

Datasheet for MiniGUI V3.0.x

MiniGUI is “a cross-operating-system graphics user interface support system for embedded devices”, and “an embedded graphics middleware”. It aims to provide a fast, stable, and cross-platform Graphics User Interface (GUI) support system, which is especially fit for embedded Linux and popular real-time operating systems.

This document lists the technology features of MiniGUI V3.0.x in detail. MiniGUI V3.0.x is the latest releases of the MiniGUI. This document intends to give you an overview about MiniGUI technology features. Every effort is made to make this document as complete as possible. You can also refer to *MiniGUI Technology White Paper V3.0* for more explanation about MiniGUI technology features.

Copyright Claim

Copyright © 2008~2017, Beijing FMSoft Technologies Co., Ltd.

All rights reserved. By whatever means you get the entire or partial text or photograph data in this documentation, no matter mechanical or electronic, you are only authorized by Beijing FMSoft Technologies Co., Ltd. the reading right. Any format conversion, redistribution, dissemination, and copying its entire or partial content, or using text or photograph therein for commercial purpose without written permission will be regarded as tortuous, and may result in severe civil or criminal punishment.

For additional contact information, please visit FMSoft's URL:

<http://www.minigui.com>

Contents

Datasheet for MiniGUI V3.0.x	1
Copyright Claim	1
Operating Systems Supported	3
MiniGUI Runtime Modes	4
Hardware Support.....	5
Footprint of MiniGUI.....	7
Graphics Sub-System	8
General Features	8
Image Formats	8
Fonts and Char Sets	8
Windowing Sub-System	11
General Features	11
Controls/Widgets.....	11
Common Dialog Boxes.....	12
Input Method.....	13
Look and Feel.....	13
Customization of MiniGUI desktop.....	13
Other Features of MiniGUI.....	14

Operating Systems Supported

MiniGUI is a complete and self-contained embedded Graphics User Interface (GUI) support system, which is designed and optimized for embedded systems. MiniGUI provides support for multiple (real-time) embedded operating systems. The OSes supported by MiniGUI include Linux, uClinux, eCos, uC/OS-II, VxWorks, pSOS, Nucleus, ThreadX, and OSE. SDK for Win32 platform is available also; it can facilitate the development and debugging of embedded applications.

Table 1: Supported Operating Systems and Runtime Modes

Operating Systems Supported	Runtime Mode(s) Supported
Linux	MiniGUI-Processes MiniGUI-Threads MiniGUI-Standalone
uClinux	MiniGUI-Threads MiniGUI-Standalone
RTOS (VxWorks, eCos, uC/OS-II, pSOS, Nucleus, ThreadX, and OSE)	MiniGUI-Threads

MiniGUI Runtime Modes

MiniGUI has three runtime modes. Different from the general-purpose operating systems like Linux, the traditional embedded operating systems have some particularities. For example, uClinux, uC/OS-II, eCos, and VxWorks usually run on non-MMU CPUs, without support for processes that have separate address spaces but only threads or tasks. Therefore, those runtime environments are entirely different from Linux. We can configure and compile MiniGUI into three runtime modes for different operating systems: MiniGUI-Threads, MiniGUI-Processes, and MiniGUI-Standalone.

Table 2: MiniGUI Runtime Modes

Runtime modes	Description	System Requirements
MiniGUI-Threads	A program running on MiniGUI-Threads can create multiple overlapped windows in different threads/tasks, and all windows running in the same address space. MiniGUI-Threads is fit for real-time operating systems such as eCos, uC/OS-II, and VxWorks. By using POSIX Threads, MiniGUI-Threads can run on Linux/uClinux as well.	<ul style="list-style-type: none">• Multiple Tasks/Threads• Semaphore/Mutex
MiniGUI-Processes	Every task in MiniGUI-Processes is a single process; multi-windows can be created for each process. At present, a complete multi-processes windowing system has already been implemented. MiniGUI-Processes are suitable for embedded system with full UNIX-like operating system, such as Linux.	<ul style="list-style-type: none">• Support for Multi-Process• Shared Memory• UNIX domain socket• Semaphore
MiniGUI-Standalone	A single-task version of MiniGUI. This mode is useful for some systems, which are lack of stable thread support, like uClinux.	<ul style="list-style-type: none">• Timer

MiniGUI can run on almost all operating systems¹ under MiniGUI-Standalone mode. MiniGUI-Threads is suitable for real-time embedded operating systems, which provide support for multi-task, or general-purpose operating systems like Linux/UNIX. Moreover, MiniGUI can run on only UNIX-like operating systems under MiniGUI-Processes mode. No matter in which mode, MiniGUI provides for application the furthest API compatibility; only a few initialization interfaces are different among different modes.

¹ Only providing support for Linux/uClinux at present.

Hardware Support

At present, MiniGUI has been proven to be capable of running smoothly on the embedded systems with such CPUs/MPUs as are based on i386, ARM, MIPS, PowerPC, and those used in low-end devices, such as DragonBall, ColdFire etc.

MiniGUI uses Graphics Abstract Layer (GAL) and Input Abstract Layer (IAL) to support different output devices and input devices. Abstract layers of graphics and input, placing no influence on API of the upper software modules, greatly facilitate the porting and debugging of MiniGUI itself as well as applications. By writing a GAL engine or IAL engine, MiniGUI can support a variety of video devices and/or input devices.

The GAL offers hardware acceleration support and makes the best use of video memory; the GDI interface based on GAL provides complete graphics APIs for applications. The GDI interface supports Alpha blending, bitmap rotating/stretching, transparent bit blitting, raster operation, as well as the advanced 2D (two-dimension) graphics functions (ellipse, polygon, pen, and brush).

We can also implement some software GAL engines and IAL engines. For example, the auto-test of application can be achieved by Random IAL engine for simulating real user input. Another example, you can support YUV output equipment by using Shadow NEWGAL engine. The Shadow engine also provides support for those graphics chips whose frame buffer cannot be accessed directly; provides support for the video modes less than 8-bpp (bits-per-pixel); and provides the function of screen rotation etc.

MiniGUI has built the support for multiple PC keyboard layouts, including American PC, French, German, Italian, Spanish, Arabic, and so on.

MiniGUI can also provide support for slave screens. If your system has multiple video devices, you can use one device as the master screen of MiniGUI to create main windows and controls and the other devices as the slave screens. By using GDI APIs of MiniGUI, you can also render text, output graphics to the slave screens.

Table 3: Hardware and Output/Input Devices

Device Type	Description
Architectures	ix86, ARM, MIPS, PowerPC, DragonBall, ColdFire, and so on.
Typical CPUs/MPUs	Pentium, xScale, S3C24x0, JZ47x0, TI DaVinci, EM86xx, HI35x0, and so on.
Output Devices	VGA, LCD, TV, OSD, and so on. No limit for resolution and color depth.
Input Devices	Keyboard, remote controller, keypad, mouse, touch screen, and so on.

Table 4: Software GAL engines and IAL engines

Engine	Description	Comment
Dummy GAL Engine	This GAL engine creates a virtual frame buffer in system memory. Therefore, we can run MiniGUI on a system by using Dummy GAL engine when the real video device is not ready.	
Shadow GAL Engine	This GAL engine creates a virtual frame buffer in system memory, but it will call sub-driver of this engine to update the pixels in the virtual frame buffer to the real video device. We can use this engine to support special video device like one in YUV color space or rotate the screen of the device.	
MLShadow GAL Engine	This GAL engine can creates multiple virtual frame buffers in system memory, and blends and updates the pixels in different buffers to the real video device. By using this engine, we can create a transparent or semitransparent surface on the screen.	New feature of MiniGUI V3.0.x
Dummy IAL Engine	This IAL engine does not do anything. If you use this engine, your MiniGUI application will not receive any input. Therefore, you can run MiniGUI on a system by using Dummy IAL engine when the real input device is not ready.	
Auto IAL Engine	You can use this IAL engine to generate pre-defined input events, so you can do auto-test of your application.	
Random IAL Engine	This engine will generate random input events, so you can use this engine to test your application.	

Footprint of MiniGUI

The typical system resources needed by MiniGUI itself are 1024KB of static memory (FLASH) and 1024KB of dynamic memory (RAM). Table 5 gives the system resources needed by MiniGUI and its application on different operating systems.

Table 5: Footprint of MiniGUI

Operating System	Minimal (FLASH/RAM)	Recommended (FLASH/RAM)
Linux	1024KB/1024KB	2048KB/4096KB
uClinux	700KB/512KB	1024KB/2048KB
RTOSes	700KB/512KB	1024KB/2048KB

Graphics Sub-System

Graphics sub-system is one of the most important sub-systems of MiniGUI. This sub-system provides APIs for application to draw graphics, to fill images, and to render text etc.

General Features

MiniGUI provides support for complete GDI APIs. You can use these APIs to do raster operations, create complex regions, draw or fill ellipses, arcs, and polygons, etc. There are advanced 2D graphics functions available by using C99 math library. By using advanced 2D graphics, you can create abstract graphics objects, like pen and brush.

Image Formats

MiniGUI provides support for almost all popular image file types including GIF, GIF89a, JPEG, PNG, Win32 BMP, etc.

Table 6: Image Types

Image Type	Description	Comments
Windows BMP	This is an image file format defined by Windows.	
GIF	This is a popular image file used on INTERNET.	
GIF89a	This is an extension of GIF to provide animation.	
JPEG ²	This is an image file format used widely by hand-held devices such as digital camera, mobile phone.	
PNG ³	This is a popular image file format intending to replace GIF.	

Fonts and Char Sets

² Support for JPEG image format is implemented by using libjpeg.

³ Support for PNG image format is implemented by using libpng.

MiniGUI provides support for multiple character sets and multiple fonts. Table 7 gives the font types supported by MiniGUI; Table 8 illustrates the char sets/encodings supported by MiniGUI.

Note that MiniGUI provides support for Arabic and Hebrew text. To support these two languages, MiniGUI provides APIs for BIDI (bi-direction) text processing.

Enhance the font and text render, including UPF (UNICODE pre-rendered font), an upgrade of QPF; VBF V3, the upgrade version of VBF V2, BITMAP font, the font glyphs can be defined by customized bitmaps, and provides APIs for BIDI text processing.

Table 7: Font Types

Font Type	Description	Comment
RBF	This font type can be used to define simple font in which all glyphs have the same width.	
VBF V3	This font type can be used to define a font in which the glyphs have different width. We have upgraded VBF to version 3, which is more efficient under MiniGUI-Processes runtime mode.	
QPF	QPF is the Qt pre-rendered font which is defined by Qt/Embedded.	
UPF	UPF is the UNICODE pre-rendered font which is defined by FMSoft to support UNICODE encoded glyphs.	
TTF	MiniGUI uses FreeType 1 or FreeType 2 to render TrueType fonts. MiniGUI can also render glyph by using sub-pixel anti-alias technology.	

Table 8: Char sets/Encodings

Char sets/Encodings	Comment
ISO8859-1,2,3,4,5,7,9,10,11,12,13,14,15,16	
ISO8859-6	Arabic (BIDI)
ISO8859-8	Hebrew (BIDI)

GB2312	Simplified Chinese
GBK	Simplified Chinese and traditional Chinese
GB18030-0	The char set standard defined by P. R. China
BIG5	Traditional Chinese
UNICODE UTF-8/UTF-16LE/UTF-16BE	Three popular encodings of UNICODE V2.0.
Shift-JIS (JISX0201 and JISX0208)	
EUC Korean (KSC5636 and KSC5601)	
EUC Japanese (JISX0201 and JISX0208)	

Windowing Sub-System

Windowing sub-system is another most important sub-system of MiniGUI. This sub-system provides APIs for applications to create window, control/widget, to update a window, to handle the messages etc.

General Features

The windowing sub-system of MiniGUI provides the following general features for application:

- 1) Mature multi-window mechanism and messaging mechanism.
- 2) Support for dialog box and message box.
- 3) Other GUI elements, including menu, acceleration key, caret, timer, etc.
- 4) Support for non-rectangular windows⁴, such as main window with round corners, irregular window and control.
- 5) Support for transparent controls.
- 6) Double buffered main window⁵. When a MiniGUI main window has double buffer, you can get the rendering result of the main window in your own buffer. By using double buffer technology, you can use an advance 2D graphics interface (mGPlus) or 3D rendering library (OpenGL ES) to get the 3D user experience easily.

Controls/Widgets

The controls/widgets provided by MiniGUI are listed in Table 9.

Table 9: Controls/Widgets

⁴ New feature of MiniGUI V3.0.x.

⁵ New feature of MiniGUI V3.0.x.

Control/Widget	Comment
Static	
Button	
Single-line edit box	
Multi-line edit box	
List box	
Combo box	
Progress bar	
Property sheet	
Tool bar	
Track bar	
Scroll bar	New feature of MiniGUI V3.0.x
Tree view	
List view	
Month calendar	
Grid view	
Icon view	
Animation	

Common Dialog Boxes

Common dialogs include Open File Dialog Box, Color Selection Dialog Box, and Font Selection Dialog Box.

Input Method

MiniGUI provides API for external input method module. You can use MiniGUI's IME API to implement soft-keyboard, hand-writing, and other input methods.

Look and Feel⁶

MiniGUI V3.0 introduces a new technology for users to customize the appearance of MiniGUI windows and controls. You can customize your own MiniGUI appearance by defining a new Look and Feel (LF) Renderer and customizing the metrics, color, font, and icon of window elements (caption, border, scrollbar, and so on).

MiniGUI includes four built-in LF renderers, which are illustrated in Table 10.

Table 10: Built-in Look and Feel Renderer

LF Renderer	Description	Comments
FLAT	This renderer can be used to support gray screen.	
CLASSIC	This renderer gives a look and feel which is similar with Windows 95 appearance.	
FASHION	This renderer gives a look and feel which is similar with Windows XP appearance.	Implemented by using mGPlus component.
SKIN	This renderer renders MiniGUI windows/controls by using pre-defined images.	

Customization of MiniGUI desktop⁷

Users can customize the MiniGUI desktop by user defined icons, and respond the event of desktop.

⁶ New feature of MiniGUI V3.0.x

⁷ New feature of MiniGUI V3.0.x

Other Features of MiniGUI

- 1) Support for built-in resources. You can compile the resources (bitmaps, icons, and fonts) into the library, so it is unnecessary to read the resources from files. Thus, MiniGUI can be used on some embedded systems without file systems.
- 2) Providing API to calibrate touch screen.
- 3) Special support for embedded systems, including the common I/O operations, byte-orders related functions, mouse (or touch-panel) position calibration etc.
- 4) Support for universal virtual frame buffer programs.
- 5) Seven MiniGUI components:
 - a) mGp is a printing component of MiniGUI, which provides a printing function to MiniGUI applications. At present, mGp supports Epson, HP and some other printers.
 - b) mGi is a input component of MiniGUI, which provides the framework of soft-keyboard and hand writing input methods. It also supplies an IME container for users to add self-defined IME to it. On the other hand, you can use self-defined keyboard bitmap for the soft-keyboard and add self-defined translation method to it.
 - c) mG3d is a 3D rendering component of MiniGUI. Using this component, you can render 3D objects in your applications.
 - d) mGUtils is a new MiniGUI component which provides API of common dialog boxes, such as ColorSelectionDialogBox, FileOpenDialogBox, and so on. It also contains the implementation of old mywins and virtual console of MiniGUI older version.
 - e) mGPlus is a new MiniGUI component which provides support for advanced graphics functions like path, gradient, and color combination.
 - f) mGNCS is the new control set introduced by miniStudio. We strongly encourage and recommend that you use mGNCS as the application framework for your new MiniGUI application.
 - g) mGEff provides an animation framework for MiniGUI applications. mGEff provides a lot of stable and efficient effectors, which can be used to implement the animations like flipping, zooming, scrolling, and so on.