

# MiniGUI 在 Sitsang 评估板上的移植

(陈云川 电子科技大学 Intel PCA 实验室 2007 年 4 月 20 日)

## 1 MiniGUI 体系结构

MiniGUI 的体系结构如图 1 所示。

MiniGUI应用程序接口 (API)	
MiniGUI核心 (core)	
GAL	IAL

图 1 MiniGUI 体系结构

## 2 准备工作

在开始具体的移植工作之前，先做如下假设：

- 所有我们需要的文件都放在宿主机上的/root/minigui/目录下；
- 已经将宿主机的/root/目录导出为 NFS 文件系统；
- 已在Sitsang板上建立ARM-Linux操作系统，请参见<sup>[1]</sup>；
- 你已经知道如何通过 minicom 连接到 Sitsang 评估板上；
- 已删掉目标板上的/rd/test.avi，为 MiniGUI 留出足够空间。

在/root/minigui/目录下，需要的三个软件包分别是：

```
[root@cyc minigui]# ls  
libminigui-1.3.3.tar.gz  minigui-res-1.3.3.tar.gz  tslib.tar.bz2
```

注意：加粗字体表示输入，下同。

## 3 为 MiniGUI 提供触摸屏输入引擎 (IAL)

GAL 和 IAL 是 MiniGUI 的两个基础设施，MiniGUI 的高度可移植性在很大程度上也是由于这两个接口提供了独立于硬件的抽象能力。但是 GAL 和 IAL 并不是生来就存在的，它们也需要通过底层代码来实现，通常把实现这两个接口的底层代码称之为“图形引擎”和“输入引擎”。在 Sitsang 评估板上移植 MiniGUI 需要为之提供针对触摸屏的输入引擎。这里我

们采用 tslib 这个函数库来为 MiniGUI 提供输入引擎。因此我们首先就需要交叉编译 tslib。具体步骤如下：

### 1、解压 tslib.tar.bz2

```
[root@cyc minigui]# tar jxvf tslib.tar.bz2
```

解压之后将在/root/minigui/目录下生成一个子目录 tslib/。

### 2、configure

```
[root@cyc minigui]# cd tslib
[root@cyc tslib]# ./autogen.sh
[root@cyc tslib]# ./configure --host=arm-linux --prefix=/root/minigui/tslib/arm-linux-build
```

执行 autogen.sh 脚本的目的是生成 configure 脚本。然后执行 configure 脚本生成 Makefile 文件。Configure 脚本的 **--host=arm-linux** 选项指定针对 ARM-Linux 进行交叉编译，选项 **--prefix=/root/minigui/tslib/arm-linux-build** 指定将最后生成的各种文件都安装到 **/root/minigui/tslib/arm-linux-build** 目录下。注意：后面要用到这个目录。

### 3、make and make install

```
[root@cyc tslib]# make
[root@cyc tslib]# make install
```

自此，tslib 的交叉编译就完成了，编译生成的文件被安装到了 **/root/minigui/tslib/arm-linux-build/**目录下，如下所示：

```
[root@cyc tslib]# cd arm-linux-build/
[root@cyc arm-linux-build]# ls
bin  etc  include  lib  share
```

其中，include/子目录下是 tslib 的头文件，lib/子目录下是编译生成的动态共享库文件。

## 4 交叉编译 MiniGUI

接下来就可以交叉编译 MiniGUI 了。首先仍然要切换当前目录到/root/minigui/下面。

```
[root@cyc minigui]# pwd
/root/minigui
[root@cyc minigui]# ls
libminigui-1.3.3.tar.gz  minigui-res-1.3.3.tar.gz  tslib  tslib.tar.bz2
```

### 1、解压

```
[root@cyc minigui]# tar zxvf libminigui-1.3.3.tar.gz
```

解压之后将生成 libminigui-1.3.3/子目录。

## 2、configure

```
[root@cyc libminigui-1.3.3]# ./configure --enable-lite --enable-smdk2410ial --host=arm-linux --target=arm-linux --prefix=/root/minigui/libminigui-1.3.3/arm-linux-build CFLAGS="-I/root/minigui/tslib/arm-linux-build/include/ -L/root/minigui/tslib/arm-linux-build/lib / -lts"
```

说明一下 configure 的参数：

**--enable-lite**                    使用 MiniGUI-Lite  
**--enable-smdk2410ial**    采用 SMDK2410 输入引擎  
**--host=arm-linux --target=arm-linux**    指定交叉编译的目标平台  
**--prefix=.....**            指定交叉编译得到的文件安装的位置  
**CFLAGS=.....**            指定到哪里去找 tslib 的头文件和动态共享库文件

## 3、make and make install

```
[root@cyc libminigui-1.3.3]# make  
[root@cyc libminigui-1.3.3]# make install
```

自此，libminigui 的交叉编译就完成了，编译生成的文件被安装到了 /root/minigui/libminigui-1.3.3/arm-linux-build 目录下，如下所示：

```
[root@cyc libminigui-1.3.3]# cd arm-linux-build/  
[root@cyc arm-linux-build]# ls  
etc   include   lib
```

自此，MiniGUI 的交叉编译工作就完成了。生成的三个子目录中的内容都是我们需要的。

# 5 拷贝核心库到 Sitsang 评估板

下一步就是要将交叉编译生成的MiniGUI文件拷贝到Sitsang评估板上。我们通过NFS文件系统来完成这一工作。如果你对如何配置NFS不清楚的话，请参考<sup>[1]</sup>。这里假设宿主机的IP地址为 202.115.26.224。

首先通过 minicom 连接到 Sitsang 评估板上。然后，将宿主机的/root/目录挂载到 Sitsang 评估板的/mnt 目录上（注意，以下操作都是在 Sitsang 评估板上进行的）：

```
[root@Sitsang2 root]$mount -t nfs -o nolock 202.115.26.224:/root /mnt
```

将宿主机的/root/目录挂载到 Sitsang 评估板上之后,就可以将生成 MiniGUI 文件拷贝到目标板上了。要拷贝的内容主要有两项,一是 MiniGUI 配置文件 MiniGUI.cfg,另外则是 MiniGUI 运行核心共享库和资源文件。MiniGUI.cfg 拷贝到目标板的/etc/目录下,运行时共享库和资源文件则拷贝到目标板的/usr/local/lib/目录下。具体步骤如下所示:

```
[root@Sitsang2 root]$cd /mnt/
[root@Sitsang2 mnt]$cd minigui/
[root@Sitsang2 minigui]$cd libminigui-1.3.3
[root@Sitsang2 libminigui-1.3.3]$cd arm-linux-build/
[root@Sitsang2 arm-linux-build]$cd etc/
[root@Sitsang2 etc]$cp MiniGUI.cfg /etc/
[root@Sitsang2 etc]$cd ..
[root@Sitsang2 arm-linux-build]$cd lib/
[root@Sitsang2 lib]$cp -r ./* /usr/local/lib/
```

## 6 拷贝资源文件到 Sitsang 评估板

除了必须的核心库文件之外,还必须将 MiniGUI 需要的一些资源文件拷贝到目标板上。首先在宿主机上解压:

```
[root@cyc root]# cd minigui/
[root@cyc minigui]# ls
libminigui-1.3.3 libminigui-1.3.3.tar.gz minigui-res-1.3.3.tar.gz tslib tslib.tar.bz2
[root@cyc minigui]# tar zxvf minigui-res-1.3.3.tar.gz
```

解压之后将生成一个 minigui-res-1.3.3/子目录。在这个目录下,存在下面这样几个目录:

```
[root@cyc minigui]# cd minigui-res-1.3.3
[root@cyc minigui-res-1.3.3]# ls -F
bmp/ config.linux cursor/ font/ icon/ imetab/ Makefile README
```

bmp/目录下是 MiniGUI 需要的位图文件;cursor/目录下是光标文件;font/目录下是字体文件;icon/目录下是图标文件;imetab/目录下是输入法码表文件。这几个目录都要拷贝到目标板上才行。切换到 Sitsang 评估板,完成如下拷贝操作:

```
[root@Sitsang2 minigui-res-1.3.3]$cp -r bmp/ cursor/ font/ \
> icon/ imetab/ /usr/local/lib/minigui/res/
```

自此,在 Sitsang 评估板上运行 MiniGUI 所需的一切资源都已齐全。正是:万事具备,只欠东风。

## 7 修改 MiniGUI.cfg

接下来还需要做的事情是修改拷贝到 Sitsang 评估板的/etc/目录下的 MiniGUI.cfg 文件，这个文件控制着 MiniGUI 启动的方方面面。

```
[root@Sitsang2 etc]$vi MiniGUI.cfg
```

将 MiniGUI.cfg 中关键的几项修改成下面这样：

```
[system]
# GAL engine
gal_engine=fbcon

# IAL engine
ial_engine=SMDK2410

mdev=/dev/input/event1
mtype=IMPS2

[fbcon]
defaultmode=640x480-16bpp
```

## 8 编译 MiniGUI 服务器程序 mginit

在宿主机上，进入/root/minigui/mginit/目录，编译服务器程序 mginit:

```
[root@cyc minigui]# pwd
/root/minigui
[root@cyc minigui]# ls
libminigui-1.3.3      mginit              minigui-res-1.3.3.tar.gz  tslib.tar.bz2
libminigui-1.3.3.tar.gz  minigui-res-1.3.3  tslib
[root@cyc minigui]# cd mginit/
[root@cyc mginit]# arm-linux-gcc -o mginit mginit.c \
> -I/root/minigui/libminigui-1.3.3/arm-linux-build/include \
> -I/root/minigui/tslib/arm-linux-build/include \
> -L/root/minigui/libminigui-1.3.3/arm-linux-build/lib \
> -L/root/minigui/tslib/arm-linux-build/lib \
> -lmgext -lminigui -lts
```

上述命令将生成 mginit 程序，这个程序是 MiniGUI 的服务器程序，当我们要运行 MiniGUI-Lite 程序时，必须先启动这个服务器程序 mginit。因此，下面切换到 Sitsang 评估板上，以后台运行方式启动 mginit（注意：下面的命令是在 Sitsang 评估板上运行的）：

```
[root@Sitsang2 root]$cd /mnt/minigui/mginit/
[root@Sitsang2 mginit]$/etc/rc.d/init.d/qtopia stop
Shutting down Qtopia services:          killall: qpe: no process killed
ok
[root@Sitsang2 mginit]$/mginit &
[1] 171
```

这里解释一下为什么要`/etc/rc.d/init.d/qtopia stop`。默认情况下，Sitsang 评估板在启动 ARM-Linux 之后就会启动 qtopia——基于 QT Embedded 的窗口环境。显然，这会和我们的 MiniGUI 冲突，于是应该先将其停止。然后再启动我们的 mginit。并且 mginit 一定要以后台方式启动，因为这样才有机会执行后面的操作。如果 MiniGUI 正确移植到了 Sitsang 评估板上的话，将看到在显示屏的正中显示出了 Linux 的 logo——可爱的企鹅。这就是 MiniGUI 的桌面了。

## 9 helloworld

下面以一个简单的 helloworld 程序结束本文。在宿主机上：

```
[root@cyc minigui]# pwd
/root/minigui
[root@cyc minigui]# ls
helloworld          libminigui-1.3.3.tar.gz  minigui-res-1.3.3      tslib
libminigui-1.3.3    mginit                  minigui-res-1.3.3.tar.gz  tslib.tar.bz2
[root@cyc minigui]# cd helloworld/
[root@cyc helloworld]# arm-linux-gcc -o helloworld helloworld.c \
> -I/root/minigui/libminigui-1.3.3/arm-linux-build/include \
> -L/root/minigui/libminigui-1.3.3/arm-linux-build/lib \
> -L/root/minigui/tslib/arm-linux-build/lib \
> -lmgext -lminigui -lts
```

然后，切换到 Sitsang 评估板上，执行编译生成的 helloworld 程序：

```
[root@Sitsang2 helloworld]$/helloworld
```

如果 MiniGUI 已经移植成功，那么将看到一个简单的窗口，在窗口区域显示了几行文字，同时每隔几秒钟就会更新一次文字信息。在控制台（即 minicom 中）按下 Ctrl-C 可以终止 helloworld，同时 mginit 不受影响。

## 参考文献

- [1] 电子科技大学——Intel PCA 联合实验室。PCA 技术培训教程。2003 年 11 月。
- [2] 北京飞漫软件技术有限公司。MiniGUI 编程指南 for MiniGUI Ver 1.3.x。2003 年 10 月

- [3] 北京飞漫软件技术有限公司。MiniGUI 用户手册 for MiniGUI Ver 1.3.x。2003 年 10 月
- [4] 北京飞漫软件技术有限公司。MiniGUI API Reference Documentation for MiniGUI Ver 1.3.x。2003 年 10 月