

基于 MiniGUI 的嵌入式图形界面的研究与实现

张 磊,王亚刚

(西安邮电大学 计算机学院,陕西 西安 710061)

摘 要:随着 Linux 系统从服务器延伸到 PC、嵌入式设备的快速发展,便携式智能设备的应用更加普遍。图形界面对于用户变得日益重要,轻量级图形用户界面的开发也日趋重要,成为当前研究的一个热门领域并且拥有广阔的应用前景。文中对比了几种主流的嵌入式图形界面,分析了 MiniGUI 体系结构及底层实现,详细介绍了 MiniGUI 的库文件与资源文件的移植方法与运行环境的配置,提出了 MiniGUI V3.0.12 移植的一般方法,并实现了 MiniGUI 到基于 ARM-Linux 系统的开发板 S3C6410 的移植,完成了嵌入式图形界面的构建。

关键词:图形界面;嵌入式系统;MiniGUI;Linux

中图分类号:TP316.2

文献标识码:A

文章编号:1673-629X(2015)02-0194-05

doi:10.3969/j.issn.1673-629X.2015.02.044

Research and Implementation of Embedded GUI Based on MiniGUI

ZHANG Lei, WANG Ya-gang

(College of Computer, Xi'an University of Posts & Telecommunications,
Xi'an 710061, China)

Abstract: With the rapid development of embedded Linux system from server to PC and portable device, the application of portable smart devices are more common. The development of embedded GUI (Graphical User Interface) based on Linux and the lightweight GUI are becoming increasingly important, and have become a hot area of current research with wide application prospect. Based on the comparison of several mainstream embedded graphical interfaces and analysis of the architecture and underlying implementation, the transplantation method and configuration of running environment for MiniGUI library files and resource files is introduced in detail, a general process of MiniGUI V3.0.12 transplantation is proposed, and a MiniGUI based on ARM-Linux systems development board S3C6410 transplanting is implemented, completing the construction of embedded graphical interface.

Key words: graphical interface; embedded-systems; MiniGUI; Linux

0 引 言

随着集成电路及手持设备的发展,嵌入式系统对轻量级 GUI 的需求越来越迫切。近来的市场需求显示,越来越多的嵌入式系统,包括 PDA、机顶盒、手机等均要求有一个高性能、高可靠的 GUI 的支持。另外,在工业实时控制系统中对 GUI 的要求也越来越高。这些系统一般不希望建立在庞大累赘的、非常消耗系统资源的操作系统和 GUI 之上,比如 Windows 或 X Windows。由于工业控制实时系统的发展对轻型 GUI 的需求更加突出,所以轻型、占用资源少、高性能、高可靠性和可配置的 GUI 在嵌入式系统或者实时系统中的地位将越来越重要,在 Linux 上实现 GUI 嵌入

式图形软件系统的重要性也显现出来。

1 嵌入式 GUI 的对比

GUI 的广泛应用是当今计算机发展的重大成就之一,它极大地方便了非专业用户的使用。人们从此不再需要死记硬背大量的命令,取而代之的是可以通过窗口、菜单、按键等方式来方便地进行操作。常见的几种主流的嵌入式 GUI^[1]有 OpenGUI、Qt/Embedded、MiniGUI^[2]。OpenGUI 基于一个用汇编实现的 x86 图形内核,提供了一个高层的 C/C++ 图形/窗口接口。它的运行速度非常快,能够在 MS-DOS、QNX 和 Linux 下运行,但是可移植性不理想。Qt/Embedded 是一个

收稿日期:2014-03-06

修回日期:2014-06-12

网络出版时间:2014-12-27

基金项目:国家自然科学基金重点项目(61136002)

作者简介:张 磊(1989-),男,硕士,研究方向为计算机系统结构、嵌入式 GUI 与驱动开发;王亚刚,副教授,硕士生导师,研究方向为计算机系统结构、并行编译技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20141227.1343.025.html>

专门为小型设备及高端设备提供图形用户界面的应用框架和窗口系统。它运行稳定并且速度非常快,但是它的结构过于复杂,很难进行底层的扩充、定制并且资源消耗较大。MiniGUI 是基于实时嵌入式或实时系统的轻量级 GUI^[3]。它的图像抽象层对顶层 API 基本没有影响,但大大方便了 MiniGUI 应用程序的移植、调试等工作,能够在基于 ARM、MIPS、StrongARM 以及 PowerPC 等嵌入式系统上流畅运行,并且支持多字体和多字符集。表 1 是对它们的比较。

表 1 嵌入式 GUI 对比

	OpenGUI	QT/Embedded	MiniGUI
API 是否完备	是	是	是
可配置和可定制性	一般	一般	较好
函数库大小	300 K	1.5 M	500 K
可移植性	一般	较好	很好
授权条款	LGPL	QPL/GPL	GPL
稳定性	较好	较好	较好
效率	最好	一般	较好

目前,MiniGUI 已经被各大嵌入式设备运营商广泛应用于他们的产品中^[4-5]。中科院某开发组已经成功地 MiniGUI 移植到了他们的 POSIX 兼容系统上。并且 MiniGUI 能够在 CPU 主频为 30 MHz,仅有 4 MB RAM 的系统上正常运行,这是一些嵌入式产品的图形系统所无法达到的。

2 MiniGUI 的体系结构

2.1 MiniGUI 整体架构

在软件体系上 MiniGUI 介于内核和应用之间,本质上是一个软件“中间件”。MiniGUI 具有良好的软件体系架构,通过自带的 ANSIC 库以及相关的 API 实现 GUI 系统所需的功能,然后通过可移植层的抽象封装,将与操作系统、底层硬件设备的交互实现细节隐藏起来,上层应用无需与底层硬件平台有直接的调用关系,所以有较好的可移植性与可配置性,也更利于后期设计开发上层应用软件。

图 1 是 MiniGUI 的软件结构。

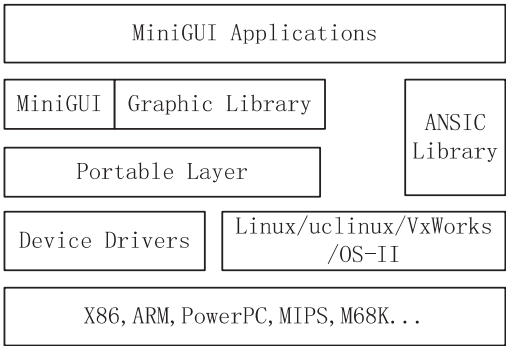


图 1 MiniGUI 软件结构

MiniGUI 采用分层设计与传统的 Client/Server 消息管理模式,服务器负责窗口 Z 序管理、鼠标和键盘的输入事件截获和分发,客户通过发请求,要求服务器完成某个操作,服务器完成任务后将结果发送给客户。在最底层 GAL (Graphics Abstract Layer) 和 IAL (Input Abstract Layer) 提供底层图形接口以及鼠标和键盘的驱动,并为 MiniGUI 提供了底层的 Linux 控制台或者 X Windows 上的图形接口以及输入接口;中间层是 MiniGUI 的核心层,其中包括了窗口系统必不可少的各个模块;顶层是 API,即编程接口。其中抽象层 GAL 与 IAL 是整个 MiniGUI 的图形库的核心。

2.2 GAL 与 IAL

实现 MiniGUI 的 GAL 与 IAL 层的底层代码称为“图形引擎”或“输入引擎”,类似操作系统中的驱动程序。利用 GAL 和 IAL,MiniGUI 可以在许多已有的图形函数库上运行,比如 SVGALib 和 LibGGI。并且可以非常方便地将 MiniGUI 移植到其他 POSIX 系统上,只需要根据抽象层接口实现新的图形引擎即可。

相比图形来讲,将 MiniGUI 的底层输入与上层相隔显得更为重要。在基于 Linux 的嵌入式系统中,图形引擎可以通过 FrameBuffer 而获得,而输入设备的处理却没有统一的接口。MiniGUI 系统维护一个已注册图形引擎数组,保存每个图形引擎数据结构的指针。系统利用一个指针保存当前使用的图形引擎。每个图形引擎的数据结构定义了该图形引擎的一些信息,比如标识符、属性等,更重要的是,它实现了 GAL 所定义的几个接口,包括初始化和终止、图形上下文管理、画点处理函数、画线处理函数、矩形框填充函数、调色板函数等等。如果在某个实际项目中所使用的图形硬件比较特殊,现有的图形引擎均不支持。这时,可以按照 GAL 所定义的接口实现自己的图形引擎,并指定 MiniGUI 使用这种私有的图形引擎即可。

在 GAL 层实现中有软件/硬件 surface 操作,即简单的赋值操作与色块的传送。同时有些硬件还支持一些像素混合的加速功能,如 alpha、colorkey 等。不管是软件还是硬件的 surface,屏幕的图像数据就是内存中的那些 RGB 数值,所以直接修改它们,就能立即显示到屏幕上^[6-7]。

2.3 MiniGUI 的 API

MiniGUI 顶层的 API 函数接口通过调用 GDI (Graphic Device Interface) 进行基本的图元绘制操作而不去关心底层的操作^[8]。MiniGUI 提供一些基本的绘图函数^[9],如 FillBox (HDC hdc, int x, int y, int w, int h)。它的基本实现流程如图 2 所示。

在 MiniGUI 中,每个绘图函数在绘图的时候都要去调用设备上下文 DC。FillBox 的作用是使用 DC 的

画刷颜色来填充 DC 中的指定矩形区域。根据 DC 的剪切域来填充矩形;根据 DC 中 surface 的类型选用填充函数;填充函数完成图像数据填充;on-screen 数据被修改直接表现在屏幕上。

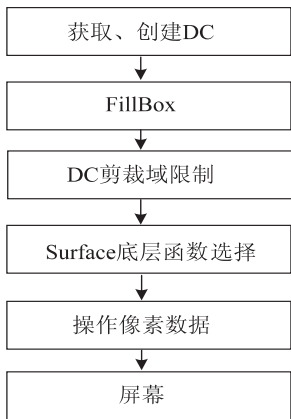


图 2 实现流程

3 MiniGUI 的移植

尽管 MiniGUI 支持多种不同的平台体系结构,但由于嵌入式硬件系统的千差万别,因此在特定的目标开发板上正确地运行 MiniGUI 还要做适当的移植修改^[10-11]。一般来说,系统的差异主要在以下几个方面:用户定制的硬件或者所使用的图形引擎没有被 MiniGUI 支持;缺少与 MiniGUI 匹配的驱动;用户需要添加新的图形引擎硬件加速功能。

3.1 移植方法与环境

一般的移植首先要选择合适的宿主机并在主机上建立交叉编译环境,其次修改与平台相关的配置文件并交叉编译相关资源文件,然后将编译好的动态链接库和资源文件拷贝到事先移植好的目标板的文件系统中,最后就可以在目标板上独立运行相应的测试程序。其中串行终端作为移植及调试过程中目标板系统信息输出的主要途径,可以使用 Windows 中的超级终端或者 Linux 下的 minicom 等终端程序。

对于 MiniGUI 的移植方法大概分为两种:一种是首先在模拟器上运行 MiniGUI,编写测试用例进行测试,最后以模拟器为参考进行相应的硬件平台的移植;另一种是在 PC 机上安装 MiniGUI 图形库,并编写测试用例进行相应的调试作为嵌入式平台移植的有效参考。为特定的开发板或者系统移植安装 MiniGUI 及相类似的图形库可以参考其他已经移植成功的开发板的程序跟配置文件,这样可以大大减少移植的工作量,缩短开发周期。

MiniGUI 在移植^[12-13]前首先要准备好相应的目标板,支持的 Linux 内核,BootLoader 引导程序^[14],支持的文件系统及相应的驱动程序。MiniGUI 的移植一般

包括以下几个步骤:获取 MiniGUI 源码包;建立 GNU 交叉编译开发环境;根据移植目标板的配置对 MiniGUI 源代码进行配置修改;在目标板与开发主机间建立连接进行调试。其中目标开发板的选择与源码包的选择尤为重要。在移植过程中 MiniGUI 的移植环境如图 3 所示。

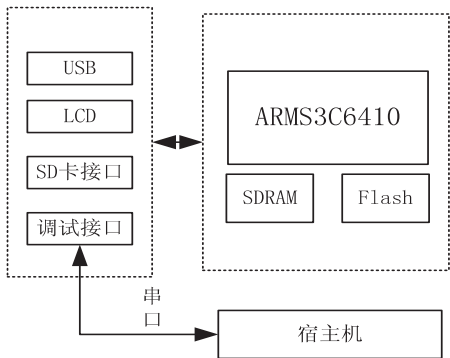


图 3 移植环境

结合以上分析,文中移植 MiniGUI 一般需要注意的事项为:

(1)明确目标硬件板的配置,满足 MiniGUI 图形库所需的最低硬件资源条件。对于 S3C6410 开发板来说,它采用 ARM11 的处理器,支持 Linux 操作系统,有 JTAG 调试接口与 480 * 273 的 LCD。

(2)宿主机的选择根据已有的资源情况进行选择,交叉编译工具要尽量选择功能齐全的、高版本的。

(3)要选择合适的源码包。对于不同版本的源码包,编译过程中会出现互不兼容的情况,所以选择合适的源码包很有必要,大大减少移植过程中会出现的错误。

3.2 移植的一般过程

结合以上对 MiniGUI 移植一般过程的介绍,文中搭建的开发环境主要有四方面。主机系统:centos 6.0, Linux 的 Kernel 版本为 2.6.32-358;交叉编译工具为:arm-linux-gcc-4.5.1;软件资源:MiniGUIV3.0.12 相关源码包;硬件平台:友善之臂 ARM6410。

3.2.1 交叉编译工具链的搭建

GUI 的编译通常在 PC 机上执行,这要求编译器本身能够在 PC 机上执行,同时编译链接源代码生成的二进制文件必须能在目标机上执行。将交叉编译工具解压到指定目录下,并修改 home 目录下的环境变量配置文件,添加交叉编译工具的路径。例如,添加 export PATH = \$ PATH:/opt/FriendlyARM/toolschain/4.5.1/bin。

3.2.2 交叉编译与安装移植

首先,获取 MiniGUI 源码包,libminigui-gpl-3.0.12;核心库,minigui-res-be-3.0.12;资源库,zlib-1.2.6;zlib 资源库,libpng-1.2.37;png 资源库,jpeg-8c;

jpeg 资源库, freetype-2.3.9-fm20100818: 字体库, lib-mgplus-1.2.4: 图形桌面中间生成库, mg-samples-3.0.12: 图形桌面。

其次, 按顺序交叉编译 zlib、png、jpeg、freetype、lib-minigui-gpl、minigui-res-be 和 libmgplus 库。

最后, 编译 mg-samples 库。

所有的库的交叉编译大体相似, 以下用 libminigui-gpl-3.0.12 核心库和 mg-samples-3.0.12 为例说明编译过程。交叉编译时首先进入相应资源库的文件中, 进行相应的编译配置, 然后进行安装。

在编译时首先使用所需的交叉编译工具, 并添加所依赖的头文件及库文件并使用 `--prefix=/usr/local/minigui` 将所有的库编译到指定的目录下, 其次添加在配置文件中要启用的选项。将编译选项配置完成后, 最后运行安装。编译过程中要确保各个图形库都要配置成功, 否则后面的编译将不会通过。在整个交叉编译过程中所要修改的平台相关的配置为:

(1) MiniGUI 根据不同硬件平台的需求提供三种不同的运行模式: 第一个为 MiniGUI-Thread 多线程模式; 第二个为 MiniGUI-Processes 多进程模式^[15]; 第三个为 MiniGUI-Standalone 独立任务模式。文中在编译时采用多进程模式, 在编译多进程核心库的时候要启用 `--enable-procs` 多进程模式。

(2) 如果所选择移植的硬件平台没有 MiniGUI 支持的图形引擎, 可以在 MiniGUI 的 GAL 层按照其所支持的其他图形引擎的结构添加适合目标平台的图形引擎, 正是由于 MiniGUI 结构灵活, 所以具有很好的可移植性。

(3) 编译好所有的资源库文件后会在 prefix 指定的目录文件下生成平台相关的 MiniGUI.cfg 资源配置文件。此时根据不同硬件平台资源特性进行相应的配置。其中有引擎、字体、所支持的图像资源等层。特别重要的是 GAL 层引擎与 IAL 层引擎的配置, 并根据实际的显示器大小进行显示分辨率配置, 指定所需资源文件的路径。文中将图形抽象层引擎 GAL_engine 配置成基于 Framebuffer 的图形引擎 fbcon, 将输入抽象层引擎 IAL_engine 配置成默认的输入引擎 console。设置所使用的 fbcon 的屏幕分辨率为 480 * 272。

①交叉编译 MiniGUI 核心库配置如下所示:

```
CC=arm-linux-gcc
CFLAGS="-I/usr/local/minigui/include" LIBS="-L/
usr/local/minigui/lib"
./configure --prefix=/usr/local/
minigui --build=i386-linux
--host=arm-linux
--target=arm-linux
--with-osname=linux
```

```
--with-style=classic
--with-targetname=fbcon
--enable-autoial --enable-procs
```

②交叉编译 mg-samples 库如下所示:

```
CC=arm-linux-gcc
CFLAGS="-I/usr/local/minigui/include"
LIBS="-L/usr/local/minigui/lib"
./configure --prefix=/usr/
local/minigui --build=i386-linux
--host=arm-linux
--target=arm-linux
```

相关资源库都交叉编译成功之后, 再交叉编译 mg-sample 库。

③将交叉编译好的资源拷贝到开发板。

首先, 把在宿主机上编译好的位于 /usr/local/minigui 下的三个文件夹中的相关资源 etc、lib、share 拷贝到目标开发板相应的位置。

其中, 把宿主机中编译好的 MiniGUI 动态链接库拷贝到目标开发板的连接库/lib下; 将配置好的 MiniGUI.cfg 文件拷贝到开发板的根目录下的/etc中; 根据 MiniGUI.cfg 文件里面的资源路径, 在目标开发板上建立相应的路径并放入编译好的资源。由于在系统初始化的时候会自动寻找/etc下的 MiniGUI.cfg 配置文件, 并根据配置加载资源, 并且应用程序在运行的时候会从动态链接库中加载需要的库。

最后, 修改开发板上图形界面的启动项, 并添加编译好的 MiniGUI 的图形界面应用程序。

3.3 移植结果

重新启动 ARM 开发板, 将会在开发板上的 LCD 界面上看到基于 MiniGUI 的图形界面, 此图形界面运行流畅, 完全满足嵌入式图形界面的要求。MiniGUI 的图形界面如图 4 所示。



图4 MiniGUI 的图形界面

所有的移植工作完成之后可以在宿主机上开发自己的应用实例。交叉编译自己开发的应用程序, 将生成可执行文件并放到指定路径, 并修改桌面客户程序的配置文件 mginit.rc, 就可以在开发板上运行。图 5

为对第 2 节 MiniGUI 的 API 实例结构 FillBox 的一个验证示例。

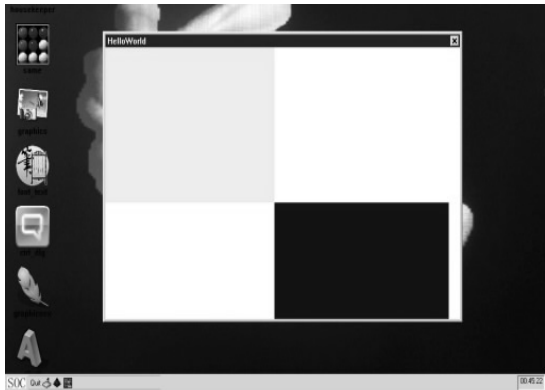


图 5 FillBox 验证示例

4 结束语

MiniGUI 现在已发展成为一个比较成熟的图形用户界面支持系统,在嵌入式 Linux 系统下采用 MiniGUI 可以比较快捷地完成相应的图形用户信息交互界面的开发。通过文中分析与实践,成功将 MiniGUI 移植到 ARMS3C6410 开发板上,并通过大量绘图测试验证 MiniGUI 的稳定性与流畅性,为嵌入式图形界面的移植提供了一个成功的范例,同时也为移植到其他开发板积累了经验。

参考文献:

[1] 孙少华,徐立中.面向嵌入式 Linux 系统的图形用户界面设计[J].微机发展(现更名:计算机技术与发展),2005,15(10):123-125.

(上接第 193 页)

[4] 曹树金,马利霞.论本体与本体语言及其在信息检索领域的应用[J].情报理论与实践,2004,27(6):632-637.

[5] Hebel J, Fisher M, Blace R, et al. Web 3.0 与 Semantic Web 编程[M].北京:清华大学出版社,2010.

[6] Horrocks L, Patel-Schneider P F, Boley H. SWRL: a semantic web rule language combining OWL and RuleML[EB/OL]. 2004. <http://www.w3.org/Submission/SWRL/>.

[7] 李景,孟宪学,苏晓路.领域本体的构建方法与应用研究[M].北京:中国农业科学技术出版社,2009.

[8] 张梅,郝佳,闫艳,等.基于本体的知识建模技术[J].北京理工大学学报,2010,30(12):1405-1408.

[9] 陈述彭.遥感大辞典[M].北京:科学出版社,1990.

[10] 陆登槐.农业遥感的应用效益及在我国的发展战略[J].农业工程学报,1998,14(3):64-70.

[11] 和海霞,范一大,杨思全,等.航天光学遥感在自然灾害管理中应用能力评述[J].航天器工程,2012,21(4):117-

[2] 魏永明,钟书毅,潘为国.嵌入式系统开发原理、工具及过程[M].北京:北京飞漫科技技术有限公司,2005.

[3] 于秀霞.嵌入式监控系统中 MiniGUI 的编程与实现[J].长春大学学报,2009,19(12):58-60.

[4] 张仁斌,朱飞,吴燎原.基于 MiniGUI 的 POS 机图形界面设计与实现[J].计算机工程与应用,2008,44(专刊):249-251.

[5] 陈文智.嵌入式系统开发原理与实践[M].北京:清华大学出版社,2005.

[6] AlexBuell. FrameBufHowTo[EB/OL]. 2000-02-27. <http://www.linuxeden.com/html/newbie/20040328/21521.html>.

[7] Daniel P B, Marco C. Understanding the Linux kernel[M]. 2nd ed. [s.l.]: O. Reilly, 2002.

[8] 周立功. ARM 嵌入式 MiniGUI 初步与应用开发范例[M].北京:北京航空航天大学出版社,2006.

[9] 刘峥嵘,张智超,许振山.嵌入式 Linux 应用开发详解[M].北京:机械工业出版社,2005.

[10] 北京飞漫软件. MiniGUI 编程手册 V1.3[EB/OL]. 2005-10-28. <http://www.minigui.com>.

[11] 刘昌盛,郭勇,谢习华.嵌入式 Linux 环境下 MiniGUI 的研究与移植[J].微计算机信息,2008,24(20):103-105.

[12] 郑端建,郭磊,魏世民. MiniGUI 图形库在嵌入式 Linux 平台上的移植与实现[J].仪表技术,2008(10):10-11.

[13] Tanenbaum A S. Modern operating system[M].北京:机械工业出版社,2002.

[14] Denk W. The DENX U-Boot and Linux Guide (DULG) for TQM8xxL[EB/OL]. 2004-11-18. <http://www.denx.de/wiki/bin/view/DULG/Manual>.

[15] 王文启,韩秀玲,孙波.基于 MiniGUI 的多进程图形用户界面的研究[J].微计算机信息,2007,23(3-2):78-80.

122.

[12] 张薇,杨思全,王磊,等.合成孔径雷达数据减灾应用潜力研究综述[J].遥感技术与应用,2012,27(6):904-911.

[13] 北京宇视蓝图信息技术有限公司.北京一号小卫星数据特征及应用前景展望[EB/OL]. 2008. <http://www.blmit.com.cn/document/lunwen.pdf>.

[14] 中国资源卫星应用中心.资源 02B 卫星介绍[EB/OL]. 2009-05-08. <http://www.cresda.com/n16/n1130/n1567/8370.html>.

[15] 中国资源卫星应用中心. HJ-1-A、B 卫星介绍[EB/OL]. 2009-05-08. <http://www.cresda.com/n16/n1130/n1582/8384.html>.

[16] 中国资源卫星应用中心. HJ-C 卫星[EB/OL]. 2010-01-05. <http://www.cresda.com/n16/n1130/n1597/49620.html>.

基于MiniGUI的嵌入式图形界面的研究与实现

作者：[张磊](#)，[王亚刚](#)，[ZHANG Lei](#)，[WANG Ya-gang](#)
作者单位：[西安邮电大学 计算机学院, 陕西 西安, 710061](#)
刊名：[计算机技术与发展](#)[ISTIC](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2015 (2)

引用本文格式：[张磊](#). [王亚刚](#). [ZHANG Lei](#). [WANG Ya-gang](#) [基于MiniGUI的嵌入式图形界面的研究与实现](#) [期刊论文] - [计算机技术与发展](#) 2015 (2)