

µC/GUI™ Embedded Graphical User Interface™

DESCRIPTION

µC/GUI is universal graphical software for embedded applications that provides an efficient processor and LCD controller-independent GUI to applications using a graphical LCD. Designed for single and multi-task environments, µC/GUI is adaptable to nearly any size physical or virtual display with an LCD controller and CPU, and is delivered with all source code.

µC/GUI is compatible with nearly all CPUs and, unlike other GUIs that require a C++ compiler, µC/GUI is written entirely in ANSI-C. Processors ranging from 8- to 32-bits run µC/GUI. 16-bit CPUs (or better) achieve optimal performance.

FEATURES

- There are three types of basic package: Monochrome, gray scale or color. Each basic package provides support for:
 - 2D Graphics
 - Font support with basic fonts charset ISO 8859-1
 - Touch Screen support
- The basic packages includes one seat of the bitmap converter tool.
- Additional fonts can be added. Usage of an optional FontConverter tool is required. The FontConverter tool converts any TrueType Font into a C-file in the format used by C/GUI.
- Optional touch-screen support contains a low-level driver to handle analog input (from two 8-bit or better A/D converter channels), debouncing, and touch screen calibration.
- Written in ANSI C
- The basic package includes multiple displays/layers support.

APPLICATIONS

- Avionics
- Medical Equipment/Devices
- Data Communications Equipment
- White Goods (Appliances)
- Mobile Phones, PDAs, MIDs
- Industrial Controls
- Consumer Electronics
- Automotive

Who should use this GUI?	Developers who want to add a state of the art graphical user interface with or without touch-screen support, that want to save time on current and future embedded projects, and who want the best GUI written in C.
Supported LCD Controllers	See complete list in LCD Driver Chart
Color resolution	1 to 32 bits per pixel
Display size	Any
Small systems (no window manager) footprint	<ul style="list-style-type: none"> • RAM: 100 bytes • Stack: 600 bytes • ROM: 10-25 kb (depending on the functionality used)
Big systems (including window manager and widgets) footprint	<ul style="list-style-type: none"> • RAM: 2-6 kb (depending on number of windows required) • Stack: 1200-1800 bytes • ROM: 30-60 kb (depending on the functionality used)
Number of widgets/controls	22
Skinning	Customize widgets appearance
Multilayer	Access to more than one display/layer
Execution models	Single task system (superloop) Multitask system (any RTOS): <ul style="list-style-type: none"> • One task calling C/GUI • Multiple tasks calling C/GUI
Foreign Language Support	All languages Unicode and UTF-8 encoding Arabic language support Text rendering with Shift-JIS (Japanese Industry Standard) encoding.
Pointer Input Devices	Touch-screen, mouse, and keyboard support
Animation	Animated GIF Sprites
Multiple buffering	Avoid flickering and tearing effects
Memory Partitions	RAM Management Yes

Micrium

OPTIONAL ADD-ON MODULES

Optional add-on modules allow for memory footprint and feature customization based on the application. Modules include:

Memory Devices	Prevents the display from flickering when using drawing operations of overlapping items.
Antialiasing	Smooths curves and diagonal lines by “blending” the background color with that of the foreground.
Window Manager	Allows for creation and handling of different windows of any size. <ul style="list-style-type: none">• Widgets (or controls) – Windows with object-type properties: buttons, radio buttons, scroll bars, check boxes, list boxes, etc.• Dialogs – Typically windows that appear in order to request user input. They may contain multiple widgets, requesting data through a variety of selections, or take the form of a message box to provide information such as a note or warning.
Virtual Screen	A display area greater than the physical size of the display, requiring additional video memory. Allows for instantaneous switching between screens, even on slow CPUs.
Virtual Network Computing – VNC	VNC is a client-server system based on a simple display protocol that allows users to view a computing 'desktop' environment not only on the machine where it is running, but from anywhere on the Internet and from a wide-variety of machine architectures.

SIMULATOR

µC/GUI includes a suite of development tools to facilitate project development. µC/GUI includes a simulation environment under MSVC++ to write and test the entire user interface on the PC. All routines are identical to the embedded application, regardless of the CPU or LCD used, making debugging and development easy and convenient. Generating screen shots of the LCD that can be inserted into documentation is a snap.

MEMORY REQUIREMENTS

The memory requirements of the GUI components have been measured on a system as follows:

ARM7, IAR Embedded workbench V4.42A, Thumb mode, Size optimization

Component	ROM	RAM	Description
Window manager	+ 6.2 kB	+ 2.5 kB	Additional memory requirements of a “Hello World” application when using the window manager. The configured GUI_ALLOC_SIZE for the measurement was 2000.
Memory devices	+ 4.7 kB	+ 7 kB	Additional memory requirements of a “Hello World” application when using memory devices. The configured GUI_ALLOC_SIZE for measurement was 8000.
Antialiasing	+ 4.5 kB	+ 2* LCD_XSIZE	Additional memory requirements for the antialiasing software.
Driver	+ 2 - 8 kB	20 Bytes	Memory requirements the driver depend on configured driver and data cache is used or With a data cache the driver requires more RAM. For details please refer the driver documentation.
Multilayer	+ 2 - 8 kB		If working with a multi or a multi-display configuration, additional memory for each additional layer is required, because each layer requires its own driver.
Core	5.2 kB	80 Bytes	Memory requirements typical “Hello world” application without using additional software items.
Core / JPEG	12 kB	38 kB	Basic routines for drawing JPEG files.
Core / GIF	3.3 kB	17 kB	Basic routines for drawing GIF files.
Core / Sprites	4.7 kB	16 Bytes	Routines for drawing sprites and cursors.
Widgets	4.5 kB		The approximate basic ROM requirement for widgets depending on individual core functions used by widgets.
Widget / BUTTON	1 kB	40 Bytes	See Note 1

Micrium

Component	ROM	RAM	Description
Widget / CHECKBOX	1 kB	52 Bytes	See Note 1
Widget / DROPDOWN	1.8 kB	52 Bytes	See Note 1
Widget / EDIT	2.2 kB	28 Bytes	See Note 1
Widget / FRAMEWIN	2.2 kB	12 Bytes	See Note 1
Widget / GRAPH	2.9 kB	48 Bytes	See Note 1
Widget / GRAPH_DATA_XY	0.7 kB		See Note 1
Widget / GRAPH_DATA_YT	0.6 kB		See Note 1
Widget / HEADER	2.8 kB	32 Bytes	See Note 1
Widget / LISTBOX	3.7 kB	56 Bytes	See Note 1
Widget / LISTVIEW	3.6 kB	44 Bytes	See Note 1
Widget / MENU	5.7 kB	52 Bytes	See Note 1
Widget / MULTIEDIT	7.1 kB	16 Bytes	See Note 1
Widget / MULTIPAGE	3.9 kB	32 Bytes	See Note 1
Widget / PROGBAR	1.3 kB	20 Bytes	See Note 1
Widget / RADIOBUTTON	1.4 kB	32 Bytes	See Note 1
Widget / SCROLLBAR	2 kB	14 Bytes	See Note 1
Widget / SLIDER	1.3 kB	16 Bytes	See Note 1
Widget / TEXT	0.4 kB	16 Bytes	See Note 1

NOTE: The listed memory requirements of the widgets contain the basic routines required for creating and drawing the widget. Depending on the specific widget, there are several additional functions available, which are not listed in the table.

STACK REQUIREMENTS

Basic stack requirement: 600 bytes (approximately)

With window manager: + 600 bytes

With memory devices: + 200 bytes (recommended)

FONT MEMORY REQUIREMENTS

The following table describes the standard fonts shipped with μ C/GUI. For each font, a measurement in pixels and the ROM size in bytes are provided. The following parameters are used in the measurement diagrams:

Element	Meaning
F	Size of font in Y.
B	Distance of base line from the top of the font.
C	Height of capital characters.
L	Height of lowercase characters.
U	Size of under length used by letters such as "g", "j" or "y".

Font name	Measurement in pixels	ROM size in bytes
GUI_Font8_ASCII	F: 8, B: 7, C: 7, L: 5, U: 1	1562
GUI_Font8_1	F: 8, B: 7, C: 7, L: 5, U: 1	1562+ 1586
GUI_Font10S_ASCII	F: 10, B: 8, C: 6, L: 4, U: 2	1760
GUI_Font10S_1	F: 10, B: 8, C: 6, L: 4, U: 2	1760+ 1770
GUI_Font10_ASCII	F: 10, B: 9, C: 8, L: 6, U: 1	1800
GUI_Font10_1	F: 10, B: 9, C: 8, L: 6, U: 1	1800+ 2456
GUI_Font13_ASCII	F: 13, B: 11, C: 8, L: 6, U: 2	2076
GUI_Font13_1	F: 13, B: 11, C: 8, L: 6, U: 2	2076+ 2149
GUI_Font13B_ASCII	F: 13, B: 11, C: 8, L: 6, U: 2	2222
GUI_Font13B_1	F: 13, B: 11, C: 8, L: 6, U: 2	2222+ 2216
GUI_Font13H_ASCII	F: 13, B: 11, C: 9, L: 7, U: 2	2232
GUI_Font13H_1	F: 13, B: 11, C: 9, L: 7, U: 2	2232+ 2291
GUI_Font13HB_ASCII	F: 13, B: 11, C: 9, L: 7, U: 2	2690
GUI_Font13HB_1	F: 13, B: 11, C: 9, L: 7, U: 2	2690+ 2806
GUI_Font16_ASCII	F: 16, B: 13, C: 10, L: 7, U: 3	2714
GUI_Font16_1	F: 16, B: 13, C: 10, L: 7, U: 3	2714+ 3850
GUI_Font16_HK		6950
GUI_Font16_1HK	F: 16, B: 13, C: 10, L: 7, U: 3	120+ 6950+ 2714+ 3850
GUI_Font16B_ASCII	F: 16, B: 13, C: 10, L: 7, U: 3	2690
GUI_Font16B_1	F: 16, B: 13, C: 10, L: 7, U: 3	2690+ 2790
GUI_FontComic18B_ASCII	F: 18, B: 15, C: 12, L: 9, U: 3	3572

Micrium

Font name	Measurement in pixels	ROM size in bytes
GUI_FontComic18B_1	F: 18, B: 15, C: 12, L: 9, U: 3	3572+ 4334
GUI_Font20_ASCII	F: 20, B: 16, C: 13, L: 10, U: 4	4044
GUI_Font20_1	F: 20, B: 16, C: 13, L: 10, U: 4	4044+ 4244
GUI_Font20B_ASCII	F: 20, B: 16, C: 13, L: 10, U: 4	4164
GUI_Font20B_1	F: 20, B: 16, C: 13, L: 10, U: 4	4164+ 4244
GUI_Font24_ASCII	F: 24, B: 19, C: 15, L: 11, U: 5	4786
GUI_Font24_1	F: 24, B: 19, C: 15, L: 11, U: 5	4786+ 5022
GUI_Font24B_ASCII	F: 24, B: 19, C: 15, L: 11, U: 5	4858
GUI_Font24B_1	F: 24, B: 19, C: 15, L: 11, U: 5	4858+ 5022
GUI_FontComic24B_ASCII	F: 24, B: 20, C: 17, L: 13, U: 4	6146
GUI_FontComic24B_1	F: 24, B: 20, C: 17, L: 13, U: 4	6146+ 5598
GUI_Font32_ASCII	F: 32, B: 26, C: 20, L: 15, U: 6	7234
GUI_Font32_1	F: 32, B: 26, C: 20, L: 15, U: 6	7234+ 7734
GUI_Font32B_ASCII	F: 32, B: 25, C: 20, L: 15, U: 7	7842
GUI_Font32B_1	F: 32, B: 25, C: 20, L: 15, U: 7	7842+ 8118

DRIVER BENCHMARK

Benchmarks are used to measure the speed of display drivers on available targets. This benchmark is in no way complete, but gives an approximation of the length of time required for common operations on various targets.

Bench1: Filling: Bench the speed of filling. An area of 64*64 pixels is filled with different colors.

Bench2: Small fonts: Bench the speed of small character output. An area of 60*64 pixels is filled with small-character text.

Bench3: Big fonts: Bench the speed of big character output. An area of 65*48 pixels is filled with big character text.

Bench4: Bitmap 1bpp: Bench the speed of 1bpp bitmaps. An area of 58*8 pixels is filled with a 1bpp bitmap.

Bench 5: Bitmap 2bpp: Bench the speed of 2bpp bitmaps. An area of 32*11 pixels is filled with a 2bpp bitmap.

Bench6: Bitmap 4bpp: Bench the speed of 4bpp bitmaps. An area of 32*11 pixels is filled with a 4bpp bitmap.

Bench7: Bitmap 8bpp: Bench the speed of 8bpp bitmaps. An area of 32*11 pixels is filled with an 8bpp bitmap.

Bench8: Device-dependent bitmap, 8 or 16 bpp: Bench the speed of bitmaps 8 or 16 bits per pixel. An area of 64*8 pixels is filled with a bitmap. The color depth of the tested bitmap depends on the configuration. For configurations <= 8bpp, a bitmap with 8 bpp is used; 16bpp configurations use a 16- bpp bitmap.

Driver Benchmarks

CPU	LCD Controller (Driver)	bpp	Bench1 Filling	Bench2 Small fonts	Bench3 Big fonts	Bench4 Bitmap 1bpp	Bench5 Bitmap 2bpp	Bench6 Bitmap 4bpp	Bench7 Bitmap 8bpp	Bench8 DDP bitmap
V850SB1 (20MHz)	S1D13806 (1300)	8	16.7M	339K	1.59M	240K	459K	459K	83K	1.25M
V850SB1 (20MHz)	S1D13806 (1300)	16	8.33M	326K	1.49M	391K	388K	388K	214K	806K
ARM720T (50 MHz)	(internal) (3200)	16	7.14M	581K	1.96M	694K	645K	645K	410K	2.94M
ARM92EJ-S (200 Mhz)	(internal) (3200)	16	123M	5.21M	7.59M	2.27M	2.21M	221M	1.77M	15.2M

Micrium

IMAGE DRAWING PERFORMANCE

The purpose of the following table is to show the drawing performance of various image formats supported by μ C/GUI. The measurement for the table is performed on an ARM922T CPU (Sharp LH7A404) running with 200MHz and with 15 bpp display color depth (fixed palette = 555) using the LCDLin driver:

Image format	Megapixels/second
Internal bitmap format: 1bpp .C. file	17.186
Internal bitmap format: 4bpp .C. file	3.897
Internal bitmap format: 8bpp .C. file	4.017
Internal bitmap format: 8bpp .C. file, without palette	4.478
Internal bitmap format: 16bpp .C. file, high color	13.363
Internal bitmap format: 16bpp .C. file, high color	1.336
Internal bitmap format: 24bpp .C. file, true color	1.671
Internal bitmap format: RLE4 .C. file	6.144
Internal bitmap format: RLE8 .C. file	6.806
Internal bitmap format: RLE16 .C. file	3.74
BMP file 8bpp	4.115
BMP file 16bpp	1.134
BMP file 24bpp	1.544
BMP file 32bpp	1.525
BMP file RLE4	6.998
BMP file RLE8	6.345
GIF file	1.285
JPEG file, gray	0.516
JPEG file, gray, progressive	0.438
JPEG file, H1V1	0.402
JPEG file, H1V1, progressive	0.28
JPEG file, H2V2	0.602
JPEG file, H2V2, progressive	0.431

For pricing, delivery, and ordering information, please contact Micrium at +1 954-217-2037, or visit Micrium's website at: www.micrium.com.

Micrium

For the way Engineers work