

### 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》V3.0-5。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 |
|---|---|---|---|---|
| kbdfrpc | _MGKBDLAYOUT_FRPC | frpc | 包含对French PC (non-US102 keys)键盘布局的支持 | 关闭 |
| kbdfr | _MGKBDLAYOUT_FR | fr | 包含对支持法语键盘布局的支持 | 关闭 |
| kbdde | _MGKBDLAYOUT_DE | de | 包含对支持德语键盘布局的支持 | 关闭 |
| kbddelatin1 | _MGKBDLAYOUT_DELATIN1 | delatin1 | 包含对支持德语Latin1键盘布局的支持 | 关闭 |
| Kbdit | _MGKBDLAYOUT_IT | it | 包含对支持意大利语键盘布局的支持 | 关闭 |
| Kbdes | _MGKBDLAYOUT_ES | es | 包含对支持西班牙语键盘布局的支持 | 关闭 |
| kbdescp850 | _MGKBDLAYOUT_ESCP850 | escp850 | 包含对支持西班牙语CP850键盘布局的支持 | 关闭 |
| kbdhebrewpc | _MGKBDLAYOUT_HEBREWPC | hebrewpc | 包含对支持希伯来语PC键盘布局的支持 | 关闭 |
| kbdarabicpc | _MGKBDLAYOUT_ARABICPC | arabicpc | 包含对支持阿拉伯语PC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 | _MGINCORE_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 | _MGHAVE_TRACE_MSG | 追踪MiniGUI的消息 | 关闭 |
| msgstr | _MGHAVE_MSG_STRING | 追踪消息时，输出信息中包含消息的字符串名称 | 关闭 |
| dblclk | _MGMISC_DOUBLE_CLICK | 可以支持鼠标双击 | 开启 |
| cursor | _MGHAVE_CURSOR | 包含对鼠标光标的支持 | 开启 |
| clipboard | _MGHAVE_CLIPBOARD | 包含对剪贴板的支持 | 开启 |
| savebitmap | _MGMISC_SAVEBITMAP | 支持SaveBitmap的相关函数 | 开启 |
| aboutdlg | _MGMISC_ABOUTDLG | 包含About对话框 | 开启 |
| savescreen | _MGMISC_SAVESCREEN | 支持屏幕截图 | 开启 |
| fixedmath | _MGHAVE_FIXED_MATH | 使用定点数学函数 | 开启 |
| adv2dapi | _MGHAVE_ADV_2DAPI | 支持高级2D图形API | 开启 |
| splash | _MG_ENABLE_SPLASH | 启动画面 | 开启 |
| screensaver | _MG_ENABLE_SCREENSAVER | 屏幕保护程序 | 开启 |

`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。

`splash`和`screensaver`选项分别用来定义 MiniGUI 的启动画面以及内建的屏幕保护程序程序。在实际项目中，通常可以关闭这两个选项。

### 2.2.8 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 定义的两种点阵字体格式）、 UPF字体、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 |
| --- | --- | --- | --- |
| latin2support | _MGCHARSET_LATIN2 | 包含对East European (Latin2, ISO-8859-2) 字符集的支持 | 关闭 |
| latin3support | _MGCHARSET _LATIN3 | 包含对South European (Latin3, ISO-8859-3) 字符集的支持 | 关闭 |
| latin4support | _MGCHARSET _LATIN4 | 包含对North European (Latin4, ISO-8859-4)字符集的支持 | 关闭 |
| cyrillicsupport | _MGCHARSET _CYRILLIC | 包含对Cyrillic (ISO-8859-5) 字符集的支持 | 关闭 |
| arabicsupport | _MGCHARSET _ARABIC | 包含对Arabic (ISO-8859-6) 字符集的支持 | 关闭 |
| greeksupport | _MGCHARSET _GREEK | 包含对Greek (ISO-8859-7) 字符集的支持 | 关闭 |
| hebrewsupport | _MGCHARSET _HEBREW | 包含对Hebrew (ISO-8859-8) 字符集的支持 | 关闭 |
| latin5support | _MGCHARSET _LATIN5 | 包含对Turkish (Latin5, ISO-8859-9) 字符集的支持 | 关闭 |
| latin6support | _MGCHARSET _LATIN6 | 包含对Nordic (Latin6, ISO-8859-10) 字符集的支持 | 关闭 |
| thaisupport | _MGCHARSET _THAI | 包含对Thai (ISO-8859-11) 字符集的支持 | 关闭 |
| latin7support | _MGCHARSET _LATIN7 | 包含对Latin7 (ISO-8859-13) 字符集的支持 | 关闭 |
| latin8support | _MGCHARSET _LATIN8 | 包含对Latin8 (ISO-8859-14) 字符集的支持 | 关闭 |
| latin9support | _MGCHARSET _LATIN9 | 包含对Latin 9 (ISO-8859-15, West Extended) 字符集的支持 | 开启 |
| latin10support | _MGCHARSET _LATIN10 | 包含对Latin 10 (ISO-8859-16, Romanian) 字符集的支持 | 关闭 |
| gbsupport | _MGCHARSET _GB | 包含对EUC编码的GB2312字符集的支持 | 开启 |
| gbksupport | _MGCHARSET _GBK | 包含对GBK字符集的支持 | 开启 |
| gb18030support | _MGCHARSET _GB18030 | 包含对GB18030-0字符集的支持 | 关闭 |
| big5support | _MGCHARSET _BIG5 | 包含对BIG5字符集的支持 | 关闭 |
| euckrsupport | _MGCHARSET _EUCKR | 包含对EUC编码的KSC5636和KSC5601字符集的支持 | 关闭 |
| eucjpsupport | _MGCHARSET _EUCJP | 包含对EUC编码的JISX0201和JISX0208字符集的支持 | 关闭 |
| shiftjissupport | _MGCHARSET _SHIFTJIS | 包含对Shift-JIS编码的JISX0201和JISX0208字符集的支持 | 关闭 |
| unicodesupport | _MGCHARSET _UNICODE | 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 | _MGFONT _RBF | 包含对RBF字体格式的支持 | 开启 |
| rbfvgaoem | _MGINCORERBF_LATIN1_VGAOEM | 包含RBF ISO8859-1 8x16 fixed内建字体 | 开启 |

| | | | |
|---|---|---|---|
| rbfterminal | _MGINCORERBF_LATIN1_TERMINAL | 包含RBF ISO8859-1 12x24 fixed内建字体 | 开启 |
| rbffixedsys | _MGINCORERBF_LATIN1_FIXEDSYS | 包含RBF GB2312 12x12 fixed/song内建字体 | 开启 |
| vbfsupport | _MGFONT_VBF | 包含对VBF字体的支持 | 开启 |
| fontsserif | _MGINCOREFONT_SANSSERIF | 包含sansserif内建VBF字体 | 开启 |
| fontcourier | _MGINCOREFONT_COURIER | 包含courier内建VBF字体 | 开启 |
| fontsystem | _MGINCOREFONT_SYSTEM | 包含system 内建VBF字体 | 关闭 |
| upfsupport | _MGFONT_UPF | 支持飞漫Unicode Prerendered Font（UPF）字体 | 开启 |
| fonttimes | _MGINCOREFONT_TIMES | 包含 Times 内建 UPF 字体 | 开启 |
| qpfsupport | _MGFONT_QPF | 包含对Qt Prerendered Font(QPF)字体的支持 | 关闭 |
| ttfsupport=ft1 | _MGFONT_TTF | 使用 FreeType 1支持TrueType字体 | 关闭 |
| ttfsupport=ft2 | _MGFONT_FT2 | 使用 FreeType 2支持TrueType字体 | 关闭 |
| ttfcachesize=256 | _MGTTF_CACHE_SIZE | TrueType缓存大小 | 256 |
| mttfcachenum=10 | _MGMAX_TTF_CACHE | TrueType 缓存数量 | 10 |

latin2support、latin3support、cyrillicsupport、arabicsupport、greeksupport、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。

gbsupport、gbksupport、gb18030support、big5support、euckrsupport、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。

rbfvgaoem、rbfterminal、rbffixedsys等配置选项，可控制是否在MiniGUI 库中内建对应的RBF点阵字体。这些编译配置选项默认是打开的，这样，MiniGUI 在不装载任何字体时，仍然可以正常运行。

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VFB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、fontsystem 编译配置选项可控制是否在 MiniGUI 库中内建SanSerif、Courier及System VBF字体。这些内建字体选项默认是打开的，且不受 incoreres 选项的影响。

upfsupport配置选项控制是否在 MiniGUI 库中包含对 FMSoft Unicode Prerendered Font（UPF）字体的支持。因为 UPF 字体使用 UNICODE 编码，因此，允许对 UPF 字体的支持，将自动开启 MiniGUI 的 UNICODE 字符集支持。

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 字符集支持。

`ft2support`配置选项用来控制是否在 MiniGUI 库中包含对 FreeType2库的支持。MiniGUI 可以使用 FreeType2 版本 2.3.4 来提供对矢量字体的渲染。如果您的系统中没有安装 FreeType2，配置脚本将自动关闭该选项。

`ttfsupport`配置选项（with选项，可选`ft1/ft2/none`三者之一）用来控制是否在 MiniGUI 库中包含对TrueType 字体的支持，以及使用FreeType1还是FreeType2库。需要注意的是，FreeType 2 和 FreeType 1 不兼容。当指定使用 ft1或者ft2作为渲染引擎时，可通过 `--with-ft1-includes` 或者 `--with-ft2-includes` 来指定FreeType1或者FreeType2函数库对应的头文件。

`ttfcachesize`和`mttfcachenum` 配置选项确定在使用FreeType1库时的缓存大小及数量。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 | _MGMAX_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 | _MGTTF_CACHE_SIZE | 64 | 默认值 |
| --with-ttfcachesize=128 | | 128 | |
| --with-ttfcachesize=256 | | 256 | |
| --with-ttfcachesize=512 | | 512 | |
| --with-ttfcachesize=1024 | | 1024 | |

### 2.2.9 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，可从MiniGUI官网下载。

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 | _MGIMAGE_GIF | 包含对GIF文件的支持 | 开启 |
| jpgsupport | _MGIMAGE_JPG | 包含对JPG文件的支持 | 开启 |

| | | | |
|---|---|---|---|
| pngsupport | _MGIMAGE_PNG | 包含对PNG文件的支持 | 开启 |
| pcxsupport | _MGIMAGE_PCX | 包含对PCX文件的支持 | 关闭 |
| lbmsupport | _MGIMAGE_LBM | 包含对LBM/PBM文件的支持 | 关闭 |
| tgasupport | _MGIMAGE_TGA | 包含对TGA文件的支持 | 关闭 |

### 2.2.10 外观风格相关的选项和宏

在 MiniGUI 3.0中，我们引入了 Look and Feel（简称 LF）概念。将原有 flat、classic、fashion窗口风格抽象为新的 LF 渲染器（LFRDR），保留了flat、classic渲染器，同时引入了 skin 渲染器，而原有Fashion风格则通过 mGPlus 实现。其中classic 渲染器是内建的，而flat和skin渲染器可通过配置选项来控制。表 2.17 给出了外观渲染器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 |
|---|---|---|---|
| --enable-flatlf | _MGLF_RDR_FLAT | 简洁的平面风格 | 打开 |
| --enable-skinlf | _MGLF_RDR_SKIN | 皮肤，窗口组件由位图定义 | 打开 |

### 2.2.11 原生控件相关的选项和宏

MiniGUI 提供了针对所有原生控件的配置选项，MiniGUI原生控件指包含在 MiniGUI 核心库中的控件。从MiniGUI 3.0 开始，我们通过 mGNCS 组件提供了一套新的控件集，新控件集的设计精良，接口优雅，可完全替代 MiniGUI的原生控件集。因此，**我们强烈建议新的 *MiniGUI* 应用开发使用 *mGNCS*，而不再使用 *MiniGUI* 内置控件。**因为有了更好的mGNCS，MiniGUI 大部分原生控件的配置选项默认是关闭的。若您的应用程序使用了这些控件，请自行开启相关的配置项。
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 | _MGCTRL_STATIC | 包含STATIC控件 | 开启 |
| ctrlbutton | _MGCTRL_BUTTON | 包含BUTTON控件 | 开启 |
| ctrlsledit | _MGCTRL_SLEDIT | 包含Single-Line EDIT控件 | 开启 |
| ctrlbidiedit | _MGCTRL_BIDIEDIT | 包含双向显示 EDIT控件 | 关闭 |
| newtextedit | _MGCTRL_TEXTEDIT<br>_MGCTRL_TEXTEDIT_USE_NEW_IMPL | 新的 TEXTEDIT 控件实现 | 开启 |
| ctrllistbox | _MGCTRL_LISTBOX | 包含LISTBOX控件 | 开启 |
| ctrlpgbar | _MGCTRL_PROGRESSBAR | 包含PROGRESSBAR控件 | 开启 |
| ctrlcombobox | _MGCTRL_COMBOBOX | 包含COMBOBOX控件 | 开启 |
| ctrlpropsheet | _MGCTRL_PROPSHEET | 包含PROPSHEET控件 | 开启 |
| ctrltrackbar | _MGCTRL_TRACKBAR | 包含TRACKBAR控件 | 关闭 |

| | | | |
|---|---|---|---|
| ctrlscrollbar | _MGCTRL_SCROLLBAR | 包含SCROLLBAR控件 | 关闭 |
| ctrlnewtoolbar | _MGCTRL_NEWTOOLBAR | 包含NEWTOOLBAR控件 | 关闭 |
| ctrlmenubtn | _MGCTRL_MENUBUTTON | 包含MENUBUTTON控件 | 关闭 |
| ctrlscrollview | _MGCTRL_SCROLLVIEW | 包含SCROLLVIEW和SCROLLWINDOW控件 | 关闭 |
| ctrltextedit | _MGCTRL_TEXTEDIT | 包含基于SCROLLVIEW的TEXTEDIT控件 | 关闭 |
| ctrlmonthcal | _MGCTRL_MONTHCAL | 包含MONTHCALENDAR控件 | 关闭 |
| ctrltreeview | _MGCTRL_TREEVIEW | 包含TREEVIEW控件 | 关闭 |
| ctrlspinbox | _MGCTRL_SPINBOX | 包含SPINBOX控件 | 关闭 |
| ctrlcoolbar | _MGCTRL_COOLBAR | 包含COOLBAR控件 | 关闭 |
| ctrllistview | _MGCTRL_LISTVIEW | 包含LISTVIEW控件 | 关闭 |
| ctrliconview | _MGCTRL_ICONVIEW | 包含ICONVIEW控件 | 关闭 |
| ctrlgridview | _MGCTRL_GRIDVIEW | 包含GRIDVIEW控件 | 关闭 |
| ctrlanimation | _MGCTRL_ANIMATION | 包含ANIMATION控件并提供对GIF89a文件的支持 | 打开 |

### 2.2.12 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（`printf`、`sprintf` 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有用。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 | _MGUSE_OWN_MALLOC | 使用MiniGUI实现的malloc函数族 | 关闭 |
| ownstdio | _MGUSE_OWN_STDIO | 使用MiniGUI实现的stdio格式化输入输出函数族 | 关闭 |
| ownpthread | _MGUSE_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 中的宏：`__MINIGUI_LIB__`。

表2.20 其他编译宏

| 宏 | 注释 | 备注 |
|---|---|---|

| | | |
|---|---|---|
| __MINIGUI_LIB__ | 编译MiniGUI库的宏 | 使用非 GNU 的 makefile 时，必须定义这个宏 |

从 MiniGUI 3.0 开始，可通过 configure 选项指定 MiniGUI 函数库的名称后缀，默认情况下，MiniGUI 函数库的名称根据运行模式而变化，比如 libminigui-ths.so、libminigui-procs.so、libminigui-sa.so 等，分别对应MiniGUI-Threads、MiniGUI-Processes 和 MiniGUI-Standalone运行模式。

您可以通过 `--with-libsuffix` 选项指定一个特殊的函数库名称后缀。

## 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-dblclk \
    --disable-cursor \
    --disable-mousecalibrate \
    --disable-clipboard \
    --disable-adv2dapi \
    --disable-splash \
    --disable-screensaver \
    --disable-flatlf \
    --disable-skinlf \
    --disable-rbfvgaoem \
    --disable-rbfterminal \
    --disable-vbfsupport \
    --disable-qpfsupport \
    --disable-upfsupport \
    --disable-bmpfsupport \
    --disable-latin9support   \
    --disable-gbsupport       \
    --disable-gbksupport      \
    --disable-unicodesupport  \
    --disable-savebitmap \
    --disable-jpgsupport \
    --disable-pngsupport \
    --disable-gifsupport \
    --disable-aboutdlg \
    --disable-savescreen \
    --disable-mousecalibrate \
    --disable-ctrlanimation \
    --disable-ctrlnewtextedit \
    --disable-consoleps2 \
    --disable-consoleimps2 \
    --disable-consolems \
    --disable-consolems3 \
    --disable-consolegpm
```

利用这个脚本可将 MiniGUI 配置成仅支持 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VGAOEM内建字体。
- ✓ 不包含Terminal内建字体。
- ✓ 不包含对Unicode Prerendered Font (UPF)字体的支持。
- ✓ 不包含对Qt Prerendered Font (QPF)字体的支持。
- ✓ 不包含对TrueType字体的支持。
- ✓ 不包含位图字体支持。
- ✓ 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。
- ✓ 不支持屏幕截图。
- ✓ 不支持高级2D图形API。
- ✓ 不包含 TEXTEDIT控件。
- ✓ 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。
- ✓ 不包含Flat和Skin外观渲染器。

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 ANIMATION控件的话，就可以在上述脚本中加入 `--disable-ctrlanimation` 选项，这样编译出来的MiniGUI函数库就不含有动画控件，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：

```
...

/* MiniGUI configure file name */
#define ETCFILENAME "MiniGUI.cfg"

...

/* 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 3

/* Micro version of MiniGUI */
#define MINIGUI_MICRO_VERSION 13

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

...

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

/* Define if support BIG5 charset */
/* #undef _MGCHARSET_BIG5 */

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

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

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

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

/* Define if support GB18030 charset */
/* #undef _MGCHARSET_GB18030 */

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

/* Define if support Greek charset */
/* #undef _MGCHARSET_GREEK */

/* Define if support Hebrew charset */
/* #undef _MGCHARSET_HEBREW */

/* Define if support Latin 10 charset */
/* #undef _MGCHARSET_LATIN10 */

/* Define if support Latin 2 charset */
/* #undef _MGCHARSET_LATIN2 */

/* Define if support Latin 3 charset */
/* #undef _MGCHARSET_LATIN3 */

/* Define if support Latin 4 charset */
/* #undef _MGCHARSET_LATIN4 */
```

```
/* Define if support Latin 5 charset */
/* #undef _MGCHARSET_LATIN5 */

/* Define if support Latin 6 charset */
/* #undef _MGCHARSET_LATIN6 */

/* Define if support Latin 7 charset */
/* #undef _MGCHARSET_LATIN7 */

/* Define if support Latin 8 charset */
/* #undef _MGCHARSET_LATIN8 */

/* Define if support Latin 9 charset */
/* #undef _MGCHARSET_LATIN9 */

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

/* Define if support Thai charset */
/* #undef _MGCHARSET_THAI */

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

/* Define if include GPM mouse subdriver */
/* #undef _MGCONSOLE_GPM */

/* Define if include IMPS2 mouse subdriver */
/* #undef _MGCONSOLE_IMPS2 */

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

/* Define if include MS3 mouse subdriver */
/* #undef _MGCONSOLE_MS3 */

/* Define if include PS2 mouse subdriver */
/* #undef _MGCONSOLE_PS2 */

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

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

/* Define if include BIDISLEDIT control */
/* #undef _MGCTRL_BIDISLEDIT */

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

/* Define if include COMBOBOX control */
```

```
#define _MGCTRL_COMBOBOX 1

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

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

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

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

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

/* Define if include MENUBUTTON control */
/* #undef _MGCTRL_MENUBUTTON */

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

/* Define if include NEWTOOLBAR control */
/* #undef _MGCTRL_NEWTOOLBAR */

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

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

/* Define if include SCROLLBAR control */
/* #undef _MGCTRL_SCROLLBAR */

/* Define if include SCROLLVIEW control */
/* #undef _MGCTRL_SCROLLVIEW */

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

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

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

/* Define if include TEXTEDIT control */
/* #undef _MGCTRL_TEXTEDIT */

/* Define if use new implementation of TEXTEDIT control */
/* #undef _MGCTRL_TEXTEDIT_USE_NEW_IMPL */
```

```
/* Define if include TRACKBAR control */
/* #undef _MGCTRL_TRACKBAR */

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

/* Define if include TREEVIEWRDR control */
/* #undef _MGCTRL_TREEVIEW_RDR */

/* Define if support Bitmap fonts */
/* #undef _MGFONT_BMPF */

/* Define if support TrueType font based on FreeType2 */
/* #undef _MGFONT_FT2 */

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

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

/* Define if support SEF scripteary font */
/* #undef _MGFONT_SEF */

/* Define if support TrueType font */
/* #undef _MGFONT_TTF */

/* Define if include ttf cache */
/* #undef _MGFONT_TTF_CACHE */

/* Define if support UPF font */
/* #undef _MGFONT_UPF */

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

/* Define if include NEWGAL engine for BF533 OSD via SPI */
/* #undef _MGGAL_BF533 */

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

/* Define if include custom NEWGAL engine */
/* #undef _MGGAL_CUSTOMGAL */

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

/* Define if include ST7167 subdriver for NEWGAL engine of DirectFB */
/* #undef _MGGAL_DFB_ST7167 */

/* Define if include dummy NEWGAL engine */
#define _MGGAL_DUMMY 1
```

```
/* Define if include NEWGAL engine for EM85xx OSD */
/* #undef _MGGAL_EM85XXOSD */

/* Define if include NEWGAL engine for EM85xx YUV */
/* #undef _MGGAL_EM85XXYUV */

/* Define if include NEWGAL engine for EM86xx GFX */
/* #undef _MGGAL_EM86GFX */

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

/* Define if include GDL Video NEWGAL engine */
/* #undef _MGGAL_GDL */

/* Define if include Hi35XX Video NEWGAL engine */
/* #undef _MGGAL_HI3510 */

/* Define if include Hi35XX Video NEWGAL engine */
/* #undef _MGGAL_HI3560 */

/* Define if include Hi3560A Video NEWGAL engine */
/* #undef _MGGAL_HI3560A */

/* Define if include NEWGAL engine for mb93493 YUV FrameBuffer driver */
/* #undef _MGGAL_MB93493 */

/* Define if include MLShadow NEWGAL engine */
/* #undef _MGGAL_MLSHADOW */

/* Define if include mstar NEWGAL engine */
/* #undef _MGGAL_MSTAR */

/* Define if include nexus NEWGAL engine */
/* #undef _MGGAL_NEXUS */

/* Define if include PC Virtual FrameBuffer NEWGAL engine */
#define _MGGAL_PCXVFB 1

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

/* Define if include RTOS Virtual FrameBuffer NEWGAL engine */
/* #undef _MGGAL_RTOSXVFB */

/* Define if include s3c6410 NEWGAL engine */
/* #undef _MGGAL_S3C6410 */

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

/* Define if include sigma8654 NEWGAL engine */
```

```
/* #undef _MGGAL_SIGMA8654 */

/* Define if include NEWGAL engine for STGFB */
/* #undef _MGGAL_STGFB */

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

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

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

/* Define if include advanced 2D graphics APIs */
/* #undef _MGHAVE_ADV_2DAPI */

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

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

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

/* Define if support menu */
#define _MGHAVE_MENU 1

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

/* Define if include message string names */
/* #undef _MGHAVE_MSG_STRING */

/* Define if PCIAccess lib is available */
/* #undef _MGHAVE_PCIACCESS */

/* Define if trace message dispatching of MiniGUI */
/* #undef _MGHAVE_TRACE_MSG */

/* Define if include the 2440 IAL engine */
/* #undef _MGIAL_2440 */

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

/* Define if include IAL engine for Cisco touchpad */
/* #undef _MGIAL_CISCO_TOUCHPAD */

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

```
/* Define if include console (Linux console) IAL engine */
#define _MGIAL_CONSOLE 1

/* Define if include IAL engine for customer's board */
/* #undef _MGIAL_CUSTOM */

/* Define if include the DAVINCI6446 IAL engine */
/* #undef _MGIAL_DAVINCI6446 */

/* Define if include the DFB IAL engine */
/* #undef _MGIAL_DFB */

/* Define if include dlcustom IAL engine */
/* #undef _MGIAL_DLCUSTOM */

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

/* Define if include IAL engine for iPAQ H3600 */
/* #undef _MGIAL_IPAQ_H3600 */

/* Define if include IAL engine for iPAQ H5400 */
/* #undef _MGIAL_IPAQ_H5400 */

/* Define if include the JZ4740 IAL engine */
/* #undef _MGIAL_JZ4740 */

/* Define if include the lide IAL engine */
/* #undef _MGIAL_LIDE */

/* Define if include IAL engine for MStar */
/* #undef _MGIAL_MSTAR */

/* Define if include IAL engine for net's board */
/* #undef _MGIAL_NET */

/* Define if include IAL engine for Nexus */
/* #undef _MGIAL_NEXUS */

/* Define if include the QEMU IAL engine */
/* #undef _MGIAL_QEMU */

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

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

/* Define if include IAL engine for TSLIB */
/* #undef _MGIAL_TSLIB */

/* Define if include the WVFB IAL engine */
/* #undef _MGIAL_WVFB */
```

```
/* Define if support GIF bmp file format */
/* #undef _MGIMAGE_GIF */

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

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

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

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

/* Define if support TGA bmp file format */
/* #undef _MGIMAGE_TGA */

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

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

/* Define if include in-core font: System */
/* #undef _MGINCOREFONT_SYSTEM */

/* Define if include in-core UPF Times fonts */
/* #undef _MGINCOREFONT_TIMES */

/* Define if include in-core FixedSys RBF for ISO8859-1 */
#define _MGINCORERBF_LATIN1_FIXEDSYS 1

/* Define if include in-core Terminal RBF for ISO8859-1 */
/* #undef _MGINCORERBF_LATIN1_TERMINAL */

/* Define if include in-core VGAOEM RBF for ISO8859-1 */
/* #undef _MGINCORERBF_LATIN1_VGAOEM */

/* Define if build MiniGUI for no file I/O system (use in-core resources) */
/* #undef _MGINCORE_RES */

/* Define if use the Arabic PC keyboard layout */
/* #undef _MGKBDLAYOUT_ARABICPC */

/* Define if use the German keyboard layout */
/* #undef _MGKBDLAYOUT_DE */

/* Define if use the German-Latin1 keyboard layout */
/* #undef _MGKBDLAYOUT_DELATIN1 */

/* Define if use the Spanish keyboard layout */
```

```
/* #undef _MGKBDLAYOUT_ES */

/* Define if use the Spanish CP850 keyboard layout */
/* #undef _MGKBDLAYOUT_ESCP850 */

/* Define if use the French keyboard layout */
/* #undef _MGKBDLAYOUT_FR */

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

/* Define if use the Hebrew PC keyboard layout */
/* #undef _MGKBDLAYOUT_HEBREWPC */

/* Define if use the Italian keyboard layout */
/* #undef _MGKBDLAYOUT_IT */

/* Define if include flat Look and Feel */
/* #undef _MGLF_RDR_FLAT */

/* Define if include skin Look and Feel */
/* #undef _MGLF_RDR_SKIN */

/* MiniGUI library suffix */
#define _MGLIB_SUFFIX "ths"

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

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

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

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

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

/* 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 the unit of timer is 10ms */
#define _MGTIMER_UNIT_10MS 1
```

```
/* Define if compile max ttf cahce size for 256k */
/* #undef _MGTTF_CACHE_SIZE */

/* Define if use own implementation of malloc functions */
/* #undef _MGUSE_OWN_MALLOC */

/* Define if use own implementation of pthread functions */
/* #undef _MGUSE_OWN_PTHREAD */

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

/* Define if build the mgeff support version */
/* #undef _MG_MINIMALGDI */

/* Define if insert a productid into the library file */
/* #undef _MG_PRODUCTID */

/* Define if build MiniGUI-Standalone (back-compatibility definition) */
/* #undef _STAND_ALONE */

/* Define if use minigui_entry function in MiniGUI */
/* #undef _USE_MINIGUIENTRY */

/* 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 if compile for OSE */
/* #undef __OSE__ */

/* Define if compile for pSOS */
/* #undef __PSOS__ */

/* Define for Blackfin run uClinux */
/* #undef __TARGET_BLACKFIN__ */

/* Define for EPSON C33L05 (axLinux) */
/* #undef __TARGET_C33L05__ */
```

```
/* Define for FMSoft internal use */
/* #undef __TARGET_FMSOFT__ */

/* Define for Monaco ANVIL target */
/* #undef __TARGET_MONACO__ */

/* Define for FMSoft miniStudio */
/* #undef __TARGET_MSTUDIO__ */

/* Define for OSE on mx21 */
/* #undef __TARGET_MX21__ */

/* Define for VxWorks on PowerPC */
/* #undef __TARGET_PPC__ */

/* Define for Philips STB810 target */
/* #undef __TARGET_STB810__ */

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

/* Define for VirualFone ANVIL target */
/* #undef __TARGET_VFANVIL__ */

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

/* 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__ */

...
```

注意上述代码中省略了若干和MiniGUI接口无关的宏定义。

## 2.4 在 PC Linux 上编译和安装MiniGUI

### 2.4.1 编译和安装依赖库

在运行 MiniGUI 之前，需要安装 MiniGUI 所需的依赖库。MiniGUI 主要使用 LibFreeType、LibPNG、LibJPEG、LibZ 等第三方依赖库。

这些依赖库的源代码包基本上都使用 GNU Automake/Autoconf 脚本组织工程，通过在运行 ./configure 命令时指定特定的环境变量及某些选项来完成这些库的编译和安装。我们还可以通过在这些依赖库源码目录下运行 ./configure --help 命令，来查看各自 configure 脚本可以接受的开关参数。

目前，这些依赖库基本上是主流 Linux 发行版（如 Ubuntu、RedHat等）的标准配置，但如果要在编译 MiniGUI 时可以找到这些函数库，则需要安装这些函数库的开发包。比如在 Ubuntu Linux 上，通过执行如下的命令可安装 FreeType 2、LibPNG、LibJPEG 开发包：

```
$ sudo apt-get install libfreetype6-dev libpng12-dev libjpeg-dev
```

本节下面给出了在源代码包基础上编译、安装这些依赖库的步骤，仅供参考。

### *LibFreeType*

FreeType库是一个开源的、高质量的、且可移植的字体引擎，它提供统一的接口来访问多种字体格式文件，包括TrueType、 OpenType、Type1、CID、CFF、Windows FON/FNT、X11 PCF等。MiniGUI 使用 FreeType 库渲染 TrueType 字体。历史上，FreeType 有两个主要的版本，一个是 FreeType 1，如 FreeType v1.3.1；另外一个是 FreeType 2，如 FreeType v2.5.2。如前所述，MiniGUI 可选择使用 FreeType 1或者FreeType 2 来支持 TrueType 字体。目前，FreeType 1的开发已经停滞，而 FreeType 2则是主流。因而，若没有特殊情形，应优先使用 FreeType 2来支持 TrueType 字体。

从 MiniGUI 官网或者 FreeType官网下载FreeType 2的源代码包并解开后进入源代码目录，之后运行如下命令：

```
$ ./configure --prefix=/usr/local
$ make
$ sudo make install
```

这样 FreeType 2库和头文件就会被安装到 /usr/local 目录下。

### *LibJPEG，LibPNG，LibZ等依赖库*

运行 MiniGUI 所依赖的函数库还包括用来支持 JPEG 图片的 libjpeg、用来支持 PNG 图片的 libpng 等等。和 FreeType 库一样，常见的 Linux 发行版中均已包含这些函数库。

首先安装LibZ 库。LibZ 库提供了 Z 算法的压缩解压功能，而PNG图像格式文件使用了同样的压缩算法，所以在编译安装 LibPNG 之前，首先要安装 LibZ库。下载并解开 LibZ库的源代码包，然后进入源代码根目录，执行如下命令：

```
$ ./configure --prefix=/usr/local
$ make
$ sudo make install
```

这样 Z 库就会被安装到 /usr/local 目录下。

下载 LibPNG 函数库源代码，解开后进入源代码根目录，执行如下命令：

```
$ ./configure --prefix=/usr/local
$ make
$ sudo make install
```

这样 PNG 库和头文件就会被安装到 /usr/local 目录下。

下载 LibJPEG 函数库源代码，解开后进入源代码根目录，执行如下命令：

```
$ ./configure --prefix=/usr/local --enable-shared
```

```
$ make
$ sudo make install
```

安装过程中可能会提示说不能创建某些文件，这时需要查看你要安装的路径下有没有对应的目录，如果没有则需要自己创建。 这样 JPEG 函数库的头文件、动态库以及静态库就会被安装到 /usr/local 目录下。

### 2.4.2 编译和安装虚拟帧缓冲区程序

MiniGUI 3.0的默认虚拟帧缓冲区图形引擎为pc_xvfb，该图形引擎定义了一个不依赖于特定实现的虚拟帧缓冲区程序（XVFB）规范，在此规范之下，我们在 Linux 上可以使用gvfb程序（使用 Gtk+开发），或者使用qvfb2 程序（使用 Qt开发），将MiniGUI及其应用程序的输出结果展示在 gvfb 或者qvfb的窗口中。

#### *gvfb*

gvfb是兼容 MiniGUI 3.0 XVFB规范的虚拟帧缓冲区程序，使用 Gtk+ 2.0 开发。要编译安装 gvfb，要确保系统中已安装有 Gtk+ 2.0 的开发包。在 Ubuntu Linux 下，使用如下命令安装相应的开发包：

```
$ sudo apt-get install libgtk2.0-dev
```

然后进入 gvfb 的源代码目录，运行如下命令：

```
$ ./configure --prefix=/usr/local
$ make
$ sudo make install
```

若成功，则gvfb 程序将安装到 /usr/local/bin 目录下。

#### *qvfb2*

qvfb2 是 qvfb的升级版本，用于兼容 MiniGUI 3.0 提出的 XVFB 规范。

要编译qvfb2，需要先安装Qt开发包，并且Qt版本需要大于等于3.0.3。具体的安装过程可以参考源代码中的 README 文件。下边举例说明一下 qvfb2 在 ubuntu 环境下安装的具体过程。

```
$ sudo apt-get install build-essential xorg-dev
```

Qt3 的库及其头文件等相关内容的安装：

```
$ sudo apt-get install libqt3-headers libqt3-mt libqt3-mt-dev
```

然后进入 qvfb2 的源代码目录，运行如下命令：

```
$ ./configure --prefix=/usr/local \
                  --with-qt-includes=/usr/include/qt3/ \
                  --with-qt-libraries=/usr/lib/qt3/
$ make
$ sudo make install
```

--prefix 选项指定 qvfb2 的安装路径； --with-qt-includes 选项是指定 Qt3 的头文件路径；--with-qt-libraries 选项指定了 Qt3 的库文件路径。

若以上命令成功，则qvfb2 程序将安装到 /usr/local/bin 目录下。

### 2.4.3 编译和安装 MiniGUI

类似地，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，指定交叉编译的选项以及安装路径等。

### 2.4.4 安装 MiniGUI 资源包

MiniGUI资源包（minigui-res）也通过 GNU autoconf/automake 脚本组织，所以只要运行如下命令即可安装：

```
user$ ./configure
user$ make
user$ sudo make install
```

类似地，我们也可以使用 `--prefix` 选项指定安装路径。

### 2.4.5 编译并运行MiniGUI示例程序

按前述步骤编译并安装好 MiniGUI 之后，即可编译并运行mg-samples 中的示例程序了。默认情况下，MiniGUI 将使用 pc_xvfb 图形和输入引擎，且实际使用的虚拟帧缓冲区程序为 gvfb。

运行如下命令配置和编译 mg-samples 代码包：

```
user$ sudo ldconfig
user$ ./configure
user$ make
```

第一条命令用来刷新 Linux系统的动态库缓存系统。因为默认情况下，MiniGUI 的动态库会安装到 /usr/local/lib 目录中，系统使用缓存来维护安装到系统中的所有动态函数库清单，若不刷新这个缓存，则可能会出现无法找到新安装的动态函数库的问题。

在MiniGUI多进程运行模式下运行演示程序，需要先在启动 mginit 程序，然后再运行其他示例程序。如下是在多进程模式下运行 same 游戏的过程：

```
user$ cd mginit
user$ ./mginit &
user$ cd ../same
user$ ./same
```

多线程模式下运行演示程序，相对简单，直接运行示例程序即可。如下是在多线程模式下运行 same 游戏的过程：

```
user$ cd same
user$ ./same
```

## 2.5 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，为了便于在非GNU环境下编译MiniGUI，已经提供了如下便利条件：

- ✓ 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）；
- ✓ 为特定平台及操作系统提供了若干备用的MiniGUI配置文件mgconfig.h的模板，以config-<os>-<platform>.h的格式保存在MiniGUI源码包的build/目录下；
- ✓ 提供了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，该文件中的TARGET_RULES行是用于指定当前要使用的子编译规则文件；
- ✓ 为不同的操作系统开发环境准备了若干备用的子编译规则文件，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。

通常，我们只需要根据真实的开发环境将对应的build/ config-<os>-<platform>.h拷贝到 MiniGUI 源代码顶层目录中，并重命名为mgconfig.h，然后修改 rules.make 文件，指定其中的 TARGET_RULES为对应的子编译规则文件，即可编译MiniGUI了。比如，我们要为 PC 平台的 VxWorks 操作系统编译 MiniGUI（对应的子编译规则文件为build/rules-pc.vxworks），则按如下步骤即可完成：

将build/config-vxworks-i386.h拷贝到MiniGUI 源代码顶层目录中，并重命名为mgconfig.h（假设当前目录为MiniGUI 源代码顶层目录）：

```
cygwin$ cp build/config-vxworks-i386.h  mgconfig.h
```

按如下方式修改rules.make文件的TARGET_RULES行：

```
TARGET_RULES=build/rules-pc.vxworks
```

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，主要的工作即在于提供正确的子

编译规则文件。在根据自己的开发环境编写rules-<platform>.<os>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 | |
| 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 | |

如build/rules-pc.vxworks 文件的内容：

```
# 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

#vxworks
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
```

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-pc.vxworks` 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文件，根据真实的开发环境，编写自己的子编译规则文件。

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
```

与移植和配置相关的其它内容请参见MiniGUI编程指南 V3.0-5 的第18章“图形引擎及输入引擎”和附录A“统一的MiniGUI初始化接口”。

## 2.6 使用 Ubuntu on Windows 配置和编译MiniGUI

特别提及，从 Windows 10 以来，微软在 Windows 平台上恢复了 POSIX兼容子系统，并提供 WSL（Windows Subsystem for Linux），同时还可通过 Microsoft Store 下载 Ubuntu 发行版。这样，我们可以使用运行在 Windows 10 之上的Ubuntu 环境来配置和编译MiniGUI 了。这将带给我们很多便利，因为运行在 Windows 上的 Ubuntu 是一个完整的 GNU 开发环境，所以我们可以使用 GNU autoconf/automake 脚本针对 VxWorks 等操作系统及其开发环境配置MiniGUI。

# 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。

***MiniGUI.cfg* 配制文件**

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。

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。

**渲染器**

MiniGUI 3.0 在窗口以及控件的外观绘制的实现上，采用了与以前的版本完全不同的实现机制。以前的版本在编译前就必须进行编译配置，选定界面风格，而且只能在fashion、classic 和 flat 三种风格之间选择其一。MiniGUI 3.0 采用外观渲染器技术实现窗口以及控件外观的绘制。MiniGUI 为应用程序准备了四种渲染器，在实际应用时，择其一便可。渲染器技术的优点在于，可以通过 MiniGUI.cfg 配置文件来修改界面外观，也可以通过函数接口来控制界面外观。应用程序甚至可以定制自己的渲染器，这就为应用程序根据实际应用环境灵活定制自己的界面提供了极大的方便。有关MiniGUI渲染器接口的详细内容，请参考 MiniGUI 编程手册。

MiniGUI 渲染器对窗口以及控件的各个部分进行了属性划分，然后使用绘制接口定义如何进行绘制，形成完整的一套外观渲染器机制。MiniGUI 3.0 提供了四种渲染器：classic、flat、fashion、skin。classic 是默认渲染器，即 MiniGUI 初始化时，默认采用 classic 渲染器来绘制窗口和控件。fashion 渲染器需要组件 mGPlus 的支持，MiniGUI 本身不提供对 fashion 渲染器的支持。

应用程序可以选择为某个窗口使用特定的渲染器，并定义窗口元素的外观属性信息。应用程序也可以定义自己的渲染器来绘制界面。

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.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=pc_xvfb
defaultmode=800x600-16bpp

# IAL engine
ial_engine=pc_xvfb

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 pc_xvfb段

该段是在[system]段gal_engine的键值指定为pc_xvfb时使用，pc_xvfb 支持 Linux 及 Window 环境，并且会自动打开虚拟显示终端。#{{ifdef 是给自动转换 MiniGUI.cfg 到内嵌资源形式时使用，生成预编译控制。

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)
- ✓ window_caption：运行时使用 pc_xvfb 时的窗口标题。
- ✓ exec_file: gvfb 可执行文件的路径

MiniGUI.cfg文件pc_xvfb段内容如下：

```
#{{ifdef _MGGAL_PCXVFB
[pc_xvfb]
defaultmode=800x600-16bpp
window_caption=XVFB-for-MiniGUI-3.0-(Gtk-Version)
exec_file=/usr/local/bin/gvfb
#}}
```

### 3.1.5 rawbitmapfonts、varbitmapfonts、qpf、truetypefonts和type1fonts段

这些段用来指定要装载的设备字体[8]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、UPF、QPF、TrueType 设备字体，分别取 rbf、vbf、upf、qpf 和 ttf。
- 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，

[8]有关设备字体的详细内容，参考《MiniGUI 编程指南》。

设置为 0。

- 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

[upf]
font_number=0

[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
```

### 3.1.6 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 、upf 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[9]。

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，则用 *代替。
- 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。如果不想指定，则用 *代替。
- 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。

此外，从MiniGUI V2.0.x 版本开始MiniGUI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。

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

[9] 有关逻辑字体的及 CreateLogFontByName 函数的具体解释，请参考《MiniGUI 编程指南》。

符集宽度的两倍。

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
```

### 3.1.7 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
```

### 3.1.8 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
```

### 3.1.9 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
```

### 3.1.10 classic 段

渲染器支持四种模式，下面以 classic 模式描述所有配置元素，其它渲染器类似。
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。该段定义了如下表：

表 3.1 窗口元素划分以及他们在配置文件和代码里的名称

| 在配置文件中的名称 | 代码名称 | 说明注释 |
|---|---|---|
| caption | WE_METRICS_CAPTION | 窗口标题栏大小 |
| | WE_FONT_CAPTION | 窗口标题栏字体 |
| fgc_active_caption | WE_FGC_ACTIVE_CAPTION | 焦点状态窗口标题栏前景色 |
| bgca_active_caption | WE_BGCA_ACTIVE_CAPTION | 焦点状态窗口标题栏背景色渐变起始色 |
| bgcb_active_caption | WE_BGCB_ACTIVE_CAPTION | 焦点状态窗口标题栏背景色渐变终止色 |
| fgc_inactive_caption | WE_FGC_INACTIVE_CAPTION | 非焦点状态窗口标题栏前景色 |
| bgca_inactive_caption | WE_BGCA_INACTIVE_CAPTION | 非焦点状态窗口标题栏背景色渐变起始色 |
| bgcb_inactive_caption | WE_BGCB_INACTIVE_CAPTION | 非焦点状态窗口标题栏背景色渐变终止色 |
| menu | WE_METRICS_MENU | 菜单项、菜单栏的高度 |
| | WE_FONT_MENU | 菜单字体 |
| fgc_menu | WE_FGC_MENU | 菜单前景色 |
| bgc_menu | WE_BGC_MENU | 菜单背景色 |

| | | |
|---|---|---|
| border | WE_METRICS_WND_BORDER | 窗口边框宽度 |
| fgc_active_border | WE_FGC_ACTIVE_WND_BORDER | 焦点状态窗口边框颜色 |
| fgc_inactive_border | WE_FGC_INACTIVE_WND_BORDER | 非焦点状态窗口边框颜色 |
| scrollbar | WE_METRICS_SCROLLBAR | 滚动条大小 |
| fgc_msgbox | WE_FGC_MESSAGEBOX | 消息框前景色 |
| fgc_msgbox | WE_FONT_MESSAGEBOX | 消息框字体 |
| fgc_tip | WE_FGC_TOOLTIP | 提示框前景色 |
| bgc_tip | WE_BGC_TOOLTIP | 提示框背景色 |
| | WE_FONT_TOOLTIP | 提示框字体 |
| fgc_window | WE_FGC_WINDOW | 窗口前景色 |
| bgc_window | WE_BGC_WINDOW | 窗口背景色 |
| fgc_3dbox | WE_FGC_THREED_BODY | 三维立体框表面上符号的颜色，如对勾、箭头等的颜色。 |
| mainc_3dbox | WE_MAINC_THREED_BODY | 三维立体框边框及表面颜色 |
| fgc_selected_item | WE_FGC_SELECTED_ITEM | 选定菜单项（列表项）的前景色 |
| bgc_selected_item | WE_BGC_SELECTED_ITEM | 选定菜单项（列表项）的背景色 |
| bgc_selected_lostfocus | WE_BGC_SELECTED_LOSTFOCUS | 选定菜单项（列表项）失去焦点后的背景色 |
| fgc_disabled_item | WE_FGC_DISABLED_ITEM | 无效菜单项（列表项）的前景色 |
| bgc_disabled_item | WE_BGC_DISABLED_ITEM | 无效菜单项（列表项）的背景色 |
| fgc_hilight_item | WE_FGC_HIGHLIGHT_ITEM | 高亮菜单项（列表项）的前景色 |
| bgc_hilight_item | WE_BGC_HIGHLIGHT_ITEM | 高亮菜单项（列表项）的背景色 |
| fgc_significant_item | WE_FGC_SIGNIFICANT_ITEM | 重要菜单项（列表项）的前景色 |
| bgc_significant_item | WE_BGC_SIGNIFICANT_ITEM | 重要菜单项（列表项）的背景色 |
| bgc_desktop | WE_BGC_DESKTOP | 桌面背景色 |

MiniGUI.cfg文件 classic 段内容如下：

```
[classic]
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form.ico
icon1=failed.ico
icon2=help.ico
icon3=warning.ico
```

```
icon4=excalmatory.ico

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

# default icons for TreeView control
treefold=fold.ico
treeunfold=unfold.ico

# bitmap used by BUTTON control
radiobutton=classic_radio_button.bmp
checkbutton=classic_check_button.bmp

# background picture, use your favirate photo
bgpicture=none
bgpicpos=center

#window element metrics
caption=20
menu=25
border=2
scrollbar=16

#window element colors
fgc_active_caption=0xFFFFFFFF
bgca_active_caption=0xFF6A240A
bgcb_active_caption=0xFF6A240A

fgc_menu=0xFF000000
bgc_menu=0xFFCED3D6


fgc_msgbox=0xFF000000

fgc_tip=0xFF000000
bgc_tip=0xFFE7FFFF

fgc_active_border=0xFFCED3D6
fgc_inactive_border=0xFFCED3D6

fgc_inactive_caption=0xFFC8D0D4
bgca_inactive_caption=0xFF808080
bgcb_inactive_caption=0xFF808080

fgc_window=0xFF000000
bgc_window=0xFFFFFFFF

fgc_3dbox=0xFF000000
mainc_3dbox=0xFFCED3D6

fgc_selected_item=0xFFFFFFFF
bgc_selected_item=0xFF6B2408
```

```
bgc_selected_lostfocus=0xFFBDA69C

fgc_disabled_item=0xFF848284
bgc_disabled_item=0xFFCED3D6

fgc_hilight_item=0xFFFFFFFF
bgc_hilight_item=0xFF6B2408

fgc_significant_item=0xFFFFFFFF
bgc_significant_item=0xFF6B2408

bgc_desktop=0xFFC08000
```

### 3.1.11 默认配置文件

本节给出 MiniGUI 默认的运行时配置文件。

```
# MiniGUI Ver 3.0.x
# This configuration file is for classic window style.
#
# Copyright (C) 2002~2017 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 configruation 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=pc_xvfb
defaultmode=800x600-16bpp

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

[fbcon]
```

```
defaultmode=1024x768-16bpp

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

#{{ifdef _MGGAL_PCXVFB
[pc_xvfb]
defaultmode=800x600-16bpp
window_caption=XVFB-for-MiniGUI-3.0-(Gtk-Version)
exec_file=/usr/local/bin/gvfb
#}}

# 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

[upf]
font_number=0

[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

[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
[resinfo]
respath=/usr/local/share/minigui/res/

[classic]
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form.ico
icon1=failed.ico
icon2=help.ico
icon3=warning.ico
icon4=excalmatory.ico

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

# default icons for TreeView control
treefold=fold.ico
treeunfold=unfold.ico

# bitmap used by BUTTON control
radiobutton=classic_radio_button.bmp
checkbutton=classic_check_button.bmp

# background picture, use your favirate photo
bgpicture=none
bgpicpos=center
# bgpicpos=upleft
# bgpicpos=downleft
# bgpicpos=upright
# bgpicpos=downright
# bgpicpos=upcenter
# bgpicpos=downcenter
# bgpicpos=vcenterleft
# bgpicpos=vcenterright
# bgpicpos=none

#window element metrics
caption=20
menu=25
border=2
scrollbar=16

#window element colors
fgc_active_caption=0xFFFFFFFF
bgca_active_caption=0xFF6A240A
```

```
bgcb_active_caption=0xFF6A240A

fgc_menu=0xFF000000
bgc_menu=0xFFCED3D6


fgc_msgbox=0xFF000000

fgc_tip=0xFF000000
bgc_tip=0xFFE7FFFF

fgc_active_border=0xFFCED3D6
fgc_inactive_border=0xFFCED3D6

fgc_inactive_caption=0xFFC8D0D4
bgca_inactive_caption=0xFF808080
bgcb_inactive_caption=0xFF808080

fgc_window=0xFF000000
bgc_window=0xFFFFFFFF

fgc_3dbox=0xFF000000
mainc_3dbox=0xFFCED3D6

fgc_selected_item=0xFFFFFFFF
bgc_selected_item=0xFF6B2408
bgc_selected_lostfocus=0xFFBDA69C

fgc_disabled_item=0xFF848284
bgc_disabled_item=0xFFCED3D6

fgc_hilight_item=0xFFFFFFFF
bgc_hilight_item=0xFF6B2408

fgc_significant_item=0xFFFFFFFF
bgc_significant_item=0xFF6B2408

bgc_desktop=0xFFC08000

#{{ifdef _MGLF_RDR_FLAT
[flat]
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form-flat.ico
icon1=failed-flat.ico
icon2=help-flat.ico
icon3=warning-flat.ico
icon4=excalmatory-flat.ico

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

```
# default icons for TreeView control
treefold=fold-flat.ico
treeunfold=unfold-flat.ico

# bitmap used by BUTTON control
radiobutton=flat_radio_button.bmp
checkbutton=flat_check_button.bmp

# background picture, use your favirate photo
bgpicture=none
bgpicpos=center

#window element metrics
caption=20
menu=25
border=1
scrollbar=16

#window element colors
fgc_active_caption=0xFFFFFFFFF
bgca_active_caption=0xFF000000
bgcb_active_caption=0xFF000000

fgc_inactive_caption=0xFF000000
bgca_inactive_caption=0xFFFFFFFF
bgcb_inactive_caption=0xFFFFFFFF

fgc_menu=0xFF000000
bgc_menu=0xFFD8D8D8

fgc_msgbox=0xFF000000

fgc_tip=0xFF000000
bgc_tip=0xFFE7FFFF

fgc_active_border=0xFF000000
fgc_inactive_border=0xFF848284

fgc_window=0xFF000000
bgc_window=0xFFFFFFFF

fgc_3dbox=0xFF000000
mainc_3dbox=0xFFFFFFFF

fgc_selected_item=0xFFFFFFFF
bgc_selected_item=0xFF000000
bgc_selected_lostfocus=0xFFBDA69C

fgc_disabled_item=0xFF848284
bgc_disabled_item=0xFF000000

fgc_hilight_item=0xFFFFFFFF
bgc_hilight_item=0xFF664E4A
```

```
fgc_significant_item=0xFFFFFFFF
bgc_significant_item=0xFF000000

bgc_desktop=0xFFC08000

flat_tab_normal_color=0xFFC6D2CF
#}}

#{{ifdef _MGLF_RDR_SKIN
[skin]
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form.ico
icon1=failed.ico
icon2=help.ico
icon3=warning.ico
icon4=excalmatory.ico

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

# default icons for TreeView control
treefold=fold.ico
treeunfold=unfold.ico

# background picture, use your favirate photo
bgpicture=none
bgpicpos=center

#window element metrics
caption=25
menu=25
border=1
scrollbar=17

fgc_active_caption=0xFFFFFFFF
bgca_active_caption=0xFFE35400
bgcb_active_caption=0xFF686868

fgc_menu=0xFF000000
bgc_menu=0xFFD4D6FF

fgc_msgbox=0xFF000000

fgc_tip=0xFF000000
bgc_tip=0xFFFFFFFF

fgc_active_border=0xFFC8D0D4
fgc_inactive_border=0xFFC8D0D4

fgc_inactive_caption=0xFFF8E4D8
```

```
bgca_inactive_caption=0xFFDF967A
bgcb_inactive_caption=0xFF686868

fgc_window=0xFF000000
bgc_window=0xFFFFFFFF

fgc_3dbox=0xFF000000
mainc_3dbox=0xFFD8E9EC

fgc_selected_item=0xFFFFFFFF
bgc_selected_item=0xFFC56A31
bgc_selected_lostfocus=0xFFD8E9EC

fgc_disabled_item=0xFF99A8AC
bgc_disabled_item=0xFFFFFFFF

fgc_hilight_item=0xFFFFFFFF
bgc_hilight_item=0xFFC56A31

fgc_significant_item=0xFFFFFFFF
bgc_significant_item=0xFFC56A31

bgc_desktop=0xFF984E00

skin_bkgnd=skin_bkgnd.bmp
skin_caption=skin_caption.gif
skin_caption_btn=skin_cpn_btn.gif

#for scrollbar
skin_scrollbar_hshaft=skin_sb_hshaft.bmp
skin_scrollbar_vshaft=skin_sb_vshaft.bmp
skin_scrollbar_hthumb=skin_sb_hthumb.bmp
skin_scrollbar_vthumb=skin_sb_vthumb.bmp
skin_scrollbar_arrows=skin_sb_arrows.bmp

#for border
skin_tborder=skin_tborder.bmp
skin_bborder=skin_bborder.bmp
skin_lborder=skin_lborder.bmp
skin_rborder=skin_rborder.bmp

skin_arrows=skin_arrows.gif
skin_arrows_shell=skin_arrows_shell.bmp

skin_pushbtn=skin_pushbtn.gif
skin_radiobtn=skin_radiobtn.gif
skin_checkbtn=skin_checkbtn.bmp

#for treeview
skin_tree=skin_tree.bmp

skin_header=skin_header.bmp
skin_tab=skin_tab.gif
```

```
#for trackbar
skin_tbslider_h=skin_tbslider_h.gif
skin_tbslider_v=skin_tbslider_v.gif
skin_trackbar_horz=skin_tb_horz.gif
skin_trackbar_vert=skin_tb_vert.gif

#for progressbar
skin_progressbar_htrack=skin_pb_htrack.gif
skin_progressbar_vtrack=skin_pb_vtrack.gif
skin_progressbar_hchunk=skin_pb_htruck.bmp
skin_progressbar_vchunk=skin_pb_vtruck.bmp
#}}


[fashion]
# Note that max number defined in source code is 5.
iconnumber=5
icon0=form.ico
icon1=failed.ico
icon2=help.ico
icon3=warning.ico
icon4=excalmatory.ico

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

# default icons for TreeView control
treefold=fold.ico
treeunfold=unfold.ico

# bitmap used by BUTTON control
radiobutton=fashion_radio_btn.bmp
checkbutton=fashion_check_btn.bmp

# background picture, use your favirate photo
bgpicture=none
bgpicpos=center

#window element metrics
caption=25
menu=25
border=1
scrollbar=17

fgc_active_caption=0xFFFFFFFF
bgca_active_caption=0xFFE35400
bgcb_active_caption=0xFFFF953D

fgc_menu=0xFF000000
bgc_menu=0xFFFFE4BF
```

```
fgc_msgbox=0xFF000000

fgc_tip=0xFF000000
bgc_tip=0xFFFFFFFF

fgc_active_border=0xFFC8D0D4
fgc_inactive_border=0xFFC8D0D4

fgc_inactive_caption=0xFFF8E4D8
bgca_inactive_caption=0xFFDF967A
bgcb_inactive_caption=0xFFEBB99D

fgc_window=0xFF000000
bgc_window=0xFFEBB99D

fgc_3dbox=0xFF000000
mainc_3dbox=0xFFD8E9EC

fgc_selected_item=0xFFFFFFFF
bgc_selected_item=0xFFC56A31
bgc_selected_lostfocus=0xFFD8E9EC

fgc_disabled_item=0xFF99A8AC
bgc_disabled_item=0xFFFFFFFF

fgc_hilight_item=0xFFFFFFFF
bgc_hilight_item=0xFFC56A31

fgc_significant_item=0xFFFFFFFF
bgc_significant_item=0xFFC56A31

bgc_desktop=0xFF984E00
```

## 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_etc_sections 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 _etc_sections 定义如下：

```
static ETCSECTION _etc_sections [] = {
    {0, 5, "system", _system_keys,_system_values },
    {0, 1, "fbcon", _fbcon_keys,_fbcon_values },
    {0, 2, "qvfb", _qvfb_keys,_qvfb_values },
#ifdef _MGGAL_PCXVFB
    {0, 3, "pc_xvfb", _pc_xvfb_keys,_pc_xvfb_values },
#endif
    {0, 1, "rtos_xvfb", _rtos_xvfb_keys,_rtos_xvfb_values },
#ifdef _MGGAL_SHADOW
    {0, 3, "shadow", _shadow_keys,_shadow_values },
#endif
#ifdef _MGGAL_MLSHADOW
    {0, 4, "mlshadow", _mlshadow_keys,_mlshadow_values },
#endif
    {0, 12, "systemfont", _systemfont_keys,_systemfont_values },
    {0, 1, "rawbitmapfonts", _rawbitmapfonts_keys,_rawbitmapfonts_values },
    {0, 1, "varbitmapfonts", _varbitmapfonts_keys,_varbitmapfonts_values },
    {0, 1, "upf", _upf_keys,_upf_values },
    {0, 1, "qpf", _qpf_keys,_qpf_values },
    {0, 1, "truetypefonts", _truetypefonts_keys,_truetypefonts_values },
    {0, 1, "mouse", _mouse_keys,_mouse_values },
    {0, 2, "event", _event_keys,_event_values },
    {0, 25, "cursorinfo", _cursorinfo_keys,_cursorinfo_values },
    {0, 1, "resinfo", _resinfo_keys,_resinfo_values },
    {0, 45, "classic", _classic_keys,_classic_values },
#ifdef _MGLF_RDR_FLAT
    {0, 46, "flat", _flat_keys,_flat_values },
#endif
#ifdef _MGLF_RDR_SKIN
    {0, 71, "skin", _skin_keys,_skin_values },
#endif
    {0, 45, "fashion", _fashion_keys,_fashion_values }
};
```

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 3.0.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：

```
static ETC_S _ETC = {
    0,
    sizeof(_etc_sections)/sizeof(ETCSECTION),
    _etc_sections
};
```

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

### 3.2.3 mgetc.c文件清单

```
/************************************************
* This is inside mode of system res configuation *
* It's generated by the mgcfg-trans, version 1.0 *
* author : dongjunjie in feynman                 *
* please donnot modify this file, if you want,   *
* please change your input file and regenerate   *
* this file                                      *
************************************************/
#include <stdio.h>
#include "common.h"
#include "minigui.h"

#ifdef _MGINCORE_RES
```

```
// This configuration file is for MiniGUI V3.0.x
//
// Copyright (C) 2002~2008 Feynman Software
// Copyright (C) 1998~2002 Wei Yongming.
//
// Web: http://www.minigui.com
//
// 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 configruation 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.
//
// Section: system
static char* _system_keys[]={
// GAL engine and default options
    "gal_engine",
    "defaultmode",
// IAL engine
    "ial_engine",
    "mdev",
    "mtype"
};
static char* _system_values[]={
// GAL engine and default options
    "pc_xvfb",
    "800x600-16bpp",
// IAL engine
    "pc_xvfb",
    "/dev/input/mice",
    "IMPS2"
};
// Section: fbcon
static char* _fbcon_keys[]={
    "defaultmode"
};
static char* _fbcon_values[]={
    "1024x768-16bpp"
};
// Section: qvfb
static char* _qvfb_keys[]={
    "defaultmode",
    "display"
};
```

```
static char* _qvfb_values[]={
    "640x480-16bpp",
    "0"
};
#ifdef _MGGAL_PCXVFB
// Section: pc_xvfb
static char* _pc_xvfb_keys[]={
    "defaultmode",
    "window_caption",
    "exec_file"
};
static char* _pc_xvfb_values[]={
    "800x600-16bpp",
    "XVFB-for-MiniGUI-3.0-(Gtk-Version)",
    "/usr/local/bin/gvfb"
};
#endif
// Section: rtos_xvfb
static char* _rtos_xvfb_keys[]={
    "defaultmode"
};
static char* _rtos_xvfb_values[]={
    "800x600-16bpp"
};
#ifdef _MGGAL_SHADOW
// Section: shadow
static char* _shadow_keys[]={
    "real_engine",
    "defaultmode",
    "rotate_screen"
};
static char* _shadow_values[]={
    "pc_xvfb",
    "800x600-16bpp",
    "normal"
};
#endif
#ifdef _MGGAL_MLSHADOW
// Section: mlshadow
static char* _mlshadow_keys[]={
    "real_engine",
    "defaultmode",
    "def_bgcolor",
    "double_buffer"
};
static char* _mlshadow_values[]={
    "qvfb",
    "800x600-16bpp",
    "0x00FF00",
    "enable"
};
#endif
// The first system font must be a logical font using RBF device font.
```

```
// Section: systemfont
static char* _systemfont_keys[]={
    "font_number",
    "font0",
    "font1",
    "font2",
    "font3",
    "font4",
    "default",
    "wchar_def",
    "fixed",
    "caption",
    "menu",
    "control"
};
static char* _systemfont_values[]={
    "5",
    "rbf-FixedSys-rrncnn-8-16-ISO8859-1",
    "*-FixedSys-rrncnn-*-16-ISO8859-1",
    "*-Courier-rrncnn-*-16-ISO8859-1",
    "*-SansSerif-rrncnn-*-16-ISO8859-1",
    "*-System-rrncnn-*-16-ISO8859-1",
    "0",
    "4",
    "1",
    "4",
    "2",
    "3"
};
// Section: rawbitmapfonts
static char* _rawbitmapfonts_keys[]={
    "font_number"
};
static char* _rawbitmapfonts_values[]={
    "0"
};
// Section: varbitmapfonts
static char* _varbitmapfonts_keys[]={
    "font_number"
};
static char* _varbitmapfonts_values[]={
    "0"
};
// Section: upf
static char* _upf_keys[]={
    "font_number"
};
static char* _upf_values[]={
    "0"
};
// Section: qpf
static char* _qpf_keys[]={
    "font_number"
```

```
};
static char* _qpf_values[]={
    "0"
};
// Section: truetypefonts
static char* _truetypefonts_keys[]={
    "font_number"
};
static char* _truetypefonts_values[]={
    "0"
};
// Section: mouse
static char* _mouse_keys[]={
    "dblclicktime"
};
static char* _mouse_values[]={
    "300"
};
// Section: event
static char* _event_keys[]={
    "timeoutusec",
    "repeatusec"
};
static char* _event_values[]={
    "300000",
    "50000"
};
// Section: cursorinfo
static char* _cursorinfo_keys[]={
// Edit following line to specify cursor files path
    "cursorpath",
    "cursornumber",
    "cursor0",
    "cursor1",
    "cursor2",
    "cursor3",
    "cursor4",
    "cursor5",
    "cursor6",
    "cursor7",
    "cursor8",
    "cursor9",
    "cursor10",
    "cursor11",
    "cursor12",
    "cursor13",
    "cursor14",
    "cursor15",
    "cursor16",
    "cursor17",
    "cursor18",
    "cursor19",
    "cursor20",
```

```
    "cursor21",
    "cursor22"
};
static char* _cursorinfo_values[]={
// Edit following line to specify cursor files path
    "/usr/local/share/minigui/res/cursor/",
    "23",
    "d_arrow.cur",
    "d_beam.cur",
    "d_pencil.cur",
    "d_cross.cur",
    "d_move.cur",
    "d_sizenwse.cur",
    "d_sizenesw.cur",
    "d_sizewe.cur",
    "d_sizens.cur",
    "d_uparrow.cur",
    "d_none.cur",
    "d_help.cur",
    "d_busy.cur",
    "d_wait.cur",
    "g_rarrow.cur",
    "g_col.cur",
    "g_row.cur",
    "g_drag.cur",
    "g_nodrop.cur",
    "h_point.cur",
    "h_select.cur",
    "ho_split.cur",
    "ve_split.cur"
};
// Section: resinfo
static char* _resinfo_keys[]={
    "respath"
};
static char* _resinfo_values[]={
    "/usr/local/share/minigui/res/"
};
// Section: classic
static char* _classic_keys[]={
// Note that max number defined in source code is 5.
    "iconnumber",
    "icon0",
    "icon1",
    "icon2",
    "icon3",
    "icon4",
// default icons for new OpenFileDialogBox
    "dir",
    "file",
// default icons for TreeView control
    "treefold",
    "treeunfold",
```

```
// bitmap used by BUTTON control
    "radiobutton",
    "checkbutton",
// background picture, use your favirate photo
    "bgpicture",
    "bgpicpos",
// bgpicpos=upleft
// bgpicpos=downleft
// bgpicpos=upright
// bgpicpos=downright
// bgpicpos=upcenter
// bgpicpos=downcenter
// bgpicpos=vcenterleft
// bgpicpos=vcenterright
// bgpicpos=none
//window element metrics
    "caption",
    "menu",
    "border",
    "scrollbar",
//window element colors
    "fgc_active_caption",
    "bgca_active_caption",
    "bgcb_active_caption",
    "fgc_menu",
    "bgc_menu",
    "fgc_msgbox",
    "fgc_tip",
    "bgc_tip",
    "fgc_active_border",
    "fgc_inactive_border",
    "fgc_inactive_caption",
    "bgca_inactive_caption",
    "bgcb_inactive_caption",
    "fgc_window",
    "bgc_window",
    "fgc_3dbox",
    "mainc_3dbox",
    "fgc_selected_item",
    "bgc_selected_item",
    "bgc_selected_lostfocus",
    "fgc_disabled_item",
    "bgc_disabled_item",
    "fgc_hilight_item",
    "bgc_hilight_item",
    "fgc_significant_item",
    "bgc_significant_item",
    "bgc_desktop"
};
static char* _classic_values[]={
// Note that max number defined in source code is 5.
    "5",
    "form.ico",
```

```
    "failed.ico",
    "help.ico",
    "warning.ico",
    "excalmatory.ico",
// default icons for new OpenFileDialogBox
    "folder.ico",
    "textfile.ico",
// default icons for TreeView control
    "fold.ico",
    "unfold.ico",
// bitmap used by BUTTON control
    "classic_radio_button.bmp",
    "classic_check_button.bmp",
// background picture, use your favirate photo
    "none",
    "center",
// bgpicpos=upleft
// bgpicpos=downleft
// bgpicpos=upright
// bgpicpos=downright
// bgpicpos=upcenter
// bgpicpos=downcenter
// bgpicpos=vcenterleft
// bgpicpos=vcenterright
// bgpicpos=none
//window element metrics
    "20",
    "25",
    "2",
    "16",
//window element colors
    "0xFFFFFFFF",
    "0xFF6A240A",
    "0xFF6A240A",
    "0xFF000000",
    "0xFFCED3D6",
    "0xFF000000",
    "0xFF000000",
    "0xFFE7FFFF",
    "0xFFCED3D6",
    "0xFFCED3D6",
    "0xFFC8D0D4",
    "0xFF808080",
    "0xFF808080",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFFCED3D6",
    "0xFFFFFFFF",
    "0xFF6B2408",
    "0xFFBDA69C",
    "0xFF848284",
    "0xFFCED3D6",
```

```
    "0xFFFFFFFF",
    "0xFF6B2408",
    "0xFFFFFFFF",
    "0xFF6B2408",
    "0xFFC08000"
};
#ifdef _MGLF_RDR_FLAT
// Section: flat
static char* _flat_keys[]={
// Note that max number defined in source code is 5.
    "iconnumber",
    "icon0",
    "icon1",
    "icon2",
    "icon3",
    "icon4",
// default icons for new OpenFileDialogBox
    "dir",
    "file",
// default icons for TreeView control
    "treefold",
    "treeunfold",
// bitmap used by BUTTON control
    "radiobutton",
    "checkbutton",
// background picture, use your favirate photo
    "bgpicture",
    "bgpicpos",
//window element metrics
    "caption",
    "menu",
    "border",
    "scrollbar",
//window element colors
    "fgc_active_caption",
    "bgca_active_caption",
    "bgcb_active_caption",
    "fgc_inactive_caption",
    "bgca_inactive_caption",
    "bgcb_inactive_caption",
    "fgc_menu",
    "bgc_menu",
    "fgc_msgbox",
    "fgc_tip",
    "bgc_tip",
    "fgc_active_border",
    "fgc_inactive_border",
    "fgc_window",
    "bgc_window",
    "fgc_3dbox",
    "mainc_3dbox",
    "fgc_selected_item",
    "bgc_selected_item",
```

```
    "bgc_selected_lostfocus",
    "fgc_disabled_item",
    "bgc_disabled_item",
    "fgc_hilight_item",
    "bgc_hilight_item",
    "fgc_significant_item",
    "bgc_significant_item",
    "bgc_desktop",
    "flat_tab_normal_color"
};
static char* _flat_values[]={
// Note that max number defined in source code is 5.
    "5",
    "form-flat.ico",
    "failed-flat.ico",
    "help-flat.ico",
    "warning-flat.ico",
    "excalmatory-flat.ico",
// default icons for new OpenFileDialogBox
    "folder-flat.ico",
    "textfile-flat.ico",
// default icons for TreeView control
    "fold-flat.ico",
    "unfold-flat.ico",
// bitmap used by BUTTON control
    "flat_radio_button.bmp",
    "flat_check_button.bmp",
// background picture, use your favirate photo
    "none",
    "center",
//window element metrics
    "20",
    "25",
    "1",
    "16",
//window element colors
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFF000000",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFFD8D8D8",
    "0xFF000000",
    "0xFF000000",
    "0xFFE7FFFF",
    "0xFF000000",
    "0xFF848284",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFFFFFFFF",
```

```
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFFBDA69C",
    "0xFF848284",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFF664E4A",
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFFC08000",
    "0xFFC6D2CF"
};
#endif
#ifdef _MGLF_RDR_SKIN
// Section: skin
static char* _skin_keys[]={
// Note that max number defined in source code is 5.
    "iconnumber",
    "icon0",
    "icon1",
    "icon2",
    "icon3",
    "icon4",
// default icons for new OpenFileDialogBox
    "dir",
    "file",
// default icons for TreeView control
    "treefold",
    "treeunfold",
// background picture, use your favirate photo
    "bgpicture",
    "bgpicpos",
//window element metrics
    "caption",
    "menu",
    "border",
    "scrollbar",
    "fgc_active_caption",
    "bgca_active_caption",
    "bgcb_active_caption",
    "fgc_menu",
    "bgc_menu",
    "fgc_msgbox",
    "fgc_tip",
    "bgc_tip",
    "fgc_active_border",
    "fgc_inactive_border",
    "fgc_inactive_caption",
    "bgca_inactive_caption",
    "bgcb_inactive_caption",
    "fgc_window",
    "bgc_window",
    "fgc_3dbox",
```

```
    "mainc_3dbox",
    "fgc_selected_item",
    "bgc_selected_item",
    "bgc_selected_lostfocus",
    "fgc_disabled_item",
    "bgc_disabled_item",
    "fgc_hilight_item",
    "bgc_hilight_item",
    "fgc_significant_item",
    "bgc_significant_item",
    "bgc_desktop",
    "skin_bkgnd",
    "skin_caption",
    "skin_caption_btn",
//for scrollbar
    "skin_scrollbar_hshaft",
    "skin_scrollbar_vshaft",
    "skin_scrollbar_hthumb",
    "skin_scrollbar_vthumb",
    "skin_scrollbar_arrows",
//for border
    "skin_tborder",
    "skin_bborder",
    "skin_lborder",
    "skin_rborder",
    "skin_arrows",
    "skin_arrows_shell",
    "skin_pushbtn",
    "skin_radiobtn",
    "skin_checkbtn",
//for treeview
    "skin_tree",
    "skin_header",
    "skin_tab",
//for trackbar
    "skin_tbslider_h",
    "skin_tbslider_v",
    "skin_trackbar_horz",
    "skin_trackbar_vert",
//for progressbar
    "skin_progressbar_htrack",
    "skin_progressbar_vtrack",
    "skin_progressbar_hchunk",
    "skin_progressbar_vchunk"
};
static char* _skin_values[]={
// Note that max number defined in source code is 5.
    "5",
    "form.ico",
    "failed.ico",
    "help.ico",
    "warning.ico",
    "excalmatory.ico",
```

```
// default icons for new OpenFileDialogBox
    "folder.ico",
    "textfile.ico",
// default icons for TreeView control
    "fold.ico",
    "unfold.ico",
// background picture, use your favirate photo
    "none",
    "center",
//window element metrics
    "25",
    "25",
    "1",
    "17",
    "0xFFFFFFFF",
    "0xFFE35400",
    "0xFF686868",
    "0xFF000000",
    "0xFFD4D6FF",
    "0xFF000000",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFFC8D0D4",
    "0xFFC8D0D4",
    "0xFFF8E4D8",
    "0xFFDF967A",
    "0xFF686868",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFF000000",
    "0xFFD8E9EC",
    "0xFFFFFFFF",
    "0xFFC56A31",
    "0xFFD8E9EC",
    "0xFF99A8AC",
    "0xFFFFFFFF",
    "0xFFFFFFFF",
    "0xFFC56A31",
    "0xFFFFFFFF",
    "0xFFC56A31",
    "0xFF984E00",
    "skin_bkgnd.bmp",
    "skin_caption.gif",
    "skin_cpn_btn.gif",
//for scrollbar
    "skin_sb_hshaft.bmp",
    "skin_sb_vshaft.bmp",
    "skin_sb_hthumb.bmp",
    "skin_sb_vthumb.bmp",
    "skin_sb_arrows.bmp",
//for border
    "skin_tborder.bmp",
    "skin_bborder.bmp",
```

```
    "skin_lborder.bmp",
    "skin_rborder.bmp",
    "skin_arrows.gif",
    "skin_arrows_shell.bmp",
    "skin_pushbtn.gif",
    "skin_radiobtn.gif",
    "skin_checkbtn.bmp",
//for treeview
    "skin_tree.bmp",
    "skin_header.bmp",
    "skin_tab.gif",
//for trackbar
    "skin_tbslider_h.gif",
    "skin_tbslider_v.gif",
    "skin_tb_horz.gif",
    "skin_tb_vert.gif",
//for progressbar
    "skin_pb_htrack.gif",
    "skin_pb_vtrack.gif",
    "skin_pb_htruck.bmp",
    "skin_pb_vtruck.bmp"
};
#endif
// Section: fashion
static char* _fashion_keys[]={
// Note that max number defined in source code is 5.
    "iconnumber",
    "icon0",
    "icon1",
    "icon2",
    "icon3",
    "icon4",
// default icons for new OpenFileDialogBox
    "dir",
    "file",
// default icons for TreeView control
    "treefold",
    "treeunfold",
// bitmap used by BUTTON control
    "radiobutton",
    "checkbutton",
// background picture, use your favirate photo
    "bgpicture",
    "bgpicpos",
//window element metrics
    "caption",
    "menu",
    "border",
    "scrollbar",
    "fgc_active_caption",
    "bgca_active_caption",
    "bgcb_active_caption",
    "fgc_menu",
```

```
    "bgc_menu",
    "fgc_msgbox",
    "fgc_tip",
    "bgc_tip",
    "fgc_active_border",
    "fgc_inactive_border",
    "fgc_inactive_caption",
    "bgca_inactive_caption",
    "bgcb_inactive_caption",
    "fgc_window",
    "bgc_window",
    "fgc_3dbox",
    "mainc_3dbox",
    "fgc_selected_item",
    "bgc_selected_item",
    "bgc_selected_lostfocus",
    "fgc_disabled_item",
    "bgc_disabled_item",
    "fgc_hilight_item",
    "bgc_hilight_item",
    "fgc_significant_item",
    "bgc_significant_item",
    "bgc_desktop"
};
static char* _fashion_values[]={
// Note that max number defined in source code is 5.
    "5",
    "form.ico",
    "failed.ico",
    "mg_help.ico",
    "warning.ico",
    "excalmatory.ico",
// default icons for new OpenFileDialogBox
    "folder.ico",
    "textfile.ico",
// default icons for TreeView control
    "fold.ico",
    "unfold.ico",
// bitmap used by BUTTON control
    "fashion_radio_btn.bmp",
    "fashion_check_btn.bmp",
// background picture, use your favirate photo
    "none",
    "center",
//window element metrics
    "25",
    "25",
    "1",
    "17",
    "0xFFFFFFFF",
    "0xFFE35400",
    "0xFFFF953D",
    "0xFF000000",
```

```
    "0xFFFFE4BF",
    "0xFF000000",
    "0xFF000000",
    "0xFFFFFFFF",
    "0xFFC8D0D4",
    "0xFFC8D0D4",
    "0xFFF8E4D8",
    "0xFFDF967A",
    "0xFFEBB99D",
    "0xFF000000",
    "0xFFEBB99D",
    "0xFF000000",
    "0xFFD8E9EC",
    "0xFFFFFFFF",
    "0xFFC56A31",
    "0xFFD8E9EC",
    "0xFF99A8AC",
    "0xFFFFFFFF",
    "0xFFFFFFFF",
    "0xFFC56A31",
    "0xFFFFFFFF",
    "0xFFC56A31",
    "0xFF984E00"
};
static ETCSECTION _etc_sections [] = {
    {0, 5, "system", _system_keys,_system_values },
    {0, 1, "fbcon", _fbcon_keys,_fbcon_values },
    {0, 2, "qvfb", _qvfb_keys,_qvfb_values },
#ifdef _MGGAL_PCXVFB
    {0, 3, "pc_xvfb", _pc_xvfb_keys,_pc_xvfb_values },
#endif
    {0, 1, "rtos_xvfb", _rtos_xvfb_keys,_rtos_xvfb_values },
#ifdef _MGGAL_SHADOW
    {0, 3, "shadow", _shadow_keys,_shadow_values },
#endif
#ifdef _MGGAL_MLSHADOW
    {0, 4, "mlshadow", _mlshadow_keys,_mlshadow_values },
#endif
    {0, 12, "systemfont", _systemfont_keys,_systemfont_values },
    {0, 1, "rawbitmapfonts", _rawbitmapfonts_keys,_rawbitmapfonts_values },
    {0, 1, "varbitmapfonts", _varbitmapfonts_keys,_varbitmapfonts_values },
    {0, 1, "upf", _upf_keys,_upf_values },
    {0, 1, "qpf", _qpf_keys,_qpf_values },
    {0, 1, "truetypefonts", _truetypefonts_keys,_truetypefonts_values },
    {0, 1, "mouse", _mouse_keys,_mouse_values },
    {0, 2, "event", _event_keys,_event_values },
    {0, 25, "cursorinfo", _cursorinfo_keys,_cursorinfo_values },
    {0, 1, "resinfo", _resinfo_keys,_resinfo_values },
    {0, 45, "classic", _classic_keys,_classic_values },
#ifdef _MGLF_RDR_FLAT
    {0, 46, "flat", _flat_keys,_flat_values },
#endif
#ifdef _MGLF_RDR_SKIN
```

```
    {0, 71, "skin", _skin_keys,_skin_values },
#endif
    {0, 45, "fashion", _fashion_keys,_fashion_values }
};

////////////////////////////////////////////////

static ETC_S _ETC = {
    0,
    sizeof(_etc_sections)/sizeof(ETCSECTION),
    _etc_sections
};

GHANDLE __mg_get_mgetc (void)
{
    return (GHANDLE) &_ETC;
}

#endif /* _MGINCORE_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

[upf]
font_number=0

[qpf]
```

```
font_number=0

[truetypefonts]
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 Win32[10]。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。

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。

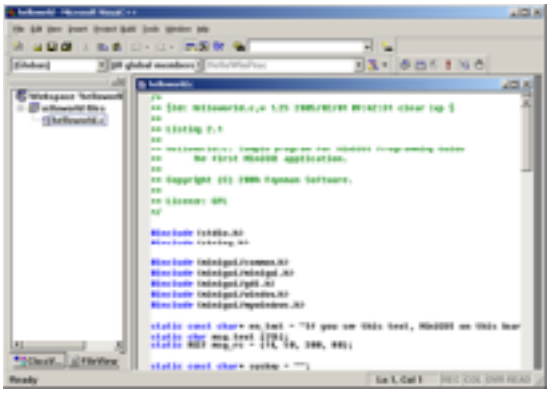

图 4.1 打开 MiniGUI 的 helloworld 工程

编译后，需首先运行开发包中的 `wvfb` 程序，然后运行 `hellowolrd` 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。

[10] 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 字体的支持（这些特性均需要第三方函数库的支持）。


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 的版权拥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拥有者，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，您可以到以下网址查阅:

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GPU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 libttf 版本太高的原因造成的。MiniGUI 支持 Freetype 1.3.1 版本及 Freetype2 2.3.5。请使用 `—with-ttfsupport=ft1/ft2/no` 等选项允许 ft1/ft2/禁止 TrueType 字体支持。具体情况请参考本文档的 MiniGUI 编译安装小节。

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 mGUtils 组件中。如果要使用，则要包括头文件：<mgutils/mgutils.h>，并且在编译的时候要链接 -lmgutils 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.6 运行时问题

15. 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。

16. 在 Linux 上，我运行 mg-samples 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 mg-samples 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。

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

答：在 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.7 常见错误信息

18. 在 Linux 上，为什么我运行 mg-samples 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-Processes，则应该首先运行 mginit 程序。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 mg-samples 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。

19. 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。

20. MiniGUI-Processes 运行模式下，我在运行 mg-samples 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。

21. 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。

22. 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。

23. 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。