

基于 MPlayer 的多媒体应用开发指南

2011-08-15

(本手册适用于 Mini6410/Tiny6410)



Copyright © 2007-2010 FriendlyARM

All rights reserved.



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

简介

本手册由广州友善之臂计算机科技有限公司(简称“友善之臂”)创建和维护,并作为标准用户手册的一个补充,仅供嵌入式爱好者学习参考使用,友善之臂目前并不对本手册的内容提供任何解释和解答服务,用户可以在论坛中反馈你所遇到的问题和疑问,我们将在以后的更新中修正或者采纳您的建议,本手册主要以首页日期为版本标志。

本手册由友善之臂软件开发工程师编写制作,以 Mini6410 和 Tiny6410 作为开发平台,介绍如何使用 mplayer 来开发多媒体应用,文档将以一个广告机和一个视频播放器的示例来进行讲解,两个示例都是使用 Qt4 来编写用户界面,然后通过调用 mplayer 来进行广告和视频的播放,除了简单的视频播放和控制外,示例程序中还将演示如何使用 overlay 技术(多 framebuffer)来实现 logo 或者字幕在视频上的叠加显示,另外,视频播放器还支持电视输出功能,实用性很强,用户可以参考这两个示例,开发产品所需的广告机程序或者专用的视频播放器等等。

Mini6410 是一款十分精致的低价高品质一体化 ARM11 开发板,由广州友善之臂设计、生产和发行销售。它采用三星 S3C6410 作为主处理器,在设计上承袭了 Mini2440 “精于心,简于形”的风格,而且布局更加合理,接口更加丰富,十分适用于开发 MID、汽车电子、工业控制、导航系统、媒体播放等终端设备;也可适用于 高校教学、嵌入式培训、个人研究学习和 DIY 等。

Tiny6410 是由友善之臂出品的一款以 ARM11 芯片(三星 S3C6410)作为主处理器的嵌入式核心板,它采用高密度 6 层板设计,尺寸为 64 x 50mm,它集成了 128M DDR RAM, 256M/1GB SLC Nand Flash 存储器,采用 5V 供电,在板实现 CPU 必需的各种核心电压转换,还带有专业复位芯片,通过 2.0mm 间距的排针,引出各种常见的接口资源,以供不打算自行设计 CPU 板的开发者进行快捷的二次开发使用。

因为 Mini6410 和 Tiny6410 的硬件接口和资源都是基本相同的,所以本手册完全适用于 Mini6410 和 Tiny6410 开发板。

为了方便用户,本手册的所用到的源代码放在光盘 A 的“开发文档和教程\专题 06 基于 MPlayer 的多媒体应用开发指南\源代码”目录下。

在移植之前需要想先体验一下效果的用户,只要烧写最新版本的 Linux 映像即可(详见:<http://www.arm9home.net/read.php?tid-14308.html>),里面内置的 AdsDemo 程序就是一个简单的广告机示例程序。

我们欢迎各位网友复制传播本手册,但不得擅自摘抄部分或全部内容用作商业用途,违者必究,友善之臂保留本手册的解释和修改权。

友善之臂公司网址: <http://www.arm9.net>

本手册由 ARM9 之家论坛(<http://www.arm9home.net>)发布,转载请注明出处,手册内难免有遗漏和不足之处,欢迎大家提出宝贵意见,请发邮件至: qt_friendlyarm@163.com。



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

更新说明

2011-08-15	1) 增加视频播放器示例; 2) mplayer 增加电视输出功能;
2011-08-07	本手册第一次发布



追求卓越 创造精品

TO BE BEST

TO DO GREAT

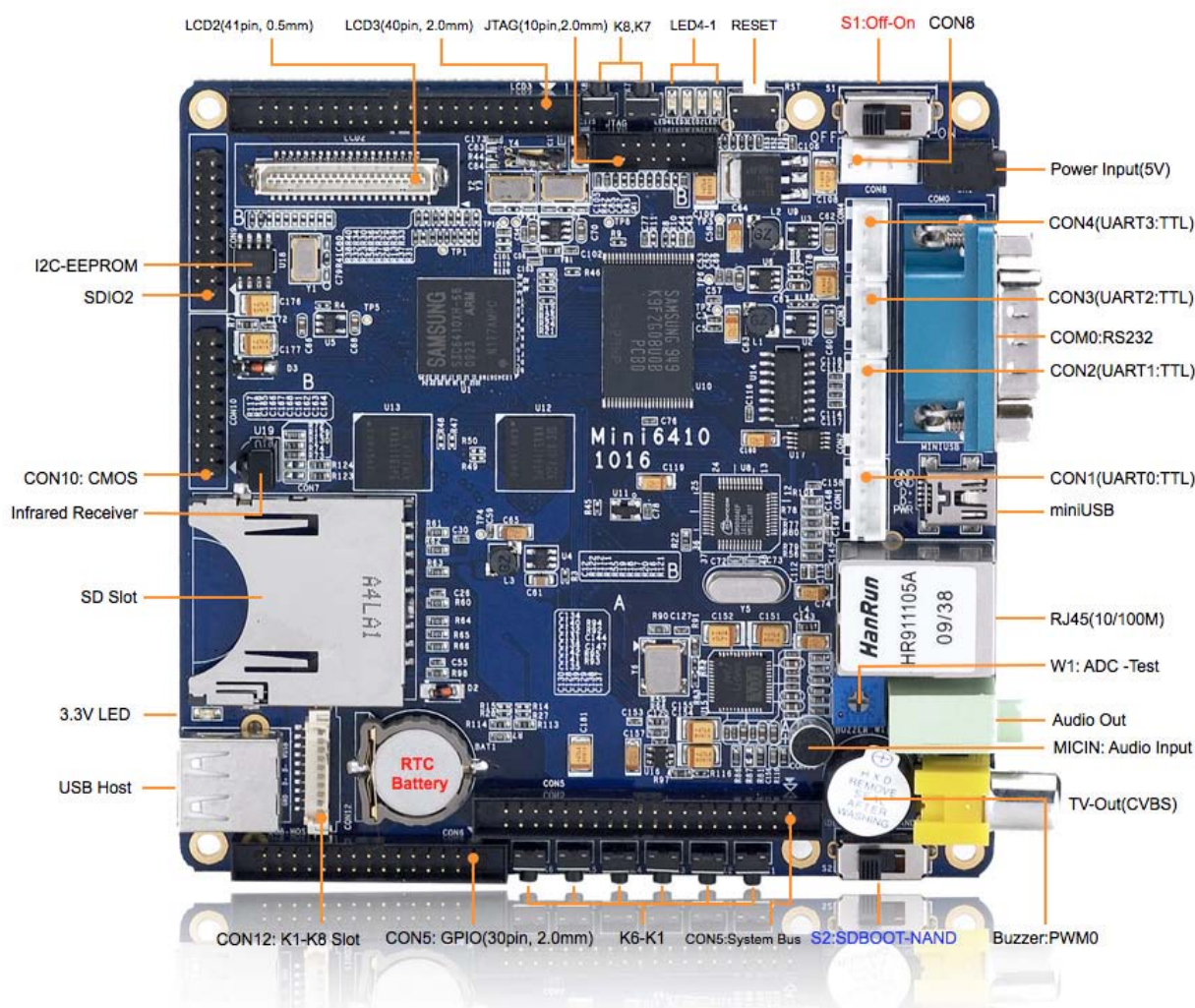
广州友善之臂计算机科技有限公司

目 录

基于MPLAYER的多媒体应用开发指南	- 1 -
目 录	- 4 -
第一章 MINI6410 和TINY6410 开发板简介	- 5 -
1.1 MINI6410 开发板	- 5 -
1.2 TINY6410 开发板	- 6 -
第二章 MPLAYER介绍	- 8 -
2.1 友善之臂对MPLAYER增加的新特性介绍	- 8 -
2.2 在命令行上体验MPLAYER的功能	- 10 -
第三章 使用Qt4 开发简单的广告机	- 12 -
3.1 基本原理解析	- 12 -
3.2 广告机的界面实现	- 13 -
3.3 实现视频在Qt4 窗口上的播放	- 15 -
3.4 实现界面与MPLAYER的交互	- 16 -
3.5 实现LOGO与字幕在视频中的叠加显示	- 17 -
3.6 示例源代码的编译和运行	- 18 -
3.7 实现开机自动运行广告程序	- 19 -
第四章 开发简单的视频播放器	- 20 -
4.1 基本原理讲解	- 20 -
4.2 播放器的界面实现	- 21 -
4.3 示例源代码的编译和运行	- 21 -
第五章 将视频压制成可供MPLAYER硬解播放的视频	- 23 -

第一章 mini6410 和tiny6410 开发板简介

1.1 Mini6410 开发板



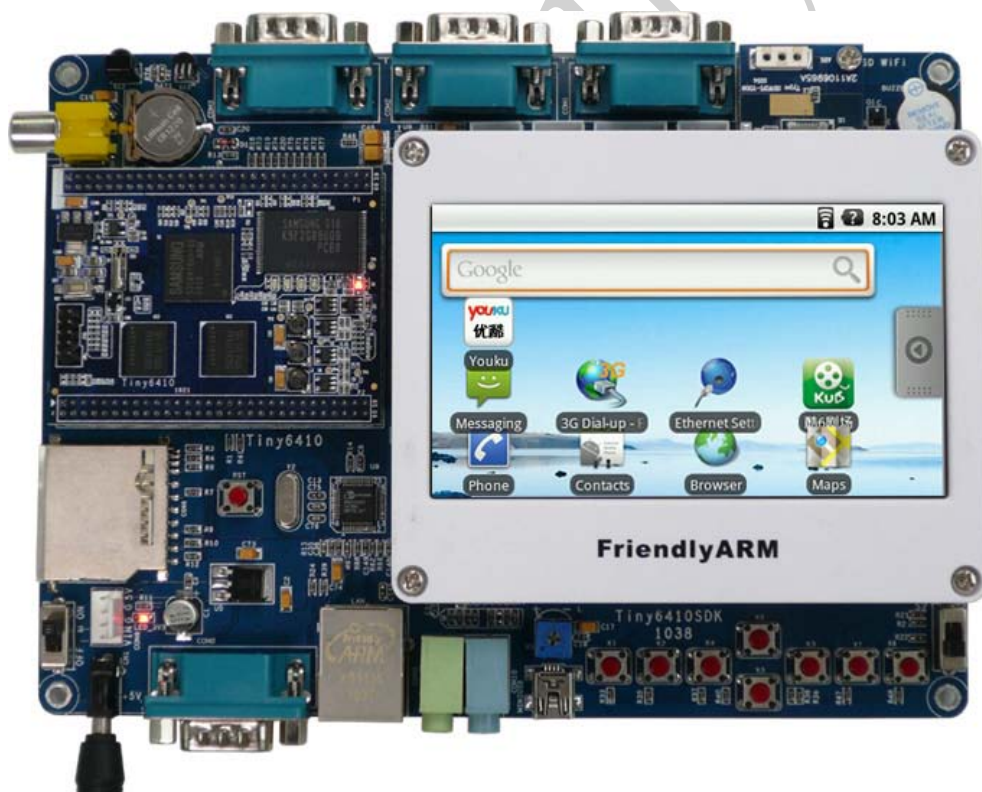
Mini6410 是一款十分精致的低价高品质一体化 ARM11 开发板，由广州友善之臂设计、生产和发行销售。它采用三星 S3C6410 作为主处理器，在设计上承袭了 Mini2440 “精于心，简于形”的风格，而且布局更加合理，接口更加丰富，十分适用于开发 MID、汽车电子、工业控制、导航系统、媒体播放等终端设备；也可适用于高校教学、嵌入式培训、个人研究学习和 DIY 等。

具体而言, Mini6410 具有双 LCD 接口、4 线电阻触摸屏接口、100M 标准网络接口、标准 DB9 五线串口、Mini USB 2.0-OTG 接口、USB Host 1.1、3.5mm 音频输出口、在板麦克风、标准 TV-OUT 接口、弹出式 SD 卡座、红外接收等常用接口; 另外还引出 4 路 TTL 串口、CMOS Camera 接口、40pin 总线接口、30pin GPIO 接口(可复用为 SPI、I2C、中断等, 另含 3 路 ADC、1 路 DAC)、SDIO2 接口(可接 SD WiFi)、10pin Jtag 接口等; 在板的还有蜂鸣器、I2C-EEPROM、备份电池、AD 可调电阻、8 按键(可引出)、4LED 等; 所有这些, 都极大地方便了开发者的评估和使用, 再加上我们按照 Mini6410 尺寸专门定制的 4.3"LCD 模块, 真正做到让您“一手掌握所有”!

我们还充分地发挥了 6410 支持 SD 卡启动这一特性, 使用我们精心研制的 superboot, 无需连接电脑, 只要把目标文件拷贝到 SD 卡中(可支持大于 2G 的高速大容量卡), 你就可以在开发板上极快极简单地自动安装或运行各种嵌入式系统(WindowsCE6/Linux/Android/Ubuntu 等); 甚至无需烧写, 就可以在 SD 卡上直接运行它们, 这一切, 简直太酷了!

要了解 Mini6410 开发板的详细信息, 请访问: <http://www.arm9.net/mini6410-feature.asp>

1.2 Tiny6410 开发板





追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

Tiny6410 是一款以 ARM11 芯片(三星 S3C6410)作为主处理器的嵌入式核心板,该 CPU 基于 ARM1176JZF-S 核设计,内部集成了强大的多媒体处理单元,支持 Mpeg4, H.264/H.263 等格式的视频文件硬件编解码,可同时输出至 LCD 和 TV 显示;它还并带有 3D 图形硬件加速器,以实现 OpenGL ES 1.1 & 2.0 加速渲染,另外它还支持 2D 图形图像的平滑缩放,翻转等操作。

Tiny6410 采用高密度 6 层板设计,尺寸为 64 x 50mm,它集成了 128M DDR RAM, 256M/1GB SLC Nand Flash 存储器,采用 5V 供电,在板实现 CPU 必需的各种核心电压转换,还带有专业复位芯片,通过 2.0mm 间距的排针,引出各种常见的接口资源,以供不打算自行设计 CPU 板的开发者进行快捷的二次开发使用。

Tiny6410SDK 是采用 Tiny6410 核心板的一款参考设计底板,它主要帮助开发者以此为参考进行核心板的功能验证以及扩展开发。该底板具有三 LCD 接口、4 线电阻触摸屏接口、100M 标准网络接口、标准 DB9 五线串口、Mini USB 2.0 接口、USB Host 1.1、3.5mm 音频输入输出口、标准 TV-OUT 接口、SD 卡座、红外接收等常用接口;另外还引出 4 路 TTL 串口 另 1 路 TV-OUT、SDIO2 接口(可接 SD WiFi)接口等;在板的还有蜂鸣器、I2C-EEPROM、备份电池、AD 可调电阻、8 个中断式按键等。

在布局上安排上,我们尽量考虑把常用尺寸的 LCD 模块能够固定在底板上,比如 3.5", 4.3" LCD, 7" LCD 等,这样用户在使用时不至于把各种电线搅在一起,更增加了开发套件的便携性。

我们还充分地发挥了 6410 支持 SD 卡启动这一特性,使用我们精心研制的 Superboot,无需连接电脑,只要把目标文件拷贝到 SD 卡中(可支持高达 32G 的高速大容量卡),你就可以在开发板上极快地简单地自动安装各种嵌入式系统(WindowsCE6/Linux/Android/Ubuntu /uCOS2 等);甚至无需烧写,就可以在 SD 卡上直接运行它们!

要了解 Tiny6410 开发板的详细信息,请访问: <http://www.arm9.net/tiny6410.asp>



第二章 MPlayer介绍

MPlayer是一款开源的多媒体播放器,可在各主流作业系统使用,例如Linux和其他类Unix作业系统、微软的视窗系统及苹果电脑的Mac OS X系统。MPlayer基于命令行界面,但在各作业系统可选择安装不同的图形界面。

2.1 友善之臂对MPlayer增加的新特性介绍

友善之臂在MPlayer集成实现了6410的硬件解码的播放能力,可以流畅播放分辨率为720x480的H.264/H.263/Mpeg4格式的视频文件,同时,为了方便客户能够更好地利用MPlayer进行二次开发,我们为MPlayer增加了以下一些参数选项,目的是方便将视频定位显示于屏幕上的任意位置,缩放至任意大小等等:

新增的参数选项	作用说明
-videoframe fullscreen:keepratio	全屏模式显示视频,但保持视频原有的长宽比例 (注意,这里的全屏会受参数-screenrect约束) (show fullscreen and keep film aspect ratio.)
-videoframe fullscreen:stretch	全屏模式显示视频,但不保持视频比例,视频将被拉伸并填充到边界 (注意,这里的全屏会受参数-screenrect约束) (show fullscreen and stretches image to the screen's size.)
-videoframe zoom:<percent>	按指定的百分比缩放视频。 (set videoframe's size according to zoom ratio. (Example: -videoframe zoom:200))
-screenrect <x,y,w,h>	限制视频显示在屏幕上的指定位置,如果不指定该参数,默认是视频显示将占据整个屏幕。 (set video's display area: video is limited to the specified area. (Example: -screenrect 50,50,320,240))
-framebuffer-index <index>	指定将视频输出到某一个FrameBuffer上,友善之臂6410开发板支持4个FrameBuffer叠加,也就是说,



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

	<p>屏幕上最终看到的图像是 4 个 FrameBuffer 的图像叠加后的图像。</p> <p>如果不指定该参数,将使用索引为 1 的 FrameBuffer,而 Qt/Qtopia 使用的是 0 的 FrameBuffer,也就是说,仍有二个 FrameBuffer 是空闲的,在本文档的示例中,我们将它们用于显示台标 (LOGO) 和字幕。</p> <p>(set video's display on specified framebuffer, index range 0~3.)</p>
<code>-tvout <off tonly tvandlcd></code>	<p>该参数可选择将视频输出到电视,有三个值可选: off、tonly 和 tvandlcd,默认为 off 即不输出到电视,如果选择 tonly,视频被输出到电视,但 LCD 不显示视频,当选择 tvandlcd 时,视频将同时输出到 LCD 和电视上。</p>

(注意: 只有播放支持硬解的视频时,上面的参数才有效,因此,请参考本文档的章节五,将你的视频转换成可硬解播放的格式)

下面用一幅图来总结友善之臂为 MPlayer 增加的新特性,在下图中,右边是最终要实现的产品效果,包括有如下内容: 有用 Qt4 编写的程序界面,以及在 Qt4 界面上显示 MPlayer 播放的视频,另外,结合多 Frame Buffer 特性,在视频上方和下方分别显示了 Logo 和字幕:

4) 在第3个Frame Buffer上显示Logo

1) Qt4的界面，占据全屏，显示在第1个Frame Buffer



最终显示效果



2) Mplayer视频画面，使用-screenrect参数指定了位置，显示在第2个Frame Buffer上。

3) 在第4个Frame Buffer上显示字幕。

2.2 在命令行上体验MPlayer的功能

新版本的 mplayer 位于/bin/目录下，因此，直接输入 mplayer 加文件名即可播放视频啦，如下所示：

```
# mplayer /sdcard/demo.mp4
```

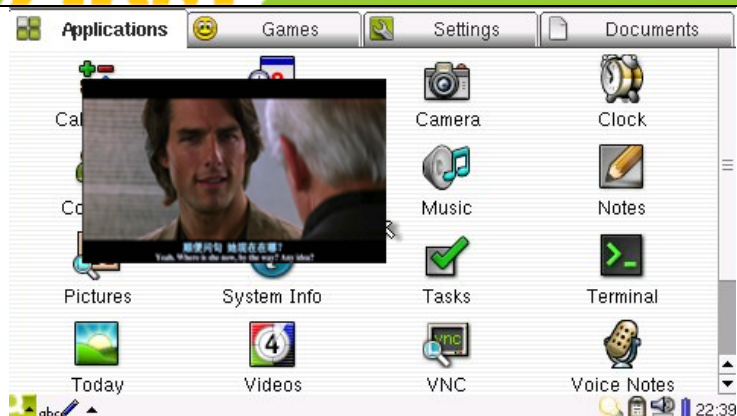
如果要播放网络上的视频，则将文件名改成网址即可，如下所示：

```
# mplayer http://192.168.1.100/demo.mp4
```

视频默认将占据全屏进行播放，如果不想全屏，则可使用 screenrect 参数指定视频的位置即可，如下所示：

```
# mplayer -screenrect 50,50,200,120 /sdcard/demo.mp4
```

你会发现视频很酷的漂在 qtopia 的界面上面，如下图所示：



在上图中，由于 MPlayer 默认将图标显示在第 2 个 Frame Buffer 上，因此不会破坏 Qt/Qttopia 的画面。

下面命令将在电视上播放视频，如下所示：

```
# mplayer /sdcard/demo.mp4 -tvout tvonly
```

第三章 使用Qt4 开发简单的广告机

在开始之前，读者需要先做好以下工作：

- 1) 安装 Fedora;
- 2) 安装友善之臂提供的 mini6410 交叉编译器;
- 3) 烧写最新 Linux 系统的 Image 到开发板;
- 4) 参考文档《Qt4 和 Qtopia 编程开发指南》搭建 Qt4 开发环境，安装 QtCreator 等，我们的示例程序是使用 QtCreator 开发的。

3.1 基本原理解析

如何实现广告机的图形界面？

Mplayer 是不带图形界面的，我们将使用 Qt4 来设计广告机的图形界面。

如何将 MPlayer 视频显示在 Qt4 的界面上？

MPlayer 不支持将视频显示在嵌入式版 Qt4 的窗口上，而是直接显示在 Frame Buffer 上，但我们可以结合多 Frame Buffer 特性，以及 screenrect 参数，将视频定位到 Qt4 某个窗口的某个区域上面，如下图所示：



Qt4 如何与 MPlayer 交互？

一旦启动了 MPlayer，MPlayer 将以单独的进程形式存在，Qt4 程序将以用管道发送命令给 MPlayer 的方式来操纵 MPlayer，诸如暂停播放，设置音量等等。

这种交互的方式称为 MPlayer 的 slave 模式，就是 mplayer 在运行过程中能够接收用户的输入命令行，具体支持哪些命令行，可以通过 `mplayer -input cmdlist` 这条命令来得到。

3.2 广告机的界面实现

本示例要实现的广告机界面如下图所示，界面比较简单，分为两个部分：1) 屏幕右侧将显示一排可以点击的图形按钮；2) 左侧则用于广告视频的显示：



图形按钮组的实现

屏幕右侧的图形按钮实现起来并不难，首先需要准备好每一个按钮未点击时的图片素材，以及点击后的图片素材，并按一定的规则命名，如下图所示：



本示例中，我们为 4 寸屏和 7 寸屏分别准备了一套图片素材，并将它们添加到程序的资源文件 buttonImages.qrc 中。

程序运行时，会加载这些图片到内存中，然后在窗口的 `MainWidget::paintEvent` 函数中进行按钮的绘制。当用户点击图形按钮时，在 `MainWidget:mousePressEvent` 函数中处理按钮的点击事件，具体细节请参考示例源代码。

全屏显示窗口

广告机示例程序是一个全屏显示的程序，即窗口没有标题栏、边框和菜单等，要实现全屏很简单，只要在 `main` 函数中，用 `w.showFullScreen();` 显示窗口即可。

3.3 实现视频在Qt4 窗口上的播放

在 Qt4 中, 启动一个新程序以及跟踪它的状态, 最方便的方式就是使用 QProcess 类, 因此, 我们需要在窗口的构造函数中实例化一个 QProcess 对象, 用于启动 MPlayer:

```
m_mplayerProcess = new QProcess(this);
connect(m_mplayerProcess, SIGNAL(readyReadStandardOutput()),
        this, SLOT(catchOutput()));
connect(m_mplayerProcess, SIGNAL(finished(int, QProcess::ExitStatus)),
        this, SLOT(mplayerEnded(int, QProcess::ExitStatus)));
```

如上述的代码所示, 除了实例化了一个 QProcess 对象外, 我们还连接了它的两个信号, 其中, readyReadStandardOutput()信号会在程序有输出发射, 而 finished 信号则在程序结束时发射, 本示例中, 我们主要关心的是 finished 信号, 因为当 MPlayer 进程结束时则意味着一个视频已播放完成, 我们将在一个广告播放完成时, 自动播放下一个广告。

如果你要实现一个播放器, 需要关心播放进度等信息, 则需要连接 readyReadStandardOutput()信号并解析 MPlayer 的输出信息。

下面的代码将启动 MPlayer, 并传递相应的参数给 MPlayer, 注意, -slave 和 -quiet 是两个比较主要的参数, 用于指示 MPlayer 工作在 slave 模式, 其它的参数在文档前面已经有介绍过了:

```
QStringList args;
args << "-framebuffer-index" << "1";
args << "-videoframe" << "fullscreen:keepratio";

QString screenRect =
QString("%1,%2,%3,%4").arg(m_videoWindowRect.left()).arg(m_videoWindowRect.top()).arg(m_videoWindowRect.width()).arg(m_videoWindowRect.height());
args << "-screenrect";
args << screenRect;

args << "-slave";
args << "-quiet";
args << m_movieFiles[m_currentMovieIndex];

m_mplayerProcess->setProcessChannelMode(QProcess::MergedChannels);
m_mplayerProcess->start(MPLAYER_PATH, args);
```

```
if(!m_mplayerProcess->waitForStarted(3000)) {  
    qDebug("startMplayer failed");  
    return false;  
}
```

上面已经有说明过，MPlayer 播放视频时，视频将在另一个 FrameBuffer 上显示，代码中，m_videoWindowRect 变量存储的是经过计算的用于显示视频的区域，它作为参数值传递给 MPlayer，保证了视频不会覆盖到图形按钮的位置。

3.4 实现界面与MPlayer的交互

虽然 MPlayer 与 Qt4 程序之间使用的管道方式的进程间通讯方式，但是幸运的是，QProcess 为我们隐藏并处理了这些细节，因此，与 MPlayer 交互非常简单，给 MPlayer 发命令只需要简单地调用 write 函数即可：

```
m_mplayerProcess->write("quit\n");
```

上面的代码是告诉 MPlayer 退出，即停止播放。

3.5 实现Logo与字幕在视频中的叠加显示



如上图所示，我们使用的广告机示例程序使用了 4 个 Frame Buffer，分别用于显示 Qt4 程序界面、MPlayer 视频图像、Logo 和字幕。

4 个 Frame Buffer 的设备文件分别为：

/dev/fb0	通常被 Qtopia/Qt4 占用
/dev/fb1	MPlayer 默认占用此 Frame Buffer，但可另外指定
/dev/fb2	未使用，本例中用于显示 Logo
/dev/fb3	未使用，本例中用于显示字幕

下面的代码演示打开 /dev/fb2，然后在上面显示一幅预览准备好的图像作为 Logo：

```
m_fd2 = ::open(FB_DEV_NAME2, O_RDWR | O_NDELAY);
if (m_fd2 >= 0) {
    s3c_win_info_t fb_info_to_driver;
    fb_info_to_driver.Bpp = 16; //BPP
    fb_info_to_driver.LeftTop_x = 5;
    fb_info_to_driver.LeftTop_y = 5;
    fb_info_to_driver.Width = 80;
    fb_info_to_driver.Height = 18;

    int fb_size = fb_info_to_driver.Width * fb_info_to_driver.Height * 2; // RGB565
    char* fb_addr = (char *) mmap(0, fb_size, PROT_READ | PROT_WRITE, MAP_SHARED, m_fd2, 0);
    if (fb_addr == NULL) {
        ::close(m_fd2);
        m_fd2 = -1;
        qDebug("mmap fb2 failed!");
        return ;
    }

    if (ioctl(m_fd2, SET_OSD_INFO, &fb_info_to_driver) == -1) {
        qDebug("SET_OSD_INFO fb2 failed!");
    }

    if (ioctl(m_fd2, SET_OSD_START) == -1) {
        qDebug("SET_OSD_START fb2 failed!");
    }

    // show logo at the video corner (left/top)
    memcpy(fb_addr, gImage_logo, sizeof(gImage_logo));
}
```

3.6 示例源代码的编译和运行

广告机的示例程序是开源的，源代码放在光盘 A 的“开发文档和教程\专题 06 基于 MPlayer 的多媒体

应用开发指南\源代码”目录下。

源代码可使用 Qt Creator 进行交叉编译（需预先配置好环境），也可以在命令行上进行编译，关于 Qt Creator 的配置和使用，可参考另一份文档《Qt4 和 Qtopia 编程开发指南》。

将广告机的源代码进行编译后，可得到一个名为 adsdemo 的可执行文件，将它替换掉板子上的 /usr/local/adsdemo 目录下的同名文件即可，示例程序中用到的广告视频位于 /usr/local/ads 目录下。

要启动广告机程序，只要在“友善之臂”程序组中，找到 AdsDemo 的图标点击即可：



要退出广告机程序，点击视频窗口即可。

另外，在板子上，/bin 目录下有一个脚本 /bin/adsdemo 用于配置 adsdemo 所需的 Qt4 执行环境，并最终启动 adsdemo 程序。

3.7 实现开机自动运行广告程序

可手动编辑 /etc/init.d/rcS 文件，将倒数第 3 行的 /bin/qtopia & 改为 /bin/adsdemo & 即可。

第四章 开发简单的视频播放器

4.1 基本原理讲解

本示例中要实现的播放器界面如下图所示，播放器只实现了基本的功能，例如打开视频文件、播放、暂停和恢复播放、进度显示和进度定位、将输出切换到电视输出等。



播放器的原理与广告机其实基本相同，只是播放器会有更多的播放控制处理，例如播放进度的显示、暂停和恢复播放、将视频输出到电视等，因此，本章节只讨论与广告机有差异的部分。

如何显示播放进度？

我们会在界面上放置一个滑块控件，用于显示进度。

首先，在开始播放一个视频时，向 mplayer 发送命令 `get_time_length` 来查询视频的时长，

然后开启一个 timer，在播放的过程中，每隔 1 秒钟就给 mplayer 发送命令 `get_time_pos` 来查询当前的播放进度，并将进度更新到滑块控件上即可。

Mplayer 返回查询结果时，`catchOutput` 函数会被调用，可查看源代码了解相关的细节。

如何暂停和恢复播放？

播放过程中，通过给 mplayer 发送命令 `pause` 来暂停播放，暂停播放的状态下重新发送一次 `pause` 则会恢复播放。

如何在播放中途将视频切换至电视输出？

由于 tvout 的设置是在 mplayer 的命令行参数中传入的，所以，在播放中途，要将视频输出到电视就需要重新启动 mplayer，重新启动 mplayer 默认情况下会导致视频从头开始播放，但我们可以在启动之前先保存一下当前的进度，然后在重新启动后，通过给 mplayer 发 seek 命令定位到上一次的播放位置，从而使得切换输出模式而不影响播放进度。

重新启动 mplayer 传入 tvout 的参数，则可将视频输出到电视，有二种模式，一种是只把画面输出到电视，LCD 不显示，另一种时，电视和 LCD 同时输出，后者要占资源多一些。

全屏播放和音量控制？

示例源代码中并没有开发这两个功能，而是留给用户自己去实现，有问题可到<http://www.arm9home.net>论坛进行交流。

4.2 播放器的界面实现

播放器的界面很简单，是直接使用 QtCreator 的界面设计器进行设计的，可用 QtCreator 打开示例工程，打开 mainwidget.ui 了解相关的细节。

4.3 示例源代码的编译和运行

视频播放器的示例程序是开源的，源代码放在光盘 A 的“开发文档和教程\专题 06 基于 MPlayer 的多媒体应用开发指南\源代码”目录下。

源代码可使用 Qt Creator 进行交叉编译（需预先配置好环境），也可以在命令行上进行编译，关于 Qt Creator 的配置和使用，可参考另一份文档《Qt4 和 Qtopia 编程开发指南》。

将视频播放器的源代码进行编译后，可得到一个名为 myplayer 的可执行文件，将它替换掉板子上的 /usr/local/myplayer 目录下的同名文件即可。

要启动视频播放器程序，只要在“友善之臂”程序组中，找到 MyPlayer 的图标点击即可：



另外，在板子上，/bin 目录下有一个脚本/bin/myplayer 用于配置 myplayer 所需的 Qt4 执行环境，并最终启动 myplayer 程序。

第五章 将视频压制成可供MPlayer硬解播放的视频

将视频转换成 MP4 格式

使用软件 Aimersoft iPhone Converter Suite 可以将视频转换成 6410 上可硬解播放的 MP4 视频格式, 这个软件本来是用于 iPhone 视频转换的, 但由于 iPhone 1 代用的 CPU 就是 6410, 所以也适用于 Mini6410/Tiny6410, 注意, 我们使用的 Aimersoft iPhone Converter Suite 版本是 1.1.32, 我们推荐你使用这个版本, 因为 iPhone 在不断升级换代, 不保证其它版本能够使用。

Aimersoft iPhone Converter Suite 的主界面如下图所示:



在界面上点击 iPhone Video Converter, 弹出如下界面:



在界面上通过点击“打开”来选择你要转换的视频文件，然后在界面下方的方案中选择“Apple TV MPEG-4 720X432(*.mp4)”，然后点击“设置”按钮，弹出视频格式设置对话框，按下图红色标出的设置项进行设置：



点击“确定”回到主界面，点右下角的 Start 开始转换视频。

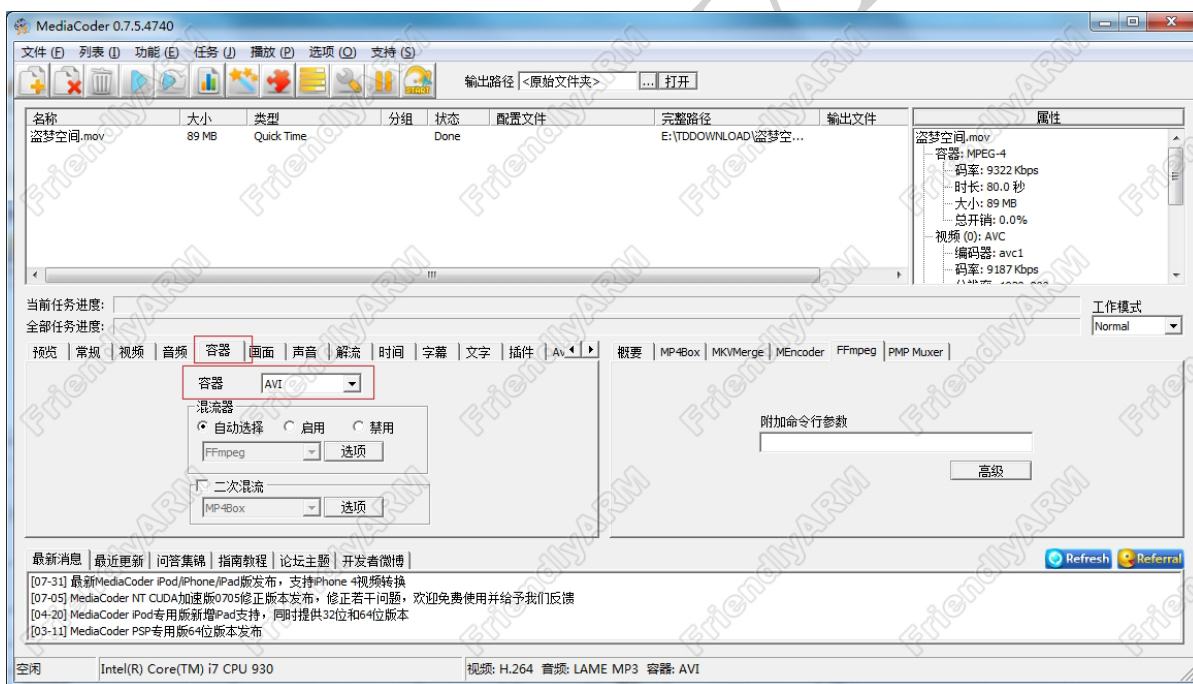
将视频转换成 H264 格式

使用软件 MediaCoder 可以将视频转换成 6410 上可硬解播放的 H264 视频格式，在编写本文时我们使用的 MediaCoder 版本是 0.7.5.4740，它是一个免费的软件，可以到它的官方网站 <http://www.mediacoder.cn/> 下载。

启动 MediaCoder，主界面如下图所示，请在主界面中，点击左上角的“+”号选择要转换的视频，然后，请参考下图中红色标出的视频配置进行设置：



在上图中，我们将视频格式设置为 H.264、二次编码、编码器使用 x264、规格为 Baseline 等，这些设置是根据 6410 的硬解能力来设定的，继续参考下图配置其它的视频参数：



在上图中，将容器设置为 avi，再点击“画面”选项卡，参考下图将视频设置为最大 720x400，并根据比例动态设置高度，6410 支持的最大硬解分辨率了是 720x480 30fps，所以不能设置超过这个规格的分辨率：

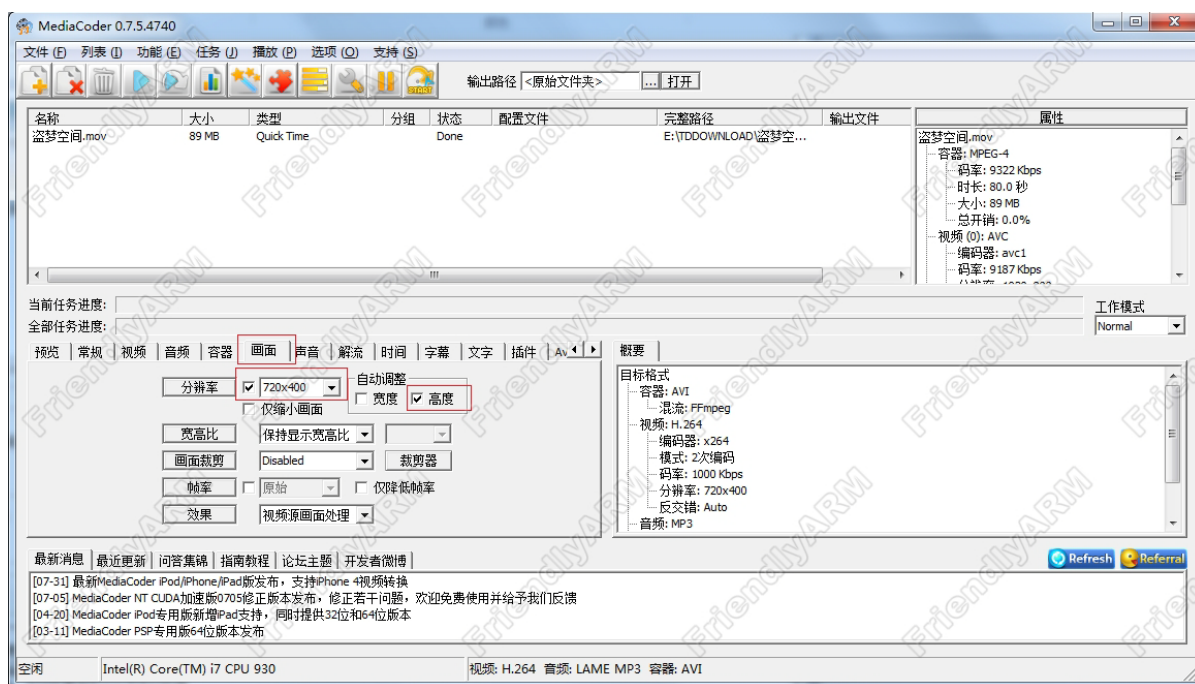


追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司



一切就绪后，点击工具栏上的 Start 开始转换视频。

更多开发相关的专题文档，敬请期待...

请密切留意我们的论坛 (<http://www.arm9home.net>) 上发布的最新消息。