



UCGUI 简介

译者:UCGUI

邮箱:UCGUI@163.com

主页:<http://www.ucgui.com>

版本:v1.0.0.0

1.0.....UCGUI 整体简介.	
.....UCGUI 简介.	
.....本文档的目的	
.....前提学习要求.	
1.1.....要求	
.....目标硬件系统.	
.....开发环境(编译器).	
1.2.....UCGUI 特性.	
.....示例.	
1.3.....评估板.	
1.4.....如何使用本手册.	
.....印刷上的字体说明.	
1.5.....屏幕及坐标系统.	
1.6.....LCD 控制器的连接类型.	
1.7.....数据类型.	

UCGUI

UCGUI是一种嵌入式应用中的图形支持系统.它设计用于为任何使用LCD图形显示的应用提供高效的独立于处理器及 LCD 控制器的图形用户接口,它适用单任务或是多任务系统环境,并适用于任意 LCD 控制器和 CPU 下任何尺寸的真实显示或虚拟显示.

它的设计架构是模块化的,由不同的模块中的不同层组成,由一个LCD驱动层来包含所有对 LCD 的具体图形操作,UCGUI可以在任何的 CPU 上运行,因为它是 100%的标准 C 代码编写的.

UCGUI 能够适应大多数的使用黑白或彩色 LCD 的应用,它提供非常好的允许处理灰度的颜色管理.还提供一个可扩展的 2D 图形库及占用极少 RAM 的窗口管理体系.

本文档的目的

本文档描述如何在嵌入式应用中安装,配制,使用 UCGUI 的图形用户接口,并讲解 UCGUI 的内部设计架构.



前提

本文档假定你已经备坚实的 C 语言程序设计方面的知识, 如果你觉得自己这方面还不够, 那么我们推荐 Kernighan 和 Richie 的"C 语言程序设计"给你, 它描述了最新的 C 标准, 即 ANSI C 标准, 本文档不须要具备汇编语言方面的知识.

第一章

1.1 要求

对于开发 UCGUI 图形应用不须什么目标系统, 大部分的图形应用开发都可以在模拟器下进行; 但是最终的目的是通常还是在目标系统上运行程序.

目标系统(硬件)

你的目标系统必须具备如下几点:

- [1].CPU(8/16/32/64 位)
- [2].必要的 RAM 和 ROM 存储
- [3].LCD 显示器(任何类型及分辨率的)

对于内存的需求取决于你选用的 UCGUI 的功能模块以及你所使用的目标系统上的编译器的效率. 内存的占用量无法估计准确的值, 下面就一些的数值适用于多数的目标系统.

小型系统(不含窗口管理功能)

- [1].RAM:100 字节
- [2].堆栈:500 字节
- [3].ROM:10~25K(取决于选用的 UCGUI 功能模块)

大型系统(包含窗口管理及各种窗体控件功能)

- [1].RAM: 2-6 kb (决于选用的应用中建立窗口的数量)
- [2].堆栈: 1200 bytes
- [3].ROM: 30-60 kb (决于选用的 UCGUI 功能模块)

还要注意 ROM 的需求量随着你在应用程序中使用的字体数目而增长, 以上的所有值都是粗糙的估计, 并不准确.

开发环境(编译器)

目标系统中采用的什么样的 CPU 并不重要, 但必须要有与所用 CPU 相对应的 C 编译器, 如果你所使用的编译器有什么局限性, 请联系我们, 我们会告知你这些局限性会不会在你编译程序时产生问题, 大多数的 16/32/64 位的 CPU 或 DSP 上的编译器都可以正常使用, 大部分 8 位的编译也都可以正常编译.



并不须要 C++编译器, 不过它也可以正常使用, 如果有须求的话, 应用程序也可以在 C++环境下正常编译使用.

1.2 UCGUI 的特性

UCGUI 的设计目标是为使用 LCD 作为图形显示装置的应用提供高效的/与 LCD 控制器独立及处理器独立的图形用户接口. 它适合于单任务环境及多任务环境, 如私用的操作系统或是商业的 RTOS(实时操作系统). UCGUI 以 C 源码形式提供, 并适用于任意 LCD 控制器和 CPU 下任何尺寸的真实显示或虚拟显示.它包含以下特性:

一般特性

- [1] 适用任何 8/16/32 位 CPU, 只要有相对应的标准 C 编译器.
- [2] 任何的控制器的 LCD 显示器(单色,灰度,颜色), 只要有适合的 LCD 驱动可用.
- [3] 在小模式显示时无须 LCD 控制器.
- [4] 所有接口支持使用宏进行配制.
- [5] 显示尺寸可定制.
- [6] 字符和位图可在 LCD 显示器上的任意起点显示,并不仅局限于偶数对齐的地址起点.
- [7] 程序在大小和速度上都进行了优化.
- [8] 编译时允许进行不同的优化.
- [9] 对于缓慢一些的 LCD 控制器, LCD 显存可以映射到内存当中, 从而减少访问次数到最小并达到更高的显示速度.
- [10]清晰的设计架构.
- [11]支持虚拟显示, 虚拟显示可以比实际尺寸大(即放大).

图形库

- [1] 支持不同颜色深度的位图.
- [2] 提供可用的位图转换工具.
- [3] 图形运算时绝对不含浮点运算.
- [4] 快速画点/线(不含浮点运算).
- [5] 高速画圆及多边形.
- [6] 多种画图模式.

字体集

- [1] 为基础应用提供多种不同字体:4*6, 6*8, 6*9,8*8, 8*9, 8*16, 8*17, 8*18, 24*32, 以及 8, 10, 13, 16 等几种高度(象素单位)的均衡字体(proportional fonts). 更详细的信息, 请参考第 25 章:"标准字体".
- [2] 可以方便的加入及链接进自定义字体.
- [3] 只有应用程序中用到的字体被实际链接进最后的执行映象文件中, 因此保证占用最



小数量的 ROM.

[4] 提供可用的字体转换工具.任何宿主系统(如微软 windows 系统)上的可用字体均可以经转换后使用.

字符串/数值输出

[1] 支持数值的任何字体下的十进制/二进制/十六制显示.

[2] 支持数值的任何字体下的十进制/二进制/十六制编辑输入.

窗体管理器

[1] 齐全的窗口管理, 包括剪切, 在窗体客户区外

[2] 窗体可以移动及改变大小.

[3] 支持窗口回调函数(可选功能).

[4] 窗体占用最低 RAM(每个窗体占用 20 个字节).

可选的类似 PC 机的窗体控件

[1] 可用的窗体控件(窗体对象, 也称作控件), 操作简便而且容易使用.

触摸屏及鼠标支持

[1]对于窗体控件如按钮, UCGUI 提供触摸屏及鼠标支持.

PC 下的工具

[1] 模拟器及查看器.

[2] 位图转器工具.

[3] 字体转换工具.

示范样例

为了给你一个更好的关于 UCGUI 可以完成什么的概念. 我们编写了不同的可用范例程序, 执行映像文件在 Sample\exe 下, 源码在 Sample 目录下. Sample\GUIDemo 下包含了展示更多 UCGUI 特性的范围范例程序.

1.3 评估板

一个完整的评估板包括有 LCD 显示器的演示板/C 编译器/完整的工程示例, 评估板的目的主要是测试和演示 UCGUI, 并可以熟悉 UCGUI 的开发.



评估板

[1] 三菱 M30803 CPU/SED 13705 LCD 控制器(包含图表及文档说明).

[2] LCD(320*240 分辨率)单色, 1/4 VGA 彩色显示或者 TFT.

更详细的说明, 请查看 www.micrium.com 网站.

1.4 如何使用此文档

本文档介绍了如何安装、配制、使用 UCGUI, 并描述了 UCGUI 的内部设计架构以及所有提供的功能(即应用程序开发手册, 简称 API 接口).

在实际使用 UCGUI 之前, 你应该阅读或者至少概览一遍整个手册以对 UCGUI 有一个整体的认识, 建议按照以下的步骤进行 UCGUI 的学习:

[1] 获取 UCGUI 源码到硬盘中.

[2] 通读本文档第二章:"UCGUI 开篇".

[3] 使用 UCGUI 模拟器以慢慢熟悉用它可以做什么(参考第三章:"UCGUI 模拟器").

[4] 使用文档其它部分提供的知识扩展你的程序.

文档排版中使用符号标志的约定

[参看具体的 E 文档]

风格	作用
----	----

[1] 正文.

[2]

[3] API 参数.

[4] 程序示例源码.

[5] 已经加入存在的应用中的源码.

1.5 屏幕及坐标体系

屏幕由许多可以单独控制的点组成, 这些点称为像素, 大部分 UCGUI 提供的文本显示及画图函数均可以在任意指定的点进行画写.

水平刻度为 X 轴, 垂直刻度为 Y 轴. 由 X 轴及 Y 轴坐标组成(x,y)二维坐标来描述屏幕中一点, 在需要 X,Y 坐标的作为参数的函数中 X 坐标在 Y 坐标之前传递. 屏幕(或者窗口)左上角点的坐标为(0,0). X 轴右方向为正, Y 轴下方向为正. 上图描述了 UCGUI 中的会标体系及 X 轴 Y 轴. 所有传入 API 参数中的坐标值的单位均为像素.



1.6 LCD 显示器到微控制器连接方式

UCGUI 提供所有 LCD 的访问支持, 实际上对几乎所有 LCD 控制器的都能提供独立透明的访问支持. 了解详细情况, 请参考第 20 章:"初级 UCGUI 配制". 如果你的 LCD 控制器不被支持的话请与我们联系, 我们目前正在为市场上所有的 LCD 控制器编写驱动, 也许我们已经可以提供你打算使用的 LCD 控制器的驱动. 通常的为你应用写访问 LCD 的驱动是非常简单的工作. 如果你目标硬件中需要的话, Micrium 公司可提供专业化的用户支持服务.

LCD 显示器控制器是以何种方式连接到系统中以及它是如何被程序访问的都不重要, 通常可以有几种连接方法, 在驱动程序中提供实现大部分的相同访问接口. 驱动程序通常情况下无须做什么大的修改, 可以在 LCDConf.h 文件中进行适合你的硬件的配制修改. 有关如何配制的情况可参考在第 22 章:"LCD 驱动程序"中的解释说明. 最通常的 LCD 访问方法如下所描述, 如果你只想知道如何使用 UCGUI, 以下小节可跳过.

[1] LCD 控制器总线型连接

LCD 控制器直接连接到系统总线上, 即意味着可以象 RAM 一样访问它, 这是一种非常高效的访问方法, 是强烈推荐使用的方法. LCD 显存地址定义在 LCDSEG 当中, 为了能访问 LCD 显存, 必须知道 LCD 显存在物理空间中的映射区域, 这种连接方式的不同 LCD 控制器的驱动通用的.

[2] LCD 控制器端口型连接

对于速度缓慢一些的控制器的使用在快速的处理器上, 使用端口连接方式是唯一的解决方案. 这种连接方法的缺点是访问 LCD 的速度比总线型连接慢, 不过加上访问缓存后可以最大化的提高 LCD 的访问速度, 从而使 LCD 画面更新不会明显的下降. 编写驱动时所要做的就是写出设置及读写与 LCD 控制器连接的硬件端口的宏. 这种连接方式对于不同的 LCD 控制器需要不同的驱动程序.

[3] 无须控制器的连接

LCD 显示器也可以不用 LCD 控制器而直接连接到系统上, LCD 显示数据可以直接通过一个 4 位或是 8 位的移位寄存器来传送, 这种方式的硬件解决方案的优点是成本低, 但是缺点是占用大部分的处理器时间, 其占用率在不同的 CPU 上从 20% 到差不多 100% 不等; 对于速度慢的处理器, 这是根本不能接受的. 这种连接方式根本不须要指定的驱动程序, 因为 UCGUI 只是简单的将要显示的数据传送到 LCD 显存当中, 其余就是编写周期性的将 LCD 显存中的数据传送到 LCD 中显示的硬件图形显示相关部分.

样例中提供了用 C 及优化过的 M16C 及 M16C/80 汇编语言编写的传送显存中数据到 LCD 中显示的代码.



1.7 数据类型

因为 C 并没有提供在不同的平台中固定长度的数据类型, 所以 UCGUI 定义了大部分自己的数据类型如下表所示:

数据类型	定义	注解
I8	signed char	8-bit 位有符号整数
U8	unsigned char	16-bit 无符号整数
I16	signed short	16-bit 有符号整数
U16	unsigned short	16-bit 无符号整数
I32	signed long	32-bit 有符号整数
U32	unsigned long	32-bit 无符号整数
I16P	signed short	16-bit 有符号整数
U16P	unsigned short	16-bit 无符号整数

对于多数 16/32 位的定义都可以非常方便的设置, 如果你在自己的应用程序部分定义改变或是重新定义自己的数据类型, 推荐放在 LCDConf.h 配置文件内.