

Datasheet for MiniGUI V2.0.4/V1.6.10

Beijing Feynman Software Technology Co., Ltd.

Main Technical Features of MiniGUI

First, MiniGUI is a complete and self-contained embedded graphics support system, which is designed and optimized for embedded systems. Its main features are as the follow:

- 1) Support for multiple real-time embedded operating systems¹. The Oses supported by MiniGUI include Linux, uClinux, eCos, uC/OS-II, VxWorks, pSOS, Nucleus, ThreadX, and OSE. SDK on Win32 platform is available also; it can facilitate the development and debugging of embedded applications.
- 2) Support for multiple runtime modes. You can configure and compile MiniGUI as one of three runtime modes: MiniGUI-Threads, MiniGUI-Processes² and MiniGUI-Standalone.
- 3) 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.
- 4) Mature multi-window mechanism and messaging mechanism.
- 5) Commonly used controls (widgets), including static label, button, single-line and multi-line edit boxes, list box, combo box, progress bar, property sheet, toolbar, track bar, tree view, list view, month calendar, grid view, animation, icon view, and so on.

¹ MiniGUI V2.0.x provides support for multi-process-based operating systems, like Linux; MiniGUI v1.6.x provides support for traditional real-time embedded operating systems, which are multi-thread- or multi-task- based. Except the difference of runtime modes supported, these two versions have the almost same features.

² This runtime mode is provided by MiniGUI V2.0.4 for Linux.

- 6) Support for dialog box and message box.
- 7) Other GUI elements, including menu, acceleration key, caret, timer, etc.
- 8) Support for skin. You can now use skin to build your dashy UI. It can be used to develop multi-media applications.
- 9) Support low end devices such as single color LCD and high end devices, and also support specific video devices (such as YUV device) by using graphics engine, which runs under the graphics abstract layer of MiniGUI.
- 10) Support for enhanced 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 on C99 math library. We can even implement these advanced graphics interfaces on low-end video devices by using Shadow engine which runs under the graphics abstract layer of MiniGUI.
- 11) Support for Windows resources file, for example, Windows bitmap, icon, cursor, etc.
- 12) Support for almost all popular image file types including GIF, JPEG, PNG, Win32 BMP, TGA, etc.
- 13) Support for multiple character sets and multiple fonts. At present, what are supported include such character sets as ISO8859-1 ~ ISO8859-15, GB2312, BIG5, EUCKR, UNICODE, SHIFT-JIS; bitmap fonts such as Qt Pre-rendered fonts, and vector fonts such as TrueType as well as Adobe Type1. MiniGUI V2.0 can flip and zoom in font automatically. For some special video equipment like TV set, MiniGUI also provides the anti-alias function for font rendering.
- 14) Support for multiple PC keyboard layouts, including American PC, French, German, Italian, Spanish, and so on.
- 15) Support for input method for multi-byte languages like Chinese, Korean, and Japanese. Moreover, input method for simplified Chinese (GB2312) is built-in.
- 16) Special support for embedded systems, including the common I/O operations, byte-orders related functions mouse (or touch-panel) position calibration etc.

- 17) 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.

Next, many technical innovations are worthy of noting in the course of over nine years of development. It is these technical innovations that enable MiniGUI to become more suitable for real-time embedded systems and exhibit agreeable flexibility, applicable to various high-end or low-end embedded systems, including hand-held devices, set-top boxes, game terminals, and so forth. These technical innovations include:

- 1) Abstract layer of graphics and input that, placing no influence on API of the top layer, greatly facilitates the porting and debugging etc. of MiniGUI itself as well as applications. At present, MiniGUI has been proven to be capable of running smoothly on the embedded systems with such CPUs as are based on i386, ARM (44B0, MX1, StrongARM, xScale), MIPS, PowerPC, and those used in low-end devices, such as DragonBall, ColdFire etc. The engine based on software structure can be implemented by using graphics and input abstract layer. 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 can not be accessed directly; provides support for the video modes less than 8-bpp (bits-per-pixel); and provides the function of screen rotation etc.
- 2) An optimized architecture to support multiple charsets and multiple fonts. It is very easy to add a new font type and/or a new charset support in MiniGUI. Support for various charsets is achieved by creating specific logical fonts in different charsets and/or encodings such as GB2312, BIG5, EUCKR, Shift-JIS, UNICODE UTF-8/UTF-16, etc. In a single MiniGUI application, it is very easy to display characters in different languages. Different from the traditional multi-charset implementations, which are achieved by UNICODE, MiniGUI's implementation consumes fewer resources and has more flexibility.
- 3) 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.

- MiniGUI-Threads: A program running on MiniGUI-Threads can create multiple cascaded windows in different threads, and all the windows belong to a single process. MiniGUI-Threads is fit for some real-time systems on Linux/uClinux, eCos, uC/OS-II, and VxWorks.
- MiniGUI-Processes³: Opposition to MiniGUI-threads. Every task in MiniGUI-Processes is a single process; multi-windows can be created for each process. At present, a complete multi-processes windows system has already been implemented. MiniGUI-Processes are suitable for embedded system with full UNIX performance, such as Linux.
- MiniGUI-Standalone: A single process version of MiniGUI. This mode is useful for some systems, which lack of PThread support, like some uClinux system.

MiniGUI can run on almost all operating systems⁵ 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 which mode, MiniGUI provide for application the furthest compatibility; only a few initialization interfaces are different among different modes.

In addition, aiming at the high-end video equipments and advanced embedded operating systems like Linux, we implement the new graphics abstract layer (NEWGAL), enhance the graphics interfaces and other features of MiniGUI-Processes mode. These features include:

- 1) The NEWGAL offers hardware acceleration support and makes the best use of video memory; the new GDI interface based on GAL is further enhanced. The new GDI interface supports Alpha

³ The MiniGUI-Processes run-time mode of MiniGUI V2.0 provides full-featured windowing solution for multi-processes environment. Only provides support for Linux/uClinux at present.

⁵ Only provides support for Linux/uClinux at present.

blending, bitmap rotating/stretching, transparent bit blitting, raster operation, YUV overlay, as well as the advanced graphics functions (ellipse, polygon) etc.

- 2) Support for advanced 2D graphics abstract objects, like pen, brush based on the NEWGAL.
- 3) Make use of NEWGAL, it can support access to frame buffer indirectly, or implement the convert from RGB color space to other color space (like YUV). And it provides the convenient to support special LCD controller and/or video equipments.
- 4) MiniGUI-Processes runtime mode. MiniGUI V2.0 implements a complete windowing system in multi-process environment; windows come from different processes can display on the same desktop.
- 5) MiniGUI-Processes supports the concept of Layer. The windows created by different applications can be displayed in different layer. And the server can create global windows (which can display in all layers).

The enhancement and improvement of the new version of MiniGUI, especially the full-featured multi-process support in version 2.0.x., will remarkably drive applications of MiniGUI. The new version of MiniGUI will make it easy to support multi-media applications, game applications, and other advanced and complex embedded applications.

Footprint of MiniGUI

The minimal system resources needed by MiniGUI itself are 700KB of static memory (FLASH) and 1MB of dynamic memory (RAM). The following table gives the system resources needed by MiniGUI and its application on different operating systems:

Operating System	Minimal (FLASH/RAM)	Recommended (FLASH/RAM)
Linux	4/8	8/16
uClinux	2/4	4/8
eCos, VxWorks, ThreadX, uC/OS-II and so on	2/2	4/4

Feature List of MiniGUI

MiniGUI's some features depend on the ability of underlying operating system, the following table lists the relation between the concrete features and operating system:

Types of operating system MiniGUI features		Linux	uClinux	uC/OS-II	eCos	VxWorks	ThreadX pSOS Nucleus OSE
Requirements for system	Frame Buffer driver	√	√				
	Keyboard and touch screen driver	√	√	√	√	√	√
	LCD controller driver			√	√	√	√
	Use automake/autoconf tools	√	√				
	Use autoconf tools to generate mgconfig.h, then modify it manually	√	√	√	√	√	√
	Modify the mgconfig.h specified for different platform manually	√	√	√	√	√	√
Compilation way	Use automake/autoconf tools to generate makefiles	√	√		√		
	Use the makefiles written			√	√	√	√
	Use IDE like ADS and Tornado			√		√	√

Runtime mode	MiniGUI-Threads	√	√	√	√	√	√
	MiniGUI-Processes (V2.0.4)	√	√ ⁶			√ ⁷	
	MiniGUI-Standalone	√	√				
Complete windowing/messaging APIs		√	√	√	√	√	√
Underlying graphics abstract interfaces		√	√	√	√	√	√
Top-level graphics interfaces		√	√	√	√	√	√
Advanced 2D graphics interfaces ⁸		√	√		√	√	√
Interfaces to calibrate touch screen		√	√	√	√	√	√
Support for layout of keyboard		√	√	√	√	√	√
Input method		√	√	√	√	√	√

Way of using resources	In-core	√	√	√	√	√	√
	Non-incore ⁷	√	√				
Charset and encode	ISO8859-1~ISO8859-16	√	√	√	√	√	√
	GB2312	√	√	√	√	√	√
	GBK	√	√	√	√	√	√
	GB18030-0	√	√	√	√	√	√
	BIG5	√	√	√	√	√	√
	UNICODE UTF-8/UTF-16	√	√	√	√	√	√
	Shift-JIS (JISX0201 and JISX0208)	√	√	√	√	√	√
	EUC (KSC5636 and KSC5601)	√	√	√	√	√	√
	EUC (JISX0201 and JISX0208)	√	√	√	√	√	√

Font format	RBF	√	√	√	√	√	√
	VBF	√	√	√	√	√	√
	QPF	√	√	√	√	√	√
	TTF	√	√				

⁶ Though the MiniGUI-Processes can run on the uClinux, but we don't suggest user do that, because the uClinux lacks support of dynamic libraries.

⁷ MiniGUI-Processes mode only supports Linux.

⁸ It is possible to implement this function if the make tool provides the complete C99 math library.

⁹ Including in the complete C99 math library, it can implement this function

⁷ It is possible to implement the non-incore way of resources in operating system including file system.

	Adobe Type1	√	√				
Control style	Classic	√	√	√	√	√	√
	Flat	√	√	√	√	√	√
	Fashion	√	√	√	√	√	√
Image format	Windows BMP	√	√	√	√	√	√
	GIF	√	√	√	√	√	√
	JPEG ¹⁰	√	√		√	√	√
	PNG ¹¹	√	√		√	√	√
	PCX	√	√	√	√	√	√
	PBM	√	√	√	√	√	√
	TGA	√	√	√	√	√	√
Basic controls ¹²		√	√	√	√	√	√

Extension library	Extension library controls ¹³	√	√	√	√	√	√
	Skin interfaces	√	√	√	√	√	√
	MyWins interfaces	√	√	√	√	√	√
	Common dialogs ¹⁴	√	√				
	GIF89a and animation control	√	√	√	√	√	√
Virtual console		√					

¹⁰ Support for JPEG image format is implemented by libjpeg.

¹¹ Support for PNG image format is implemented by libpng.

¹² Basic Controls include: static control, button control, edit control, list box control, combo box control, process bar control, toolbar control, menu button control, track bar control, property sheet control, scroll view control, and scroll window control.

¹³ Extended library controls include, month calendar control, tree view control, spin box control, list view control, icon view control, animation control, and grid view control.

¹⁴ Common dialogs include Open File Dialog Box and Color Selection Dialog Box.

Feynman's Products Related to MiniGUI

MiniGUI Value-added Release (MiniGUI-VAR)

MiniGUI Value-Added Release (MiniGUI-VAR) is a value-added product for clients who develop commercial, proprietary (non-GPL) software based on MiniGUI. This product contains source code and you can develop your own applications based on it. The table below gives the products and the runtime modes provided by MiniGUI for the OS:

Product and Version	Runtime Mode(s) Supported
MiniGUI-VAR for Linux V2.0.4	MiniGUI-Processes MiniGUI-Threads MiniGUI-Standalone
MiniGUI-VAR for uClinux V1.6.10	MiniGUI-Threads MiniGUI-Standalone
MiniGUI-VAR for VxWorks V1.6.10	MiniGUI-Threads
MiniGUI-VAR for ThreadX V1.6.10	MiniGUI-Threads
MiniGUI-VAR for Nucleus V1.6.10	MiniGUI-Threads
MiniGUI-VAR for OSE V1.6.10	MiniGUI-Threads
MiniGUI-VAR for eCos V1.6.10	MiniGUI-Threads
MiniGUI-VAR for uC/OS-II V1.6.10	MiniGUI-Threads
MiniGUI-VAR for pSOS V1.6.10	MiniGUI-Threads

For more information about MiniGUI-VAR, please refer to:

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

MiniGUI Components Products

Except for the MiniGUI-VAR products, Feynman Software also provides some MiniGUI component products for customers. For more information, please refer to:

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

★★★

北京飞漫软件技术有限公司



Address: ROOM 508B-C, FLOOR 5, CITIC GUOAN SHUMAGANG,
No.32, HAIDIAN SOUTH ROAD, HAIDIAN DISTRICT,
BEIJING, P. R. China (100080)

Tel: 86-10-82652690/2692

Fax: 86-10-82652690/2692-88

Website: www.minigui.com