MiniGUI在AT91RM9200开发板上的移植

■ 北京理工大学　方宁　马忠梅

摘　要　简要介绍Atmel公司生产的基于ARM9内核的AT91RM9200开发板的结构及其部分特性；详细介绍如何将MiniGUI图形用户界面移植到AT91RM9200开发板上的全过程。

关键词　AT91RM9200DK　MiniGUI　图形用户界面

引　言

近年来的市场需求显示，越来越多的嵌入式系统，包括PDA、机顶盒、DVD/VCD播放机、WAP手机等均要求提供一个方便、简洁的可视化操作界面或是全功能的Web浏览器。而这些都要求有一个高性能、稳定可靠的GUI(图形用户界面)来提供支持。本文正是针对这样的需求，介绍如何在高端的基于ARM9内核的AT91RM9200开发板上移植MiniGUI图形界面的全过程。

1　AT91RM9200开发板性能特点简介

AT91RM9200DK是Atmel公司最新推出的一款高性能开发板，此开发板基于ARM920T的32位RISC微控制器——AT91RM9200。AT91RM9200芯片基于ARM9结构，运算速度高于200MIPS。此款芯片非常适用于系统控制以及通信领域，同时，也适合于汽车、医药等领域。AT91RM9200开发板的结构图如图1所示。

AT91RM9200DK的主要特点包括：①开发板基于AT91RM9200芯片，ARM920T处理器内有2个16KB Cache,以及内存管理单元；②存储资源包括16KB的SRAM、128KB的Boot ROM、2MB的并行Flash存储器、32MB的SDRAM、128KB的EEPROM、8MB的串行DataFlash存储器；③处理器最高可达200MIPS@ 180MHz，采用0.18μm工艺；④外部总线接口支持SDRAM, Burst Flash, Compact Flash, Smart Media;⑤具有USB host/device端口；⑥介质独立接口/简化的介质独立接口(MII/RMII)的以太网介质访问控制——10/100；⑦3个SSCs支持I^2S和TDM；⑧SPI，MCI(SDCard和MMC兼容)，TWI；⑨4个USART，支持IrDA；⑩2个晶振，2个锁相环电路和实时时钟；⑪1.65~3.6V的I/O选项，内核工作电压为1.65~1.95V；⑫具有JTAG/ICE接口和边界扫描特性。

另外，此款开发板提供了Linux操作系统。AT91RM9200的Linux包括业界标准Linux v2.4.19内核和ARM Linux支持组织维护的ARM9 Linux兼容性能，还包括一套AT91RM9200 Linux代码模块，其中包括外设驱

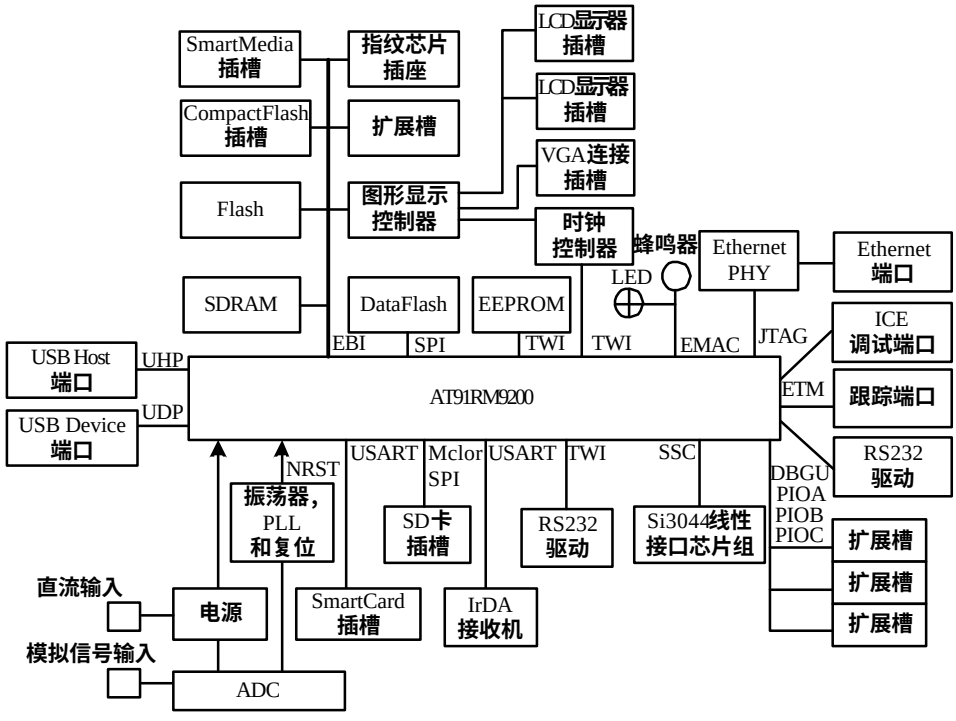

图1　AT91RM9200开发板结构图

动器和专为AT91RM9200提供的内核补丁。可以看出AT91RM9200开发板可以满足绝大多数的日常应用以及工控要求。然而，对于一个高性能的面向用户的嵌入式产品，用户图形界面(GUI)是非常必要的。

2 图形用户界面MiniGUI简介

下面简要介绍一下MiniGUI。MiniGUI是遵循GPL条款发布的自由软件，其目标是为基于Linux的实时嵌入式系统提供一个轻量级的图形用户界面支持系统。与QT/Embedded、MicoroWindows等其它GUI相比，MiniGUI的最显著特点就是轻型、占用资源少。据称MiniGUI能够在CPU主频为30MHz，仅有4MB RAM的系统上正常运行，这是其它多种GUI所无法达到的。历经四年的发展，MiniGUI已经非常成熟和稳定，并且在许多实际产品和项目中得到了应用。目前，它的最新稳定版是1.3.1。

3 MiniGUI在AT91RM9200开发板上的移植过程

3.1 构建Linux交叉编译环境

首先，需要在宿主机上建立MiniGUI的交叉编译环境。通常使用的交叉编译工具链是cross-2.95.3.tar.bz2。然而，在这里需要注意的是，AT91RM9200开发板上的Linux使用的Glibc库是2.1.3版本，而cross-2.95.3.tar.bz2里面包含的库是2.2.3版本。所以，如果你希望交叉编译出的程序可以在开发板上顺利执行的话，在编译时必须使用与板载Linux的库相同的版本。我在这里使用arm-linux-binutils-2.10-1.i386.rpm、arm-linux-gcc-2.95.2-2.i386.rpm、arm-linux-glibc-2.1.3-2.i386.rpm来构建交叉编译环境，这些资源可以免费得到(ftp://ftp.arm.linux.org.uk/pub/armlinux/toolchain)。在宿主机上执行rpm指令，安装这3个rpm包，这时你会发现在/usr/local/目录下多了一个arm-linux目录，里面包含有arm-linux-gcc等指令程序及2.1.3版本的Glibc库等资源。然后，在PATH中添加/usr/local/arm-linux/bin路径，这样交叉编译环境就构建完成了。

3.2 交叉编译MiniGUI

下面将要开始关键的一步，就是进行MiniGUI的交叉编译。首先，从网上(http://www.minigui.com/download/cindex.shtml)得到MiniGUI的源程序包，包括三个部分：libminigui-1.3.0.tar.gz——MiniGUI函数库源代码；minigui-res- 1.3.0.tar.gz——MiniGUI所使用的资源，包括基本字体、图标、位图等；mde-1.3.0. tar.gz——MiniGUI的综合演示程序。(在这里使用的是MiniGUI的1.3.0版本)

首先进行MiniGUI函数库的编译和安装。解开libminigui-1.3.0.tar.gz软件包，进入该目录，运行./configure脚本："CC=arm-linux-gcc ./configure --target=arm-linux --prefix=/usr/local/arm-linux/arm-linux --build=arm-linux --host=i686-pc-linux-gnu --disable-lite"。这里需要指明的是，CC用来指定所使用的编译器，这里使用的arm-linux-gcc就是在第一部分中安装RPM包时，被安装到宿主机上的交叉编译器；--target选项用于指定目标平台，这里使用arm-linux；--prefix选项用于指定MiniGUI函数库的安装路径，默认的安装路径是/usr/local，而在这里应该是所使用的交叉编译环境中系统头文件目录include和库目录lib所在的目录，本例中就是/usr/local/arm-linux/arm-linux(这样一来，MiniGUI的函数库和头文件也将被分别安装到交叉编译环境中的lib和include目录中。在后续进行MiniGUI应用程序的交叉编译时，就可以正确地找到MiniGUI的头文件和函数库了)；--build选项与--arget一样为arm-linux；--host选项用来指明宿主机的类型，这里使用i686-pc-linux-gnu即可；最后一个选项--disable-lite用来指定生成基于线程的MiniGUI-Threads版本而不生成基于进程的MiniGUI-Lite版本。目前，MiniGUI函数库有两种版本：Threads版本和Lite版本，1.3.0版默认生成Threads版本。为了简便起见，本例中编译生成Threads版本，两种版本之间的具体差异请参看MiniGUI用户手册。

如果运行./configure脚本成功，就会生成定制的Makefile文件，然后可以继续执行make和make install命令编译并安装libminigui。在执行make install操作时，用户必须具用root权限。安装成功后，MiniGUI的函数库和头文件以及配置文件等资源将被安装到/usr/local/arm-linux/arm-linux目录中，具体情况为：函数库被装在lib/子目录中；头文件被装在include/子目录中；手册被装在man/子目录中；配置文件被装在etc/子目录中。

下面进行MiniGUI资源的编译安装。解压minigui-res-1.3.0.tar.gz，进入相应目录。这里需要说明的是，在执行make install操作之前，需要对目录中的configure.linux文件做一些修改。打开configure.linux文件，你会发现prefix选项部分的默认值为$(TOPDIR)/usr/local，需要将这里修改为prefix=$(TOPDIR)/usr/local/arm-linux/arm-linux，这样执行make install操作之后，安装脚本会自动把MiniGUI资源文件安装到/usr/local/arm-linux/arm-linux/lib/minigui/res/目录下。

最后，编译并安装MiniGUI的演示程序mde。解压mde-1.3.0.tar.gz，进入相应目录。首先修改目录下的configure.in文件，将其中AC_CHECK_HEADERS(minigui/commmon.h, have_libminigui=yes, foo=bar)改为AC_CHECK_HEADERS($prefix/include/minigui/common.h, have_libminigui=yes, foo=bar)，以防止编译时系统无法找

到头文件，从而把have_libminigui赋值为no。编译时，屏幕上会出现“MiniGUI is not properly installed on the system. You need MiniGUI-Lite Ver 1.2.6 or later for building this package. Please configure and install MiniGUI-Lite Ver 1.2.6 first.”的警告。然后执行autogen.sh脚本，重新生成configure脚本，执行“CC=arm-linux-gcc ./configure --target=arm-linux--prefix=/usr/local/arm-linux/arm-linux--build=arm-linux --host=i686-pc-linux-gnu --disable-lite”各选项的意义与前面部分相同。执行成功后，在该目录下会生成一个新的Makefile文件。修改Makefile文件中的COMMON_SUBDIRS部分，将其中的notebook、tools、controlpanel例子暂时删除，否则编译这几个程序时会提示找不到popt.h和libpopt.so。而这些popt头文件和库文件需要自己从网上下载到相应的目录当中。执行make操作，完成演示程序的编译。

这时可以发现，在每一个demo目录下都会生成一个可执行程序。可不要急着在宿主机上运行这些可执行程序，因为这些交叉编译出的程序是将在你的开发板上执行的。下面可以将MiniGUI的库函数、配置文件等资源连同demo程序一起拷贝到AT91RM9200开发板上试验一下了。

3.3 拷贝MiniGUI资源到开发板

进入/usr/local/arm-linux/arm-linux目录,在etc,lib子目录下有我们需要拷贝到目标机上的资源。但是你会发现，子目录lib中的MiniGUI库文件很大，很难全部拷贝到AT91RM9200开发板上。解决的办法是，首先对这些库文件执行arm-linux-strip操作,arm-linux-strip指令会除去文件中的调试信息，使文件体积大大缩小。另外需要注意的是，有些库函数是链接文件，如果单纯的拷贝，会将原先的链接信息丢失，造成不必要的麻烦。使用tar命令将所需拷贝的资源打包，其中包括etc子目录下的配置文件MiniGUI.cfg；lib子目录下的libmgext*、libminigui*、libvcongui*和minigui子目录；mde-1.3.0/housekeeper/下的可执行程序housekeeper(这里用housekeeper作为演示程序)。可以将这些资源做进ramdisk文件系统中，也可以在AT91RM9200开发板运行时，通过ftp从宿主机进行下载。解压后将MiniGUI的配置文件MiniGUI.cfg放入/usr/local/etc目录中,MiniGUI的库文件放入/usr/local/lib目录中，housekeeper程序可以随便放置。在执行程序之前，还有一件重要的事情要做，就是在开发板上的Linux中配置好MiniGUI的运行环境。

3.4 板载Linux的环境配置

AT91RM9200开发板上的Linux中，自带有VESA FrameBuffer设备驱动程序，并且初始状态已经激活。这样，MiniGUI就可以使用FrameBuffer作为图形引擎进行图像显示。否则，需要首先安装一种图形引擎，例如QVFB、SVGALib、LibGGI等，或自己编写一个图形引擎供MiniGUI使用。

FrameBuffer是Linux内核中的一种驱动程序接口，这种接口将显示设备抽象为帧缓冲区，用户可以将它看成是显示内存的一个映像。将其映射到进程地址空间之后，就可以直接进行读写操作，而写操作可以立即反映在屏幕上。首先，建立FrameBuffer驱动程序的设备文件：进入/dev目录，执行操作mknod fb0 c 29 0。这样就建立了设备文件fb0，供MiniGUI使用。同样在/dev目录下执行指令mknod tty0 c 5 0和mknod mouse c 10 1,建立终端设备文件和鼠标驱动程序的设备文件。修改/usr/local/etc目录下运行时的配置文件MiniGUI.cfg，将其中fbcon的defaultmode设置为合适的显示模式。此例中将它设为了640*480-16bpp。如果开发板没有连接鼠标、触摸屏等输入设备，可以先将MiniGUI.cfg中的输入引擎ial_engine设置为dummy。Dummy IAL引擎是个什么工作都不做的引擎，使用这个引擎，可以首先将MiniGUI在目标板上运行起来，然后再进一步参照其它引擎编写适合自己的引擎。

最后，将键盘和鼠标连接至AT91RM9200开发板的串口上，执行housekeeper程序。顺利的话，屏幕上将出现搬运工的游戏界面，并且可以使用鼠标和键盘来进行游戏控制。到此，MiniGUI已经成功移植到了AT91RM9200开发板上。此后，可以继续增添MiniGUI库函数及各种资源，并且编写自己的应用程序，使开发板上的界面更加美观和完善。

结 语

本文中所介绍的MiniGUI移植过程已通过实践检验。对于其它型号的开发板而言，移植过程大体相似。可以肯定，带有简洁、美观图形用户界面的嵌入式系统将会在更多的领域具有更好的实用价值。

参考文献

1 MiniGUI用户手册v1.3-1. 北京飞漫软件技术有限公司. 2003.10

2 AT91RM9200 Development Kit User Guide. Atmel Corporation.2003.7

马忠梅：副教授、硕士生导师。主要从事单片机与嵌入式系统的研究。方宁：硕士研究生，研究方向为嵌入式系统。

（收稿日期：2004-04-14）