火龙果•整理
uml.org.cn

杭州電子科技大学

硕 士 学 位 论 文

题　　目：车载多媒体播放器的研究与开发

研 究 生　　尹　克
专　　业　　电 路 与 系 统
指导教师　　高 明 煜 教 授

完成日期　　2007 年 12 月

火龙果•整理
uml.org.cn

杭州电子科技大学硕士学位论文

车载多媒体播放器的研究与开发

研 究 生： 尹　　克

指导教师： 高 明 煜　教授

2007 年 12 月

火龙果•整理
uml.org.cn

Dissertation Submitted to Hangzhou Dianzi University

for the Degree of Master

The Research and Development of Vehicle Multimedia Player

Candidate: Yin Ke

Supervisor: Prof. Gao Mingyu

December，2007

火龙果•整理
uml.org.cn

杭州电子科技大学
学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：　　日期：　　年　　月　　日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：　　日期：　　年　　月　　日

指导教师签名：　　日期：　　年　　月　　日

火龙果•整理
uml.org.cn

摘　　要

汽车已经成为人们必不可少的交通工具，随着出行时间的增加，人们对新型车载信息娱乐功能的需求也日益增长；然而，车载信息娱乐市场呈现出相当大的差异性，一些国际上知名的汽车制造商已经在其概念车、高档车上安装了多媒体设备，实现了 GPS 导航、音视频播放、Internet 浏览、电子邮件收发等功能，而国内仍然是以数字调频接收机、CD 播放机等为主。因此，车载多媒体系统有广阔的开发前景和深远的经济意义。

本文设计了一种新型的车载多媒体播放器，以 ARMv5TE 结构体系的嵌入式处理器 PXA270 为核心，取代了传统的机械式 CD、VCD 播放器机芯，以可重复擦写 SD 卡代替光盘等传统存储设备，并采用 800*480 分辨率的数字 LCD 屏显示影视图像，使用触摸屏配合图形界面进行人机交互。另外，设计了以单片机 MEGA48 为控制核心的音频输出模块，可选配无线调频发射芯片 RS300A，连接车载原配音频功放系统，获得较好的音质欣赏效果；或选配无线调频芯片 TEA5767，实现 FM 接收功能。

多媒体播放器的应用程序研究与开发是本文的重点。该软件运行在 Windows CE.NET 嵌入式操作系统之上，使用安装了定制 SDK 开发包的 Embedded Visual C++环境进行编写、调试。采用 DirectShow 技术和模块化设计的思想，将播放器的控制与功能分离，分别设计了系统主控程序、图形用户界面和音视频解码器模块。系统主控程序是多媒体播放器主要的控制模块，使用事件-消息机制控制图形界面的显示状态，创建 Filter Graph 连接各个音视频解码模块，具有良好的通用性和扩展性。为实现 MPEG-4 视频文件的流畅播放，使用 DirectShow Filter 技术移植了 XviD 视频解码算法，实现了 Windows CE.NET 系统下使用的解码器 XidDFilter。最后对多媒体播放器进行了功能测试和解码性能分析，并使用 MMX 技术和高级语言编程技巧对解码器进行了优化，最后给出了测试结果。测试证明，多媒体播放器长时间工作性能稳定，能够流程播放 MPEG-4 格式的影音文件。

关键词：多媒体，播放器，嵌入式系统，应用程序设计

火龙果•整理
uml.org.cn

ABSTRACT

At present, automobile has become the most important transport. People spend more and more time on the trip. Thus the demands for new kinds of vehicle information and entertainment equipments grow day by day. However the development of vehicle information and entertainment markets is uneven from places to place. Some international famouse manufacturers have fit multimedia equipments on their concept cars and luxury vehicle, which provide functions such as GPS navigation, media player, Internet brower, Email and so on. At the same time, the main domestic entertainment devices are still FM receivers and CD plyers. Therefore, automotive multimedia systems have broad prospects of development and far-reaching economic significance.

This paper designs a new kind of multimedia player used in vehicle. It adopts high performance, lower power SOC PXA270 that based on ARMv5TE architecture as its core processor, instead of traditional mechamical CD and VCD player. And it chose SD Card to replace traditional memory. The new kind of multimedia player has a 800*480 pixels LCD with a touchscreen and a friendly GUI interface. Further more, it has an audio module, which used MEGA48 as its controller. The audio module with a low-power FM transmitter IC RS300A to send FM signals. The original-packaged audio amplifier can brocast these audio signals and perform beautilfully. Under the same controller MEAG48, theIC RS300A can be repleaced by FM transmitter IC TEA5767.Thus this module become a FM stereo radio.

The research and development of mutimedia player's application program is the important part of the paper. This program runs on the Windows CE.NET embedded oprating system, and it was made and debugged under the Embedded Visual C++ environment with a special SDK. Adopting the DirectShow technology and the modularization idea, the application is made up of the system manager program, GUI, and audio and vedio decoders. The system manager program is the main contol module of the player. It control the appreance of the GUI interface, create Filter Graph and manager the audio and video decoders. In order to play MPEG-4 video files fluently, the parper discribe how to transplant the Open source XviD vedio decoding algorithm and make out the XidDFilter with DirectShow Filter technology. In last part of this paper it displays function tests of the multimedia player and analyzes the decoder's capability. Then it optimized the MPEG-4 video decoder with MMX technology and

火龙果•整理
uml.org.cn

advanced Language programming skills. Finaly it shows the results of the perfomance tests.

These experiments show that, the multimedia player can run steadily and dependablely. It can play MPEG-4 format vedio fluently and fulfill the request.

Keywords: Multimedia, player, embedded system, application program design

火龙果•整理
uml.org.cn

目　录

摘　　要.....I
ABSTRACT.....II
第一章 绪　论.....1
1.1 课题研究的背景及意义.....1
1.2 车载多媒体系统的研究现状.....2
1.3 车载多媒体播放的发展趋势.....2
1.4 课题的研究内容.....3
1.5 论文内容安排.....3
第二章 车载多媒体播放终端的系统构建.....5
2.1 硬件系统设计.....5
2.1.1 方案比较.....6
2.1.2 系统实现.....6
2.1.3 Intel PXA270 处理器.....7
2.1.4 无线调频发射/接收模块.....9
2.2 Windows CE 操作系统的定制.....12
2.2.1 Windows CE.NET 的结构和特点.....13
2.2.2 Windows CE.NET 平台的开发工具.....14
2.2.3 Windows CE.NET 的定制流程.....15
2.2.4 Windows CE.NET 定制过程中重要组件的配置.....16
2.2.5 多媒体播放器应用程序开发环境的配置.....21
2.3 小结.....22
第三章 多媒体播放器应用程序的设计与实现.....23
3.1 多媒体播放器的设计方法.....23
3.1.1 DirectShow 技术概述.....23
3.1.2 多媒体播放器的模块化设计.....26
3.2 系统主控模块设计.....28
3.2.1 功能模块的加载/卸载.....28
3.2.2 播放器工作状态分析.....30
3.2.3 消息循环的维护.....31
3.3 用户图形界面设计.....33
3.3.1 图形界面的结构设计.....33

火龙果•整理
uml.org.cn

3.3.2 图形界面的程序要点 34
3.4 解码器模块的实现 34
3.4.1 MPEG-4 视频 Filter 的功能分析[26] 35
3.4.2 XvidFilter 的设计 36
3.5 小结 41
第四章 多媒体播放器的测试与优化 42
4.1 功能测试 42
4.1.1 开机功能测试 42
4.1.2 无线调频发射功能测试 43
4.1.3 无线调频接收功能测试 43
4.1.4 系统设置功能测试 44
4.1.5 多媒体播放功能测试 44
4.1.6 测试小结 45
4.2 性能测试与分析 45
4.2.1.性能测试结果 46
4.2.2 多媒体播放性能测试 46
4.3 性能优化的方法 48
4.3.1 针对 PXA270 平台的 MMX 指令优化 48
4.3.2 程序的 C 语言优化 51
4.3.3 优化结果 52
4.4 小结 53
第五章 总结与展望 54
5.1 已完成的任务及亟待解决的问题 54
5.2 发展前景 54
致 谢 55
参 考 文 献 56
附 录 59

火龙果•整理
uml.org.cn

第一章 绪 论

1.1 课题研究的背景及意义

汽车已经成为人类不可缺少的交通工具。根据美国交通部 (USA Department Of Transportation)的数据显示，经常往返于北美地区的人们每周大约会在车上度过的时间共计 5 亿个小时，Entertainment on the Go 的理念迅速在汽车业中蔓延开来。据 Strategy Analytics 研究表明，汽车市场对新型车载信息娱乐功能的需求日益增长，到 2012 年底其市场规模将高达 360 亿美元。就市场成长来看，后座影音娱乐系统市场规模在 2005 年大约为 500 万台，成长率达 30%; 车载电视市场 2005 年达 170 万台，成长率达 24%[1]。

汽车的多功能化和电子化为汽车电子产业发展带来广阔空间。1999 年日本企业推出以 DVD 为主的汽车 AV 系统。进入新世纪以来，随着汽车工业与电子信息产业加速融合，汽车开始向电子化、多媒体化和智能化方向发展，从以机械产品为主，向高级的机电一体化产品方向演变。目前，车载信息娱乐系统已成为汽车电子领域最具吸引力且成长最快的产品之一。当前，高、中档轿车中的电子设备配置都在向豪华型、个性化方向发展，主流配置可大致归纳为三大类：音频(如 CD)、视频(如 DVD)以及 Navigation(导航)和 Telematics(远程信息处理)，并有形成多媒体中心整合所有娱乐系统的趋势。

目前，在全球不同地区，车载信息娱乐市场呈现相当大的差异性。从国际趋势来看，HD-CD、汽车 VCD、汽车 DVD、MP3 等正在逐渐成为主流产品，而国内仍然是以数字调谐收录机、CD 等为主。国际上知名的汽车制造商已经在其概念车、高档车上安装了多媒体设备，主要实现 GPS、视频播放、Internet 浏览、电子信息收发等功能。例如，别克荣御和奥迪 A6 商务车安装的德国蓝宝车载多媒体系统、2007 款三菱欧兰德原车选装的美国来福汽车音响；在北美，车载 DVD 播放器和广播视频通常安装在汽车后座供乘客享用，而在日本，前座地面电视接口则更为普遍，供驾驶员和前座乘客在交通高峰时收看[2]。

国内汽车制造厂商的技术主要来源于国外现有的成型技术，缺乏自主知识产权；由于成本因素，多媒体影音播放设备尚未纳入汽车的标准配置。因此，国产车辆在出厂时没有类似的车载多媒体播放设备。而收音机、CD 等音响设备的原厂配套市场也已集中在几大汽车音响巨头手中，如西门子、飞利浦、江苏天宝、深圳航盛等在国内汽车原厂配套市场的占有率超过 75%。目前，国内家电企业积

火龙果•整理
uml.org.cn

极拓展车载多媒体市场。但由于车载影音设备作为汽车的重要装饰配件，是必须与整车设计相配合的。国内家电厂商拓展车载影音市场取得了一些成就，但由于起步晚，没有形成强势品牌，尚不足以对国外品牌构成实质性威胁。

1.2 车载多媒体系统的研究现状[2]-[4]

面对巨大的市场需求，各个半导体厂商纷纷推出了各自特色的硬件平台。车载多媒体领域老牌半导体供应商有 ADI、Freescale、NXP（前身是 PHILIPS 半导体）等，它们凭借各自优势带来了独具特色的解决方案。

ADI 推出的 Blackfin 处理器 ADSP-BF539F，在同一封装内集成了闪存(512KB 或 1MB)；支持 500MHz 处理能力，可提供 CAN 和 MOST 总线连接，适用于需要大量存储的音视频、会话语音识别等应用。目前，汽车巨头沃尔沃公司已经认证了 ADSP-BF539F 的 MXVR 接口并支持其 MOST 网络，符合该公司为降低总系统成本而采用的高集成度解决方案开发车载信息娱乐系统的策略。

Freescale 推出了以 MPC5200B 处理器为核心的 Media5200 开发平台提供一个迅速开发各种新款多媒体与信息娱乐产品的系统，该处理器专门为车内信息娱乐与通讯系统所设计，处理能力达 885MIPS，支持音频压缩解码/编码以及视频解码，可在单核内同时实现前排 GPS 导航和后排娱乐功能。NXP 半导体（前身是飞利浦半导体）推出了车载视频处理器 Nexperia PNX9520 支持多重标准音视频解码、数字收音机以及先进的连接能力。它支持的视频解码标准包括 MPEG-1、MPEG-2 和 MPEG-4、Windows Media Video、H.264 及 JPEG 图像显示，它还支持面向高端画质和 MPEG 的 DVB-T 数字广播标准、AC3(杜比数字)和 Window Media Audio 解码标准。

2006 年，NXP 半导体推出的车载视频处理器 Nexperia PNX9520 支持多重标准音视频解码、数字收音机以及先进的连接能力。它支持的视频解码标准包括 MPEG-1、MPEG-2 和 MPEG-4、Windows Media Video、H.264 及 JPEG 图像显示，它还支持面向高端画质和 MPEG 的 DVB-T 数字广播标准、AC3 和 Window Media Audio 解码标准。

1.3 车载多媒体播放的发展趋势

（1）功能多样化。车载多媒体系统可以为驾驶员提供一系列服务，表现在增加语音声控、影音娱乐、GPS 导航导航服务等功能。市场调查表明功能越多越容易受到用户的欢迎。

（2）多元化信息服务。显示塞车信息，通知天气预报等信息的服务将逐渐成

火龙果•整理
uml.org.cn

为可能，甚至能与办公室和家里进行通信联络，开启家中的照明灯，启动自家的喷灌系统，控制办公室安全系统等将不在是梦想。汽车多媒体播放产品的未来发展，多元化的信息服务是其中一个必然的趋势。

（3）体积更小，携带安装方便，适用多种车型的汽车多媒体播放产品呈增长趋势。

（4）高新科技的运用。如今一些汽车电子产品都可以看到高新技术的影子，如索尼推出的影像导系统运用了三维技术。

未来的车载影音播放系统在性能上将更接近家用影音系统，集成音视频、导航器、远程信息处理为一体，能够支持以往由地面波发送的收音机和电视广播、导航系统使用的卫星信息，卫星数字广播发送的数字媒体等，兼容更多的技术标准和接口，并将导入宽带车内网络。在 HDD 等存储设备广泛普及，与内容有关的数字著作权管理方法成熟之后，多媒体信息可以通过流媒体的方式将被广泛使用。车载信息系统的应用将更加广泛应用无线传输技术，电子地图、自主导航、汽车电话、汽车防盗、行车监控、车辆派遣、行动上网、收发传真、影音服务等，都可通过良好的无线通信组件来实现。随着人们活动范围的扩大，车辆将由单纯的交通工具将集成更多功能的无缝移动信息系统。

1.4 课题的研究内容

本文为实现一种新型的车载多媒体播放终端为设计目标：以功能强大的通用处理器取代专用的播放器机芯，以小巧、可重复擦写 SD 卡可重复擦写存储卡代替光盘等存储设备；基于车载的使用环境，以视频/音频播放为核心功能，在性能上接近家用影音系统，能够提兼容可能的多音视频文件格式；具有丰富的外设接口，能够进行软件升级。

本课题完成的研究工作主要有：

（1）硬件平台系统设计与规划。

（2）嵌入式操作系统定制。

（3）嵌入式应用程序的开发，媒体播放器功能的实现。

（4）多媒体播放功能测试与性能优化。

1.5 论文内容安排

本论文共分五章。

第一章 绪论 本章主要介绍课题立项的背景，国内外发展状况及趋势，本课题的研究内容和作者的工作。

火龙果•整理
uml.org.cn

第二章 系统平台的设计 本章主要从硬件系统和软件系统两方面进行阐述。首先介绍了本课题的硬件系统设计，主要说明了 PXA270 适于多媒体处理的优势和无线调频发射模块的巧妙设计；然后定制了 Windows CE.NET 操作系统，重点阐述了内核定制的关键步骤，最后介绍了嵌入式应用程序开发环境 EVC 的配置。

第三章 多媒体播放器应用程序的设计 本章主要介绍了基于 DirectShow 技术的应用程序开发，并引入模块化设计，将播放器应用程序的设计分为三部分，分别介绍了系统管理模块、美观实用的图像界面的设计与实现，并使用 DirectShow 技术将开源软件 XviD 的视频核心算法，实现了本系统可用的 MPEG-4 视频解码器。

第四章 播放优化与测试 本章首先对播放器进行了功能测试，评测了多媒体播放器播放 MPEG-4 音视频文件的性能，然后针对硬件平台 PXA270 处理器的特性，使用 MMX 技术和 C 语言编程技巧对解码器 XvidFilter 进行了算法优化和程序优化，并给出了优化结果。

第五章 系统待解决的问题和发展前景，总结本论文所作的工作。

火龙果•整理
uml.org.cn

第二章 车载多媒体播放终端的系统构建

车载多媒体播放终端的设计目标是：基于车载的使用环境，在性能上接近家用影音系统，能够兼容多种音视频文件格式；具有丰富的外设接口，能够通过外接设备和软件升级。本章以系统的角度着眼，主要从硬件设计和软件系统设计两方面进行阐述。

2.1 硬件系统设计

车载多媒体播放终端方案应考虑以下要素[5]-[8]

（1）处理能力和支持的音视频格式。多媒体播放功能的实现要求系统能够兼容尽可能多的主流音视频文件格式，支持各种视频压缩格式，如 MPEG-1、MPEG-2、MPEG-4、VC-1、H.264MJPEG 等国际标准，也支持 RM、QUICK TIME、DivX、XviD 等不同公司的标准；支持 MP3、AAC、DOLBY、DTS 等音频压缩格式。音视频压缩算法工作量巨大，要求系统有足够高的处理能力，能够承担起运算量的要求。

（2）系统功耗。处理器主频越高处理能力越强，功耗也越大。尤其是依赖软件解码的系统扩展能力强，但功耗也越大。同时，考虑到系统内不同组件的工作电压不同，还涉及到电源管理的问题。

（3）硬件系统对操作系统和软件的支持。软件系统设计是对消费类电子产品的最大挑战，也是提升其附加价值的关键所在。但对于不同体系结构的硬件，软件系统的开发成本和难易程度是有区别的。如能缩短软件开发时间，就可以加快产品上市时间，增强产品的竞争力。

（4）外部接口。车载终端的外围接口应包括存储器接口、USB 接口、存储卡接口、摄像头接口、硬盘接口、显示接口等。丰富接口对系统调试、功能的扩展和产品更新换代意义重大。另外，随着近年来 ITS 的发展，以 3G（GPS、GIS 和 GSM）为代表的新型电子通讯产品的出现，它对汽车的综合布线和信息的共享交互提出了更高的要求，能够提供丰富的接口对车载终端设备来说至关重要。

（5）扩展能力。考虑到电子产品更新换代快、研发成本高等因素要求系统具有通用性，能够进行快速升级。

火龙果•整理
uml.org.cn

2.1.1 方案比较

基于市场的巨大潜力，老牌的半导体供应商凭借各自优势带来了独具特色的解决方案，由于其处理器源自不同的架构，所以在性能、接口等方面差异很大，以下分别进行介绍和比较。

第一类是基于通用处理器的方案，Intel 的 PXA27x 和 AMD 的 AU1200，SANSUNG 的 2410 是智能手持设备市场上用的最广泛的处理器。通用型嵌入式处理器的性能强劲，支持丰富的音视频格式，包括 AVI、MPEG4、WMV9、MP3、WMA、ASF、MJPEG 等。由于依赖软件解码，优点是扩展性强，允许设计者根据需要增加软件编解码器，缺点是功耗较大[2]。

第二类是基于 DSP+ MCU 的双核方案，如 TI 的 DM270/DM320、NXP 的 Nexperia PNX9520 和 ADI 的 Blackfin 系列 ADSP-BF539F。音视频解码是 DSP 的专长，因此，该种方案的优点是能够支持更高复杂度的视频编解码器，如 H.264；并且能够通过软件升级来支持新的编解码器，但需要另配一个 MCU(ARM7 或 ARM9)来处理系统的通用功能，如操作系统和用户接口等。这样就涉及到双核设计的复杂度和敏感的成本问题[8][9]。

第三类是基于专用处理器的方案，如 Freescale 的基于 ARM11 架构，并集成了多媒体加速器 eMMA 的 i.MX31 和 i.MX31L 处理器；安凯的 AK3220M 芯片等。由于是基于硬件实现音视频的编解码的，因此这类方案支持的音视频格式种类受到限制，如 Freescale 支持 MPEG4、H.263 的编解码，但不支持 MPEG-2。

另外，还有 FPGA+DSP 方案，FPGA 使用 Alter 的 NiosⅡ软核技术、独有的 C2H 优化以及硬件的可定制的特性，使用 DSP 作为 GPU 将发挥处理强大的图像数据处理能力，优势是显而易见的，但系统定制的灵活性大，研发难度大，功耗也较大。

经过综合比较，车载多媒体播放器的设计选择第一类方案，以通用型的处理器 PXA270 为核心，并扩展丰富的外设接口，可灵活配置外围设备增减功能，满足不同用户的需求。

2.1.2 系统实现

本课题最终实现的硬件平台以 PXA270 嵌入式理器为核心，可读取 SD 卡中存储的多媒体文件，流畅播放 MPEG-4 音视频文件，并集成了触摸屏输入、USB 主控制器、电源管理等功能。还配有无线调频发射模块，用于连接车内原配功放系统，进行音频输出，在此模块的基础上还可选配无线调频电路，实现 FM 接收功能。系统硬件结构如图 2.1 所示，器件选型如下：

- 处理器：Intel PXA270；

火龙果•整理
uml.org.cn

- 存储器：NOR FLASH（存储只读属性的映象文件）Intel E28F128J3A；
  NAND FALSH（存储大容量的文件）Samsung K9F1208；
  SDRAM（系统初始化之后程序的运行空间及堆栈区）两片 Hynix HY57V561620；
- 视频输出：7 寸 800*480 像素 TFT LCD 屏幕（16：9 宽屏）；
- 音频输出：混合信号音频解码/编码芯片 UCB1400，单片机 MEGA48，
  立体声发射芯片 RS300A；
- 用户输入：触摸屏，USB 鼠标（调试用）；
- 数据通信接口：串口，SD 卡接口，100M 以太网口，主 USB 接口，
  从 USB 接口，JTAG（调试用）；
- 选配模块：单片数字调谐调频接收芯片 TEA5767

系统硬件结构如图 2.1 所示。

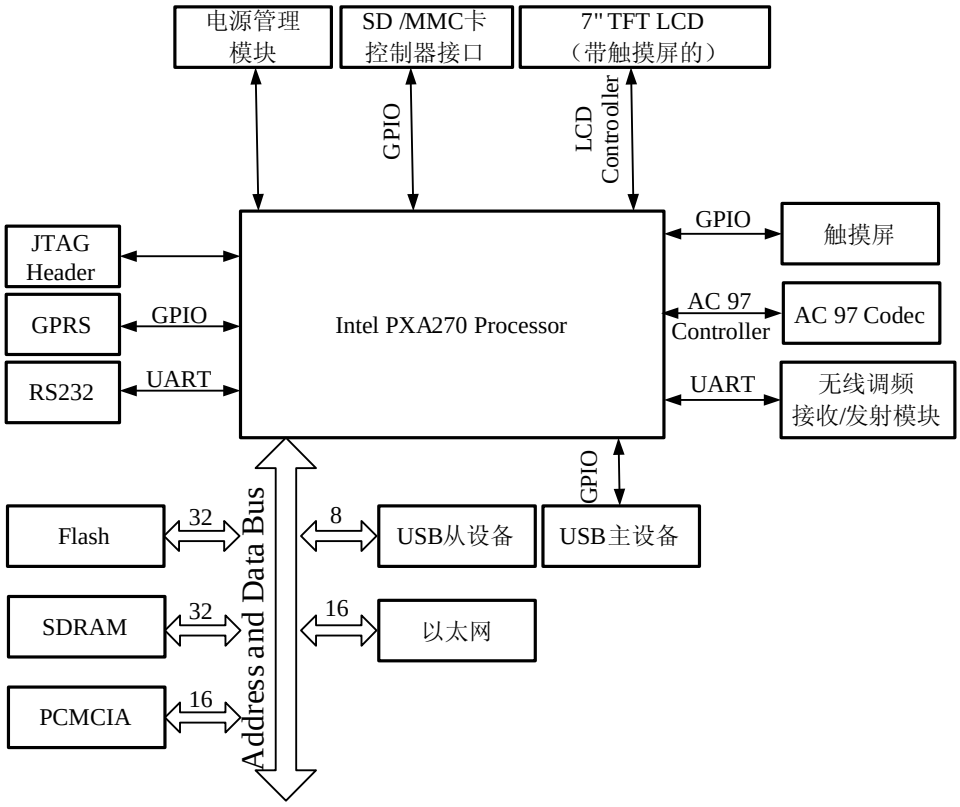

图 2.1 系统硬件结构框图

2.1.3 Intel PXA270 处理器[10]-[13]

在 2006 年之前，基于 Xscale 技术的处理器，Intel 公司发布并广泛投入使用的有 PXA25x、PXA26x、PXA27x（代码 Bulverde）三个系列；在 2006 年 10 月，Marvell 公司在收购 Intel 移动通信芯片业务，发布了第三代基于 Intel XScale 技术

火龙果•整理
uml.org.cn

的应用处理器 PXA3xx(内部称为 Monahans)，目前实现批量生产的第一种处理器是 PXA320，其最高主频为 1.2GHz，并可调整至 806MHz。使用 Marvell PXA3xx 系列处理器需要与 Marvell 公司签订保密协议，该系统处理器可供参考资料较少，研发难度较大。

本课题设计的硬件系统以 PXA270 处理器为核心，该处理器采用 Intel Xscale 体系结构，兼容 ARMv5TE ISA 指令集(不支持浮点指令集)，支持 16 位 Thumb 指令和 DSP 扩充。它是 Intel 针对高端便携式手持设备及工业设备推出的一款高性能、低功耗、功能强大的嵌入式 SOC 微处理器产品，最高主频达 624MHz。另外，与其他同类芯片相比，PXA27x 系列处理器在多媒体处理能力和功耗方面有着非常大的优势，因为它具有三项重要的技术特点，即 Quick Capture（快速拍摄）、Wireless Speed Step（无线动态节能）和 Wireless MMX（无线多媒体扩展技术）。

- Quick Capture

Quick Capture 技术为成像设备（摄像头）与无线设备（主机）提供接口，包括快速浏览、快速拍照和快速视频拍摄三种操作模式。该技术使得基于 PXA270 的成像系统能够拍摄高达 400 万象素的图像和视频，并支持低功耗、实时的回放处理，最大可提供 416Mbps 的数据传输速率。另外使用该技术，与支持 24 位色的 LCD 显示和 256KB 的片上 SRAM 帧缓冲，可加速图像回放。

- Wireless Speed Step

支持 Intel 独有的无线 Speed Step 动态电源管理技术，使处理器根据系统需要，自动切换工作频率和电压，将 PXA270 的主频在 26MHz～624MHz（最高）之间自由调节。该技术的精髓在于处理器内部的 Speed Step 控制器可以随时进行电压和频率的调节，当控制器侦测到电源模式的改变时就向操作系统发出中断请求，系统响应请求执行中断后重新定义控制器状态并将 CPU 休眠，然后 Speed Step 控制器调节电压调节器并改变 CPU 的倍频，唤醒 CPU 以新的电压和主频开始工作。这个过程少于 1ms，用户工作无需任何停顿。而多个频率断点的优势在于可以让处理器根据应用程序选择所需的运算能力，在性能和功耗间取得最理想的折衷。根测算，完全空闲时 PXA270 运行在 26MHz 的主频下，其电流将低于 0.1mA。与 PXA263 相比，PXA270 播放 MP3 音频的时间可约延长 42%，播放 QVGA 的视频时间约延长 77%。

- Wireless MMX

这种技术起源于桌面 PC 处理器，MMX 是 Intel 公司为增强 Pentium CPU 在音像、图形和通信应用方面而采取的新技术。采用这一技术的 PXA270 处理器，可使用专为音视频信号以及图像处理而设计的 57 条指令，极大地提高了处理器对多媒体文件的处理能力。相对于同频率的芯片，其处理多媒体的效率约提高了 60％左右，即使运行不使用 MMX 指令的程序，也能获得 15％左右的性能提升。

火龙果•整理
uml.org.cn

此外，PXA270 处理器具有丰富的外围接口，如 USB Host/Client、USB OTG、4 位的 SD I/O、MMC/SD 卡、CMOS/CCD、OTG、IDE、LAN、Memory Stick、USIM 卡接口、Keypad 控制器等。配合嵌入式 Linux 或 WinCE 操作系统，以 PXA270 处理器为核心的多媒体播放平台，在理论上可通过软件升级支持任何媒体格式。由于此平台能够支持外设，可连接多种模块和传感器实现行驶信息系统的复杂功能，大大简化了系统设计。

下图为 PXA270 处理器内部结构框图。

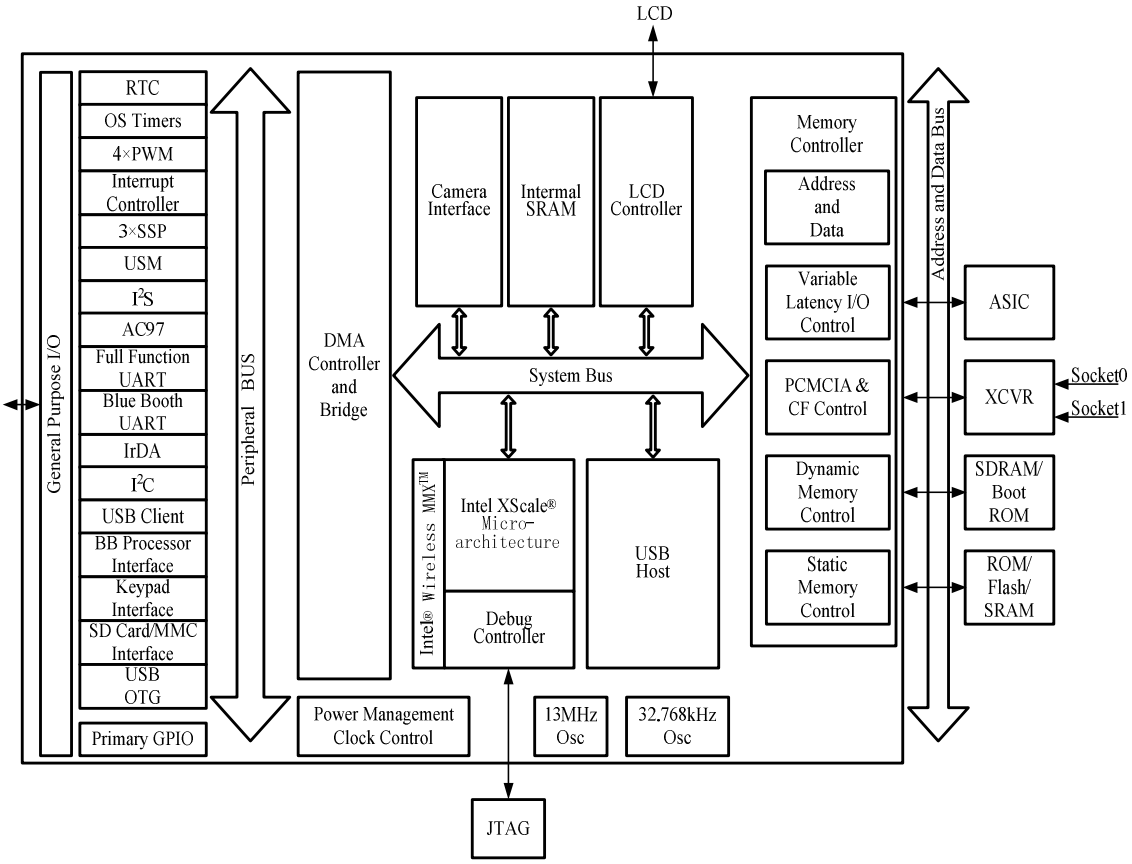

图 2.2 PXA270 处理器内部结构框图

2.1.4 无线调频发射/接收模块

由于体积和功耗的限制，多媒体播放器只可配置功率较小的扬声器，因此在车载环境下使用效果不佳。本文提出了一种巧妙的解决方法：设计一个无线调频发射模块，将系统音频以无线调频信号的形式送出，使用车载收音机接收，再通过汽车原配的功放输出，就可获得高质量的收听效果。

无线调频接收模块以单片机 MEGA48 为核心，通过串口接收 AT 命令，实现与嵌入式处理器之间的双向通信；由立体声发射芯片 RS300A 通过两根音频信号线，接收系统输出的音频信号，并负责调制发射。结构原理如图 2.3 所示。

火龙果•整理
uml.org.cn

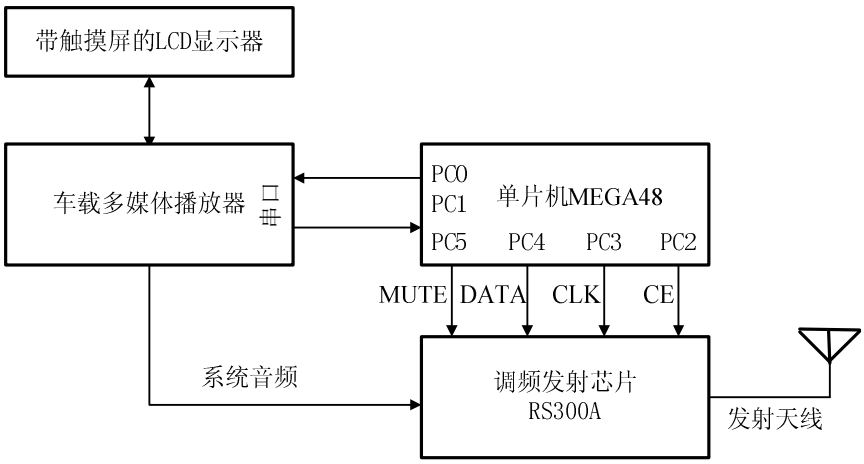

图 2.3　无线调频发射模块原理框图

以单片机为核心设计无线调频发射模块，这一设计思路十分巧妙。使用这种方法不仅大大降低了嵌入式操作系统底层驱动软件的开发难度，而且提供了一种灵活配置多媒体播放器系统外接设备，扩展系统功能的方法。在此模块的基础上，以立体声接收芯片 TEA5767 取代调频发射芯片 RS300A，改编 MEGA48 的控制程序，就可为没有配置收音机的车辆提供 FM 接收功能。结构原理如图 2.4 所示。

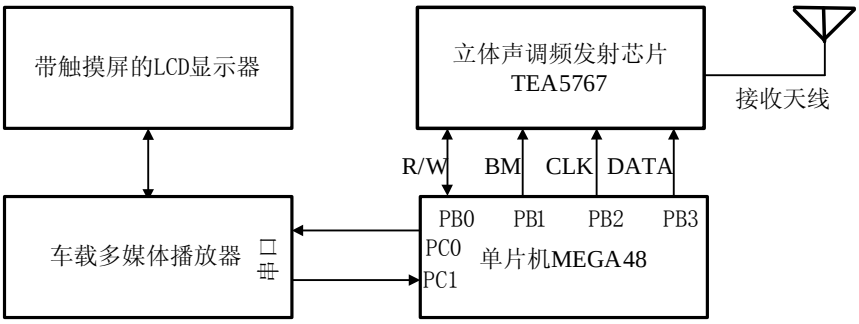

图 2.4　无线调频接收模块原理框图

2.1.4.1 无线调频发射模块的设计

无线调频发射模块的控制核心 MEGA48 是 Atmel 公司生产的 8 位微控制器，采用 RISC 精简指令集，运行速度快，性能稳定。MEGA48 在 1.8V～5.5V 电压范围内均能正常工作，片内自带 4KB 的 FLASH，256B 的 E^2PROM 以及 512B 的 SRAM；内置 6～8 路 10 位 A/D 转换器、看门狗、3 个 16 位定时/计数器、具有独立振荡器的实时技术器 RTC 和 6 路 PMW 输出，拥有五种休眠模式，引脚及中断可唤醒 MCU，具有高性能、低功耗的特点。

音频信号的调制与发射由立体声发射芯片 RS300A 完成，它的音频输入范围是 20-15kHz，调频发射频率范围为 70-120MHz，因此目前汽车上配置的调频收音机均能接收。RS300A 芯片的 21 脚和 22 脚接收多媒体播放器输出的音频立体声

火龙果•整理
uml.org.cn

信号，14-17 脚分别接单片机 PC2-PC5 口，9 脚为射频信号输出端，通过电容 C134 耦合到发射天线，电路如 2.5 所示。单片机 MEGA48 通过 SPI 接口配置 RS300A 的相关寄存器，设置调频信号的发射频率，控制程序简单不再叙述。

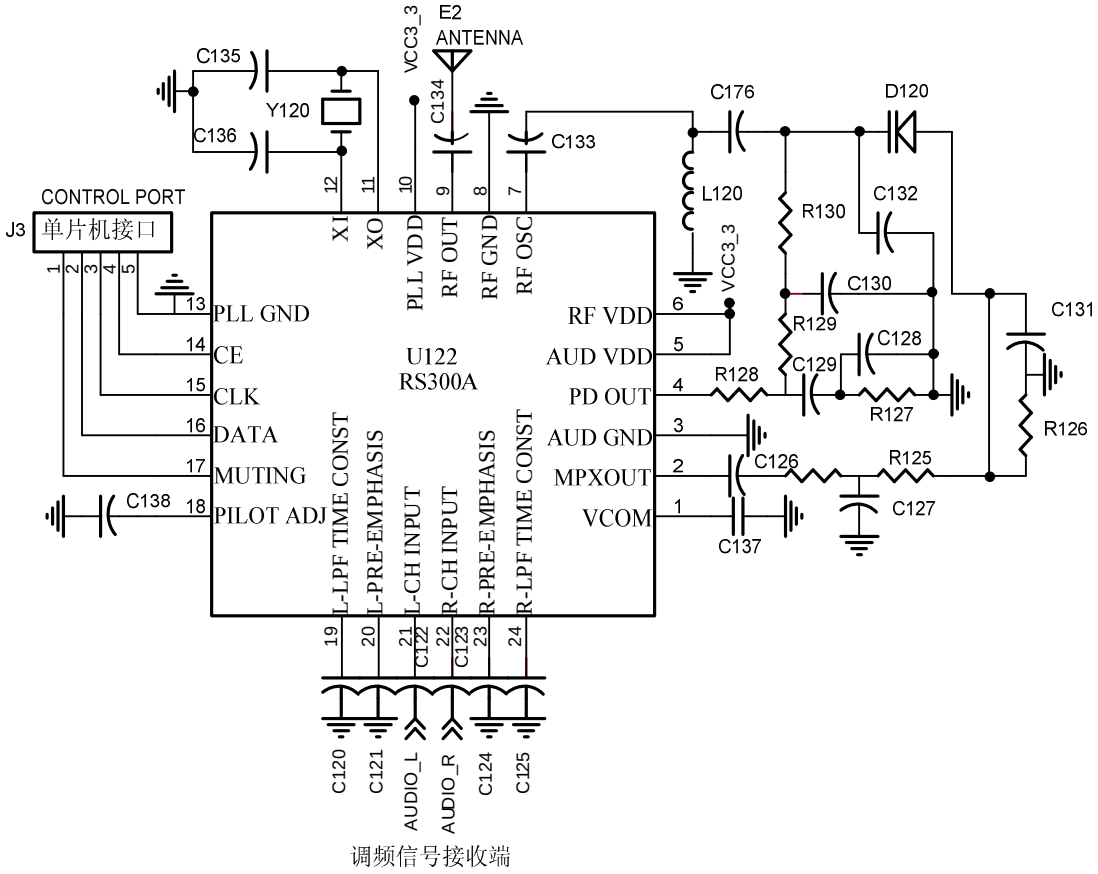

图 2.5　无线调频接收模块电路

2.1.4.2 无线调频接收模块设计

无线调频接收部分电路如图 2.6 所示，由单片机 MEGA48 控制，采用 Philips 公司的 TEA5767 实现调频接收功能。TEA5767 芯片 2、3 脚为立体声音频输出端，6-9 脚为串行通信控制端，连接到单片机 MEGA48 的 PB0-PB3 端口。

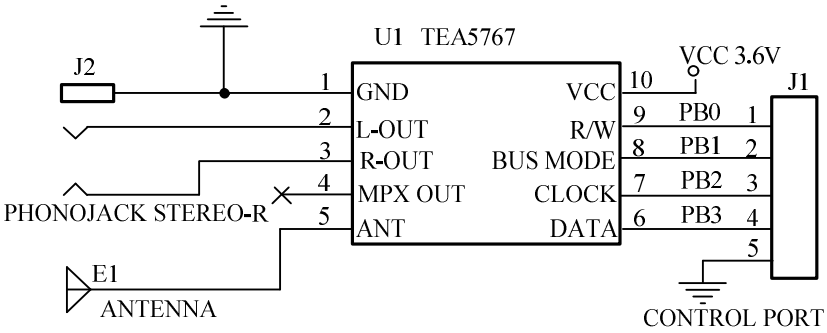

图 2.6　无线调频接收模块电路

火龙果•整理
uml.org.cn

用户在 LCD 触摸屏界面上设定的电台接收频率设置、音量大小等设置，或选择频道搜索方式（手动搜索/自动搜索），嵌入式处理器将用户指令以进行处理，通过串口传送至 MEGA48，再由 MEGA48 通过 SPI 接口配置 RS300A 相应的寄存器，进行响应[14]-[15]。

无线调频接收模块的单片机控制程序由三部分组成，分别是主程序、预定义文件、芯片配置文件。预定义文件约定了字符串指令转化格式，增强了代码可读性；芯片配置文件定义了 TEA5767 的字符转换方式和寄存器配置。无线接收模块部分，MEGA48 的主控程序部分流程如图 2.7 所示：

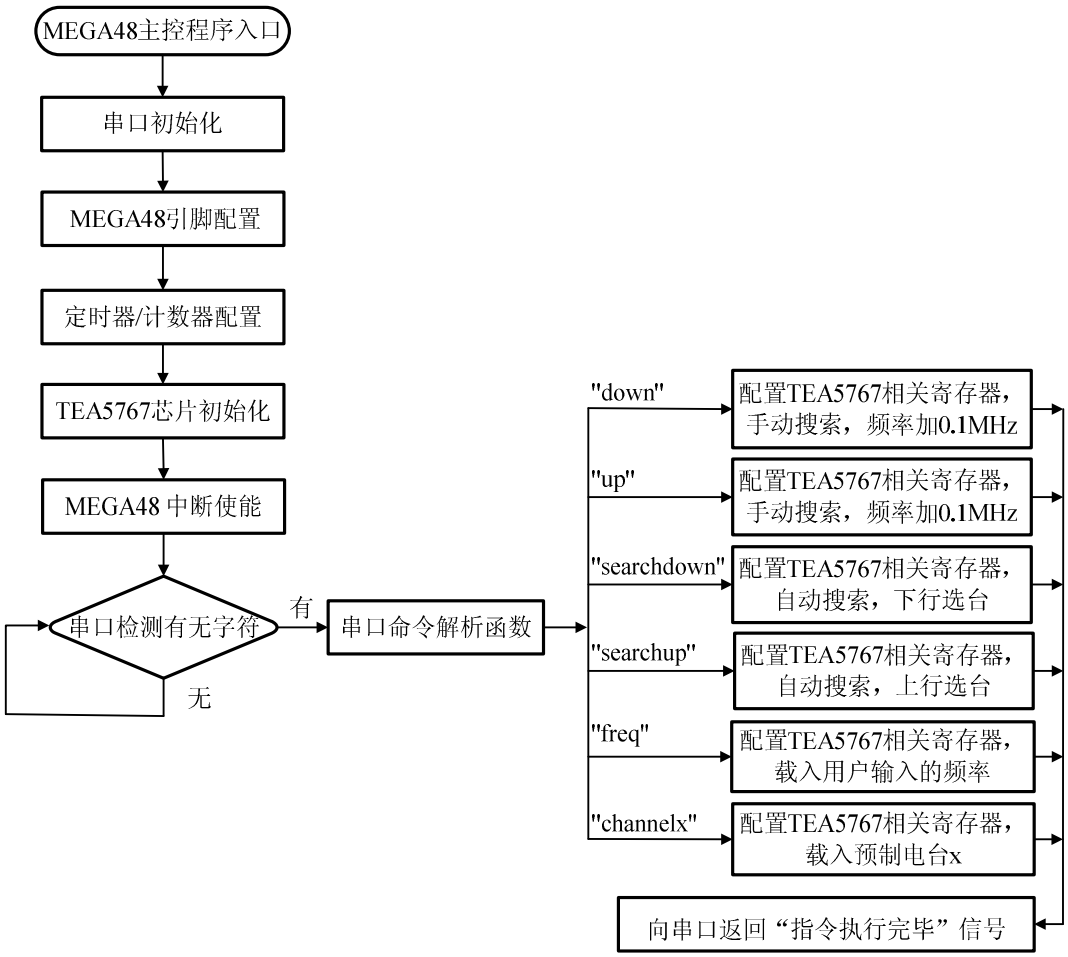

图 2.7 MEGA48 的主控程序部分流程图

2.2 Windows CE 操作系统的定制

传统的软件开发一般采用“本地”的方式，即用本机开发、本机调试和本机运行。但由于嵌入式系统的资源有限，直接在开发板的硬件系统上进行调试十分困难，因此嵌入式开发一般采用交叉编译的模式，即“宿主机—目标机”的开发模式，由功能强大、资源丰富的 PC 担任宿主机。宿主机的体系架构与目标板完全不同，因此必须完成以下几个步骤：

火龙果•整理
uml.org.cn

（1）在宿主机上建立交叉编译环境；

（2）在宿主机上完成目标板操作系统内核编译、应用程序开发、调试等；

（3）通过 JTAG 口或以太网接口，将宿主机编译连接生成的可执行文件下载到目标板上运行、调试；

（4）调试成功后，将应用程序存储到 SD 卡中，或重新编译操作系统内核，将应用程序固化到目标板的 Flash 中。

2.2.1 Windows CE.NET 的结构和特点[16]-[19]

Microsoft Windows CE.NET 是一个紧凑、高效和可扩展的嵌入式操作系统，是 Windows CE 3.0 的后继产品，专为快速建立智能移动设备和小内存占用设备提供健壮的 32 位、多线程、多任务的实时操作系统。Window CE.NET 采用了层次化的体系结构，具有可定制性、组件可裁剪、强调编程接口和支持上层应用等特点，自下而上分为四层：嵌入式系统分为四个层次，分别是硬件层、OEM 层、操作系统层和应用程序层。Windows CE.NET 结构如图 2.8 所示：

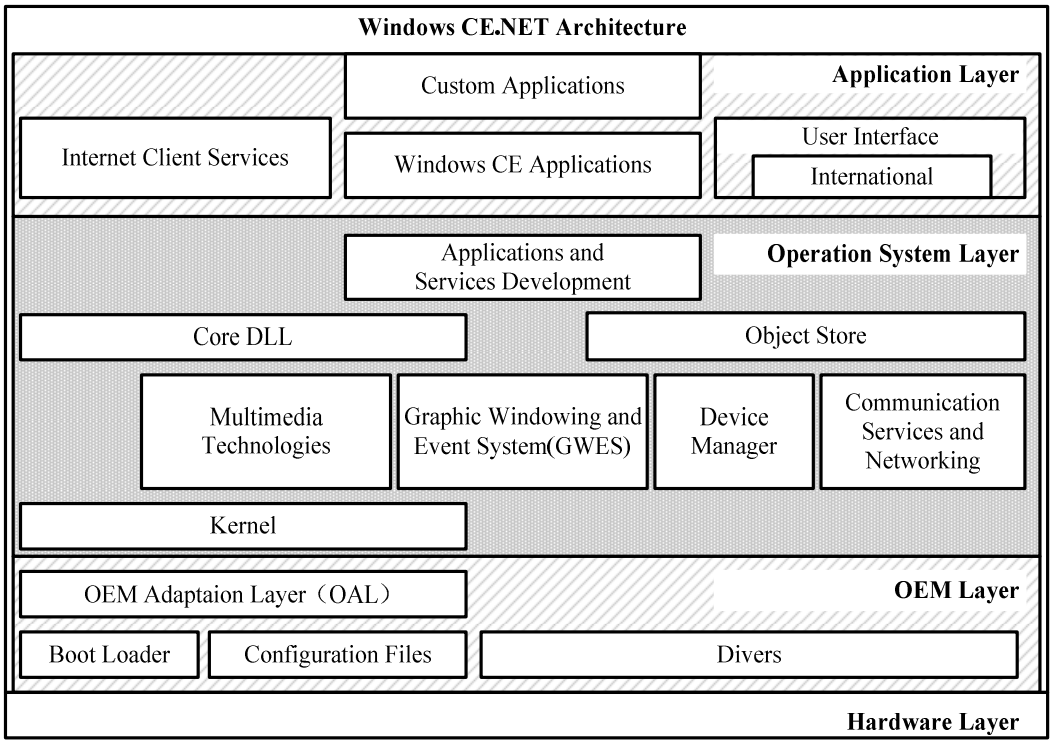

图 2.8 Windows CE 平台层次结构图

- 硬件层：Windows CE 系统所需的最小硬件配置包括 32 位处理器、用于线程调度的实时时钟、用于存储运行操作系统的存储器以及显示器、串口、网卡、键盘等外设。
- OEM 层：硬件抽象层，提供硬件与操作系统之间的接口。Windows CE 直接了支持丰富的硬件外围设备，如键盘、鼠标设备、触摸面板、串行

火龙果•整理
uml.org.cn

口、以太网、调制解调器、USB 设备、音频设备、并行口和存储设备（ATA 或闪存）。同时，Windows CE 扩展了新的外围设备类型，具有支持新的外围设备类型的巨大潜力：Windows CE 通过简洁而良好的定义了设备驱动模型，提供了易读的文档化设备驱动程序接口(DDI)，并展示了如何实现这些接口的例程代码。因此，嵌入式系统开发员（OEM 和 IHV）很容易地为运行在 Windows CE 平台上的设备实现自己的驱动软件。

- 操作系统层：Windows CE 使用模块化的设计方式，系统开发人员和应用开发人员能够根据自己的硬件和需求定制设备的操作系统，从而实现平台系统所需的最小软件模块和组件集合，从而使内存用量最小并最大可能地提高了操作系统的性能。
- 应用程序层：Windows CE 支持超过 1400 条使用最频繁的普通 Microsoft Win32 API 和其他编程接口，包括组件对象模型(COM)、Microsoft 基本类库(MFC)、Microsoft ActiveX 控制、Microsoft 活动模板库(ATL)等，基于 Microsoft 的桌面程序经过重新编译可以用于 Windows CE.NET。但由于对于 Windows CE.NET 所不支持的 API 函数，必须利用其他方法来实现；另外，Windows CE.NET 支持的内存容量和显示面积也是很有限。

由于 Microsoft 公司对嵌入式操作系统研发力量的增强，Windows CE 的优势越来越明显：实时处理能力不断增强、强大的多媒体能力、可伸缩的无线技术支持、针对小内存体积占用进行优化（最小 200kB）、具有适应性的可定制用户界面、丰富的设备驱动程序、强大的应用程序开发等。

2.2.2 Windows CE.NET 平台的开发工具[20]-[22]

Windows CE.NET 平台的开发工具 Platform Builder（简称 PB）是一个完全集成的开发环境，包括一个软件开发工具包（SDK）导出工具。Windows CE.NET 支持 Microsoft eMbedded Visual C++和 Microsoft Visual Studio .NET，为面向 Mrcrosoft .NET Compact Framework 的Web 服务和应用程序提供完整的开发环境。借助于完善的操作系统功能和开发工具，Windows CE.NET 为开发人员提供了定制有限资源的硬件系统所需的系统特性。

本课题实现的 Windows CE 4.2 版本的操作系统就是通过平台开发工具 PB 定制完成的。PB 运行在桌面 Windows 环境下，提供了开发 Windows CE 所需的设计、创建、编译、测试和调试工具。PB 允许开发人员通过平台向导（Platform Wizard）创建设备的类型，并提供了 12 个预建配置可供选择，包括数字媒体接收设备、移动手持设备、移动电话、Windows 瘦客户机等的基础配置。开发人员可根据嵌入式系统的硬件进行 WinCE 平台的 OAL 层移植、驱动程序开发、系统模

火龙果•整理
uml.org.cn

块裁剪以及应用程序开发。Microsoft 以及 Intel 等半导体厂商为常用的 CPU 和外设提供了 OEM 层源代码，开发人员可根据 PB 提供的标准 BSP 进行裁剪或添加，驱动标准的硬件设备。此外，PB 还提供了丰富的标准应用程序（如 WordPad 写字板程序等）和服务程序（如 Fat File System 文件系统等）。

2.2.3 Windows CE.NET 的定制流程[23]-[25]

创建 Windows CE 操作系统需要完成的任务有：

（1）选择操作系统的基本配置，并为特定的平台选择相应的微处理器和板级支持包 BSP（Board Support Packet），导入和硬件平台相关的.cec 文件。

（2）选择一个标准的开发版（SDB）作为模板，创建一个 WinCE 的平台；根据目标硬件设备开发 bootloader、设备驱动程序，适当裁剪和添加组件，对某些配置文件进行修改，然后封装所需要的各功能模块，编译生成镜像文件（image）。

（3）通过以太网将镜像文件下载到目标设备，可以使用 Platform Builder 提供的调试工具查看 Windows CE 的运行情况并进行调试，若需要，则进行重复配置、封装、下载及调试，直到满足要求，完成平台的创建。

（4）导出相应的软件开发工具包，运行后加入到 EVC++中，使得可以行特定硬件平台上的应用程序开发。SDK 包含程序库、头文件、示例程序源代和库函数使用文档，同时还包括编程指导和参与以及设备驱动工具包(DDK)。

Windows CE 定制流程如图 2.9 所示。

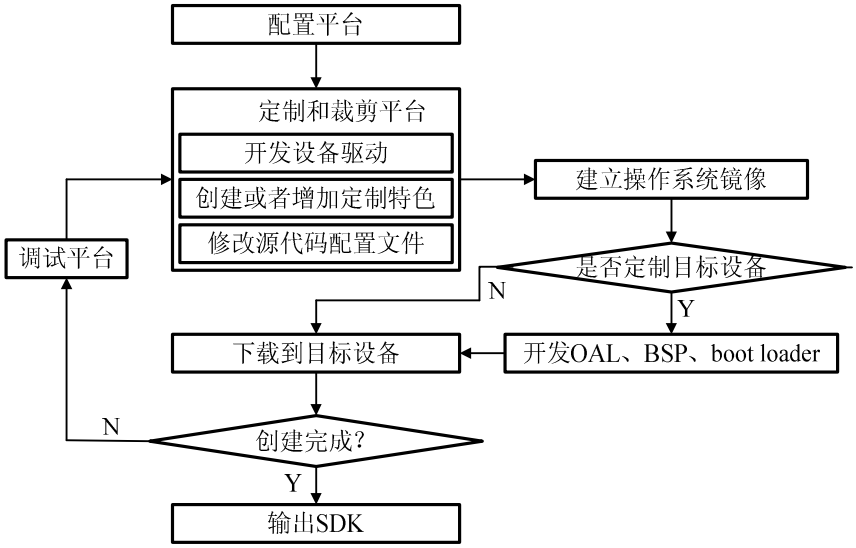

图 2.9 操作系统定制流程

火龙果•整理
uml.org.cn

2.2.4 Windows CE.NET 定制过程中重要组件的配置

在定制 Windows CE.NET 的过程中，PB 提供了很多组件可供选择，诸如基础类库、应用程序、核心服务、网络通信服务、文件系统、显示字体、所支持的国家、Internet 程序、安全性、多媒体、操作系统外壳等选择。考虑到嵌入式系统有限的存储器资源和处理能力，用户必须综合自己的硬件设置，将要进行的上层应用软件开发等，有取舍的选择必要的组件；同时，还必须对 PB 预先配置的组件属性进行裁剪和修改，卸载不必要的组件，修改硬件设备相应的配置文件。

以下小节中，将针对本文要实现的多媒体播放器的应用程序，对所定制 Windows CE.NET 内核定制过程中必须配置的组件和重要步骤进行说明。

2.2.4.1 导入 BSP（板级开发包）

BSP 包是一个主板支持软件包，包括启动程序（Boot Loader）、OEM 适配层程序（OAL）、标准开发包（SDB）和相关硬件设备的驱动程序软件包。BSP 是 Windows CE.NET 系统平台的主要部分，由资源文件和二进制文件组成。

BSP包是通过特定目录文件（CEC），加载到集成开发程环境PB中去的。CEC文件所实现的功能就是把用户自己编写的应用程序、驱动程序、第三方的驱动程序加载到Windows CE.NET特性目录中，可以在PB中能够像添加Windows CE.NET自带特性一样添加自己的特性。因此，导入BSP包之前必须先编写包含PXA270特性的CEC文件。一个CEC文件由若干单元及子单元组成，表2.1显示的是包含在CEC文件中的单元名称及说明。

表 2.1 CEC文件单元

单元名称	包含内容
CECInfo	关于一个CEC文件的基本信息，如名称、厂商、描述、版本等
ComponentType	特性组或是一个普通特性，它可以包括多个实现（implementation）
Implementation	一个实现定义了一个部件的类型
BuildMethod	建立系统的信息，用来建立或展开这个特性到操作系统镜像中
BibInfo	对于一个实现来描述这个Project.bib文件

CEC文件编写完成后，首先，在PB中导入特性文件PXA270.cec，然后执行File—>New Platform命令，在New Platform Wizard中选择INTEL PXA270BSP：ARMV4I选项，如图2.10所示。

火龙果•整理
uml.org.cn

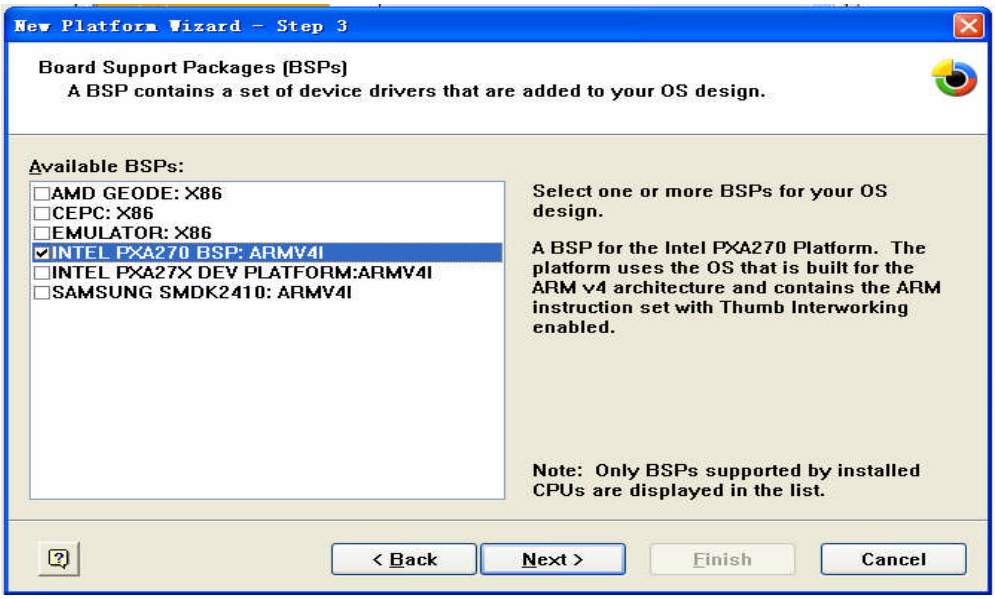

图2.10　制作映象文件之选择BSP包

在进行应用程序开发时，如需要在宿主机 PC 上进行模拟调试，还必须选择“EMULATOR：X86”选项，如图 2.11 所示。

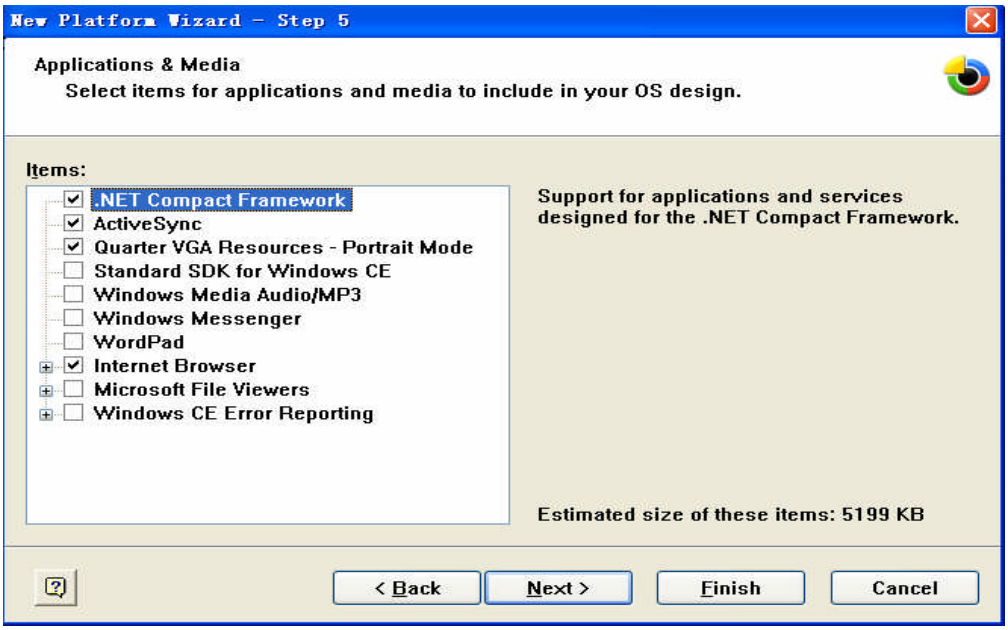

图 2.11　制作映象文件之选择配置

2.2.4.2 添加多媒体应用程序开发环境必须的属性

本课题所设计的应用程序将使用 EVC++编程环境和 DirectShow 技术，因此，还需要添见 COM、DCOM 服务和 Standard SDK for Windows CE.NET 等，按照以下步骤分别添加相应属性，如图 2.12 所示：

火龙果•整理
uml.org.cn

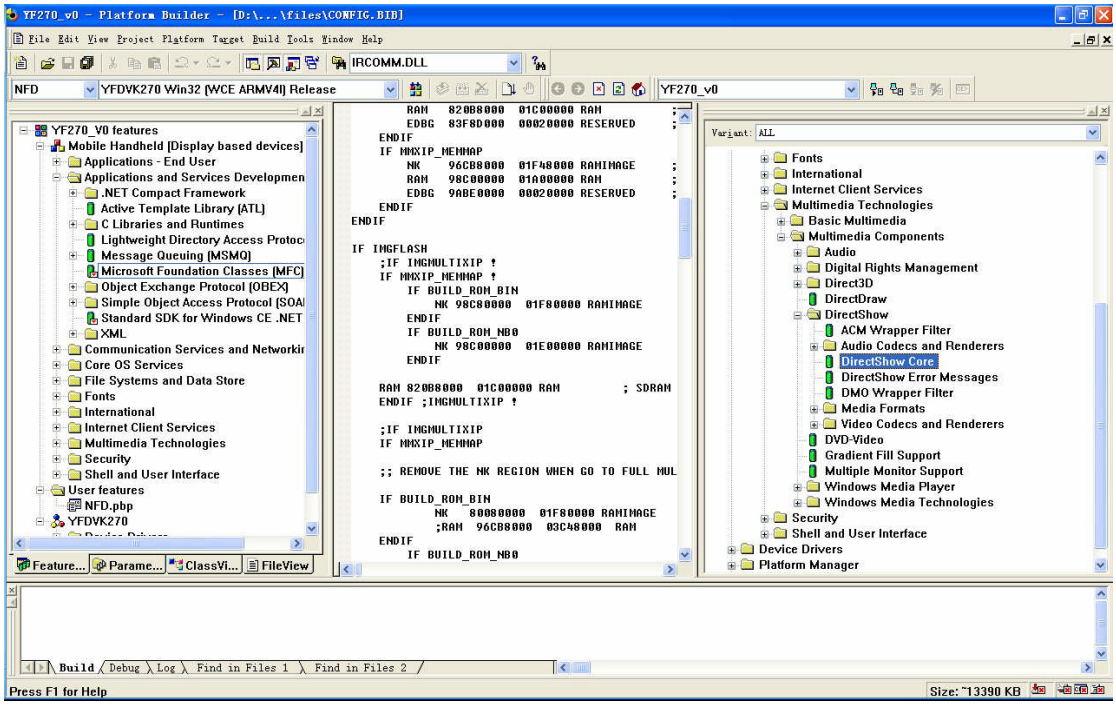

图2.12　制作映象文件之添加组件

- 添加MFC支持

在Catalog列表Core OS－>Windows CE devices－>Applications and Services Development中的Microsoft Foundation Class（MFC）组件上单击鼠标右键，在右键菜单中选择Add to OS Design，将该组件加入当前平台。

- 添加SDK支持

在Catalog列表Core OS－>Windows CE devices－>Applications and Services Development中的Standard SDK for Windows CE组件上单击鼠标右键，在右键菜单中选择Add to OS Design，将该组件加入当前平台。

- 添加多媒体技术

在Catalog列表Core OS－>Windows CE devices－>Multimedia Technologies的Basic Multimedia、Multimedia Componetnts中的DirectShow等组件加入当前平台。

2.2.4.3 添加多媒体播放系统的硬件驱动

根据硬件平台的配置，选择相应的驱动：为支持IDE设备选择FAT File System；为将操作系统的配置保存在注册表，选择Hive-base Registry；并需要支持SD卡、TFT LCD显示器、串口设备等硬件。

按照以下步骤分别添加相应属性：

- 添加FAT文件系统

在Catalog列表Core OS－>Windows CE devices－>File Systems and Data Store－>Storage Manager中的FAT File System组件上单击鼠标右键，在右键菜单中选择

火龙果•整理
uml.org.cn

Add to OS Design，将该组件加入当前平台。

- 添加SD/MMC卡支持

在Catalog列表Core OS－>Device Drivers－>Storage Devices－>Storage Devices下的Secure Digital/Multimedia Card(SD/MMC)组件上单击鼠标右键，在右键菜单中选择Add to OS Design，将该组件加入当前平台。

- 添加Third party中的相应支持

在Third Party列表BSPS－>Intel PXA270 BSP：ARMV4I－>Device Drivers下所有展开的组件中，将相应的音频、液晶、存储卡、串口等组件上一一单击鼠标右键，在右键菜单中选择Add to OS Design，将组件加入当前平台。

添加完这些特性之后，在Build菜单中选择Build Plaftorm一项，系统开始编译内核，生成。最终生成镜像文件NK.BIN，即是适用于本课题自定义设备的操作系统镜像文件，也就是自定制的Windows CE.NET内核。

2.2.4.4 修改 Windows CE.NET 配置文件

配置好以上 Windows CE.NET 的组件之后，即可通过网线将镜像文件下载到 PXA270 硬件平台作为操作系统使用。但当播放器应用程序的开发完成后，作为产品发布之前，还必须将重新编译 Windows CE.NET 内核，实现开机启动画面、自动加载播放器应用程序，对用户屏蔽掉 Windows CE.NET 操作系统。为此，需要修改一些配置文件，以下详细说明。

- 定制启动画面

由于 Windows CE 启动时，要经过耗时几秒的 EBOOT 引导加载过程，而这段时间 LCD 的显示为花屏，从而为用户带来多余的困扰。一般的处理方法是在 EBOOT 引导期间初始化 LCD，使之显示一幅静态画面，画面可以是公司商标、产品 LOGO 等简单图形，也可以是用户法律协议、或使用方法这样的文字。以下是让 EBOOT 操作 LCD 的具体修改步骤：

首先，使用 Photoshop 等图形制作工具设计启动画面 eboot_logo.bmp，加入设备名称、版权信息等标识，规格为 BMP 图标，颜色深度为 16 位真彩色。由于一个标准的 16 位分辨率为 800*480 的启动画面大小约为 600kB，而 EBOOT 的大小受到严格控制（Windows CE 4.2 的 EBOOT 要求在 128kB 以内），因此还需要通过压缩图像数据、减小分辨率的方法减少启动画面对 EBOOT 的占有率。与次对应，在显示启动画面的程序中时还必须在添加图像放大的算法，使得图像覆盖整个显示屏幕。

然后，修改 BSP 目录下的 eboot 子目录中的 MAIN.C 文件，添加定义的 LCD 显示函数 TestLCD()：

火龙果•整理
uml.org.cn


```
/* Copy kernel data to RAM & zero out BSS */
KernelRelocate(pTOC);
OEMInitDebugSerial();
TestLCD();                    //调用 LCD 屏幕绘图函数
EdbgOutputDebugString(NKSignon);
……  ……
/*********************************************************/
```

TestLCD()函数的原理是：将位图 eboot_logo.bmp 文件以一维数组的形式存储存，依照屏幕坐标，使用二重 for 循环，以“描点”的形式将 bmp 位图等比列打印在 LCD 显示屏上。这样做的好处是：显示速度快，占用 RAM 资源少，而且不易修改，有力地保障了软件版权。

- 开机自动加载多媒体播放器应用程序

由于涉及到软件版权的保护，考虑防拷贝机制，使用户不能获得 WindowsCE.NET 的系统文件访问权，因此需要在操作系统成功启动后立即加载播放器，在软件屏蔽用户对 Windows CE.NET 的直接访问权。修改如下：

首先，在 PB 中将播放器应用程序 Player.exe 添加到 Windows CE.NET 的 windows 目录中，方法是：打开工程项目，选择菜单“Platform”→“Insert”→“Insert User Feature…”选项弹出添加文件窗口，在 User Feature 栏中填入要添加的文件路径名称，但需要注意，添加用户文件会使最后生成的内核增大，可能导致编译失败。

然后，修改 Windows CE.NET 的注册表项，找到 shell.reg 注册表文件，将 Player.exe 放在注册表的键[HKEY_LOCAL_MACHINE\init]下面，如：

```
[HKEY_LOCAL_MACHINE\init]
"Launch50" = "explorer.exe"
"Depend50" = hex:14,00,le,00
"Launch55"="Player.exe"              ；添加该行自动调用 Player.exe 程序
"Depend55"=hex:14,00, 1e,00, 32, 00   ；添加该行设置 Player.exe 启动条件
/*********************************************************/
```

其中 Launch 后的数字越小，运行越早；Depend 的值说明了所加载的程序要依赖前面的那些程序。

这里需要强调的是，嵌入式应用程序开发后还应重新编译操作系统内核，以便将自行开发的应用程序（AP）编译到 Windows CE.NET 的内核 NK.bin 中，作为组件发布；也可将多媒体播放的应用程序存放在 SD 卡等外部存储器中，作为分离的嵌入式可执行文件单独运行。表 2.2 对这两种方法的优缺点进行了比较：

火龙果•整理
uml.org.cn

表 2.2 嵌入式应用程序存在形态的优缺点比较

AP 形态	优缺点
AP 与 NK.bin 分离	软件升级容易，但保密性不好，容易非法拷贝
AP 集成在 NK.bin 中	不容易非法拷贝，但升级需要更改整个操作系统文件

2.2.5 多媒体播放器应用程序开发环境的配置

本课题的宿主机运行的是 Windows CE.NET 4.2 和 Microsoft eMbedded Visual C++ 4.0 开发环境。目标开发板是自行开发的 PXA270 为核心的嵌入式系统，运行自定制的 Windows CE.NET 操作系统，通过网线、串口与宿主机相连。宿主机 PC 上进行嵌入式程序开发，必须先安装含有目标板的硬件配置和系统信息的 Windows CE.NET SDK 开发包，由于开发中用到了 DirectShow 进行多媒体开发，还必须安装 DirectShow SDK 开发包，以下小节分别介绍。

Microsoft eMbedded Visual C++是 Microsoft 公司推出的 Windows CE 程序可视化开发工具。随着 Windows CE 版本的变化，Embedded Visual C++的版本也相应的变化，EVC 3.0 只能开发 Widnwos CE 3.0 的应用程序，EVC 4.0 只能开发 Widnwos CE 4.0 的应用程序，EVC 4.0+SP1 能开发 Widnwos CE 4.1 的应用程序，EVC 4.0+SP2 能开发 Widnwos CE 4.2 的应用程序。

进行应用程序开发前，首先在桌面 PC 的 Windows XP 系统上安装 EVC 4.0+SP2，然后将 PB 定制的 SDK（*.msi）文件安装到 EVC 中，建立应用程序开发的交叉编译环境。

在进行需要注意的问题有：

（1）在安装、配置和编译 EVC 时会出错。无论出于何种原因，若第一安装 EVC 失败再次安装也很难成功，因为 EVC 的注册信息保留在注册表中无法彻底卸载。因此，在配置嵌入式系统程序开发之前，最好先为桌面 PC 的操作系统做一个镜像备份，以便在出错后快速恢复。

（2）Windows CE.NET 的编译是一个交叉编译的过程。PB 依赖运行在开发机的 Windows CE 环境来编译目标机定制的 Windows CE 系统。这相当于同时运行两个操作系统，所以编译的速度慢，需要耐心等待。

（3）选择 Built/Sysgen 实现对整个平台的编译和连接，进而生成内核镜像。这个过程如会出现编译错误重新编译时，应该选择 Clean Before Builder 之后再 Sysgen。

火龙果•整理
uml.org.cn

2.3 小结

本章从系统的角度阐述了车载多媒体播放器的硬件与软件平台设计。

硬件方面首先介绍了车载多媒体设备的选型标准，比较了主流的处理器，然后给出了本课题的硬件方案，重点阐述了 PXA270 处理器的技术特点，以及音频输出模块连接车载功放的巧妙设计。

软件方面主要介绍了 Windows CE 操作系统的定制流程，重点说明了定制过程中必要组件加的载方法和关键步骤，介绍了嵌入式应用程序开发环境 EVC++ 的配置方法。另外，为消除系统启动时的花屏现象给用户带来的困扰，保护应用软件的版权，防止盗拷贝，本章介绍了以下功能的实现：

- 定制多媒体应用程序启动画面，显示设备标识、版权信息等；
- 自动加多媒体播放器，屏蔽 Windows CE.NET 操作系统。

火龙果•整理
uml.org.cn

第三章 多媒体播放器应用程序的设计与实现

车载多媒体播放器的应用程序是在嵌入式系统上实现的，其功能的实现包括两个重要设计：一是美观实用的人机交互界面，二是强大的音视频解码能力。

本章首先对多媒体播放器的模块化设计思想进行阐述，介绍了 DirectShow 技术设计应用程序设计的思想和 Filter 插件开发的方法；然后将多媒体播放器的程序设计分为系统管理模块、图形界面模块、音视频解码模块三部分，并分别实现：

- 系统主控模块：多媒体播放器的主体部分，实现了音视频解码模块的加载、卸载，使用事件-消息机的机制控制图形用户界面模块的显示、更新。这部分重点分析了主控模块程序结构的设计，播放期间的工作状态、控制函数的编写与实现以及事件消息机制的实现。
- 用户图形界面模块：使用 Microsoft 提供的公共控件开发图形界面，由于该模块只实现了控件的图形特征，因此只简单介绍了公共控件库的加载方法。按钮等控件的功能特性由主体程序实现。
- 音视频解码模块：实质上是依据 DirectShow 标准实现的 Filter 插件。这部分首先介绍了 Filter 音视频解码插件的开发标准和步骤，然后具体实现了 MPEG-4 视频解码器 XviDFilter 的 Filter 封装。

3.1 多媒体播放器的设计方法

3.1.1 DirectShow 技术概述[26]

随着信息技术的发展，多媒体技术面临新的挑战：

- 多媒体流包含了大量的数据，这些数据要求被迅速处理以达到实时性。
- 视频流和音频流的同步。
- 流的来源非常复杂,包括本地文件、Internet 网络、摄像头、视频卡和电视广播网等。
- 流的格式也多种多样,包括 Audio2Video Interleaved(AVI)，Advanced Streaming Format (ASF)，Motion Picture Experts Group (MPEG)，and Digital Video (DV)等。
- 开发人员事先并不知道终端用户的硬件设备。

Microsoft 的 DirectShow 正是为了适应以上的挑战而设计的多媒体开发工具，Microsoft 设计它的意图就是简化多媒体应用程序的开发，使开发者不必考虑复杂的流数据格式和不同的终端设备，以及数据同步的问题。

火龙果•整理
uml.org.cn

Microsoft 通过 DirectShow 给多媒体程序开发员提供了标准的、统一的、高效的 API 接口。为实现设备无关性，使同样的操作设备代码不加修改即可在 Windows 平台上正常运行，Windows 设计时在内核与硬件之间引入了硬件抽象层，对硬件进行了屏蔽抽象，不允许用户直接访问硬件，并尽可能少的干涉系统内核运行。此举满足了系统的稳定性的要求，但使得图形应用程序只能通过 Windows 提供的图形设备接口（GDI）来完成绘图任务，如窗口绘制等，而无法直接使用新硬件的新特性。但对内存资源有限、实时性要求较高的嵌入式多媒体应用程序设计，基于 GDI 程序在速度上远远达不到要求。然而 Windows CE 提供的 DirectX 库，既保持了 Windows 的硬件无关特性，Windows CE.NET 下的 DirectX 大约与桌面 PC 机上的 8.0 版本的 DirectX 功能相当。

3.1.1.1 基于 DirectShow 技术的应用程序开发

DirectShow 技术是建立在 DirectX 的 DirectDraw 和 DirectSound 的基础之上的，它通过 DirectDraw 对显卡进行控制以显示视频，通过 DirectSound 对声卡进行控制以播放声音。DirectX 为了最大限度提高效率而允许用户直接访问硬件，如允许用户直接读写显存，因此，DirectShow 嵌入式系统的多媒体功能应用上具有速度的优势。

Microsoft 通过 DirectShow 给多媒体程序开发员提供了标准的、统一的、高效的 API 接口。DirectShow 的工作原理类似“流水线”：系统通过 Filter Graph 的模型管理数据流的处理过程，各个 Filter 在 Filter Graph 中按照一定的顺序连接成一条“流水线”参与数据的处理。

一个典型的 DirectShow 系统工作流程如图 3.1 所示。

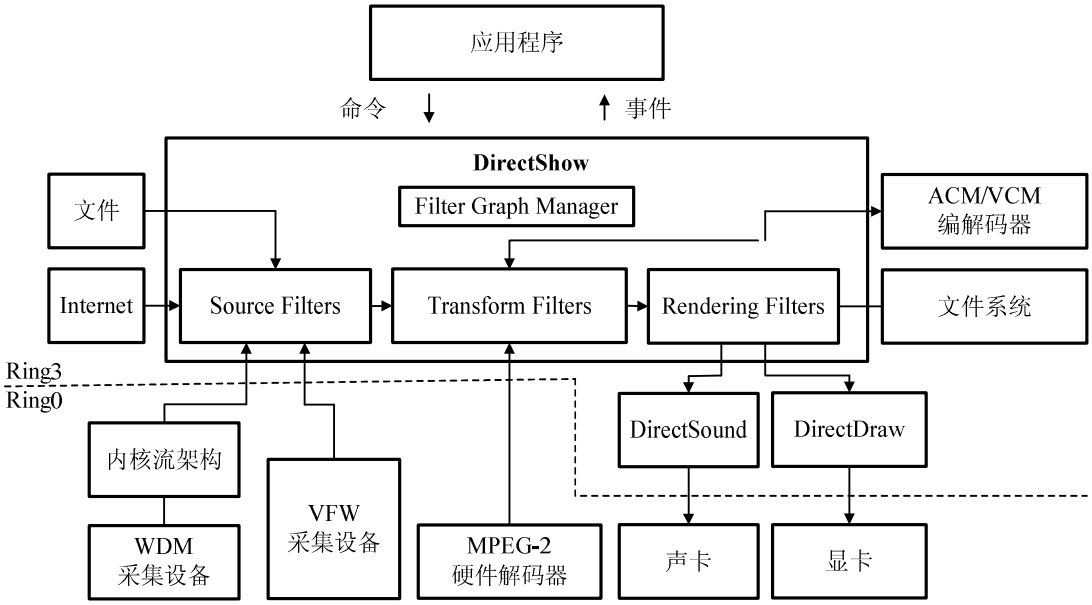

图 3.1 用户程序与 DirectShow 及外部对象的关系框

火龙果•整理
uml.org.cn

一个典型的 DirectShow 应用程序中，Filter 的一系列操作，包括建立、添加、删除等都是通过 Filter Graph Manager 来完成的。在程序开始之前首先创建 Filter Graph Manager，然后申请各种接口，应用程序通过接口的成员函数管理媒体播放的整个流程。

- DirectShow 应用程序的开发步骤如下：

（1）通过 API 函数 CoCreateInstance()创建 Filter Graph Manager 实例；

（2）通过调用 QueryInterface ( )函数来获取 Filter Graph 和 IMediaEvent 组件的指针；

（3）对 Filter Graph 进行控制并对事件作出响应。

- DirectShow 常用的接口函数

（1）IMediaControl：通过 Filter Graph 实现数据流的控制，还可以自动创建过滤器图来完成文件的播放，下文用到的成员函数见下表 3.1：

表 3.1 IMediaControl 的成员函数

成员函数	描述
Run	运行所有 Filter
Pause	暂停所有 Filter
Stop	停止所有 Filter
GetState	获取 Filter 状态
RenderFile	为指定文件自动创建 Filter Graph
AddSourceFilter	为指定文件添加 Filter

（2）IMediaSeeking：获取或设定文件当前播放位置，下文的设计用到了 SetPosition 函数。

（3）IBasicVideo：获取视频窗口的基本信息及设定视频窗口的位置，下文的设计用到了 GetVideoSize 成员函数获取视频的原始尺寸。

（4）IVideoWindow：设置视频窗口的属性，如父窗口、窗口位置以及窗口尺寸，还可将视频窗口的各项窗口消息传递给父窗口处理。

3.1.1.2 DirectShow Filter 的开发

Filter 是 DirectShow 的基本组件，按照功能大致分为三类：Source Filter、Transform Filter 和 Rendering Filters，分别介绍如下：

- Source Filter 主要负责获取数据，然后将数据往下传输；
- Transform Filter 主要负责数据的格式转换，如将数据流分离/合成、解码/编码等，然后将数据继续向下传输；
- Rendering Filters 负责数据的输出，将处理后的数据送给显卡、声卡等设

火龙果•整理
uml.org.cn

备，或输出到文件进行存储。

DirectShow 提供了大量的 Filter 用以支持最基本的应用，并且允许用户根据自己的需要按照规范编写自己的 Filter，以扩充 DirectShow 的功能。开发新的 DirectShow Filter 比较复杂，以下选择几个重要的方面进行分析：

（1）为 Filter 选择合适的基类

Microsoft 的 DirectX 9.0 为用户提供了丰富的 DirectShow Filter 基类，允许用户从基类和接口中派生，简化了 DirectShow Filter 的开发工作。简要介绍几个重要的基类：

- CBaseObject：绝大部分的 DirectShow Filter 都是从 CBaseOject 类派生的，它提供了基本的辅助调试功能。
- CUnknow：CBaseObject 类的派生类，提供了 COM 功能，包括应用技术以控制对象生存，用 QueryInterface 来得到对象接口等。
- CBaseFilter：CUnknow 的派生类，提供了 IBaseFilter 接口。
- CBasePin：CUnknow 的派生类，提供了 IBasePin 和 IQualityControl 接口。
- CSource：CBaseFilter 的派生类，提供了 Transform Filter 的基类。
- CBaseRender：CBaseFilter 的派生类，提供了 Render Filter 的基类。

（2）实例化 Filter

所有的 Filter 必须实例化，重载基类和必须的接口，这涉及到两方面的工作：一是必须为派生 Filter 类增加 CreateInstance 静态成员函数，用它调用派生类的构造函数，从而获得派生类的实例；二是告诉基类的类工厂（Class Factory）如何访问这个 CreateInstance 函数：声明一个 CFactory Template 全局数组以提供 Filter 的名称，Filter 的身份（CLSID）和指向函数 CreateInstance 的指针。

（3）重载 NonDelegatingQueryInterface 以获得新增接口

如果没有新增任何接口，可以忽略这一步。DirectShow 中定义的一个特殊的 IUnknown，称为 INonDeletegatingUnknown，当 Filter 要查询某个接口时，必须调用其方法 NonDeletegatingQueryInterface 获取接口。如果 Filter 实现了基类没有的接口，必须重载这个函数，获得新的接口。

（4）编写 Filter 类的成员函数

包括重载的一些必要的基类成员函数和 Filter 新的函数。

3.1.2 多媒体播放器的模块化设计

播放器采用模块化的设计方法，将播放器的管理控制和功能实现分别设计成独立的模块，并实现为同类功能模块实现统一的访问接口，当播放器需要使用某个功能就只需加载此模块，不再使用时就卸载此模块。这种方法的优点是：

火龙果•整理
uml.org.cn

- 程序结构清晰，设计思路清晰，调试和维护方便；
- 模块化的设计，有利播放器功能的升级和扩展；
- 各个模块功能明确，使设计人员能够集中精力进行某一领域的研究，减小了发难度和风险；
- 程序运行时灵活加载/卸载功能模块，占用资源少，运行速度快，对于资源有限的嵌入式系统来讲，这是一点尤为重要。

根据上述分析，将多媒体播放终端的应用程序设计分为三部分，分别是图形用户界面模块、系统管理主控模块和音视频解码模块，结构如图 3.2 所示：

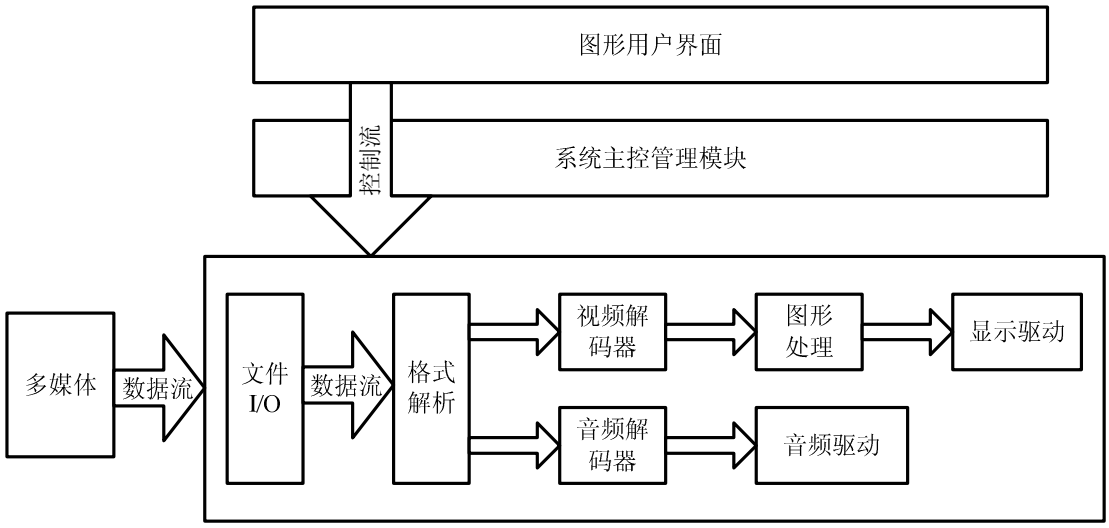

图 3.2 播放器的模块化设计结构

根据播放器的模块化设计思路和上节所引入的 DirectShow 技术，从程序设计的角度进行分析：

- 系统管理主控模块负责播放器的初始化化配置和程序结束后系统资源的回收，并协调图形用户界面和音视频解码模块之间的工作。
- 图形用户界面模块封装按钮（Button）、对话条（Dialogbar）等控件，并以资源的形式封装了控件外观。
- 音视频解码模块封装了各种音视频文件的解码算法，并提供了初始化方法、解码逻辑等的具体实现。

当用户点击图形界面按键操作时，实际就是图形用户界面模块通过事件/消息机制，将发生的状态改变如按钮的按下、弹起，进度条的拖动等事件以消息的形式发送给系统主控管理模块，由系统主控模块解析该消息，通知控件的行为改变（如变灰不可用、进度条滑块移动）等；同时系统主控模块还将调用相应的音视频解码模块进行播放。

系统主控模块对音频解码模块的控制，通过管理 Filter Graph 的相关操作来实现的。其职责是：根据用户界面接到的指令，动态的建立 Filter Graph，并管理 Filter 的运行，终止和释放。

火龙果•整理
uml.org.cn

音视频解码器模块使用 COM 技术实现，具有标准统一的接口，如果有了新的编码技术或编码器更新了版本，都可在不修改播放器实现方案的情况下，引入新的解码器插件，直接在程序中使用。

3.2 系统主控模块设计

系统主模块的工作流程如图 3.3 所示。系统主控模块是多媒体播放器的主要功能控制模块，可将图形用户界面模块和音视频解码模块可看作是系统主控模块的“功能插件”。当系统主控程序启动时，首先进行初始化，根据需要加载“功能模块”，创建 Filter Graph 管理音视频解码器，维护消息循环队列，等待事件发生。本节分别从程序结构设计、功能函数的实现、消息循环维护等角度具体分析系统主控模块的实现。

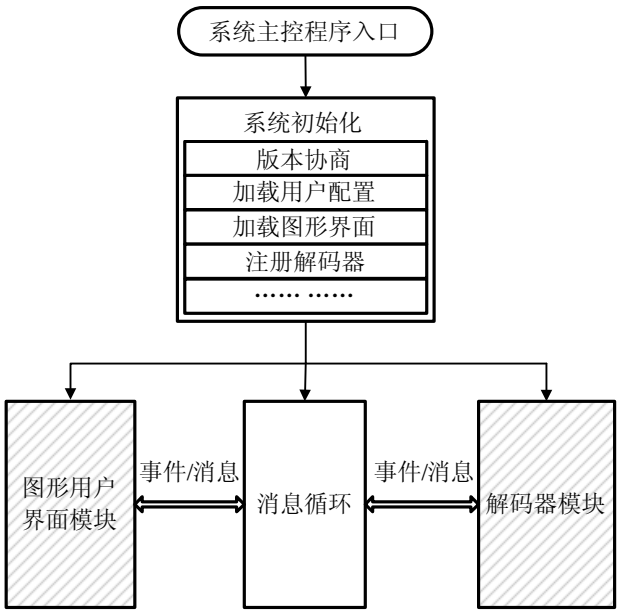

图 3.3 系统控制模块工作流程图

3.2.1 功能模块的加载/卸载

播放器采用模块化的设计方法，每个功能被设计成每个独立的模块，并且具有统一的访问接口，放器如果要使用某个模块只须加载此模块，如果不再使用就卸载此模块，因此模块的加载/卸载就当重要，下面进行详细分析一下播放器中的模块加载/卸载技术。

（1）功能模块的加载

播放器功能模块的加载过程如图 3.4 所示，播放器在每次启动时将要使用的解码模块加载到一个全局的数据结构 context 中，为了实现兼容不同的功能模块，context 需要保存的信息非常多，主要成员如表 3.2 所示：

火龙果•整理
uml.org.cn

表 3.2 context 的结构体成员

成员类型	成员名称	含义
int	Version	程序的版本
uint32_t	ProgramId	程序的唯一 ID
Const tchar_t*	ProgramName	程序名称
Const tchar_t*	ProgramVersion	程序的版本
Const tchar_t*	CmdLine	命令行参数
array	Node	节点序列（没有排序）
array i	NodeClassPri	节点序列（按照节点优先级排序）
array	NodeModule	模块节点序列
int	LoadModuleNo	当前加载的模块 ID
void*	LoadModule	加载模块
struct	node*Player;	播放器模块
int	StartUpMemory	起始内存大小
bool_t	InHibernate	休眠标志
bool_t	WaitDisable	等待标志

- A 阶段，播放器刚刚启动，尚未加载任何解码模块，所以全局的数据结构是空的；
- B 阶段，播放器加载了所需要的解码模块，由于功能模块是静态定义的，在编译的时候已经存在，加载的过程实际上是连接功能模块的过程；
- C 阶段，当播放器要用到某个解码模块时，首先查找该模块在内存中的位置，然后通过该模块提供的构造函数构造应用实例，就可调用该模块所有的功能函数，根据需要动态申请功能模块所使用的数据结构（包括私有数据或接口等）。

（2）功能模块的卸载

当播放器不再使用某个模块，那么 C 阶段所构造的功能模块就没有存在的必要了，播放器将释放 C 阶段动态申请的数据结构（包括私有函数或接口），回到 B 阶段；由于功能模块是静态定义的，因此所连接成的树状结构仍然保存在内存中，因此在播放器运行过程中，不断的在 B 阶段和 C 阶段之间循环。当用户退出播放器时，则调用功能模块自定义的析构函数，从 B 阶段回到 A 阶段，并释放所有资源。

火龙果•整理
uml.org.cn

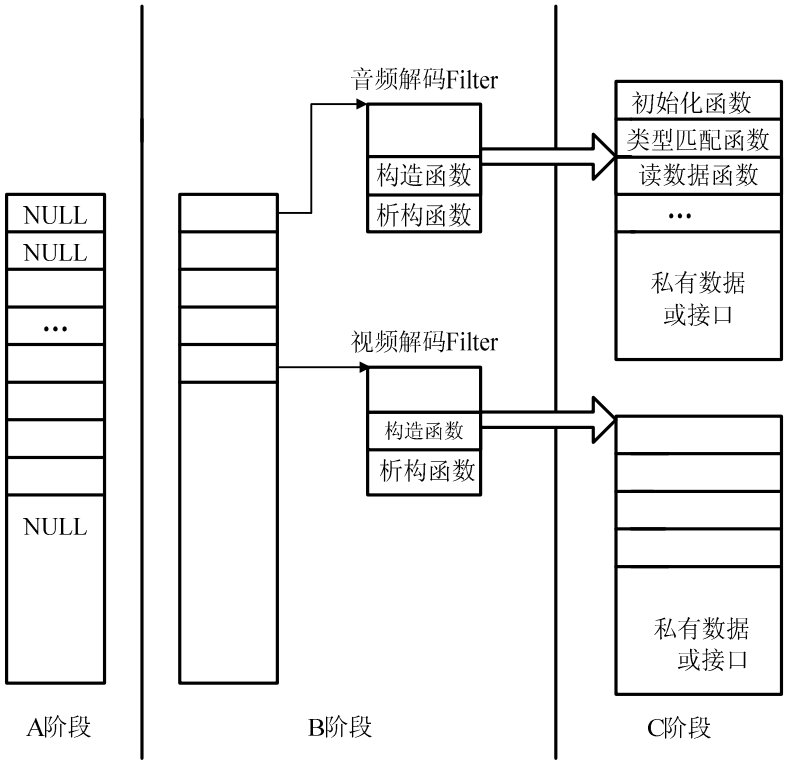

图 3.4 模块加载模型

3.2.2 播放器工作状态分析[27]-[29]

当播放器完成初始化后，系统主控程序就进入播放准备状态。下面具体分析在播放媒体文件的过程中，需要实现的几点功能：

- 允许用户打开指定文件
- 用户选择播放功能时播放媒体文件。
- 当用户选择停止时，复位媒体文件。
- 退出时释放全部资源。

根据以上几点要求，需要编写函数实现其功能。

（1）主控程序开始运行后，建立窗口，初始化 DirectShow。InitGraphBuilder 函数负责创建 Filter Graph 的实例，申请接口。使用 SUCCEEDED( )宏判断函数返回值即返回申请接口的结果，如果函数执行成功，返回 TRUE，否则为 FALSE。并定义 CurrentState 宏指示当前播放器的状态，INIT（初始化）、RUNNING（播放中）、PAUSE（暂停）或是 STOP（停止）。

（2）当用户点击菜单中的“打开文件”时调用 AddFileToPlayer 函数打开指定文件，并初始化视频窗口和主窗口，建立两者之间的联络。打开文件功能是通过 API 函数 GetOpenFileName 来实现的，此函数只有一个参数，为一个包含操作信息的结构体，其中的 lpstrFile 域存放文件的完整路径以文件名。调用此函数时 Windows CE 将弹出一个标准的“打开文件”对话框。文件打开后，让 Filter Graph

火龙果•整理
uml.org.cn

自动建立，只要安装了相应的解码器 Filter，就可播放所打开的媒体文件。使用函数 IGraphBuilder::RenderFile 来判断系统中是否有对应的解码器 Filter。

如果能够播放，就将 CurrentState 设置为 STOP，并建立与视频父窗口的关系，内容主要是两项：一是将视频窗口消息交给父窗口处理，由父窗口中处理，即相应用户通过触摸屏发送的暂停、全屏等命令；二是讲视频窗口的媒体消息传给父窗口，媒体消息是指媒体播放中途无法继续、媒体播放完毕等等，媒体消息由自定义消息实现。

（3）不同的视频文件显示窗口有不同尺寸，为保证视频画面能够正常显示，由 IBasicVideo::GetVideoSize 函数获取原始视频大小。若返回数值为 E_NOINTERFACE，则表示此文件为音频文件，或无法正常解码；若返回值为原始视频尺寸，则按照此尺寸重新定义视频窗口大小。

视频窗口大小的定义方法是用视频的高度加上菜单栏、命令条、进度栏以及窗口下框尺寸得到的实际窗口高度，用视频宽度加上两倍的窗口侧边框的尺寸得到窗口宽度。调用 API 函数 GetSystemMetrics 获得视频大小，调用 CommanBar_Height 获得命令条的高度。若原始视频尺寸过大，超过屏幕，按比例缩小。再调用 SetWindowPos 即指定窗口位置，将窗口置于屏幕中央。

（4）正确载入音视频文件后，用户可以使用播放、暂停、停止等命令。为协调控制播放功能的函数，定义 CurrentState 表示当前状态。若需执行某条功能函数，先判断当前状态，若当前状态不允许执行此函数，则直接返回。三种功能分别对应 IMediaControl 接口中的 Run、Pause 和 Stop 成员函数，执行一条功能函数后，再调用 CurrentState 变量对应播放器当前工作状态。

（5）当程序执行结束，释放申请的各类指针及变量，这个功能通过 ReleaseAll( )函数实现的。

3.2.3 消息循环的维护

3.2.3.1 消息函数的设计机制

消息循环处理是播放器运行重要的流程，直接影响播放器操作功能的正确性以及响应的及时性。为了使程序结构清晰设置消息处理函数时采用可扩展的循环检测。

（1）定义消息与消息函数关联的结构体：

（2）建立消息映射表：

（3）实现消息函数。

火龙果•整理
uml.org.cn

3.2.3.2 消息函数的实现[26][28]

具体消息的响应函数在必要时对消息参数进行检测分析，调用相应的功能函数相应特定的用户操作。下面阐述重要的消息响应函数的实现：

（1）窗口重绘消息函数的实现

窗口重绘消息函数十分重要，它直接关系到播放器外观的显示效果。在具体实现该函数时关键的一点是把握在必要时重绘失效的窗口区域，主要的逻辑实现托付给图形用户界面模块完成。

（2）控件按下的消息函数实现

在控件按下消息的处理中，主要通过把控件参数传递图形用户界面模块提供的控件 API 函数完成，方法是得到事件类型，然后改变当前控件按钮的外观。

（3）控件弹开消息函数的实现

在控件弹开消失的处理中，首先通过图形用户界面模块的 API 函数恢复当前控件按钮的外观；然后通过判断控件事件的类型来调用相应的功能函数相应用户的操作；最后将事件类型设置为空类型，结束该次控件事件的响应。

（4）播放事件通知消息函数的实现

该消息函数主要是处理播放器播放过程中产生的事件。主要通过检测用户设置的播放模式等来设置播放顺序的控制。

- 播放、暂停、停止功能的实现

通过 DirectShow 提供的接口实现以上几种功能，首先在创建 Filter Graph Manager 接口的函数里面定义 IMediaControl *mMediaControl；然后调用 IMediaControl 的接口实现播放、暂停、停止功能。主要函数如下

播放功能：mMediaControl->Run();

停止功能：mMediaControl->Stop();

暂停功能：mMediaControl->Pause();

- 全屏播放功能的实现

使用 Filter Graph Manager 上 IVideoWindow 接口的 put_FullScreenMode()方法，调用如下函数响应用户指令：

VideoWindow->put_FullScreenMode(inEnabled *OATRUE:OAFALSE);

- 进度条的实现

根据 Filter Graph Manager 上的 IMediaSeeking 接口，为进度条控件添加一个控制类 CSliderCtrl mSliderGraph，然后在初始化函数 OnInitDialog 中初始化进度控件：mSliderGraph.SetRanger(0,100)。用一个定时器，不断调用 IMediaSeeking：：GetCurrentPositon()接口方法获得当前的播放时间点，然后根据媒体文件总时间长度，计算进度的百分比，再转化为进度条当前的位置，调用 CSliderCtrl::SetPos()

火龙果•整理
uml.org.cn

函数设置。

实现进度条位置拖动，实际上是进度即时显示的反过程。当用户拖动进度块到一个新位置后，需要使用方法 CSliderCtrl::SetPos 得到这个进度值，然后转换为 Filter Graph 的播放进度时间点，并调用接口方法 IMediaSeeking::SetPosition()进行设置，于是，Filter Graph 就可以从新的位置继续播放了。

- 音量控制的实现

根据 Filter Graph Manager 上的 IBasicAudio 接口方法，首先定义 IBasicAudio 对象指针 mBasicAudio，调用方法 put_Volume(inVolume)控制音量，参数 inVolume 是音量值，在-10000 到 0，-10000 表示静音，0 表示最大音量。

系统音量的控制方法是通过操作系统内部控制逻辑相关连的结构体来实现的：MIXERCONTROLDERAILS_SIGNED，MIXERCONTROLDETAILS，改变两个结构体的参数设置，不但可以控制音量，还能够调解音频控制器的状态。

3.3 用户图形界面设计

3.3.1 图形界面的结构设计[30]-[34]

由于车载使用的环境特殊和嵌入式设备的资源有限，在 7 寸触摸屏上实现简单快捷的人机交互，使界面和按键功能的设计图形化的界面设计必须科学，既不会占用过多的资源，又不容易发生用户误操作，能够满足用户的使用习惯和审美心理。一般，传统的界面设计使用 MFC 库可以简化应用程序的框架设计，但在 Windows CE 系统下，运行负责、功能全面的 MFC 应用程序不但没有必要，而且程序的反应速度也不能满足需求。Microsoft 公司开发了一系列称为公共控件的附加控件，将这些公共控件包容到 DLL 中提供给开发者使用。伴随公共控件 DLL 的每次发布，新控件和新特性被加到旧的控件中。公共控件同标准 Windows 控件比，成熟度会低一些。但因公共控件同操作系统核心是分离的，装载需要的控件不仅可以减少界面资源占用的宝贵内存，而且通过加载不同的控件实现“换肤”技术，满足不同用户的审美要求。

由于公共控件 DLL 提供了控件接口的 API 函数，而将对话框、字符串、位图、图标以及菜单等资源作为 DLL 的一个只读数据段，因此在图形界面模块被编译后，资源也被链接到模块中。由于资源使用了一个与编译器无关的位置，用来存储常量数据，资源并不编译在程序里，所以改变控件外观不用重新编译程序。图 3.5 是多媒体播放的界面。

火龙果•整理
uml.org.cn

图 3.5 播放器界面设计

3.3.2 图形界面的程序要点

加载图形用户界面必须实现两个关键的步骤：

首先，调用 void InitCommonControls(void)装载动态库并注册公共控件类；

然后，初始化这些控件，使用函数

BOOL InitCommonControlsEx(LPINITCOMMONCONTROLSEX lpInitCtrls);

只装载应用程序初始化选择的公共控件，减少资源对内存消耗。

这样就可以在应用程序中调用控件了。在本课题实现的每个控件都有一个可定制风格标志集，用来配置控件的外观和行为。针对每个控件的消息会被发出，用来配置和操纵控件并让控件执行某些动作。

公共控件和标准 Windows 控件之间的一个主要差别是事件通知或服务请求是通过 WM_NOTIFY 消息来发出，而标准控件则是通过 WM_COMMAND 消息发出的。同通过 WM_COMMAND 消息发出的通知相比，采用这种技术可以使通知能够包含更多的信息，并且允许对 WM_NOTIFY 消息扩展和改编。消息的实现已在消息一节介绍过，不再重复。

3.4 解码器模块的实现

在多媒体应用中，MPEG-4 以其高压缩比和高性能倍受人们的青睐，成为多媒体应用中的主流视频压缩技术之一。实际的多媒体播放器部分已经注册了多种解码器，但在应用中发现无法播放后缀为.avi 的 MPEG-4 格式的音视频文件，这是因为系统没有提供相应的解码器。

目前，在诸多视频压缩标准中，MPEG-4 以其高压缩比和高性能倍受人们的青睐，成为多媒体应用中的主流视频压缩技术之一。国际标准化组织给出了 MPEG-4 编码的具体方法，目前，基于 MPEG-4 标准的实用编解码器主要有 ISO

火龙果•整理
uml.org.cn

MPEG-4 编码器、DivX 和 XviD。ISO MPEG-4 编码器可以提供标准的编码效果，它的编码速度取决于用户及其运算速度和硬盘的传输速度，由于其设置过于专业，所以用得不多；DviX 核心源自微软的 MS MPEGV1、MS MPEGV2 和 MS MPEGV3 系列编解码内核，即大家经常看到的流媒体文件的编解码方案，但经过了修改。DivX 起初是完全开放源代码的，但 2001 年 7 月的 DivX 拥有者公司停止开放源代码，并先后发布了收费的 DIVX4 和 DIVX5。此后，继续热衷于源代码开放的爱好者重新聚拢开发力量于 2002 年 2 月发布了基于 OpenDivX Encore2 的 XviD 编码器，并且使其性能、效率得到了极大的提升，被认为是目前最快的 MPEG-4 video compressor-decompressor(MPEG-4 video codec)，而且 XviD 编解码器是针对 Windows 用户设计的，它的源代码完全开放，功能非常强悍，图像压缩后质量接近 DVD 画质，但数据量只有 DVD 的 1/8[38]-[42]。

本课题要在 Windows CE 操作系统上实现 MPEG-4 解码器，因此 XviD 编解码器是最好的模型。在设计多媒体播放器的结构时，采用的是基于 DirectShow 技术的模块化架构，只要将 XviD 的核心算法封装成为 Filter 插件，作为一个独立的 Filter 加入到 Filter Graph Manager 中，就能够实现 MPEG-4 视频解码功能。

XviD 编解码器的 API 接口定义得非常清晰，其三个核心函数（xvid_global()，xvid_decore()，xvid_encore()）具有统一的形式，其中 xvid_decore()实现了 MPEG-4 视频解码函数。XviD 视频编解码器作为 MPEG-4 框架的子集，主要实现在 Simple level 上。XviD 的视频解码器主要包括变长解码（VLD），反扫描（IZZ），反 ACDC 预测（IACDC），反 DCT（IDCT），运动补偿等模块。

下文将根据开发音视频解码 Filter 一般过程设计 MPEG-4 视频解码 Filter：首先进行需求分析，为 Filter 选择合适的基类；再利用 Filter 应用框架将开源的音视频解码的核心算法进行封装，实现 Filter 的应用逻辑；最后实现可在播放器进行挂载的接口 Pin，进行注册。

3.4.1 MPEG-4 视频 Filter 的功能分析[26]

在进行设计时，先对音视频解码 Filter 进行功能分析，确定 Filter 所要实现的应用逻辑，然后按照前文所述的 Filter 开发的一般步骤，构造 Filter 的应用框架，需要考虑以下要素：

（1）功能分析

- XVIDFilter 实现 MPEG-4 视频流解码；
- XVIDFilte 架构符合 Microsoft Windows CE.NET DirectShow Architechture，以便在本课题设计的系统平台上使用；
- 符合系统主控模块的接口构架；

火龙果•整理
uml.org.cn

（2）Filter 模型选择

XVIDFilter 需要接收输入数据，经过处理后输出数据，其中需实现一个线程用于解码，一个线程用于解码输出。

（3）定义输入和输出

Tranform Filter 类型的 Filter 必须实现输入 Pin 和输出 Pin，以下定义输入 Pin 的数量，输出 Pin 的数量，以及各 Pin 上支持的媒体类型等。XVIDFilter 有一个输入 Pin，支持 xvid，divx 格式的视频数据；输出 Pin 支持 YUV420 格式的视频数据；同时，输入 Pin 还需要能够将传入的 Sample 数据组成一个连续的数据流，供解码线程读取。

（4）接口定义

接口是音频解码 Filter 与主调者系统主控模块之间的桥梁，通过接口可以得到 Filter 的属性、状态等参数，控制 Filter 的行为如初始化、开启、关断等。

（5）自定义功能

根据具体的应用进行拓展。

3.4.2 XvidFilter 的设计

根据功能分析，将所设计的 MPEG-4 视频解码 XvidFilter 命名为 CXvidFilter 类，选择 CSource 类作为其父类；另外定义输入 Pin 类名为 CXvidInputPin，输出 Pin 类名为 CDecodedStream，它们的继承关系如图 3.6 所示：

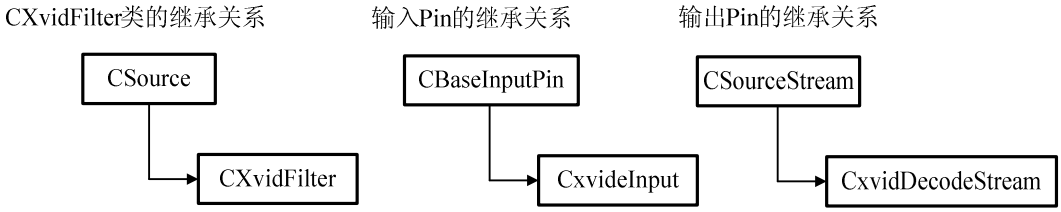

图 3.6　XvidFilter 应用框架的类继承关系

由于 Filter 是一个支持“输入－内部处理－输出”的典型过程的应用框架，如图 3.7 所示，在这个框架之下，更重要的是 Filter 内部处理的“应用逻辑”。因此，在进行设计时将 Filter 的“应用框架”与“应用逻辑”分离进行设计，具体含义如下：

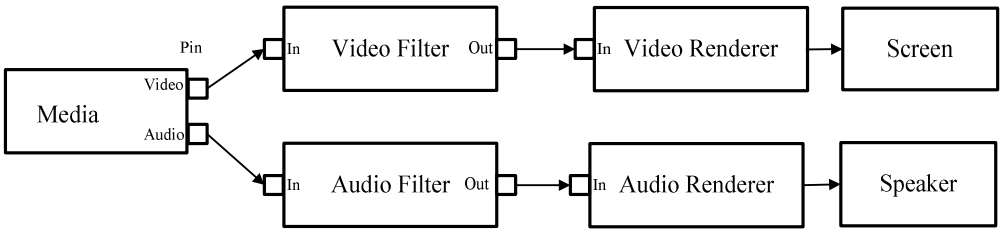

图 3.7　音视频解码 Filter 构建的 Filter Graph

火龙果•整理
uml.org.cn

- Filter 框架是指完成 Filter 功能的函数集合，这些函数包括实现输入 Pin 和输出 Pin 的媒体类型检查、决定 Sample 内存的大小等功能的纯虚函数、虚函数，还包括控制 Filter 类状态的函数，如初始化、反初始化函数等。由于基类 Filter 已将 Filter 框架以纯虚数或虚函数实现，因此 Filter 派生类可直接从基类继承获得。
- Filter 的应用逻辑是指对输入数据进行处理的一套控制策略，不关心如何输入、处理后又是如何输出的，这些问题由 filter 框架负责。Filter 应用逻辑需要关心的只是数据处理的算法实现、效率优化等问题。

3.4.2.1 XvidFilter 应用逻辑函数的实现

定义 CXvidFilter 的应用逻辑类 CXvidController 是一个 CXvidFilter 的对象实例，实现了 Xvid 格式的视频解码内核，该类定义如下：

```
Class CXvidController
{
private:
    long      mOutputImageSize;   //输出图像帧大小
public：
    CXvidController();
    ~ CXvidController();
public:
    bool Initialize(void);   //初始化
    bool Uninitialize(void);   //反初始化
    void SetOutputType(int inType);   //设置输出 Pin 的图像格式
    void SetOutputImageSize(long inImageSize);   //设置输出图像的大小

    void BeginFlush(void);   //开始数据对齐
    void EndFlush(void);
    void BeginEndOfSteam(void);
    void EndEndOfSteam(void);
    void FlushAllPending(void);   //使数据线程脱离循环等待状态

    void ReceiveXvid(unsigned char *inData, long inLength);
    //输入 Pin 调用该函数接收 Sample 数据
    void GetDecoded(unsigned char *outPicture);
```

火龙果•整理
uml.org.cn


```
    //输出 Pin 调用该函数取出解码后的图像帧数据

    Bool IsCacheInputWaiting(void);   //返回内部缓冲状态
    Bool IsCacheOutputWaiting(void);
    Bool IsCacheEmpty(void);
    void IsCacheInputWaiting(void);

    void SequenceHeaderChecking(void);   //检查 Xvid 码流的序列头
    Bool LocatePictureHeader(void);   //定位 Xvid 流的图片头
    Bool DecodeOnePicture(void);   //解码出一帧图像
}
```

由于 Xvid 算法是以 C 语言实现的，因此设计到 C 与 C++语言混合编程的问题，在 CXvidController.cpp 的文件头部，还必须加上如下定义：

```
#define GLOBAL
Extern "C"
{
#include "global.h"
#include "getbit.h"
#include "smartcache.h"
}
extern "C" int xvid_decore(void *handle, int opt, void *pararm1, void param2);
/*********************************************************/
```

CXvidController 类的核心是从 OpenDivxEncore2 的移植来的解码算法，调用 CXvidController 类解码一帧图像的流程如图 3.8 所示[43]-[47]。

火龙果•整理
uml.org.cn

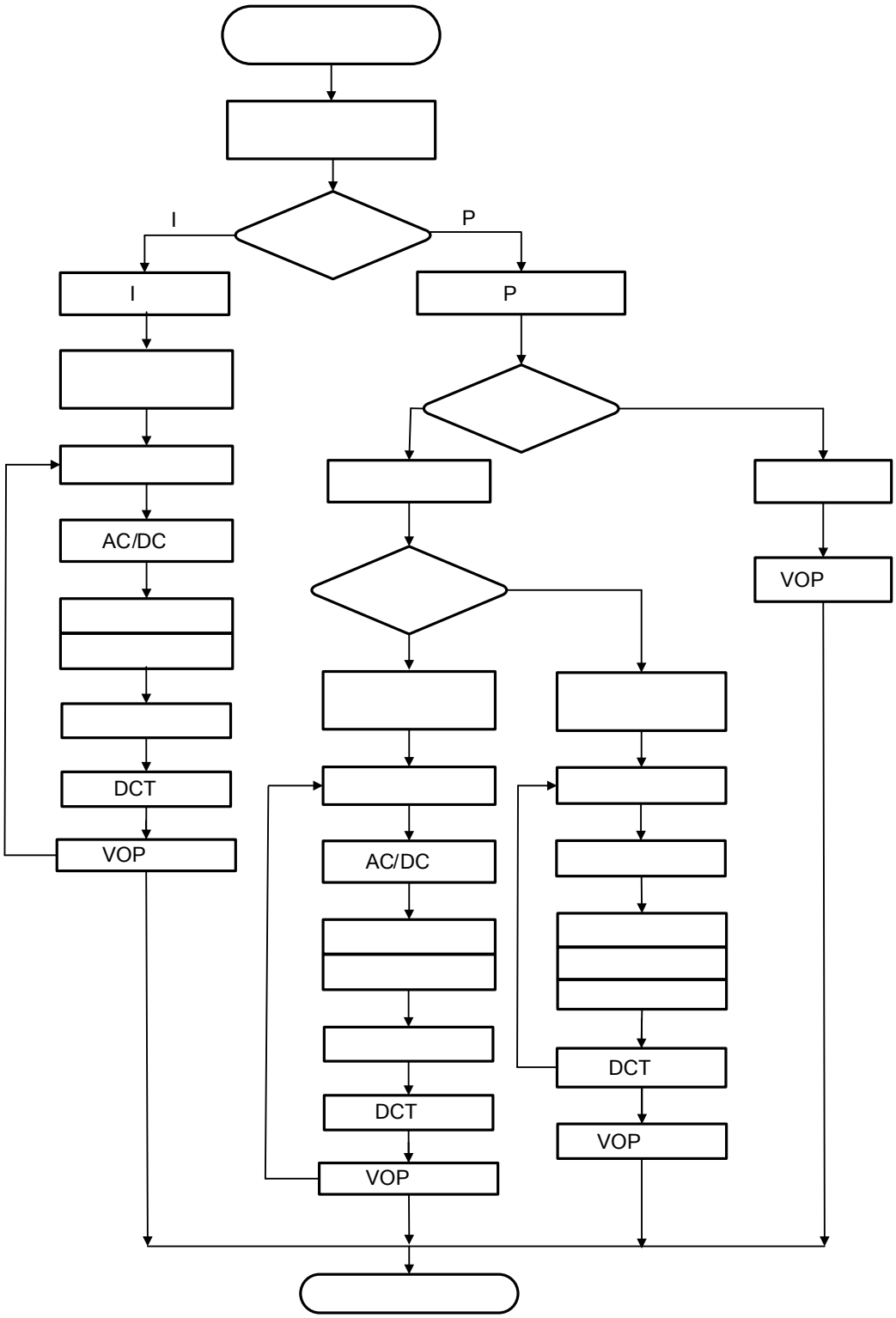

图 3.8　XvidFilter 解码一帧码流的流程图

3.4.2.2 XvidFilter 应用框架函数的实现

CXvidFilter 类的应用框架函数即是除去应用逻辑类以外的函数，以下对输入 Pin 类和输出 Pin 类上实现的重要函数进行分析，再对 XvidFilter 进行注册。

火龙果•整理
uml.org.cn

（1）输入 Pin 类 CXvidInputPin 重载了基类函数，如 CheckInputType 函数用于检查输入 Pin 连接的媒体类型，如媒体类型不匹配，返回错误信息。但其他函数如检查 Sample 中的媒体类型 Receive 函数、媒体流的控制函数 EndOfStream、EndFlush、NewSegment、获取媒体流参数并进行解码器初始化的 CompleteConnect 函数等都“委托”CXvidFilter 进行处理。

（2）输出 Pin 类 CDecodedStream 主要功能是控制整个 MPEG-4 的解码过程，主要实现的函数有：

CDecodedStream::DoBufferProcessingLoop()

//根据 CXivdFilter 的运行状态（播放或是暂停）接收未解码的码流

CDecodedStream::DeliverCurrentPicture()

//发送当前解码出来的图像帧 Sample

CDecodedStream::NonDelegatingQueryInterface()

//向 XvidFilter 上级 Filter 传递随机定位操作

（3）自定义接口：为了完成解码功能还必须实现 CreateInstance 以提供 COM 的入口，实现 NonDelegatingQueryInterface 以“暴露”支持的接口。

（4）生成 CLSID：使用 Guidgen 给 XvidFilter 生成一个 128 位的 ID 号，然后利用 DEFINE_GUID 宏在 Filter 的头文件声明该 Filter 的 CLSID：DEFINE_GUID(CLSID_MPEG4_Filter, 0xaff8f523, 0xdb02, 0x495c, 0xa1, 0xa1, 0x60, 0x5a, 0x50, 0x55, 0x0, 0xa5);

（5）Filter 的注册

Filter 在使用之前必须先注册。填写必须要的 Filter 注册信息，定义两个导出函数 DllRegisterServer 和 DllUnregisterServer，完成插件的注册和反注册。

通过 EVC 编译好 XvidFilter 插件后，可通过远程调试工具 Remote Registry Editor 查看 Windows CE 的注册表中的注册信息。

3.4.2.3 XvidFilter 开发总结

（1）解码线程中的解码内核设计成一个应用控制类，封装具体的解码函数实现。这样 Filter 框架只需要通过这个控制类交互，从而降低了编程的复杂度，并使得程序结构清晰。

（2）要为输入 pin 配置充足的缓冲区，当输入 Pin 接收 Sample 输入时，直接可将 Sample 数据放到这个缓冲区，等待解码线程读出，并从中取出足够的数据以完成图像帧的解码；若缓冲区不够大，数据包过小，解码器无法获取足够连续的数据，就不能流畅播放。由于 Filter 的输入 Pin 和输出 Pin 使用不同的线程，因此，还需要进行线程同步。

火龙果•整理
uml.org.cn

（3）由于 XVID 解码算法使用的是 C 代码，而 Filter 开发使用的是 C++代码，涉及到混合编程的问题。因此设计 C++风格的 Filter，将解码器上可能被调用的 C 函数在类中封装起来，供系统主控框架调用，既降低了编程复杂度，提高了开发效率，又使程序段保持了清晰的结构。

3.5 小结

本章设计了车载多媒体播放终端的应用软件部分，主要实现了多媒体播放功能。首先对多媒体播放功能需要完成的任务进行了需求分析，然后划分为系统主控模块、解码器模块、用户图形界面模块三个模块，并分别实现；利用 DirectShow 技术，将开源代码 XVID 的视频解码核心算法封装成了 CXvidFilter 类，实现了 MEPG-4 视频文件的流畅播放。

火龙果•整理
uml.org.cn

第四章 多媒体播放器的测试与优化

本章对多媒体播放器应用程序进行了测试和优化。首先，对媒体播放进行了基本功能测试，然后重点对音视频文件的播放性能进行了测试和评估；接着分析了测试结果，重点是针对解码器部分进行代码优化。优化工作先从高级语言层面入手，运用 C 语言编程技巧，针对性 PXA270 处理器的特殊性能，利用 Intel MMX 指令以汇编语言优化代码，最后获得了另人满意的播放效果。

4.1 功能测试

软件测试是发现软件中错误和缺陷的主要手段，功能测试部分需要完成的任务，是根据软件开发阶段的需求分析和各个模块的功能以黑盒测试的方法，测试是否完成设计目标，以及完成该项功能的主客观满意度。这是进行下步优化的根据。本节首先给出了各个模块的测试结果，然后展示了整体性能测试的结果，最后列出了性能测试的结果。

4.1.1 开机功能测试

系统上电启动，立即显示启动界面，然后进入系统功能主界面，车载多媒体播放终端开机界面如图 4.1 所示，功能主界面如图 4.2 所示，

图 4.1 开机界面

火龙果•整理
uml.org.cn

图 4.2　多媒体播放器功能主界面截图

测试结果：

（1）按键功能都可实现；

（2）上电启动 10 秒钟内即显示开机界面，从开机界面转入功能主界面时间不超过 5 秒。

4.1.2 无线调频发射功能测试

点击触摸屏上的“调频发射”，就可通过调频收音机接收到多媒体播放器输出的音频信号。测试时，根据该模块的设定，在 FM80MHz－FM108MHz 范围内，在 10 米范围都能够接收到主机的调频信号，并能清晰外放。

4.1.3 无线调频接收功能测试

点击触摸屏上的“调频收音”，进入调频接收机功能界面，如图 4.3 所示。用户通过点击触摸屏按键对应的位置可实现相应功能：“自动搜索”即系统自动搜索电台，通过拖动电台下的频率指示条或点击“《”“》”可定位电台频率，还可将搜索到电台点“存储”在预制频道栏中，最多可保存 6 个频道。

图 4.3　调频接收机功能界面截图

火龙果•整理
uml.org.cn

测试结果：

（1）所测功能都可实现；

（2）系统响应的延迟时间较短，一般不超过 2 秒钟；

（3）可搜索的电台范围大致从 80Hz 到 108Hz。

4.1.4 系统设置功能测试

系统设置界面如图 4.4 所示，通过点击对“《”“》”两个方向控键可实现对应的功能，系统将这些设置进行保存。

图 4.4　系统设置功能界面截图

测试结果：

（1）所测功能都可实现；

（2）系统响应的延迟时间较短，一般不超过 2 秒钟；

（3）能够保存上次对系统的设置。

4.1.5 多媒体播放功能测试

点击触摸屏上的“媒体播放”，进入多媒体播放功能界面。如图 4.5 所示，通过触摸屏点击“打开文件”选择播放影片；多媒体文件播放过程中，可完成“暂停”“播放”“停止”等功能；能够通过点击音量条调节系统音量，点击进度条随机定位实现“随机播放”等功能；选择上端功能键，可终止媒体播放返回主界面，或实现全屏播放。

测试结果：

（1）所测功能都可实现；

（2）系统响应时间在 3 秒内；

（3）能够流畅播放的 MPEG-4 影音文件。

火龙果•整理
uml.org.cn

图 4.5 播放 MPEG-4 影音文件截图

4.1.6 测试小结

在测试时间的 24 小时内，媒体播放终端持续工作且错误产生，并能够按照用户指令作出反应，界面美观简洁，易用性好。

以上结果表明：多媒体播放器满足设计要求，可保证长时间连续工作性能稳定，表 4.1 给出了整体性能的测试结果。

表 4.1 多媒体播放功能整体性能测试结果

预期功能	测试结果
车载音频外放功能	满足预期设计
车载无线调频接收机功能	满足预期设计
接收用户的播放、暂停、停止等请求	满足预期设计
流畅播放 MPEG-4 视频文件	满足预期设计
声音和图形要求同步	多数情况下音频和视频同步
视频播放流畅	满足预期设计
音质无明显抖动、播放流畅	满足预期设计
用户界面友好易用，美观	满足预期设计

4.2 性能测试与分析

由于存在其他因素，如：从 SD 卡中读取媒体文件的速率，显示缓冲区的大小，以及调试代码本身都会影响播放的性能，并且 MPEG4 视频解码部分由于图像的动静程度不同，解码耗时也不同，因此不同的测试文件会得到不同的测试结果。考虑到以上因素，媒体播放的测试结果并不代表 XviDFilter 解码器的实际效率。但这种方法测试的是播放器的实际播放效果，简单直观，能够反应出解码器 XviDFilter 的真实性能，因此其结果具有研究价值。

火龙果•整理
uml.org.cn

4.2.1.性能测试结果

下文所记录的测试结果是在下列条件下获取的：

- 测试工具：Intel® VTune Performance Analys 8.0.1
- 测试环境：处理器 Intel Xscale PXA270，最高主频：624MHz
  操作系统 Windows CE 4.2
  内存 SDRAM 容量 64MB*2
- 测试文件存储器：SD 卡，卡容量 1G

（1）播放一个 MPEG-4 格式的媒体文件，视频部分采用 Xvid 编码视频文件，音频部分采用 MP3 编码。为了测试视频解码的极限，忽略音视频同步处理，结果如表 4.2 所示。

表 4.2 视频最优播放性能测试结果

比特率	图像大小	目标图像	总帧数	平均每帧耗时 (ms)	平均帧率（fps）
1064kbps	624*336	800*480	15115	33.8	29.6
	480*260	800*480	13150	33.4	29.9
		480*260	13150	30.1	32.5
	352*288	800*480	13615	33.1	30.2
		352*288	13615	29.9	33.4

（2）播放一个多媒体文件，视频部分采用 Xvid 编码，音频部分采用 MP3 编码，并进行音视频同步处理，测试结果如表 4.3 所示。

表 4.3 音视频同步播放性能测试

比特率	图像大小	目标图像	总帧数	平均每帧耗时 (ms)	帧率 (fps)
1064kbps	624*336	800*480	15115	42.4	23.6
	480*260	800*480	13150	40.9	24.4
		480*260	13150	38.6	25.9
	352*288	800*480	13615	37.7	26.5
		352*288	13615	36.1	27.7

4.2.2 多媒体播放性能测试

对多媒体播放器进行性能测试的方法是：通过分析软件 VTune 和测试用例分析解码器运行时各个函数的执行时间，并参照此结果制定解码器的优化方案。本节首先介绍了性能测试使用的 VTune 性能分析工具，然后给出了测试环境和具体

火龙果•整理
uml.org.cn

指标，最后对测试结果进行了分析。

4.2.2.1 测试工具介绍[49]

Intel® VTune Performance Analys 是 Intel 提供的性能分析工具，能够从系统到特定功能、模块以及指令等方面搜集、分析以及显示软件性能，在这个工具的帮助下找出和消除软件性能的瓶颈。它采用无打扰的技术采样应用程序的活性，提供了多种分析技术，如通过采样（sampling），直接深入到源代码分析系统活动的结果；通过调用图(callgraph)，查看看到程序中函数的调用的关系，了解最消耗 CPU 时间的程序段；通过计数器监测器(counter monitor)，检查代码的某一时间片和性能计数器的关系，同时还能将分析结果以用图表显示出来。

4.2.2.2 测试工具配置

首先用网线将目标机（本地 PC）和多媒体开发终端连接起来，然后在目标机上安装 VTune Performance Analyze 并导入 Lincese，然后将 Sampling Driver 也即 vtune.0 模块拷贝到多媒体播放终端的 SD 卡中，启动 VTune server，然后运行 PC 上的 VTune Performance Analyze 主程序采集数据。

4.2.2.3 性能测试分析[38][48]

分析测试结果可知，解码流程中最消耗时间的模块是：视频解码器、音频解码器，图形缩放处理部分以及音视频同步代码段，其中视频解码部分约占解码时间的 60%以上。对 XvidFilter 的解码算法核心进行分析，估算出各个函数模块消耗的时间，结果如表 4.4 所示。

表 4.4　XvidFilter 解码函数模块执行时间的比例

函数模块	消耗解码时间的比例
反 DCT	28％
VOP 重建	20％
运动补偿	27％
VLC 可变长解码	10％
码流分解	6％
其他	9％

从运算量分析来看，计算密集型操作反 DCT 消耗 CPU 时间最多；从解码时间来看，运动补偿和 VOP 重建等涉及存储器访问的运算模块也消耗了相当多的时间。这说明，复杂的运算和数据存取操作对 MPEG-4 视频解码的性能影响最大。

火龙果•整理
uml.org.cn

4.3 性能优化的方法

根据上节的分析结果，应从两方面对解码器进行对优化：

（1）降低运算量

通过改进算法，减少占用周期数比较多的数据处理操作，比如使用乘法个数较少的反 DCT 快速算法、避免使用浮点运算、使用查表代替计算等方法。

（2）降低对存储器的访问

从优化软件结构和数据结构方面考虑，尽量减少对同一数据的多次存取，以及利用系统的存储器组织结构优化访问方式。

由于精力有限，本课题所进行的优化是在程序设计的层面上优化，没有引入新的解码算法。考虑到解码算法部分是由开源软件 XviD 的代码移植而来，其特点是采用 C 语言编写，优点是与平台无关，便于移植。但 C 语言的代码不如使用汇编代码的执行效率高。因此，可将部分执行效率较低的函数模块，使用 PXA270 处理器支持的 MMX 指令手工将部分算法改写为汇编代码。以下分别阐述。

4.3.1 针对 PXA270 平台的 MMX 指令优化

4.3.1.1 Intel MMX 指令的技术特点

MMX 技术是 Intel 公司为加速其微处理器在多媒体和通信中的应用而首先引入的，有以下特点：

（1）单指令多数据技术。这是 MMX 技术的关键，由于多媒体和通信中用到的数据类型通常是 8 位或 16 位的，利用单指令多数据技术，可以在一条指令内同时对多个数据操作，从而使其效率成倍增长。

（2）57 条新的指令。这些指令是针对多媒体和通信算法中固有的并行运算而引入的，其最大特点是对“打包”数据进行并行处理。57 条新指令包括以下几类：算术指令、比较指令、转换指令、逻辑指令、移位指令、数据传输指令以及 MMX 状态清空（EMMX）指令。

（3）8 个 64 位宽的 MMX 寄存器（MM0～MM7）。这 8 个 64 位的通用寄存器是和原有的浮点寄存器共用，它们分别对应 8 个 80 位浮点寄存器的低位。因此，当 MMX 指令和浮点运算指令切换时，在 MMX 指令结束以后浮点运算开始之前必需用 EMMX 指令将浮点寄存器的标志位（FP tagbits）清零。

（4）4 个新的数据类型。它们是：“打包”的字节（由 8 个字节打包而成的 64 位数据）、“打包”的字（由 4 个 16 位字“打包”而成的 64 位数据）、“打包”的双字（由 2 个 32 位的双字“打包”而成的 64 位数据）和四字（quadword）。

火龙果•整理
uml.org.cn

（5）更多的指令数据缓存。指令、数据缓存分别由原来的 8k 变成 16k。

MMX 技术针对 PXA270 处理器特殊的体系结构而设计，具有上述特点，但未引入任何异常和状态信息。因此，现有的操作系统就可以支持带 MMX 指令的应用。

4.3.1.2 优化方法[50]-[54]

MPEG-4 视频解码器系统常用到的 MMX 指令有：移动、逻辑位运算、加减、乘、交织等。使用 MMX 技术进行 8 个 64 位宽的 MMX 寄存器（MM0～MM7）的并行处理，可用于优化反 DCT 模块的实现、运动补偿模块的实现和预测差值模块的实现。

举例说明，在运动估计中，由于简单有效以及易于大规模集成而广泛使用块匹配算法，其原理是将当前帧分成若干两维的 N×N 个像素的小块（如 N＝16），在运动估计中整个小块被看成一个单元，用它来和参考帧内的块匹配。匹配的准则使用绝对误差和准则（SAD），用公式（4.1）表示是：

$$SAD(x,y)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|I_n(i,j)-I_{n-1}(i+x,j+y)\right| \qquad (4.1)$$

由于像素点是 8 位的，利用 MMX 技术单指令多数据的特点，先对数据进行“打包”，然后在一条指令内并行处，代码如表 4.5 所示：

表 4.5　SAD 算法的 MMX 指令

指令	含义
MOVQ MM0, PTR0	PTR0 为当前宏块数据对应的地址， 该语句将数据打包传入 MM0 中
MOVQ MM1，PTR1	PTR1 为参考数据块对应的地址， 该语句将数据打包传入 MM1 中
MOVQ MM2, MM1	将 MM0 中数据拷贝一份， 存入 MM2 中备用
PSUBUSB MM0, MM1	MM0，MM1 中对应字节进行无符号数相减， 并将其结果钳位在无符号数范围内， 然后再存入 MM0 中
PSUBUSB MM1, MM2	功能同上
POR MM0，MM1	MM0 和 MM1 逻辑位或，并将结果存入 MM0 中
MOVQ MM2, MM0	将 MM0 中的数据拷贝一份，存在 MM2 中备用
PSRLW MM0, 8	将 MM0 中打包成字的数据逻辑右移 8 位， 结果存入 MM0

火龙果•整理
uml.org.cn

PSRLW MM2, 8	将 MM2 中打包成字的数据逻辑右移 8 位，结果存入 MM2
PADDW MM2, MM0	将 MM2，MM0 中对应字相加，并将结果存入 MM2

使用 C 语言编写的代码至少需要 16 行代码，而使用 MMX 技术只需要 11 条指令，却可使程序执行速度提升 2 倍以上。另外，在改造算法的过程中，还可使用 MMX 技术尽可能的提高内存访问的容量，以提高程序的执行效率，如：

```
//原代码段:
for (j=0; j<h; j++)
{
    d[0] = s[0];
    d[1] = s[1];
    d[2] = s[2];
    d[3] = s[3];
    d[4] = s[4];
    d[5] = s[5];
    d[6] = s[6];
    d[7] = s[7];
    d[8] = s[8];
    d[9] = s[9];
    d[10] = s[10];
    d[11] = s[11];
    d[12] = s[12];
    d[13] = s[13];
    d[14] = s[14];
    d[15] = s[15];
    s+= lx2;
    d+= lx;
}
```

```
//运用 MMX 技术，以汇编语言代替 C 的语言改写后的代码段：
__asm{
    pushf
    mov     edx,dword ptr h
    xor  ecx,ecx
    mov esi,dword ptr s
    mov     edi,dword ptr d
    mov     eax,lx2
    mov ebx,lx AGAIN:
    movq    mm0,byte ptr [esi]
    movq    mm1,byte ptr[esi+8]
    movq    byte ptr [edi],mm0
    movq    byte ptr [edi+8],mm1
    add     esi,eax
    add     edi,ebx
    add     ecx,1
    cmp     ecx,edx
    jl      AGAIN
    emms
    popf
}
```

将 8 位的写，改为 64 位的写，测试结果表明程序运行速度提升了 25%。同样的道理，在将 C 语言的代码改成汇编实现代码时要尽可能的将几个 movq 写在一起，这样做可以提高 5%左右的速度。

当然，MMX 技术并不是总能提高代码处理速度，如在传统的代码优化的方

火龙果•整理
uml.org.cn

法中，构造数组，然后将运算变为查表的方法，有的时候在 MMX 技术内反而会降低速度。这是由于使用 MMX 指令中，“访问寻址的内存导致程序执行效率下降”的速度大于“数据不对齐 8 位的速度下降”，大于“指令不配对的速度下降”。因此，尽可能的避免不去访问内存，而直接在寄存器中完成操作。

4.3.2 程序的 C 语言优化

在运用 Intel MMX 指令优化后，要进一步提高软件性能，需要对代码进行高级语言层次上的优化，主要是 C 和 C++语言的优化。处理器 PXA270 是基于 XScale 微架构技术的，采用的是 ARM V5TE 的体系。根据 ARM V5TE 的体系特点，和其流水线和缓存状况，调整高级编程语言的代码，提高编译效率需要注意以下几点。

（1）数据类型：为了避免不同编译器对数据类型定义上不同，带来处理上的分歧，将数据类型按类型定义（typedef）在一个头文件中，当移植时修改头文件，例如：

```
typedef unsigned int    UINT32;
typedef unsigned int    INT32;
typedef unsigned short    UINT16;
typedef unsigned int    INT16;
```

（2）高效地编写循环体：

- 使用减计数到零的循环结构。这样做的好处是编译器不需要分配寄存器来保存循环终止值，可以省略使与零比较的指令；
- 使用无符号的循环计数值，用(i!=0)替换循环继续的条件(i>0)，这样可以保证循环开销只有两条指令；
- 如果事先知道循环体至少会执行一次，使用 do-while 循环优于用 for 循环好，使编译器省去检查循环计数值是否为零的步骤；
- 在关键的循环中，避免使用全局变量和静态变量，使用局部变量；
- 展开重要的循环体，使每次循环能够充分利用 CPU 资源。以一个数据块拷贝函数为例，循环展开方法如下所示：

```
 void transfer-8to16copy c(int16_ t*const dst,const uint8_ t*const src,
                                        uint32 t stride)
{
int i, j;
for (j=O;j<8; j++){
      fir (i=0; i<8; i++){
```

火龙果•整理
uml.org.cn


```
            dst}j*8+i}=(intl6 t) src }j*stride+i];
                }
            }
        }
```

该函数中，内外都有 8 个循环迭代，将内循环展开，展开后的代码如下：

```
void transfer-8to16copy c(int16_ t* const dst, const uint8_t* const src,
                                uint32 t_stride)
    {
    int temp0,templ, j;
    for (j=0;j<8; j++){
        tempo j*8;
        temp 1 j * stride;
        dst[tempo+0]=(int16_ t) src [temp 1+0];
        dst[tempo+1]=(int16_ t) src [temp 1+1];
        dst[tempo+2]=(int16_ t) src [temp 1+2];
        dst[tempo+3]=(int16_ t) src [temp 1+3];
        dst[tempo+4]=(int16_ t) src [temp 1+4];
        dst[tempo+5]=(int16_ t) src [temp 1+5];
        dst[tempo+6]=(int16_ t) src [temp 1+6];
        dst[tempo+7]=(int16_ t) src [temp 1+7];
    }
```

（3）减少函数调用的次数。函数的嵌套调用对系统资源的要求过高，由于函数的调用是一个跳转过程，不仅要传递参数，还要保护跳转前的程序状态。因此对于操作简单，调用次数不多的函数，直接展开放在主调函数中实现，不仅可以减少参数的传递，避免了跳转过程。

4.3.3 优化结果

利用经过上述优化过的代码，采用相同的测试环境和相同的音视频文件进行测试，获取数据如下：

（1）播放一个 MPEG-4 格式的媒体文件，视频部分采用 Xvid 编码视频文件，音频部分采用 MP3 编码。为了测试视频解码的极限，忽略音视频同步处理，结果如表 4.6 所示。

（2）播放一个多媒体文件，视频部分采用 XVID 编码，音频部分采用 MP3 编码，让播放器自动选择解码器，测试数据如表 4.7 所示。

火龙果•整理
uml.org.cn

表 4.6　性能优化后的视频最优播放测试

比特率	图像大小	目标图像	总帧数	平均每帧耗时(ms)	帧率(fps)	提升
1064kbps	624*336	800*480	15115	28.1	35.7	20.6%
	480*260	800*480	13150	28.4	35.2	17.7%
		480*260	13150	26.4	37.8	16.3%
	352*288	800*480	13615	27.6	36.2	19.9%
		352*288	13615	25.9	38.6	15.6%

表 4.7　性能优化后的音视频同步播放测试

比特率	图像大小	目标图像	总帧数	平均每帧耗时(ms)	帧率	提升
比特率 1064kbps	图像大小	目标图像	总帧数	平均每帧耗时(ms)	帧率(fps)	提升
	624*336	800*480	15115	37.7	26.5	12.3%
	480*260	800*480	13150	36.8	27.2	11.3%
	352*288	480*260	13150	35.3	28.3	9.0%
	352*288	800*480	13615	34.2	29.2	10.3%

测试数据表明，通过算法优化和程序优化确实在一定程度上提高了多媒体播放器的性能，约提升了 10%左右。视频最优测得的性能提升高于音视频同步播放测得数据，这是因为影响播放性能的主要因素是解码器的算法，但系统从 SD 卡中获取数据、图像显示处理、音频同步等处理也会影响播放器的性能。因此，播放器性能的全面提升还需要综合考虑各种因素。

4.4 小结

本章对多媒体播放器的进行了功能测试，结果表明多媒体播放器能够通过触摸屏接收用户指令，并可在较短的时间内进行响应；无线调频模块能够在十米左右的范围内连接车载音频功放，选配的无线调频接收模块能够实现自动/手动选台、储存、加载预制频道，能够流畅播放 MPEG-4 视频文件，图形用户界面美观实用，满足了设计要求。本章重点对多媒体播放器进行了性能测试，并对视频解码器 XvidFilter 的核心算法进行了分析，结合硬件平台 PXA270 处理器的特性，使用 MMX 技术和 C 语言编程技巧对解码器 XvidFilter 进行了算法优化和程序优化，并使播放器的性能在一定程度上得到了提升。

火龙果•整理
uml.org.cn

第五章 总结与展望

5.1 已完成的任务及亟待解决的问题

本文设计实现了一种适于车载环境使用的多媒体播放器软件，运行在以PXA270 处理器为核心的硬件平台和 Windows CE.NET 操作系统上。主要完成的任务有：

（1）重新编译了 Windows CE.NET 嵌入式操作系统内核，为系统定制了启动画面，并实现了多媒体播放器应用程序的自动加载。

（2）设计了以单片机为控制核心的无线调频发射模块，使多媒体播放器能够连接汽车原配的音频功放，并使用同一控制核心实现了 FM 接收功能。

（3）采用 DirectShow 技术，设计了模块化的多媒体播放器结构，实现了结构清晰的系统主控模块程序设计和简洁实用的图形界面设计。

（4）采用 COM 组件技术实现了 MPEG-4 解码 Filter，使本系统能够流畅播放 MPEG-4 视频文件。

（5）对多媒体播放的功能进行测试，并对解码器进行了性能分析和优化。

系统亟待解决的问题有：

（1）由于多媒体播放功能使用的是软件解码，PXA270 的功耗较大，不适合使用电池供电;因此可考虑使用 PXA270 的升级产品——配有协处理器的 PXA320 处理器作为替代，则视频播放时会更加流畅。

（2）多媒体文件主要从 SD 卡中获取，下一步考虑通过无线通信技术从 Internet 上获取多媒体资源。

（3）图形用户界面还较为简单，由于时间关系，只提供了桌面配色界面，没有开发多套可供选择的皮肤，在后续开发中还需要设计更多风格的整套界面供用户选择。

5.2 发展前景

未来的车载影音播放系统在性能上将更接近家用影音系统，集成音视频、导航器、远程信息处理为一体，能够支持以往由地面波发送的收音机和电视广播、导航系统使用的卫星信息，卫星数字广播发送的数字媒体等，兼容更多的技术标准和接口，并将导入宽带车内网络。随着人们活动范围的扩大，车辆将由单纯的交通工具将集成更多功能的无缝移动信息系统。

火龙果•整理
uml.org.cn

致 谢

值此论文完成之际，我要对我的导师高明煜教授表示深深的谢意。他渊博的专业知识，严谨的治学态度，精益求精的工作作风，诲人不倦的高尚师德，朴实无华、平易近人的人格魅力使我终生受益。高老师不但为我们营造了良好的研究氛围，提供了优越的实验条件，还督促我们不断探索、积极实践，使我们可以充分发挥自己的能力，全面锻炼和提升自己的能力。正是他孜孜不倦的教诲、循循善诱的教导、由浅入深的任务布置以及理论联系实践的教学方法指导我一步步走上科研岗位。他对学术和工作上近乎苛求的勤勉一直激励着我。在硕士论文的撰写过程中，高老师倾注了大量心血，指导我最终顺利完成。所有这些都令我永生难忘！在此，我要向他表示深深的敬意和衷心的感谢！

衷心感谢黄继业、曾毓两位老师三年来对我的帮助和指导。黄继业老师实践经验丰富，善于用机智风趣的言语讲解复杂的专业知识，他扎实的硬件功底和百科全书般的专业修养使我收获颇丰。曾毓老师勤勉的工作态度和软件造诣令人敬佩，他在课题研究期间给予我很多的帮助。

衷心感谢何志伟老师对我的无私帮助与指导。何老师渊博的学识和忘我的工作精神给我留下了深刻印象。在我论文的写作过程中，他不但要完成科研任务，照顾即将分娩的妻子，还花费了大量时间和精力，一次又一次的阅读我的拙作，给了我许多宝贵的建议，使我的论文一点一滴地完善。再一次对何老师表示衷心的感谢！

我还要感谢师兄陈吉锋、徐超群、同学吴锦跃、陆杰、张思恩以及俞翔、刘伟等实验室所有的师弟师妹们，大家一起学习、讨论，创造了良好的科研氛围；真诚的感谢我的室友叶淑霞和同学蒋立飞、章潋，他们陪伴我度过了本科到研究生阶段的七年时间，让我时刻感受到友谊的温暖。

特别感谢我的父母，他们给了我无私的爱和关怀，是我最坚强的后盾。

最后，对所有参加论文评审和对本文提出宝贵意见的各位专家、教授以及老师表示衷心的感谢。

火龙果•整理
uml.org.cn

参 考 文 献

[1] 北京水清木华科技有限公司(2006). 2005-2006 年中国汽车电子产业链研究报告.

[2] 方明，裴昌幸，陈南，易运晖(2006).基于 AMD Alchemy Au1200 嵌入式处理器的媒体播放器设计.《现代电子技术》.(2)

[3] 张西光，吕志(2002).车载自主导航系统的现状和发展趋势.《导航》.

[4] 常青,杨东凯,寇艳红, etc(2005). *车辆导航定位方法及应用*.机械工业出版社.

[5] 田根，童小华，张锦（2004）. 基于 PDA 的潜入式 GIS 结构体系.《测绘工程》,13(2):15－18.

[6] Bishop Richard(2000). Intelligent Vehicle Applications Worldwide, IEEE Intelligent System, Vol. 15, pp:78-81.

[7] TOM IYAMA H, CH IKADA S, HONDA S(2005). An RTOS based Approach to Design and Validation of Embedded System[A]. IEEE International Symposium on VLSI Design, Automation and Test[C].185-187.

[8] Zhiyong Xu, Rui Min, Yiming Hu: An Analysis of Cache Performance of Multimedia Applications, IEEE Transactions on Computers, VOL.53,NO.1,2004

[9] JIN-HWAN JEONG, CHUNK YOO: Software-Based Video/Audio Processing for Cellular Phones, Telecommunication Systems 28:2, 185-210, 2005

[10] Intel Intel PXA27X Processor Family Developer's Manual.Intel(2004).

[11] Intel PXA27X Processor Family Developer's Kit.April 2004.Intel(2004).

[12] Intel PXA27X Processor Family Video Decoder Display Performance Optimization.

[13] Intel Corporation(2005).Intel(R) PXA27x Processor Family Video Decoder Display Performance Optimization Aplication Note.

[14] Freescale(2004). MC68HC908QL4 Microcontrollers Datasheet.

[15] 马殷元(2004).基于新型单片机 P89C51RD2 的倒车雷达设计.甘肃科技.（9）：32－34.

[16] 嵌入式研究网，傅曦，陈黎等（2006）.Windows CE 嵌入式开发入门——基于 Xscale 架构.人民邮电出版社.

[17] Microsoft Windows CE .NET 4.2. www.Microsoft.com.

火龙果•整理
uml.org.cn

[18]刘大鹏，马孝江(2002).基于 WindowsCE 的系统平台定制.《微型电脑应用》.(8)

[19]陆永忠，刘峰，陆永祥(2005).Windows CE. NET 在高速实时数据采集系统中可移植性的研究.《微电子学与计算机》.（7）

[20]周毓林，宁 杨，陆贵强等(2004). *Windos CE.net 内核定制及应用开发*. 电子工业出版社.

[21]张冬泉，谭南林，王雪梅等(2006). *Windows CE 实用开发技术*. 电子工业出版社.

[22]傅曦，陈黎，董磊鋆等(2006). *Windows CE 嵌入式开发入门—基于 Xscale 架构*. 人民邮电出版社.

[23]Microsoft Windows CE.NET Help. Microsoft Corp. 2003.

[24]Microsoft. Microsoft Windows CE Device Driver Kit.

[25]Microsoft. Microsoft Windows CE.Net 5.0 Documentation.

[26]陆其明(2005).*DirectShow 开发指南*.清华大学出版社.

[27]李博轩(2000).*Visual C++ 6.0 多媒体开发指南*.清华大学出版社.

[28]Embedded Visual C++ Version 4.0 help. Microsoft Corp. 2001.

[29]侯捷(2001).*深入浅出 MFC*.华中科技大学出版社.

[30]郭玉东,祁献鹏(2001).MiniGUI 一面向嵌入式系统的 GUI 系统.《信息工程大学学报》.(3).

[31]吴峰，王自强(2005).基于 FrameBuffer 的嵌入式 GUI 系统设计. 《计算机应用与软件》.(3).

[32]钱华锋，雷航(2004).面向对象嵌入式 GUI 研究和模式应用.《计算机应用》.(4).

[33]詹瑾瑜、熊光泽、孙明(2003).一种嵌入式 GUI 软件结构实现方案.电子科技大学学报》.(l).

[34]吴峰，王自强.基于 FrameBuffer 的嵌入式 GUI 系统设计.《计算机应用与软件》.(3).

[35]邓丽娜，顾伟楠(2006).基于 DirectShow 的媒体播放器的设计与实现.《计算机应用与软件》.(1).

[36]（美）Corry，Mayfield，Cadman 著. 刘云，孔雷译. *COM/DCOM 编程指南*. 清华大学出版社.

[37]Jerry D.Gibson 等(2000).*多媒体数字压缩原理与标准*.电子工业出版社.

[38]Feng P, Li Z, Lim K, etc(2003). A study of MPEG-4 rate improvements. IEEE Trans On Circuits And control scheme Systems Forand its Video Technology.(13): 440-446

[39]沙丽娜, 王宁章, 覃团发(2005). 视频压缩编码国际标准综述. 电子科技.

[40]胡晓峰,吴玲达,老松杨(2005). 多媒体技术教程（修订版）. 人民邮电出版社.

[41](美)Tay Vaughan 著.晓波,倪敏译(2004). 多媒体技术及其应用（第六版）. 清华大学出版社.

[42]Shu-Ching Chen，Mei-Ling Shyu(2002). Multimedia Data Mining for Traffic Video Sequences[J]. Journal of Intelligent Information Systems. 19(1):61-77.

火龙果•整理
uml.org.cn

[43]蔡安妮，孙景鳌 (2000).*多媒体通信技术基础*.电子工业出版社.

[44]钟玉琢,王琪,贺玉文(2000).基于对象的多媒体数据压缩编码国际标准—MPEG4 及其校验模型.科学出版社钟玉琢,王琪,贺玉文(2000).基于对象的多媒体数据压缩编码国际标准—MPEG4 及其校验模型.科学出版社.

[45]ISO/IEC 14496-2:2001(Code of Audio-visual Objects-Part 2:Visual).MPEG N4530,2001.

[46]Fernando Pereira, Touradj Ebrahimi: The MPEG-4 Book, Prentice Hall PTR,2002

[47]Iain E.G.Richardson: Video Codec Design, John Wiley & Sons, 2002.

[48]Zhiyong Xu, Rui Min, Yiming Hu: An Analysis of Cache Performance of Multimedia Applications, IEEE Transactions on Computers, VOL.53,NO.1,2004.

[49]VTune(TM) Performance Analyzer 8.0 for Windows* Release Notes, http://www.intel.com/software/products/college/gs_vtune/index.htm.

[50]JIN-HWAN JEONG, CHUNK YOO: Software-Based Video/Audio Processing for Cellular Phones, Telecommunication Systems 28:2, 185-210, 2005.

[51]王伟波，李善平(2005).WMMX 技术及应用.《计算机工程与设计》.(9) .

[52]Intel Corporation: A Fast Precise Implementation of 8*8 Discrete Cosine Transform Using the Streaming SIMD Extentions and MMX^{TM} Instructions, 1999.

[53]Intel.MMX Technology Programmers Reference Manual.Intel,1996.

[54]Intel.MMX Technology Developers Guide.Intel,1996.

火龙果•整理
uml.org.cn

附 录

作者在读期间发表的学术论文及参加的科研项目

1. 尹克，高明煜，陈吉锋(2007). 车载可视倒车装置的设计与实现.《杭州电子科技大学学报》，27(4):1-4.
2. 车辆行驶综合信息系统. 本项目为浙江省重大科技攻关项目，项目编号：2005C11019.