

文章编号:1001-9081(2011)09-2584-05

doi:10.3724/SP.J.1087.2011.02584

基于 Atom Z510 的车载多媒体播放系统

李 霞, 闵华松

(武汉科技大学 计算机科学与技术学院, 武汉 430065)

(scjlxia@126.com)

摘要:针对现有车载影音系统功能单一、更新不便的缺陷,设计并实现了一款能够同时实现音频和视频播放的高性能车载播放器。系统采用 Intel 最新推出的嵌入式处理器 Atom Z510 作核心处理器,在其上定制 XPE 作操作系统,采用 Qt Creator 作开发工具,使用图形用户接口(GUI)技术,实现了一个带有图形操作界面的播放器。该播放器既能播放本地媒体文件,也能识别 CF 卡、U 盘等移动存储设备,并能对移动设备上存储的媒体文件进行播放。

关键词:车载播放器; 图形用户接口; Atom510 处理器; XPE

中图分类号: TP311.521 **文献标志码:**A

Vehicle multimedia player based on Atom Z510

LI Xia, MING Hua-song

(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan Hubei 430065, China)

Abstract: Since the current vehicle audio system has a defect that the function of the system is limited and it is difficult to update media files for the system, the design and implementation of a high-performance vehicle player that can play both audio and video media files were given. The latest embedded processor Atom Z510 published by Intel was selected as the core processor of the system, XPE (Windows XP Embedded) was built on it, and Qt-Creator was used to design the software of the player, Graphical User Interface (GUI) technology was also used to design a graphical user interface for the player. The player can not only play local media files, but also identify the CF card, U disk and other removable storage devices and play media files stored on them.

Key words: vehicle player; Graphical User Interface (GUI); Atom Z510 processor; Windows XP Embedded (XPE)

0 引言

在车载影音行业的发展中,产品经历了几个升级阶段:20世纪90年代以前,市场是以立体声AM/FM收音机为主;90年代中期,市场出现收放机(又称卡带机);随后的十几年中,车载影音产品更新换代速度加快,经历了从卡带机向CD机的转变,从纯听觉的CD机向视听结合的DVD机转变^[1]。目前主流的车载播放器仍然是CD、DVD播放器,其存储内容不能随时更新,支持的媒体文件格式也很有限;但在消费类电子产品中,主流的便携式播放器已支持几乎所有个人计算机支持的媒体文件格式。车载信息系统的发展趋向于引入高性能的嵌入式硬件平台和操作系统,以满足车辆本身及驾乘人员的信息化需求^[2]。

本播放器基于 Intel 推出的最新高性能嵌入式处理器 Atom Z510,通过定制 Windows Embedded Standard 操作系统,集成跨平台播放器 Mplayer,实现对媒体文件的播放。同时,在核心处理器上扩展了 USB 接口和 CF 卡接口,用户可以将媒体文件存储在 CF 卡、U 盘等移动存储设备中,通过 CF 卡接口和 USB 接口接入系统,实现对 CF 卡、U 盘等外部移动存储设备中保存媒体文件的播放。

1 系统总体设计

本系统主要由硬件层、系统软件层、应用软件层三部分组成,系统结构如图 1 所示。

硬件层采用 Atom Z510 处理器作核心处理器,外接 CF 卡(或 USB 等移动存储设备)、LCD 显示屏和按键。

系统软件层采用嵌入式操作系统 Windows Embedded XP (XPE) 作操作系统,在 XPE 上安装声卡驱动、显卡驱动。

应用软件层采用 Qt Creator 设计用户图形操作界面,提供人机交互接口,并利用 QT 的信号和槽机制,调用开源多媒体播放器 Mplayer 实现对媒体文件的播放。

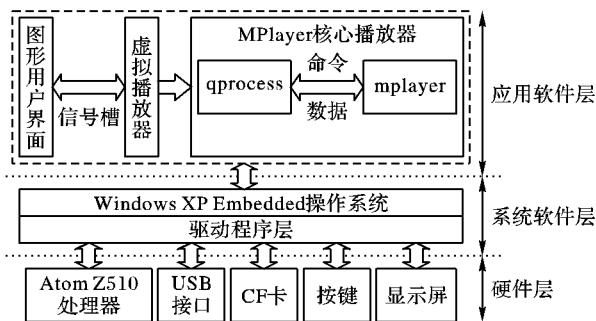


图 1 系统总体结构

2 硬件平台设计

系统硬件平台选用 Atom Z510 处理器作核心处理器。Atom Z510 集成 Intel GMA500 显示控制器,支持 2D、3D 显示加速,支持 H.264、MPEG2、*MPEG4、VC1 和 WMV 硬件解码,支持高清播放,支持双头显示^[3]。Intel Atom Z510 具有低功耗的特性^[4],可提供 6~24 V DC 输入,耗电量极微小,CPU

收稿日期:2011-01-28;修回日期:2011-03-20。

作者简介:李霞(1982-),女,湖北十堰人,硕士研究生,主要研究方向:嵌入式系统; 闵华松(1969-),男,湖北人,教授,博士,主要研究方向:嵌入式系统、可编程片上系统、机器人控制、工业以太网。

耗能也仅有2.5 W,可满足车载播放系统低功耗的需求。

系统扩展了5个USB接口和一个CF卡接口以支持移动存储设备的接入;配置一个Keypad,给用户提供按键操作^[5];选用一个Mega8单片机与核心处理器相连,用于处理按键信息;配备一块7 inch(1 inch = 2.54 cm)800×480的LCD显示屏,用于视频的显示输出。

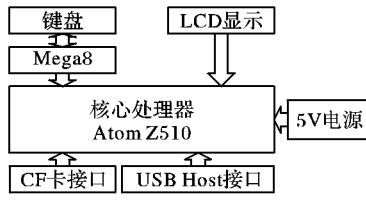


图2 车载播放系统硬件结构

按照硬件平台的功能设计分布,该硬件平台分为以下几个模块^[6]。

2.1 高速接口模块

该模块以Intel Atom Z510处理器和桥芯片SCH US15WP为核心,Atom Z510通过US15WP桥芯片访问1 GB DDR2内存,并通过该芯片控制整个系统资源。该模块具备的外设资源如下:1)US15WP桥芯片通过并行ATA接口与IDE硬盘或CF卡连接,IDE硬盘和CF卡中存储嵌入式Windows操作系统和媒体数据,系统上电后,从硬盘或CF卡引导启动,进入Windows操作系统,然后运行媒体播放程序。2)US15WP桥芯片具有8个USB2.0接口,其中一个连接触摸屏控制器,用于用户操作LCD,其余7个USB接口外置,用于U盘等即插即用设备,随时和外界进行大容量数据交互。3)US15WP芯片集成Intel GMA500显示控制器,支持2D、3D显示加速,支持H.264、MPEG2、MPEG4、VC1和WMV硬件解码,支持高清播放,支持双屏显示,支持VGA、DVI等显示端口输出,和LCD控制器连接,LCD安装在汽车的操作面板中央,方便用户观看。

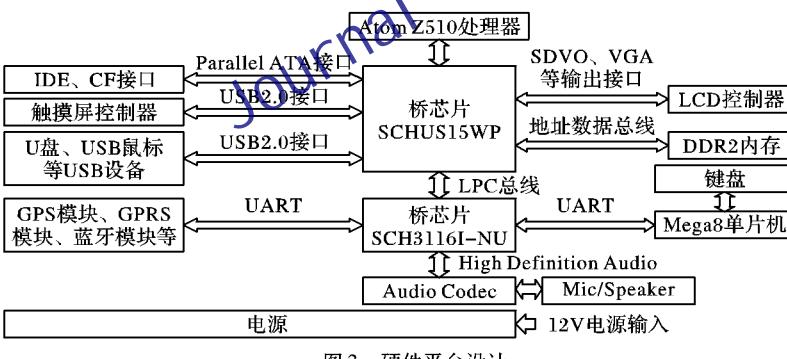


图3 硬件平台设计

2.2 低速接口模块

该模块以Intel SCH3116I-NU桥芯片为核心,外设速度相对较慢。SCH3116I-NU通过的LPC总线(Low Pin Count Bus)接口和SCH US15WP连接,作为SCH US15WP的外设。该模块具备的外设资源如下:1)SCH3116I-NU桥芯片集成6个UART接口,其中一个UART接口和Atmel Mega8单片机连接,接收单片机扫描到的按键代码,让用户通过键盘控制车载播放器;其余串口连接GPS模块、GPRS模块等,使系统具备车载导航、通信功能。2)SCH3116I-NU的High Definition Audio接口外扩REALTEK ALC883声卡芯片,使该系统具备双通道录/放音功能。

2.3 电源模块

该模块为整个系统提供电源支持。输入电压为6~24 V。

3 软件设计与实现

该播放器同时支持音频和视频播放,并且给用户提供了图形操作界面。播放器的图形操作界面采用Qt Creator来设计,通过调用Qt Creator提供的控件库根据需要设计用户界面。播放器的播放功能是通过在后台调用MPlayer来实现的,MPlayer是一款跨平台播放软件,此款软件可在各主流作业系统使用^[7],例如Unix、Windows、Mac OS X系统等。MPlayer的功能十分强大,能够播放众多格式的媒体文件^[8],MPlayer能使用众多本地的Xanim、RealPlayer和Win32 DLL编解码器^[9],播放大多数MPEG、VOB、AVI、OGG、VIVO、ASF/WMV、QT/MOV、FLI、RM、NuppelVideo、yuv4mpeg、FILM、ROQ文件。

3.1 系统支持格式与遵循标准

系统支持多数音频格式和视频格式文件。

1)支持的音频格式:AAC、AC3、ALAC、AMR、FLAC, Intel Music Coder, Monkey's Audio、MP3、Musepack、RealAudio、Shorten、Speex、Vorbis、WMA。

2)支持的视频格式:Cinepak、DV、H.263、H.264/MPEG-4 AVC、HuffYUV、Indeo、MJPEG、MPEG-1、MPEG-2、MPEG-4 Part 2、RealVideo、Sorenson、Theora、WMV。

系统支持多种主流视频编码标准,如MPEG-1、MPEG-2、MPEG-4、MJPEG、H.263、H.264等。MPEG系列标准是由ISO/IEC制定的,主要应用于视频存储(DVD)、广播电视、因特网或无线网上的流媒体等。H.26X系列标准由ITU组织制定,该标准主要针对实时视频通信的应用^[10]。

3.2 软件设计

3.2.1 软件架构设计

播放器软件部分分为三层:图形用户界面层、虚拟播放器层和核心播放器层。图形用户界面层负责管理和设置图形界

面中各图形元素在不同时刻的不同状态;虚拟播放器和核心播放器完成播放控制,提供播放控制接口供外部调用。三层之间的通信利用QT的信号和槽机制来实现,播放器软件结构如图4所示。

图形界面为用户提供了统一的操作界面,系统的图形界面主要由主界面、音频播放界面和视频播放界面组成。从主界面中可以进入音频播放界面和视频播放界面。

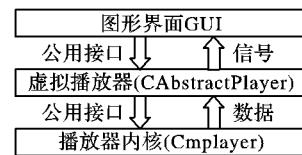


图4 播放器软件结构

虚拟播放器旨在设计一个不包含图形操作界面,但为一个播放器所必备的播放控件提供操作接口的抽象播放器,播放控件的逻辑控制操作均在对应的接口内完成。虚拟播放器提供的操作接口有:播放按钮接口BtnPlayClicked()、上一首/下一首按钮接口PlayPrev() / PlayNext()、进度条拖动接口SliderDraged()、音量加/音量减按钮接口VolumeUp() / VolumeDn(),加载播放列表接口LoadPlayList()。当用户在图形界面上点击某个播放控件时,便发出按钮单击信号,将此信号关联到处理该按钮的槽,在槽中调用虚拟播放器的相应

接口完成该按钮的逻辑控制操作。

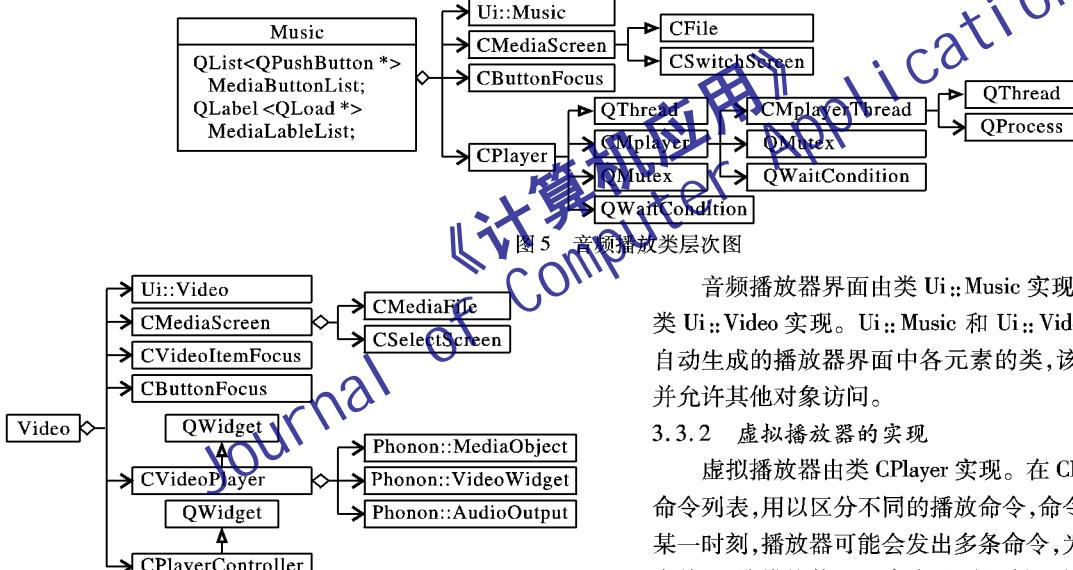
核心播放器层封装了跨平台播放器 MPlayer，提供一系列的播放控制接口函数供外部调用，实现对媒体文件的播放。核心播放器提供的接口有：播放/暂停 `bool Pause()`、上一首/
下一首 `PlayPrev() / PlayNext()`、获取播放文件名 `GetFileName()`、
获取播放总时长 `GetTimeLength()`、获取当前播放时间
`GetTimePos()`、获取播放进度 `GetPlayProgress()`、跳转到制定位置播放
`Seek(int progress)`、循环播放 `PlayLoop()` 等。

3.2.2 软件功能模块设计

从系统中抽取出来具有独立功能的模块，使用 C++ 面向对象思想，将每个功能模块均封装成一个类，提供接口供外部调用。系统划分的功能模块有：文件管理模块、屏幕切换模块、播放器核心模块、虚拟播放器模块、播放器界面管理模块和设备监测模块。下面逐一对其进行简要介绍。

1)文件管理模块(CFile):用于管理文件目录和目录中的文件,可以查询一个或多个目录中的文件,并能对文件进行指定格式的筛选。

2) 屏幕切换模块(CSwitchScreen):本模块使用图形控件来显示用户定义的对象,将需要显示的对象以每屏固定数量在屏幕上显示,如果一屏文件不足指定数量,则将用于显示对



3.3.1 播放器界面实现

图形用户界面层包括主界面、音频播放界面和视频播放界面。主界面提供音频播放和视频播放两个选项供用户选择,是系统的入口。以 Qt Creator 为基础开发,通过调用 Qt 提供的控件库来设计图形界面。音频播放界面和视频播放界面比较类似,都包括以下元素:用于显示媒体文件的按钮列表、用于显示播放进度的 SliderBar、用于进行播放控制的按钮列表(播放/暂停,上一首/下一首,音量加/音量减)、用于显示播放文件信息的 label(文件名、总时长,当前播放时间)、以及用于更新移动设备中媒体文件的刷新按钮。移动设备中存储的文件可通过刷新按钮进行更新,当有移动设备接入时,点击一下刷新按钮,就将移动设备中的媒体文件读取到媒体显示列表中,并接在原媒体列表的末尾进行显示。当移动设备拔出时,则媒体列表自动进行刷新,将移动设备中的媒体文件从媒体显示列表中移除。此外,音频播放界面还包含一个用于进行播放模式选择的 Frame。

象的控件隐藏,如果一屏显示不下,则分多屏显示。设置“上一屏”和“下一屏”按钮来在不同页面之间进行切换。

3) 媒体列表显示模块(CMediaScreen):此模块继承自类CFile 和类 CSwitchScreen,将文件管理和屏幕切换模块结合到一起,将可选择的媒体文件以列表的形式显示出来,供用户选择。

4) 设备监测模块: 实现对移动设备接入和拔出的及时监测。

5) 核心播放器模块 (CMplayer): 其中封装播放器 Mplayer, 提供媒体播放控制接口。

6) 虚拟播放器模块(CPlayer):提供播放器图形用户界面中常见播放控制按钮的操作接口。

7) 播放器界面管理模块:包含一个媒体播放列表和常见播放控制按钮,音频界面和视频界面相似,但略有差别。

3.3 软件实现

音频播放实现了对一个播放列表的播放，并提供随机、循环、单曲三种播放模式供用户选择^[11]。视频播放的播放模式为单个视频播放，当一个视频文件播放完毕时，返回到视频主界面，从媒体列表中选择下一个要播放的文件进行播放。音频播放类层次图如图 5 所示，视频播放类层次图如图 6 所示。

图5 音频播放类层次图

音频播放器界面由类 `Ui::Music` 实现, 视频播放器界面由类 `Ui::Video` 实现。`Ui::Music` 和 `Ui::Video` 是由界面编辑器自动生成的播放器界面中各元素的类, 该类初始化相应控件并允许其他对象访问。

3.3.2 虚拟播放器的实现

虚拟播放器由类 CPlayer 实现。在 CPlayer 类中设置一个命令列表，用以区分不同的播放命令，命令列如表 1 所示。在某一时刻，播放器可能会发出多条命令，为对播放命令进行有序管理，该模块使用了命令队列机制。在模块内部维护一个 QQQueue 命令队列对象 CmdQueue，每当收到一条命令，就将该命令加入到命令队列中进行排队。

表 1 播放命令列表

播放控制命令	播放控制功能	播放控制命令	播放控制功能
startmp3player	开启 Mplayer	seek	跳到指定位置播放
stop	停止播放	fastforward	快进
pause	暂停	filename	获取播放文件名
playprev	上一首	timelength	获取播放文件时长
playnext	下一首	timepos	获取当前播放位置
loadlist	加载播放列表	playprogress	获取当前播放进度

在接口函数内,首先对当前的播放状态进行判断,根据当前的播放状态决定下一步的操作。以处理播放按钮的接口函数 BtnPlayClicked() 为例,在函数中,首先判断当前 MPlayer 是否开启,如果没开启,则将 startmplayer 命令加入命令队列;否则再判断当前 MPlayer 是否处于播放状态,是则将 pause 命令加入命令队列,否则判断进度条是否被拖动且进度条当前位置

置跟上次播放暂停时位置是否相同,如果被拖动且两次位置不同则将命令 seek 加入到命令队列,否则将 pause 命令加入到命令队列。

播放按钮的接口函数逻辑控制流程如下:

```

Start;
定义变量 cmd;
If bMplayerRun = true Then
Do
  If Mplayer. IsPlay( ) = true Then
    cmd = pause;
  Else If bSliderDraged = true && SliderPos!= StopPos Then
    Do
      cmd = seek;
      bSliderDraged = false;
    Else Then
      Do cmd = pause;
    Else Then
      Do cmd = startmplayer;
  End If
  将命令 cmd 加入到命令队列 CmdQueue;
End;

```

从启动播放器到关闭播放器,整个期间,程序将一直等待并处理来自播放器的播放命令,为处理这一耗时操作,本模块自 QThread 派生,通过启用一个新的线程来专门处理这些播放控制命令。当队列中有命令时,从队列中出队一条命令,根据出队命令调用核心播放器相应播放控制接口进行播放;当命令队列中没有命令时,等待命令队列中有命令。命令队列机制保证了播放器的稳定性和流畅性。虚拟播放器的工作流程如图 7 所示。

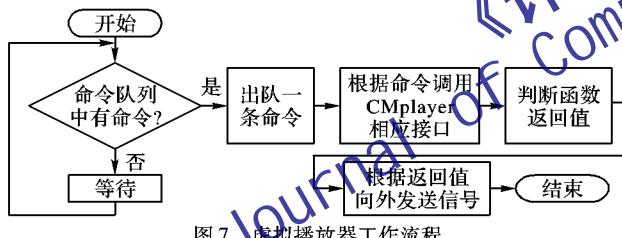


图 7 虚拟播放器工作流程

虚拟播放器模块是车载播放系统的核心模块,实际的播放逻辑控制均在此模块中实现。虚拟播放器模块将播放控制界面从播放逻辑控制中解脱出来,使得播放器图形界面层只负责 GUI 元素管理,同时也使不同的用户可根据自己的喜好设计出不同风格的播放器界面。

3.3.3 播放器内核模块的实现

播放器内核由类 CMplayer 实现,它处于播放器软件的最下层,其中封装了跨平台播放器 MPlayer,对外提供必要的播放控制接口。MPlayer 支持命令行操作模式,通过管道通信向 MPlayer 发送控制命令,从而达到对 MPlayer 的播放控制。使用 QProcess 对象来开启 Mplayer,使用 QProcess 的 Read 和 Write 方法来与 Mplayer 进行交互,实现对 Mplayer 的读和写。利用 QProcess 开启 MPlayer 的代码如下:

```

QString DefaultMplayerPath = "mplayer.exe"; // 定义 MPlayer 路径
QStringList arguments; // 声明开启 MPlayer 的参数
arguments << "-slave" << "-quiet" << "playlist.txt";
// 设置 MPlayer 启动参数,开启模式为 - slave 和 - quiet 模式,
// playlist.txt 为将要播放的播放列表文件
QProcess * pMplayerProcess = new QProcess( parent );
// 创建用于开启 MPlayer 的 QProcess 指针对象
pMplayerProcess->setProcessChannelMode( QProcess:: MergedChannels );
// 设置开启 MPlayer 的模式为 MergedChannels
pMplayerProcess->start( DefaultMplayerPath, arguments );
// 调用 QProcess 的 start 函数开启 mplayer, mplayer 路径
// 保存于变量 DefaultMplayerPath, 启动参数保存在变量 arguments 中

```

```

MergedChannels);
// 设置开启 MPlayer 的模式为 MergedChannels
pMplayerProcess->start( DefaultMplayerPath, arguments );
// 调用 QProcess 的 start 函数开启 mplayer, mplayer 路径
// 保存于变量 DefaultMplayerPath, 启动参数保存在变量 arguments 中

```

在播放器内核的接口函数中向 MPlayer 发送播放命令,实现播放控制,并从 MPlayer 读取相应数据。在读取数据时,采用超时机制,每发送一条命令,就等待读取对应的数据,收到想要数据,则返回成功,否则等待,如果等待 3 min 没有收到需要数据,则返回超时错误。图 8 为播放器内核接口函数工作流程。

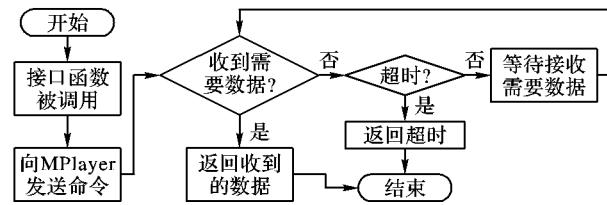


图 8 播放器内核接口函数工作流程

3.3.4 移动设备监测模块的实现

设备监测模块是通过类 CSystemMonitor 来实现的,CSystemMonitor 是从 QObject 派生而来。



图 9 设备监测消息图

在 QT 中,负责消息循环的是 QApplication 对象,在 main 函数中总是会调用 QApplication 的 exec 接口来启动消息循环,当 exec 返回时,消息循环就结束了。QApplication 是第一个接到系统消息的,随后将消息转发给顶层窗口,再由顶层窗口将消息转发到各个子窗口。

为监测硬件设备的变化,需要获取消息,并对消息进行判断,从中提取出需要的信息。为此创建一个类 CApplication,该类继承自 QApplication,让 CApplication 替代 QApplication 来接受系统消息,并对感兴趣的消息进行处理。为接收系统消息只需重载 QApplication 的虚函数 winEventFilter (MSG * msg, long * result) 即可,这个函数可获取系统消息。MSG 是 windows 的标准类型,参数 msg 中包含了消息的详细信息。在 Windows 系统中对硬件变化的监视是通过接收消息来完成的,当有硬件变化时,Windows 发送 WM_DEVICE_CHANGE 消息到窗口的消息队列。当 CApplication 类收到此消息时就向外发送 DeviceChange (Msg * msg) 信号。系统监视器只需要在此信号对应槽 DeviceChange (MSG * msg) 中接收 msg,从参数 msg 中提取出设备的详细信息,即可实现对移动设备变化的监测。

QT 是跨平台的编程环境,为不同操作系统提供了不同的接收消息的方式。对于 Windows 系统,需要覆盖 WinEventHandler 来完成对消息处理函数的自定义,而对于 Linux 系统,需要覆盖 X11EventHandler 来完成处理来自 X11 的消息的任务。如果将系统移植到 Linux 上,就不需要监视系统消息了,可以改为监视 proc 或 sys 目录,也可以得到设备更改的信息。

3.4 车载播放器效果图

车载播放器最终实现结果如图 10、11 所示,图 10 是车载播放器硬件实物图,图 11 是车载播放器的音乐播放界面截图。

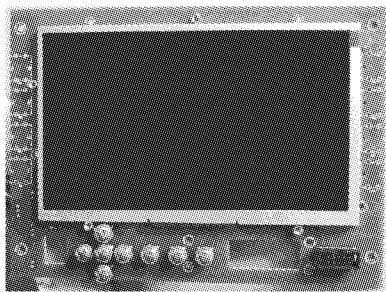


图 10 车载播放器硬件实物图



图 11 车载播放器音频播放截图

4 结语

Atom Z510 是 Intel 推出的系列嵌入式处理器之一, 搭载有最新款 SCH 控制芯片, 主要面向车载娱乐、工业控制与自动化、多媒体网络电话等嵌入式计算领域, 提供 Windows/Linux 多操作系统支持。Atom Z510 具备低功耗的特性, 支持高清, 板载 1 GB 内存, 搭载有 SCH US15WP 桥片, 提供丰富的接口, 使用户可根据需要方便的扩充自己需要的接口。Atom Z510 的这些特性, 为利用它制作一款高性能的车载播放器提供了优越条件。本文的播放器建立在 Atom Z510 基础之上, 具备图形播放界面, 操作简单便捷; 同时支持音频、视频播放,

(上接第 2573 页)

表 2 CG 协处理器的计算性能

矩阵名	计算性能/GFLOPS	实际性能占峰值的百分比/%
ash292	0.4310	47.26
rdl8001	0.6612	72.50
nasa2146	0.7676	84.17
lnsp3937	0.4905	53.78
s1rmq4m1	0.7786	85.37
nemeth24	0.7374	80.86

4 结语

本文针对 CG 迭代算法的特点, 在基于 FPGA 的可重构硬件平台上设计和实现了优化的硬件 CG 迭代求解器。有效的软硬件划分技术将 CG 代码被分割成两部分, 分别以硬件和软件的方式执行。采用 Ellpack-Itpack 格式作为矩阵存储格式, 内部电路生成行交错数据流, 提高了整个系统的计算性能。实验结果表明, 与软件实现相比, 本设计平均可以得到 2 倍左右的计算性能加速。

参考文献:

- [1] MORRIS G R, PRASANNA V K, ANDERSON R D. A hybrid approach for mapping conjugate gradient onto an FPGA augmented reconfigurable supercomputer [C]// FCCM'06: Proceedings of the 14th IEEE Symposium Field-programmable Custom Computing Machines. Washington, DC: IEEE Computer Society, 2006: 3–12.
- [2] KASBAH S J, DAMAJ I W. The jacobi method in reconfigurable hardware [C]// WCE 2007: Proceedings of the World Congress on

支持多种格式的音视频文件, 支持所有主流视频编解码标准, 支持高清, 功能强大, 具有很大的实用价值。

参考文献:

- [1] 殷建红. 国内车载娱乐信息系统发展现状及趋势 [J]. 汽车与配件, 2009(11): 40–42.
- [2] MACARIO G, TORCHIANO M, VIOLENTE M. An in-vehicle infotainment software architecture based on google android [C]// SIES '09: IEEE International Symposium on Industrial Embedded Systems. Piscataway, NJ: IEEE, 2009: 257–260.
- [3] Intel Corporation. Intel System Controller Hub [EB/OL]. [2010-01-26]. <http://download.intel.com/design/chipsets/embedded/datasheets/319537.pdf>.
- [4] 李建, 许勇, 苏平. 基于 Atom 的移动装置远程监控系统设计 [J]. 计算机系统应用, 2010, 19(4): 5–8.
- [5] 张江洪. 基于 AU6840 的车载音乐播放器系统设计 [J]. 吉首大学学报: 自然科学版, 2009, 30(2): 63–65.
- [6] 王建国, 马然. 基于 Au1200 处理器的车载多媒体电脑设计 [J]. 计算机工程, 2009, 35(23): 243–245.
- [7] 张海滨, 李挥, 吴晔, 等. 嵌入式高清播放器的设计与实现 [J]. 计算机工程与设计, 2010, 31(13): 3084–3087.
- [8] 李巍, 贾克斌. MPlayer 播放器在嵌入式平台上的应用 [J]. 测控技术, 2007, 26(增刊): 286–289.
- [9] MPlayer 工作组. MPlayer——电影播放器 [EB/OL]. [2010-10-24]. <http://www.mplayerhq.hu>.
- [10] 毕厚杰, 王健. 新一代视频压缩编码标准——H.264/AVC [M]. 2 版. 北京: 人民邮电出版社, 2009.
- [11] BICKFORD B L, BROWN A K. Vehicle audio system having random access player with play list control: US, 6185163 B1 [P]. 2001-02-06.
- [12] 梁大威, 李娟, 门爱东. 现代视频编码关键技术及其发展 [J]. 电力系统通信, 2006, 27(3): 1–4.
- [13] BUTLER N, BATOUR J-N, TUCKER D, et al. In-vehicle multimedia system: US, 7571038 B2 [P]. 2009-08-04.

Engineering. [S. l.]: Newswood Limited, 2007, 2: 823–827.

- [3] LOPEZ A R, CONSTANTINIDES G A. A high throughput FPGA-based floating point conjugate gradient implementation [C]// ARC '08: Proceedings of the 4th International Workshop on Reconfigurable Computing: Architectures, Tools and Applications, LNCS 4943. Berlin: Springer-Verlag, 2008: 75–86.
- [4] BAKOS J D, NAGAR K K. Exploiting matrix symmetry to improve FPGA-accelerated conjugate gradient [C]// FCCM 2009: 17th Annual IEEE International Symposium on Field Programmable Custom Computing Machines. Washington, DC: IEEE Computer Society, 2009: 223–226.
- [5] SUN J, PETERSON G. Sparse matrix-vector multiplication design on FPGAs [C]// FCCM 2007: 15th Annual IEEE International Symposium on Field Programmable Custom Computing Machines. Washington, DC: IEEE Computer Society, 2007: 349–352.
- [6] ZHUO L, PRASANNA V K. Scalable hybrid designs for linear algebra on reconfigurable computing systems [J]. IEEE Transactions on Computers, 2008, 57(12): 1661–1675.
- [7] ROLDAO A, CONSTANTINIDES G. A high throughput FPGA-based floating point conjugate gradient implementation for dense matrices [J]. ACM Transactions on Reconfigurable Technology and Systems, 2010, 3(1): 1–19.
- [8] WANG Y, BOYD S. Fast model predictive control using online optimization [J]. IEEE Transactions on Control Systems Technology, 2010, 18(2): 267–278.
- [9] CONSTANTINIDES G A. Tutorial paper: parallel architectures for model predictive control [C]// ECC'09: Proceedings of European Control Conference. Washington, DC: IEEE Control System Society, 2009: 138–143.