

数码相框用户手册

文件状态： <input checked="" type="checkbox"/> 草稿 <input type="checkbox"/> 正式发布 <input type="checkbox"/> 正在修改	文件标识：	
	当前版本：	1.0
	作者：	黄奕军 陈育新
	完成日期：	2011-9-23

版本历史

版本/状态	作者	参与者	起止日期	备注
0.2	黄奕军	陈育新	2011-8-22— 2011-9-23	数码相框项目包位置为： http://58.249.121.165/share/%e9%a1%b9%e7%9b%ae%e5%8c%85/%e6%95%b0%e7%a0%81%e7%9b%b8%e6%a1%86%ef%bc%88QT%e7%89%88%e6%9c%ac%ef%bc%89/

目 录

目 录.....	3
项目简介.....	4
1.移植过程（X86 平台）.....	5
1.1 移植用到的系统和开发板：.....	5
1.2 编译支持触摸屏 QT-EMBEDDED.....	5
1.2.1 解压工具链并设置环境变量：.....	5
1.2.2 编译触摸屏工具 <i>tslib</i>	5
1.2.3 编译 <i>qt-embedded</i>	6
1.3 交叉编译数码相框程序.....	9
1.4 编译内核及制作文件系统.....	10
1.4.1 编译内核.....	10
1.4.2 配置数码相框文件系统.....	11
1.4.3 内核及文件系统的烧写.....	13
2. 使用方法.....	15
2.1 图片播放功能设定.....	15
2.2 电子时钟功能设定.....	17

项目简介

基于 Qt 的数码相框主要完成的功能是：搭建数码相框基本框架，支持数码照片的显示及演示，并具有五种切换特效，能够使用触摸屏控制。

其主要特性有：

- 1) 支持*.bmp、*.jpg、*.png、*.gif 格式的数码照片的显示及演示；
- 2) 支持接入 U 盘读取图片的功能；
- 3) 具有五种图片切换特效，能够流畅的进行播放图片；
- 4) 可以使用触摸屏对整个界面进行操作和控制；
- 5) 对照片可实现幻灯片列表浏览；
- 6) 支持对图片进行放大、缩小及旋转；
- 7) 实现了电子时钟的功能，并能够对日期和时钟进行修改；
- 8) 预留了音乐播放器、视频播放器、电子书几个拓展功能的接口，便于以后进行二次开发；
- 9) 支持图片尺寸为 1M 以下图片文件，为保证图片显示流畅及视觉感受，请用户自行选择与数码相框屏幕相近像素的图片进行播放；

1.移植过程（X86 平台）

1.1 移植用到的系统和开发板：

基于 X86 平台的 ubuntu10.04 系统；
龙芯 1b 开发板；
3.5 寸液晶触摸屏；

1.2 编译支持触摸屏 QT-Embedded

1.2.1 解压工具链并设置环境变量：

从数码相框项目包里解压工具链

```
#tar zxvf gcc-3.4.6.tar.gz -C /opt/
```

此时可以在文件夹/opt/下看到/opt/gcc-3.4.6。

将交叉编译工具链添加到环境变量中

```
#vim ~/.bashrc
```

在里面加上一句：

```
export PATH=/opt/gcc-3.4.6/bin:$PATH
```

也可以在/etc/profile 里添加。

均是设置环境变量，其区别在于在第一个文件只针对当前的用户有效，第二个文件是针对所有的用户有效。要注意通过修改文件改变环境变量则定义完环境变量后一定要关闭终端，然后再打开才有效。

1.2.2 编译触摸屏工具 tslib

Tslib 是一个开源的程序，能够为触摸屏驱动获得的采样提供诸如滤波、去抖、校准等功能，通常作为触摸屏驱动的适配层，为上层的应用提供了一个统一的接口。

编译 tslib-1.4，在数码相框项目包里解压 tslib-1.4.gz

```
# tar zxvf tslib-1.4.gz
```

```
# cd tslib
```

```
# export PATH=/opt/gcc-3.4.6/bin:$PATH
```

由于使用开发板的 3.0 内核与 gcc-3.4.6 之间的版本号之间不一致，故需要修改 gcc 的版本号或者内核的版本号，不然会出现板上提示：

```
xres = 320, yres = 240
```

selected device is not a touchscreen I understand.

修改 gcc 版本号的如下：

```
#vi /opt/gcc-3.4.6/mipsel-linux/include/linux/input.h
```

将 include/linux/input.h 文件中的：

```
#define EV_VERSION          0x010000 改成
```

```
#define EV_VERSION          0x010001
```

然后执行以下 Tslib 的配置与编译

```
# ./autogen.sh
```

这里可能会遇到以下无法 autoconf 的问题，这是因为系统没有安装 autoconf，只需要执行 apt-get 指令即可解决问题：

```
#apt-get install autoconf automake libtool
```

```
# ./configure --host=mipsel-linux --prefix=/TSLIB
```

（选择 make 之后库放的位置，可以自己选择路径，此处是将其放在了根目录的下面/TSLIB）

```
# make
```

编译过程可能会出现一个错误：

```
ts_test.o: In function `main':
```

```
ts_test.c:(.text+0x3b4): undefined reference to `rpl_malloc'
```

```
futils.o: In function `open_framebuffer':
```

```
futils.c:(.text+0x2f8): undefined reference to `rpl_malloc'
```

```
collect2: ld returned 1 exit status
```

```
make[2]: *** [ts_test] 错误 1
```

只需要将./config.h 中的#define malloc rpl_malloc 注释掉就可以编译通过了。

```
# make install
```

然后可见在根文件目录下生成了 TSLIB 文件夹，包含 tslib 的可执行文件、库、头文件及配置文件。

编译完 Tslib 之后记得将 include/linux/input.h 文件中的 gcc 版本号修改回来。

1.2.3 编译 qt-embedded

在数码相框的项目包中解压 QT 的嵌入式 4.5.3 版本。

解压 qt-embedded-linux-opensource-src-4.5.3.tar.gz 到/home/1b 下，将文件夹重命名，执行

```
#mv qt-embedded-linux-opensource-src-4.5.3 qt-embedded-4.5.3
```

在 vim ~/.bashrc 中添加所需要的变量

```
export PATH=/opt/gcc-3.4.6/bin:$PATH
```

```
export QTEDIR=/home/1b/qt-embedded-4.5.3
```

```
export LD_LIBRARY_PATH=$QTEDIR/lib:$LD_LIBRARY_PATH
export PATH=$QTEDIR/bin:$PATH
export QTDIR=$QTEDIR
```

接着修改 QTDIR 文件夹下边的 mkspecs/qws/linux-mips-g++/qmake.conf 文件，注意将文件中的 mips-linux 修改成 mipsel-linux，如下图所示：

```
# modifications to g++.conf
QMAKE_CC      = mipsel-linux-gcc
QMAKE_CXX     = mipsel-linux-g++
QMAKE_CFLAGS += -mips32
QMAKE_CXXFLAGS += -mips32
QMAKE_LINK    = mipsel-linux-g++
QMAKE_LINK_SHLIB = mipsel-linux-g++

# modifications to linux.conf
QMAKE_AR      = mipsel-linux-ar cqs
QMAKE_OBJCOPY = mipsel-linux-objcopy
QMAKE_STRIP   = mipsel-linux-strip
```

然后关闭终端，使修改的环境变量生效，接着开始编译 QT-Embedded。

首先进入 QTDIR，对所要编译的 QT-Embedded 进行配置，目的在于裁剪一些我们不需要的库。

```
#cd $QTDIR
#./configure -release -shared -no-largefile -qt-sql-sqlite
-no-qt3support -no-xmlpatterns -no-mmx -no-3dnow -no-sse -no-sse2
-no-svg -no-webkit -qt-zlib -qt-gif -qt-libtiff -qt-libpng -qt-libmng
-qt-libjpeg -make libs -nomake tools -nomake examples -nomake docs
-nomake demo -no-nis -no-cups -no-iconv -no-dbus -no-openssl
-xplatform qws/linux-mips-g++ -embedded mips -qt-freetype -depths
8,16,24,32 -qt-gfx-linuxfb -no-gfx-transformed
-no-gfx-multiscreen -no-gfx-vnc -no-gfx-qvfb -qt-kbd-usb
-no-glib -qt-mouse-tslib -I/TSLIB/include -L/TSLIB/lib(此处的位置应
与之前编译 Tslib 所设的目标文件夹一致。)
```

解释一下这段配置，其中 -xplatform qws/linux-mips-g++ -embedded mips 的作用为设置目标板平台及交叉编译，而最后的 -I/TSLIB/include -L/TSLIB/lib，则是讲编译好的 Tslib 编译进 QT-Embedded。

此时系统会弹出一下信息

```
Which edition of Qt do you want to use ?
```

```
Type 'c' if you want to use the Commercial Edition.
```

```
Type 'o' if you want to use the Open Source Edition.
```

只要敲入 o 便可，跟着又会弹出

```
This is the Qt for Embedded Linux Open Source Edition.
```

```
You are licensed to use this software under the terms of
```

the Lesser GNU General Public License (LGPL) versions 2.1.
You are also licensed to use this software under the terms of
the GNU General Public License (GPL) versions 3.

Type '3' to view the GNU General Public License version 3.
Type 'L' to view the Lesser GNU General Public License version 2.1.
Type 'yes' to accept this license offer.
Type 'no' to decline this license offer.

Do you accept the terms of either license?

只要敲入 yes 便可，然后就会开始进行配置了。

最后就可以进行 make 了。过程中有可能会报一个错误，错误信息如下所示：

```
io/qfsfileengine_iterator_unix.cpp: In member function `virtual
bool QFSFileEngineIterator::hasNext() const':
io/qfsfileengine_iterator_unix.cpp:124: 错误：`FILENAME_MAX'
was not declared in this scope
io/qfsfileengine_iterator_unix.cpp:124: 警告： unused variable
'FILENAME_MAX'
make[1]: ***
[.obj/release-shared-emb-mips/qfsfileengine_iterator_unix.o] 错误 1
make[1]:正在离开目录 `/home/1b/qt-embedded-4.5.3/src/corelib'
make: *** [sub-corelib-make_default-ordered] 错误 2
```

解决方法：

```
#vim $QTDIR/src/corelib/io/qfsfileengine_iterator_unix.cpp
```

在其中添加一个语句便可，#define FILENAME_MAX 1024

接着便是长达两个多钟的编译了。

编译完后再执行

```
#make install
```

这样就能在 /usr/local/Trolltech/QtEmbedded-4.5.3-mips/ 路径下看到 QT 安装好的 QT-Embedded 了。

1.3 交叉编译数码相框程序

解压数码相框项目包中的数码相框源码包：

```
#tar zxvf digitalPhotoFrame.tar.gz
```

由于交叉编译数码相框程序需要用到前面编译好的 QT-Embedded 的开发工具，为了方便使用，可以软链接 qmake 工具到/usr/bin 目录下，并设置好交叉编译工具链的环境变量：

```
#ln -s /usr/local/Trolltech/QtEmbedded-4.5.3-mips/bin/qmake  
/usr/bin/qmake-embedded
```

```
#export PATH=/opt/gcc-3.4.6/bin:$PATH
```

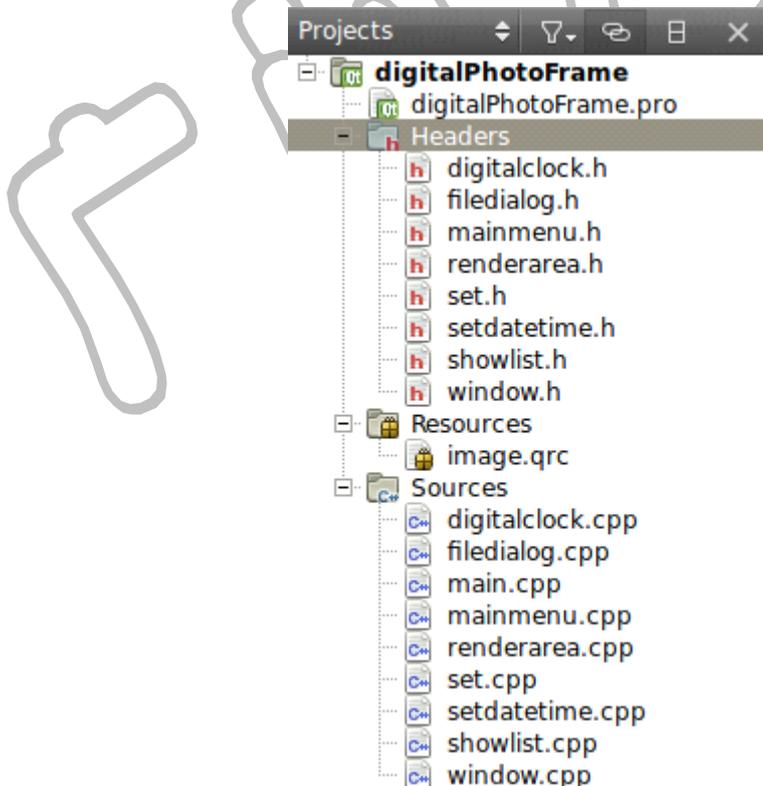
然后进入数码相框源码文件夹内进行交叉编译。

```
#cd digitalPhotoFrame
```

```
#qmake-embedded -project
```

```
#qmake-embedded
```

这样可以利用 qmake 工具生成一个名为 digitalPhotoFrame.pro 的工程文件。注意检查是否数码相框所有头文件、实现文件、资源文件均已添加到工程中。利用 QT-creator 工具打开工程文件，工程结构如下：



并利用 qmake 生成一个基于 digitalPhotoFrame.pro 工程的 Makefile 文件。检查 Makefile 文件内容以下内容是否对应交叉编译工具链：

```
CC = mipsel-linux-gcc
```

```
CXX = mipsel-linux-g++
```

这就表示这个 Makefile 文件是指定了我们需要的交叉编译工具的。

最后执行 make 就可以生成一个名为 digitalPhotoFrame 的二进制可执行文件。

```
#mipsel-linux-readelf -d digitalPhotoFrame
```

该指令可以了解这个二进制可执行文件的相关信息。

PS. 若要编译数码相框的 QT-X11 版本，则需要将数码相框源码的 main.cpp 实现文件中的 QWSServer::setCursorVisible(false) 注释掉：

```
//QWSServer::setCursorVisible(false);
```

此句代码的目的在于去除 Embedded 环境下的鼠标指针。

1.4 编译内核及制作文件系统

1.4.1 编译内核

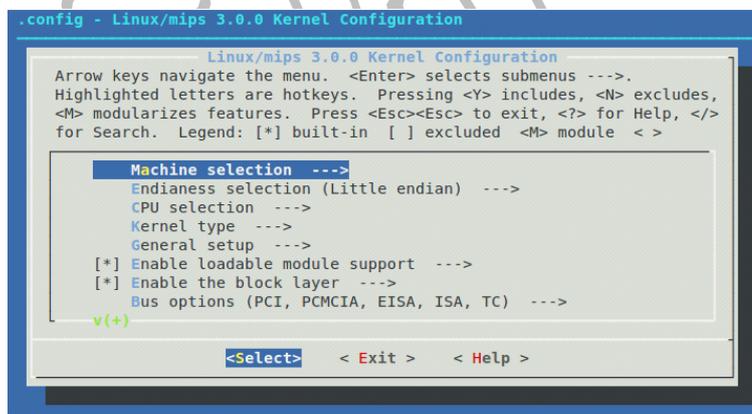
解压数码相框项目包内的内核压缩包：

```
#tar zxvf 1b-linux-3.0-digitalPhotoFrame.tar.gz
```

该内核的版本号为 3.0，在编译内核之前先对内核进行配置

```
#make menuconfig
```

可以见到一下画面：



通过此图形界面对内核进行相应的配置，主要配置目标板的驱动情况。本内核源码包已经针对 loongson-1b 开发板进行相应的配置了，可以不需要修改直接编译。

```
#make
```

如果遇到 error 的话一般情况下都是缺少一些系统库的原因，请通过搜索引擎查找相关资料，然后安装相应的库后就能编译通过了。

编译后在源码包目录下生成 vmlinux 文件。

1.4.2 配置数码相框文件系统

本项目包内提供了利用 busybox 工具制作好的纯净版文件系统，本小节主要介绍如何移植数码相框程序及相关配置。

首先解压数码相框项目包中的 ram12.tgz，解压后得到纯净的文件系统，为了方便使用，将文件系统文件夹修改名字，然后移动到指定开发路径中，操作如下：

```
#tar zxvf ram12.tgz
#mv ram12 ram-1Bfs
#mv ram-1Bfs /root/
```

此时板上文件系统文件夹在 PC 端上的路径为 /root/ram-1Bfs。

现在需要将数码相框程序移植进去，包括经过 QT-Embedded 编译生成的数码相框二进制可执行文件，以及该二进制可执行文件需要的 QT-Embedded 的动态库文件和交叉编译工具链 mipsel-linux-gcc 的动态库文件，具体的方法如下：

进入数码相框源码包，经过 QT-Embedded 交叉编译后可以生成一个二进制可执行文件，可以通过 mipsel-linux-readelf -d filename 指令来读取一个二进制文件所需要的动态库。上面显示了这个 qt 二进制文件要执行时所需要的库，还有这些动态库的默认相对路径：

```
usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/
```

要注意下边所有的库的移植之前都需要先进入文件系统文件夹中。

```
#cd /root/ram-1Bfs
```

首先进入文件系统中将数码相框交叉编译后的二进制文件拷贝到文件系统中，然后创建上边的 QT 库相对路径：

```
#mkdir -p usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/
```

将数码相框所需要的库拷到文件系统中的 QT 库路径。数码相框程序需要以下几个 QT 库：libQtGui.so.4、libQtNetwork.so.4、libQtCore.so.4，其源路径为：

```
/usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/
```

由于 QT 编译了支持触摸屏的 Tslib 进去，所以编译后的二进制文件还需要一个 libts-0.0.so.0 动态库，其位置在之前编译好的 /TSLIB/lib/ 目录下。

其他动态库为交叉编译工具链 gcc 中的库，其源路径为：

```
/opt/gcc-3.4.6/mipsel-linux/lib/
```

具体执行步骤如下所示：

```
#cd /root/ram-1Bfs
```

```
#cp /usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/+(程序需要的 QT 库) /root/ram-1Bfs/usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/
```

```
#cp /opt/gcc-3.4.6/mipsel-linux/lib/+(程序需要的 gcc 库) /root/ram-1Bfs/usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/
```

由于数码相框中需要解析图片，还需要用到 QT 中图片解析的库，先进入文件系统，然后执行以下步骤：

```
#cd /root/ram-1Bfs
#mkdir -p usr/local/Trolltech/QtEmbedded-4.5.3-mips/plugins/
#cp -rf /usr/local/Trolltech/QtEmbedded-4.5.3-mips/plugins/
imageformats/ usr/local/Trolltech/QtEmbedded-4.5.3-mips/plugins/
```

数码相框中还需要支持中英文显示的字体库，先进入文件系统，然后执行以下操作：

```
#cd /root/ram-1Bfs
#mkdir -p usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/fonts
#cp -rf /usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/fonts/
helvetica_240_50.qpf
usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/fonts/hel_240_50.qpf
#cp -rf /usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/fonts/
unifont_160_50.qpf
usr/local/Trolltech/QtEmbedded-4.5.3-mips/lib/fonts/
```

由于数码相框支持触摸屏，故还需要将 Tslib 移植到文件系统中。

```
#cp -rf /TSLIB /root/ram-1Bfs
```

至此支持数码相框需要的所有动态库已经都移植到文件系统中了。

进入文件系统文件夹，接下来进行相关文件的配置：

```
#cd /root/ram-1Bfs
```

(1) 环境变量的配置：

```
#vi /root/ram-1Bfs/etc/profile
echo "-----"
export PATH=$PATH:/usr/X11R6/bin:/usr/local/bin
echo "sh"
ifconfig lo up
ifconfig eth0 up
ifconfig eth0 192.168.1.88
export LC_ALL=zh_CN.UTF-8
export LANG_ALL=zh_CN.UTF-8
export LANG=zh_CN.UTF-8
export TSLIB_ROOT=/TSLIB
export LD_LIBRARY_PATH=$TSLIB_ROOT/lib:$LD_LIBRARY_PATH
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_TSDEVICE=/dev/input/event0
export TSLIB_CALIBFILE=/etc/pointercal
export TSLIB_CONFFILE=/TSLIB/etc/ts.conf
export TSLIB_PLUGINDIR=/TSLIB/lib/ts
```

```
export TSLIB_TSEVENTTYPE=INPUT
export QWS_MOUSE_PROTO=Tslib:/dev/input/event0
export QTDIR=/usr/local/Trolltech/QtEmbedded-4.5.3-mips
export
LD_LIBRARY_PATH=$QTDIR/lib:$TSLIB_ROOT/lib:/usr/local/lib:$LD_LIBRARY_PATH
export QWS_DISPLAY="LinuxFb:mmWidth100:mmHeight130:0"
export QT_QWS_FONTDIR=$QTDIR/lib/fonts
export QT_PLUGIN_PATH=$QTDIR/plugins/
cd /dev
mkdir input
ln -s /dev/event0 /dev/input/event0
cd /
#echo "Profile"
```

(2) Tslib 的配置:

```
#vi /root/ram-1Bfs/TSLIB/etc/ts.conf
# Uncomment if you wish to use the linux input layer event
interface
module_raw input
module pthreads pmin=1
module variance delta=30
module dejitter delta=100
module linear
```

(3) 自动挂载 u 盘的配置:

见数码相框项目包中的文档《开发板上文件系统实现 u 盘自动挂载》

至此整个数码相框的文件系统已经配置好了,我们在数码相框项目包里边提供了一个配置好的数码相框文件系统: ram-ts.tar.gz 及其两种文件系统 (yaffs2 和 cramfs) 格式的镜像文件。

1.4.3 内核及文件系统的烧写

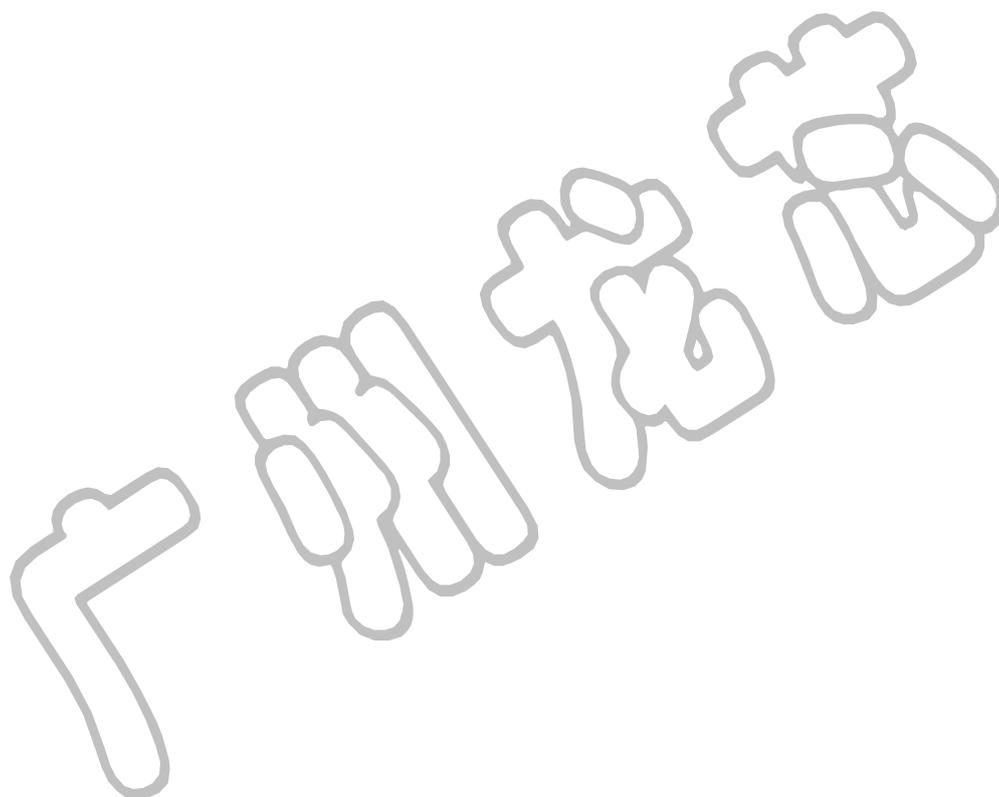
PC 端安装配置好 tftp,然后将内核及文件系统镜像放到 tftp 共享文件夹中,然后进入超级终端中,待板子进入 pmon 里面将内核及文件系统镜像烧写到板子上,详细方法请参考《1B 用户手册》中相关内容。

待烧写好数码相框需要的内核及文件系统后,重新启动,进入到板子上的文件系统,需要先对板子进行触摸屏的校准:

```
# ./TSLIB/bin/ts_calibrate
```

校准后系统会自动生成一个校准文件,位置为/etc/pointercal。

然后执行以下指令打开文件系统中的数码相框程序:



2. 使用方法

第一步：当启动了数码相框之后，数码相框会进入主界面，如下图所示：



这个主界面现在有六个图标按键，第一排从左至右分别代表的功能是：“数码相框图片播放”、“电影播放器”和“音乐播放器”。第二排从左至右分别代表的功能是：“电子书阅读”、“电子时钟”和“设置”。第一阶段完成的功能有数码相框的图片播放和电子时钟及设置时钟的功能。

点击第一个按键  进入数码相框图片播放界面。

2.1 图片播放功能设定

对于数码相框的图片播放功能，我们有以下的设定：

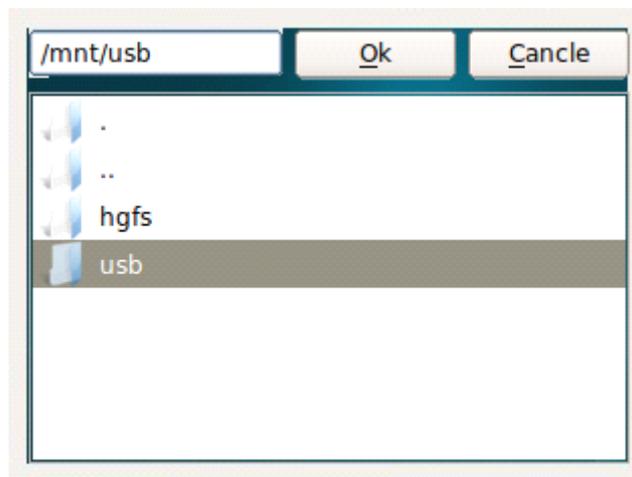
1. 设定了一个板上的默认路径（/mnt/usb）作为默认图片文件夹，进入界面默认加载此路径的第一张照片；
2. 图片下边有两行功能按钮，其功能分别为：效果选择、图片幻灯片列表、选择上一张、自动播放、选择下一张、返回主菜单、选择新图片路径、图片缩小、图片旋转、图片放大；
3. 若此文件夹内包含一层子目录内检测不到图片，则显示默认背景图片，且图片幻灯片列表、选择上一张、自动播放、选择下一张这几个按钮处于不可使用状态；
4. 功能按键在数码相框无任何输入情况下默认 5s 时间自动隐藏，隐藏后当数码相框触碰到屏幕则自动显示出来。



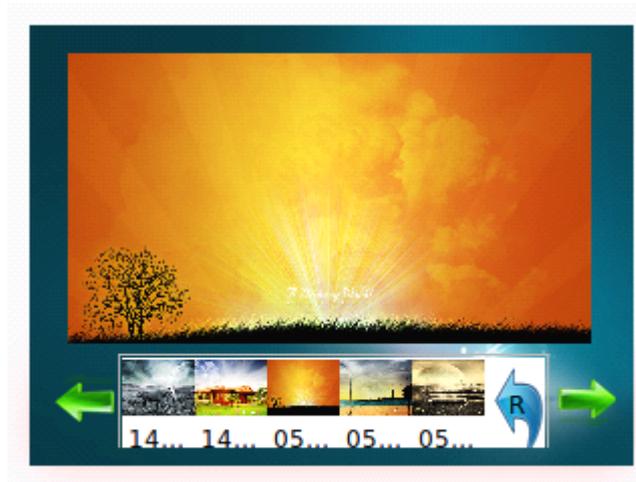
5. 浏览图片的特效有以下几种模式：直接切换、旋转缩放、淡入淡出、竖直滑动。水平滑动、卷书翻页、随机切换。



6. 当数码相框程序启动之后插入 U 盘，则需要点击 **查找** 按钮来选择 /mnt/usb 路径，才能播放 U 盘中的图片文件。其中左上角的文本框会显示当前选中的文件夹，点击 Ok 确定选择，点击 Cancel 取消查找；



7. 当点击功能按键  后，将会切换到图片幻灯片列表的方式来浏览图片，如下图所示：



8. 在图片幻灯列表，可以单击下边列表中的任意图片进行显示，也可以通过左右两个按键进行选择上一张及下一张图片。当点击  按键，则返回数码相框图片播放界面，继续全屏播放。

9. 在数码相框图片播放界面中包含了对相片的处理功能，包括对其进行放大、缩小、旋转，这些功能均不改变图片文件本身，仅仅在图片显示上做的处理。用户可以逐一的去进行尝试。

2.2 电子时钟功能设定

1. 当用户不想播放数码图片时，也可以将数码相框当作电子时钟来用，具

体操作方法是返回到主界面当中去，然后再单击时钟按钮  就可以切换到电子时钟界面中去，具体的显示效果如下所示：



2. 如果想要退出时钟显示的界面，只要点击一下屏幕便可以回到主界面。

假如需要调整时间，则可以单击主界面中的设置  按钮对系统时间进行设置。

