

摘 要

为了提高车间层员工的生产操作技巧,进而提高车间所加工产品的质量及工艺性能,本文提出了一种嵌入式视频生产示教系统,它是集嵌入式技术、多媒体技术和 GUI 技术于一体的装置,能够实现包括视频在内的多媒体信息的实时传输,示教和交互。

本文第一部分首先介绍了课题提出及其意义,对嵌入式系统的应用领域、多媒体信息技术的研究现状、GUI 技术的研究现状、生产示教系统的技术应用现状进行了分析,接着,本文说明了系统的课题来源,阐述了论文所研究的主要内容。

本文第二部分首先对嵌入式视频生产示教系统的总体需求进行了分析,阐明了嵌入式视频生产示教系统的总体运行模式及其功能模块的划分,接着,阐述了本文所设计系统的硬件架构和软件架构。

本文第三部分首先论述了系统在开发过程中所涉及到的关键技术问题、对基于 Au1200 的音视频编解码技术进行了研究、对 QT 技术在系统播放器界面开发过程中的相关技术问题进行了分析,接着阐述了系统实时示教编解码器的原理以及视频实时示教优化的必要性和可能性,对视频编解码器的算法优化进行了研究。

本文第四部分首先介绍了系统的开发工具和开发环境,对嵌入式视频生产示教终端的一体化播放界面开发、视频播放控制功能开发的关键技术问题实现过程进行了阐述,重点针对系统实时示教功能实现过程中,资源占用率过高、播放卡壳、音视频不同步的问题、对音视频编解码器进行了相关优化,从而使系统的整体性能得到了有效提高,最后,本文演示了系统的实现效果。

关键词: 生产示教, 嵌入式, QT 技术, 实时示教, 多媒体信息

ABSTRACT

In order to improve the operative skills of workshop's staff in manufacturing, the quality and processing property of converted products, this thesis puts forward the system of teaching in video manufacturing based on embedded system, it's a equipment which is synthesized by the embedded technology, the multimedia technology, and the GUI technology, it can achieve the real-time transmission, real-time teaching, real-time interactive of multimedia information including the video information.

First of all, this project and its significance are introduced in the first portion, the application area of embedded system, the situation of study about mutimedia, the situation of study about GUI technology and the situation of technology application about the system of teaching in manufacture are analyzed, and then, the source of this project are illustrated, the main research contents of this paper are expounded.

The demand of the system of teaching in video manufacturing is analyzed firstly in the second portion of this paper, the overall operational mode and the division of functional module are clarified, and then, the hardware infrastructure and software infrastructure of the system of teaching in video manufacturing are designed and expounded.

The key technologies which are involved in the process of system development are treatised firstly in the third portion, including the research in encoding and decoding about audio frequency and video technology, the QT technical matters in the process of player skin design are analyzed, and then, the necessity and possibility of optimization in video display is illustrated, after that, this paper makes a study in algorithm optimization about the technololgy in video decoding.

In the fourth portion of this paper, the development tool of this system and the development environment are introduced in the first place, the all-in-one player skin development of the system of teaching in video manufacturing, the achievement process of the key technologies involved in the control fuction's realization are expounded. In the development process of Real-Time video displaying, when the high resolution videos are displayed, the video would be stumped, the system resources occupancy rate will be too high, if this type of video is displayed for a long time, the system is easy to crashed. In order to solve this problem, this paper takes some optimization measure to the process of video decoding, after that, the performance of the system have been

effectively improved a lot. At last, this paper demonstrates the application effect of this system.

Keywords : Teaching In The Process of Production, Embedded, QT Technology,
Real-time Teaching, Multimedia Information

学位论文独创性声明

本人声明所呈交的 硕 士学位论文
《一种嵌入式视频生产系统的设计与实现》 是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名: 秦大鹏

签字日期: 2009.6.5

导师签名: 朱

签字日期: 2009.6.5

学位论文使用授权书

本人完全了解重庆大学有关保留、使用学位论文的规定。本人完全同意《中国博士学位论文全文数据库、中国优秀硕士学位论文全文数据库出版章程》(以下简称“章程”),愿意将本人的 硕 士学位论文 《一种嵌入式视频生产系统的设计与实现》 提交中国学术期刊(光盘版)电子杂志社(CNKI)在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》以及《重庆大学博硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版,并同意编入 CNKI《中国知识资源总库》,在《中国博硕士学位论文评价数据库》中使用和在互联网上传播,同意按“章程”规定享受相关权益和承担相应义务。本人授权重庆大学可以采用影印、缩印或其他复制手段保存论文,可以公开论文的全部或部分内容。

作者签名: 秦大鹏

导师签名: 朱

2009 年 6 月 5 日

备注: 审核通过的涉密论文不得签署“授权书”,须填写以下内容:

该论文属于涉密论文,其密级是 _____, 涉密期限至 _____ 年 _____ 月 _____ 日。

说明: 本声明及授权书 必须 装订在提交的学位论文最后一页。

1 绪 论

1.1 课题的提出及意义

制造业是国民经济的支柱产业，是衡量一个地区经济综合实力和现代化水平的重要指标，强大的制造业是实现工业现代化的可靠保证，在经济发展中占有十分重要的地位^[1]。世界经济发展过程表明，制造业在国民经济中占有举足轻重的地位，工业化国家的财富有60%-80%是制造业创造的。随着经济全球化和科学技术的迅猛发展，全球制造业呈现出三个特征：一是技术进步和技术创新成为制造业产业结构升级的主要推动力；二是技术竞争逐渐取代了产品竞争而成为制造业产业的主要竞争方式，发达国家依靠所掌握的核心技术占据了产业链的前端，形成了比较强的产业竞争优势；三是制造业产业转移趋势明显，全球制造业正由发达国家向发展中国家转移，新的产业分工体系逐步形成，发达国家凭借技术优势在分工体系中占据了有利的地位^[2]。

七十年代以来，随着信息时代的来临、经济的全球化、市场客户化、知识经济的挑战、高速发展的高科技的推动以及新思想、新技术不断涌现。这些新思想、新技术对制造业产生了极其深刻和全面的影响，使制造业的制造模式发生了深刻变化，现代制造业及制造技术逐渐表现出数字化、敏捷化、网络化、虚拟化、智能化、绿色化、信息化、集成化以及客户为中心的服务化的发展趋势^[3-12]。

目前，在制造企业的工厂层与设备层之间、企业内部的员工之间、信息主要以文字或者图片的方式进行传递、企业内的示教活动大多是通过图纸、电子看板等媒介，以集体教学的方式进行，音频信息的交互主要是通过电话的方式进行，视频信息由于成本比较高，在制造企业中应用的还不广泛。但是随着时代的发展，在企业生产管理的现实中，企业的战略层、战术层和运行层之间交流的，往往更加需要的是以音视频、图像、文字相结合的多媒体信息，所以工厂层和车间层都对示教信息的丰富性和易理解性提出了更高要求，因此也就对生产示教活动的多媒体信息支持能力提出了更多要求。为了更好的实现制造企业整个生产过程中的各种信息的传输、示教与交互，需要开发一种集音频、视频、图片、文字等多媒体信息为一体的嵌入式视频生产示教系统终端，该系统通过网络和工厂层相连，它可以及时的从服务器下载各种多媒体信息到终端，实现在示教终端的多媒体信息播放，并可以通过网络完成实时示教的功能，方便车间层员工之间进行更加有效沟通和讨论，方便员工实时的查阅相关生产操作流程、生产工艺流程、图纸、文档资料等多媒体信息，这对于提高员工生产操作的准确性、产品的质量和工艺性能、以及生产操作流程的可靠性都有着重要的意义。

1.2 国内外研究现状

1.2.1 嵌入式系统的应用现状

嵌入式系统即嵌入式计算机(Embedded Computer)及其应用,是指嵌入在各种设备及应用产品内部的计算机系统。嵌入式系统是随着计算机技术、微处理器技术、电子技术、通信技术、集成电路技术的发展而发展起来的,经过几十年的发展,嵌入式系统已经在很大程度改变了人们的生活、工作和娱乐方式,而且这些改变还在加速。嵌入式系统广泛应用于工业自动化、国防、交通和航天,消费类电子等领域。目前,对嵌入式系统的定义多种多样,但没有一种定义是全面的。下面给出两种比较合理的定义^[13-15]:

① 从技术的角度定义:以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

② 从系统的角度定义:嵌入式系统是设计完成复杂功能的硬件和软件,并使其紧密耦合在一起的计算机系统。术语嵌入式反映了这些系统通常是更大系统中的一个完整的部分,称为嵌入的系统。嵌入的系统中可以共存多个嵌入式系统。嵌入式系统主要由嵌入式处理器、相关支撑硬件和嵌入式软件系统(嵌入式操作系统与应用软件)构成,是集成软硬件于一体的可独立工作的系统,且软硬件根据应用的需要可裁剪。

嵌入式系统技术具有非常广阔的应用前景,它的应用领域主要包括以下几个方面^[16-23]:

① 工业控制:

基于嵌入式芯片的工业自动化设备将获得长足的发展,目前已经有大量的 8、16、32 位嵌入式微控制器在应用中,网络化是提高生产效率和产品质量、减少人力资源主要途径,如工业过程控制、数字机床、电力系统、电网安全、电网设备监测、石油化工系统。就传统的工业控制产品而言,低端型采用的往往是 8 位单片机。但是随着技术的发展,32 位、64 位的处理器逐渐成为工业控制设备的核心,在未来几年内必将获得长足的发展。

② 交通管理:

在车辆导航、流量控制、信息监测与汽车服务方面,嵌入式系统技术已经获得了广泛的应用,内嵌 GPS 模块, GSM 模块的移动定位终端已经在各种运输行业获得了成功的使用。目前 GPS 设备已经从尖端产品进入了普通百姓的家庭,只需要几千元,就可以随时随地找到你的位置。

③ 信息家电:

这将称为嵌入式系统最大的应用领域,冰箱、空调等的网络化、智能化将引

领人们的生活步入一个崭新的空间。即使你不在家里,也可以通过电话线、网络进行远程控制。在这些设备中,嵌入式系统将大有用武之地。

④ 家庭智能管理系统:

水、电、煤气表的远程自动抄表,安全防火、防盗系统,其中嵌有的专用控制芯片将代替传统的人工检查,并实现更高,更准确和更安全的性能。目前在服务领域,如远程点菜器等已经体现了嵌入式系统的优势。

⑤ POS 网络及电子商务:

公共交通无接触智能卡(Contactless Smartcard, CSC)发行系统,公共电话卡发行系统,自动售货机,各种智能 ATM 终端将全面走入人们的生活,到时手持一卡就可以行遍天下。

⑥ 环境工程与自然:

水文资料实时监测,防洪体系及水土质量监测、堤坝安全,地震监测网,实时气象信息网,水源和空气污染监测。在很多环境恶劣,地况复杂的地区,嵌入式系统将实现无人监测。

⑦ 机器人:

嵌入式芯片的发展将使机器人在微型化,高智能方面优势更加明显,同时会大幅度降低机器人的价格,使其在工业领域和服务领域获得更广泛的应用。

在这些应用中,主要着重于在控制、信息交互方面的应用。同样的道理,所有基于网络的远程控制器件都需要与嵌入式系统之间实现接口,然后再由嵌入式系统来控制并通过网络实现控制,本文所探讨的嵌入式视频生产示教系统也是基于这种机制所研发的系统。

1.2.2 多媒体信息技术的研究现状

多媒体技术是当今信息技术领域中发展最快,最活跃的技术之一,是新一代电子技术发展和竞争的焦点。多媒体技术是融计算机、声音、文本、图像、动画、视频和通信等多种功能于一体,借助日益普及的高速信息网,可实现计算机的全球联网和信息资源共享,因此被广泛应用在咨询服务、图书、教育、通信、军事,金融、医疗等诸多行业,并正在潜移默化地改变着我们的生活面貌^[15-25]。

结合本文所要解决的主要问题,本节主要阐述多媒体信息技术中音视频处理的研究现状:

① 音频处理标准^[26-33]

在语音编码技术领域,各个厂家都在大力开发与推广自己的编码技术,使得在语音编码领域编码技术产品种类繁多,兼容性差,各厂家的技术也难于尽快得到推广。所以,需要综合现有的编码技术,制定出全球统一的语言编码标准。自 20 世纪 70 年代起,CCITT 下第十五研究组和国际标准化组织(ISO)已先后推出

了一系列的语音编码技术标准。其中, CCITT 推出了 G 系列标准, 而 ISO 则推出了 H 系列标准。

1) 电话质量的语音压缩标准

主要有 ITU 的 G.711(64kb/s)、G.721(32kb/s)、G.728(16kb/s)和 G.729(8kb/s)等建议, 用于数字电话通信。

2) 调幅广播质量的音频压缩标准

调幅广播质量音频信号的频率范围是 50Hz~7kHz, 又称“7kHz 音频信号”, 主要采用 ITU 的 G.722(64kb/s)建议。这种压缩方法能够在每秒 8kB 的存储量下给出相当好的音乐信号, 也很适合于需要存储大量高质量音频信号的多媒体系统使用。用于优质语音、音乐、音频会议和视频会议等。

3) 高保真立体声音频压缩标准

高保真立体声音频信号的频率范围 50Hz~20kHz, 在 44.1 kHz 采样频率下用 16bit 量化, 信号速率为每声道 1410kb/s。主要采用 MPEG-1 或 MPEG-2 双杜比 AC3 等建议, 目前国际上比较成熟的高保真立体声音频压缩标准为 MP3 音频。

常见的音频文件格式包括: WAVE 格式、MP3 格式、Real Audio、WMA 格式、RM 格式、AC3 格式等。

② 视频处理标准^[26-33]

视频编解码方面, 由 ISO 和 ITU-T 正式公布的视频压缩标准中, 有 MPEG-X 系列和 H.26X 系列, 我国也在加紧制定 AVS 标准。

1) 视频图像数据压缩标准 MPEG

MPEG (Moving Picture Experts Group) 是运动图像专家组的简称, 可以简单的分为以下几种格式: MPEG-1, MPEG-2, MPEG-4, MPEG-7 和 MPEG-21。

MPEG-1: 制定于 1992 年, 它是针对 1.5Mbps 以下数据传输率的数字存储媒体运动图像及其伴音编码而设计的国际标准。

MPEG-2: 制定于 1994 年, 设计目标为高级工业标准的图像质量以及更高的传输率。它主要应用在 DVD/SVCD 的制作(压缩)方面, 同时在一些 HDTV (高清晰电视广播)和一些高要求视频编辑、处理上面也有相当的应用。

MPEG-4: 制定于 1998 年, MPEG-4 是为了播放流式媒体的高质量视频而专门设计的, 它可利用很窄的带度, 通过帧重建技术, 压缩和传输数据, 以求使用最少的数据获得最佳的图像质量。

MPEG-7 标准: 于 1998 年 10 月提出, 于 2001 年最终完成并公布。它被称为“多媒体内容描述接口”, 它为各类多媒体信息提供一种标准化的描述。

MPEG-21 编码标准: 2000 年开始制订, 标准号为 ISO/IEC 21000, 正式名称是“多媒体框架”。

2) H.26X 视频编解码标准^[25-33]

H.261: 它是 CCITT 于 1990 年 12 月制定的国际上第一个视频压缩标准, 主要用于电视电话和视频会议, 以满足 ISDN 日益发展的需要。

H.263: 是 ITU-T 关于低于 64kbps 比特率的窄带通道视频编码建议, 其目的是能在现有的电话网上传输活动图像。

H.263+是 IUT-T1998 年推出的 H.263 建议的第 2 版, 它在保证 H.263 标准核心句法和语义不变的基础上, 提供了 12 个新的可协商模式和其他特征, 以提高压缩效率或改善某方面的功能, 增强应用的灵活性。

H.264/MPEG-4 part 10 AVC: 是 ITU-T 和 ISO/IEC 联合制定的最新编码标准。在 ISO/IEC 中该标准命名为 AVC (Advanced Video Coding), 作为 MPEG-4 标准的第 10 个选项, 在 ITU-T 中正式命名为 H.264 标准。

3) AVS 标准

AVS 是指由国家信息产业部科学技术司于 2002 年 6 月批准成立的数字音视频编解码技术标准工作组所制定的标准。目前 AVS 标准中涉及视频压缩编码的有两个独立部分: AVS 第二部分 (AVS1-P2) 主要针对高清晰度数字电视广播和高密度存储媒体应用; AVS 第七部分 (AVS1-P7) 主要针对低码率、低复杂度较低图像分辨率的移动媒体应用。AVS 视频标准的特色是在同一编码框架下, 针对有明显不同的应用制定不同的信源压缩标准, 尽可能减少技术的冗余, 从而降低 AVS 视频产品的设计成本、实现成本和使用成本。

另外, 随着互联网的普及与发展, 利用网络传输声音与视频信号的需求也越来越大。广播电视等媒体上网后, 也都希望通过互联网来发布自己的音视频节目。但是, 音视频在存贮时文件的体积一般都十分庞大。在网络带宽有限的情况下, 花几十分钟甚至更长的时间等待一个音视频文件的传输, 不能不说是一件让人头疼的事。流媒体技术的出现, 在一定程度上使互联网传输音视频难的局面得到改善^[27-34]。

传统的网络传输音视频等多媒体信息的方式是完全下载后再播放, 下载常常要花数分钟甚至数小时。而采用流媒体技术, 就可实现流式传输, 将声音、影像或动画由服务器向用户计算机进行连续、不间断传送, 用户不必等到整个文件全部下载完毕, 而只需经过几秒或十几秒的启动延时即可进行观看。当声音视频等在用户的机器上播放时, 文件的剩余部分还会从服务器上继续下载。

由于流媒体技术在一定程度上突破了网络带宽对多媒体信息传输的限制, 因此被广泛运用于网上直播、网络广告、视频点播、远程教育、远程医疗、视频会议、企业培训、电子商务等多种领域。

1.2.3 GUI 技术的研究现状

随着计算机硬件设计水平的提高,越来越多的软件开发工作集中在图形用户接口 GUI 上。图形用户接口是指计算机与其使用者之间的对话接口,是计算机系统的重要组成部分,GUI 的广泛流行极大地方便了非专业用户的使用。人们不再需要死记硬背大量的命令,而可以通过窗口、菜单方便地进行操作。当前,GUI 在嵌入式系统中地位越来越重要。这些系统对 GUI 的基本要求包括轻型、占用资源少、高性能、高可靠性、可配置等特点。针对嵌入式应用的 GUI,主要有以下几种^[35-38]:

① OpenGUI

OpenGUI 基于一个用汇编实现的 x86 图形内核,提供了一个高层的 c/c++ 图形窗口接口,还提供了 2 维绘图原语,消息驱动的 API, BMP 文件格式支持。OpenGUI 支持鼠标和键盘的事件,并支持 8, 15, 16 和 32 位颜色模型。然而,由于其内核用汇编实现,可移植性受到影响。

② 紧缩的 X Window 系统

X Window 是 Linux 以及其它类 UNIX 系统的标准 GUI。X Windows 系统采用标准的客户/服务器(Client/server)体系结构,具有可扩展性好、可移植性好等优点,但该系统庞大、累赘和低效率。另外,该系统的源代码尚不开放,很难进行本地化的开发。

③ MicroWindows

MicroWindows 是 Century Software 主持的开放源码的嵌入式 GUI 项目,提供类似于 X 的客户/服务器(client/server)体系结构。MicroWindows 的移植性很强,它支持多种软硬件。MicroWindows 采用分层设计方法,以便不同的层面能够在需要的时候改写,充分体现了嵌入式系统定制之特色。另外, MicroWindows 提供了比较完善的图形功能,包括一些高级功能。但是作为一个窗口系统,该项目提供的窗口处理功能还需要进一步完善,还有汉化处理方面不好,键盘和鼠标等驱动还很不完善。

④ MiniGUI

MiniGUI 是由北京飞漫软件技术有限公司开发的面向实时嵌入式系统的轻量级图形用户界面支持系统。它主要运行于 Linux 控制台,遵循 LGPL 条款的纯自由软件。MiniGUI 灵活性好,可定制能力强,提供了完备的多窗口机制,消息传递机制,并支持多字符集和多字体等。然而,它的图形功能尚不完善,对图形设备的抽象层次太高。

⑤ Qt/Embedded

Qt/Embedded 是 TrollTech 公司以原始 Qt 为基础并做了许多调整以适用于嵌入

式环境。Qt/Embedded 通过 Qt API 与 Linux I/O 设施直接交互,成为嵌入式 Linux 端口。Qt/Embedded 的主要特点面向对象编程、丰富的控件资源、可移植性较好、有优越的绘图功能等。Qt/Embedded 采用 Server/Client 结构,类库完全采用 C++封装。Qt 以工具开发包的形式提供给开发者,这些开发包包括图形设计器、Makefile 制作工具、字体国际化工具和 Qt 的完备 C++类库等。它给开发者提供了丰富的窗口构件(Widgets),并且还支持窗口部件的定制,因此可以为用户提供丰富的图形界面。

基于 QT/E 的开发相当快捷和方便,可以迅速地在嵌入式设备上开发图形界面。使用 QT/E,开发者可以得到这些好处:当移植 QT/E 程序到不同平台时,只需要重新编译代码,而不需要对代码进行修改随意设置程序界面的外观,方便地为程序连接数据库,使程序本地化,将程序和 Java 集成等等。

Qt/Embedded 对嵌入式系统和设备提供以下优势^[39-40]:

- 1) 同 Qt Desktop 相同的 API: 开发者只需学习一种 API,源代码可以应用于多个目标桌面环境(Windows, X11, Mac OS X)和嵌入式 Linux 环境。
- 2) 对嵌入式系统扩展了 Qt 的功能: 开发者可以创建灵活的嵌入式程序,需要很低的内存和 flash。
- 3) 包含它自己的窗口系统: Qt/Embedded 不需要其他的图形子系统。它本身含有递交图形到 Linux 所需的一切。
- 4) 可定制应用: Qt/Embedded 是超级可定制的。
- 5) 完全组件化: 为了节省内存, Qt/Embedded 允许多余模块和系统的移除。
- 6) 完全源代码: Trolltech 提供完全的源代码给客户,使他们可以更大范围的调整性能规格和调试代码。
- 7) 同 Qtopia 的集成: Qtopia-Trolltech 的 PDA 和移动电话的软件平台和用户界面,利用 Qt/Embedded 被创建。Qt/Embedded 可以方便植入基于 Qtopia 的设备中。
- 8) 同 Java 的集成: Qt/Embedded 可以同几种 Java 虚拟机集成。Java 程序可以在基于 Qt/Embedded 的程序工作台上运行,提供同原程序相同的效果。

嵌入式系统的要求是小而快速,而 QT/E 就能帮助开发者为满足这些要求开发强壮的应用程序。QT/E 优点是模块化和可裁剪的,而且开发者可以只选取需要的一些特性,裁剪掉那些不需要的特性。这样,通过选择所需的特性,QT/E 的映像可以变的很小,最小只有 600KB 左右。

1.2.4 生产示教系统的技术应用现状分析

生产示教系统在国民经济各个领域中的应用还不广泛,其中车间层的生产示教活动,例如,车间设备的使用方法、使用规范、产品的加工过程、车间的生产

安全规范、卫生规范等，一般是采用集体教学的方式进行信息传递，而加工工艺流程、加工工艺参数等，一般是采用电子看板、图纸、图片、文字等形式对员工提供指导。然而随着时代的发展，企业的管理层和车间层之间、车间层内部之间以及员工和员工之间的信息传递、示教和沟通，往往更加需要的是音视频、图像、文字等相结合的多媒体信息。

目前生产车间的示教活动暴露出了以下缺点：

① 支持的多媒体信息相对较少：多媒体技术被广泛的应用于各个领域，但是车间层的各种生产示教活动，对于多媒体信息的支持力度却不足。

② 图纸、工艺卡片、工艺路线单等纸质文档需要经常性的更新，纸质文档和图纸具有不便于查阅，容易磨损等缺点，这给车间层的生产和管理均带来了不便之处。

③ 实时示教的能力不足：伴随着制造业的大发展，各种新技术，新工艺被不断的被应用与车间中，而目前车间层的生产示教活动，实时示教能力表现出许多不足。

基于以上嵌入式系统的发展和应用状况、多媒体信息技术的发展现状、GUI 技术的研究现状、以及目前车间层生产示教活动所表现出来的不足之处，对生产示教系统的综合分析如下：

① 制造业信息化、网络信息化、已经在全球兴起，伴随着嵌入式系统、多媒体技术、网络技术的发展，如果能够开发出一种融合几种多媒体技术于一体的嵌入式视频生产示教系统，对于提高整个车间以及车间之间的产品工艺性能、员工的操作技巧、以及提高车间所生产产品的质量都有着重要的意义。

② 多媒体技术信息在不断的发展，开发出的嵌入式视频生产示教终端设备，应该能够利用各种技术的新发展和新技术，以便使系统在车间层的应用中发挥更好的效果。

③ 嵌入式系统的资源是十分有限的，为了使研发出来的嵌入式视频生产示教系统，更加有效的实现各种多媒体信息的传输、示教与交互，需要对系统的视频处理能力进行有针对性的优化处理，只有这样，才能让系统更好的支持各种多媒体信息，从而才能使开发出的视频生产示教系统所支持的各种功能达到更好的应用效果。

④ 所研发出的嵌入式视频生产示教系统的整体界面，应当尽量友好，并且要方便车间层员工进行操作，增加界面的人性化按钮，提高界面的整体显示效果，所以在系统开发之前，就应当选用一种方便开发人员进行裁剪，方便车间层用户进行定制的 GUI 技术进行系统应用界面的开发，一个友好的车间层应用界面，能够令系统的多媒体信息支持模块更加醒目，能够让系统支持的多媒体信息格式产

生更好的示教效果。

1.3 课题来源

本课题得到以下科研项目的资助：

- ① 国家 863 重点项目“面向离散制造的可配置 MES 产品及行业解决方案（重点）”（项目编号：2007AA040701）；
- ② 重庆市科技攻关项目“车间制造过程信息化支持平台开发与应用示范”（项目编号：CSTC2007AA2013）。

1.4 论文研究目的和主要研究内容

本文的研究目的是要在制造业信息化突飞猛进的时代背景下，结合嵌入式系统的诸多优点，开发出一种嵌入式视频生产示教系统，该系统能够对车间层员工的生产加工过程、产品的生产工艺流程、提供更为有效的指导，目的是让车间员工更快更好的掌握各种生产和工艺技巧，提高加工过程的可靠性和产品的整体工艺性能，从而提高整个工厂的生产效率。论文主要的研究内容如下：

① 第 1 章，在阐述了嵌入式系统的研究现状和应用领域的基础之上，对多媒体技术的研究现状，GUI 技术的研究现状以及生产示教系统的技术应用现状进行了分析，接着阐述了本课题的项目来源，最后对论文的研究目的和主要的研究内容进行了汇总性说明。

② 第 2 章，首先从多媒体技术发展带来生产示教系统新需求的角度，以及车间层应用需求的角度，对开发的嵌入式视频生产示教系统进行了需求分析，提出了系统主要处理模块功能结构图，最后，对整个系统的软硬件架构进行了设计和说明。

③ 第 3 章，主要阐述了在系统开发的过程中所遇到的一系列关键技术问题，这些关键技术主要包括基于 Au1200 的音视频编解码技术的研究、QT 技术在系统界面以及功能丰富开发过程中的应用研究、系统实时示教功能的编解码器的原理、基于 Au1200 的视频实时示教功能的优化方法研究。

④ 第 4 章，首先阐述了系统的开发工具和开发流程，在前一章所进行的关键技术研究基础之上，对系统的关键技术功能进行了设计和实现，接着重点针对系统进行实时示教过程中，系统资源占用率过高，视频长时间播放容易发生死机的问题，进行了有针对性的优化处理，通过这些优化处理，有效的减少的系统实时示教过程中的资源占用率，使得系统总体运行情况得到了有效的提高，最后，演示系统的实现效果。

1.5 本章小结

本章首先介绍了嵌入式系统的应用现状，然后，在此基础上，对多媒体技术的研究现状、GUI 技术的研究现状，生产示教系统的技术应用现状进行了综述，结合各种技术的发展现状和技术应用现状，进行了国内外研究现状的综合分析，说明了论文研究的目的，意义以及课题来源，最后，阐述了论文的主要研究内容。

2 嵌入式视频生产示教系统的整体架构

2.1 嵌入式视频生产示教系统的总体需求分析

生产示教系统对于提高工厂车间的信息化程度具有很好的辅助效果，它能够让管理层和车间层之间、车间层之间、车间员工之间的交流更加的方便、快捷。在多媒体技术快速发展的今天，音视频、图片、文字等信息被广泛的应用于工业生产、社会生活、航天科技、医疗卫生等领域。做为生产车间信息化重要一环的生产示教系统，要尽量满足工厂车间各种不同类型的需要，做到支持音视频、图片、文字等信息。具体的需求分析如下：

① 功能需求分析：

针对现在车间层示教活动对于多媒体技术的支持能力不足的状况，本文所研发的嵌入式视频生产示教系统应当尽可能多的支持各种多媒体信息格式，对于多媒体信息的功能需求大致分为以下四个部分：

1) 音视频播放功能：音视频信息主要是加工示教过程中，嵌入式视频示教终端所需要播放的各类音视频信息，主要包括一些零件加工过程的视频、由零件到整机的装配流程视频、以及企业培训录像的循环利用视频等等，这些存储在生产示教终端中的相关视频对于提高车间员工的加工能力，车间员工所加工零件的可靠性以及工艺性都是十分有帮助的。

2) 图片播放功能：将本系统利用于车间层当中，并不是要完全的抛弃被车间员工和工程师所熟悉的各类图纸，而是要让图纸的利用和应用效果，适应时代的不断变迁和发展。在研发的视频生产示教系统中，应能够对传统零件二维、三维图片的支持。

3) 文字播放功能：在生产车间当中，尤其是对于一些大的企业集团而言，车间层之间的信息传递，以及车间层和管理层的信息传递，是比较浪费时间的，做为车间层中的一种示教信息终端，如果能够兼顾字幕信息的功能，对于车间层面，以及车间层和管理层之间的信息传递是十分有好处的，不但节约了企业的成本，更重要的是能够提高整个企业的交流效率。

4) 实时示教功能：根据车间层的具体需求，管理层能够通过嵌入式视频生产示教终端对员工的生产加工过程、产品的加工工艺流程等进行实时的示教。

② 性能需求分析：

嵌入式视频生产示教系统除了要满足支持多媒体类型多样化的要求之外，性能要比较稳定，在播放各种多媒体文件，尤其是视频文件的时候，系统的 CPU 占用率应当被控制在一个可以接收的范围之内，否则，过高的资源占用率，容易导

致系统资源占用率太高，进而可能造成系统崩溃，影响到车间层的正常使用。

③ 成本消耗需求分析：

考虑到企业的经济承受能力，嵌入式视频生产示教终端，应当在满足企业需求的前提之下，尽量采用比较经济合理的技术方案，这样才能满足企业大范围内的应用需求。

④ 操控界面需求分析：

嵌入式视频生产示教终端的操作者多为一线工人，这就要求终端的操控界面要比较人性化，只有这样，才能使员工在使用系统终端的过程中更加方便和快捷。

2.2 嵌入式视频生产示教系统的总体运行模式

嵌入式视频生产示教终端的运行模式如图 2.1 所示。此信息终端既可以通过 PSTN、ISDN、xDSL 等方式接入 Internet，也可以通过企业局域网联入 Internet。示教终端分布于车间层的各个信息节点，在各个工作车间可以设立二级服务器，以方便对本工作车间终端的管理。各管理部门均设置了工作站，工作站一般采用 PC 机，操作简单，性能较高。示教终端与服务器采用 C/S 结构，示教终端之间、示教终端与管理部门工作站之间的信息同步都必须通过连接信息服务器来进行。

在该运行模式中存在三种类型的交互：

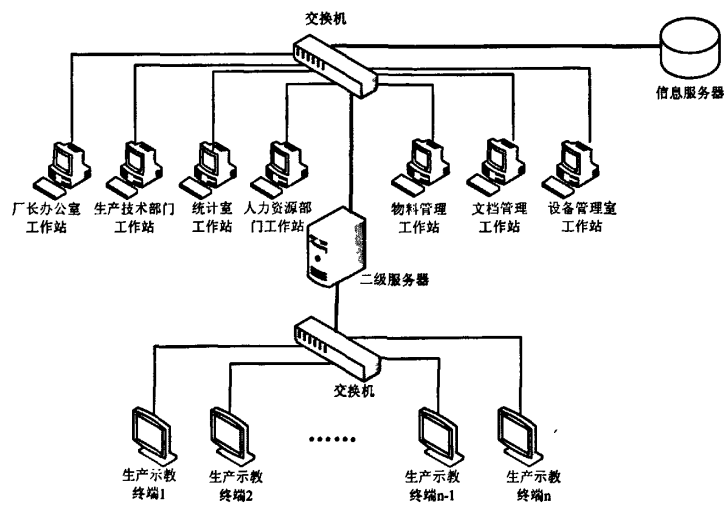


图 2.1 嵌入式视频生产示教系统的运行模式

Fig 2.1 The networking mode of the system of teaching in video manufacturing based on embedded system

① 示教终端之间的交互：任意一台联网的示教终端 A 都可以通过连接服务器，以对等的方式与另一台联网的示教终端 B 进行实时通讯。

- ② 示教终端与工作站 PC 机之间的交互：任意一台联网的示教终端 A 都可以通过连接服务器与安装了自主开发的 PC 机端软件的 PC 机 B 进行交互，其交互过程及效果和示教终端之间的交互相同。
- ③ 示教终端和服务端之间的交互：任意一台联网的示教终端 A 都可以直接与服务器进行交互。

2.3 嵌入式视频生产示教系统的功能模块划分

通过前面对系统整体需求的分析，就可以对系统的整体处理模块进行规划，具体的模块划分结构如图 2.2 所示：

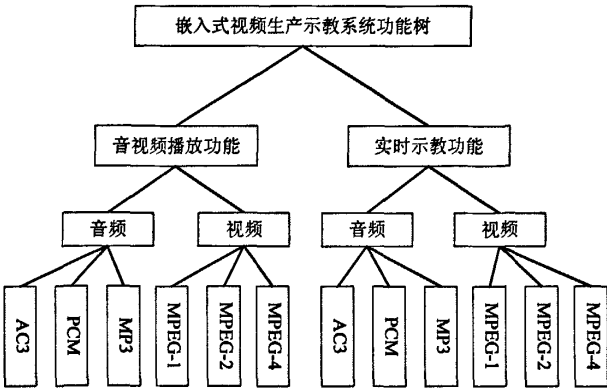


图 2.2 嵌入式视频生产示教系统的功能树

Fig 2.2 Function tree of the system of teaching in video manufacturing based on embedded system

- ① 音视频播放功能：该模块是系统的核心模块，也是系统运行时，占用系统内存最大的部分，根据处理器核心 Au1200 所支持的视频格式，主要支持的音频格式包括 AC3、PCM、MP3 格式等，视频方面支持 MPEG-1、MPEG-2、以及 MPEG-4 等。
- ② 实时示教功能：实时沟通是示教过程中一种更加有效的沟通方式，对于快速的解决技术问题，具有很好的辅助作用，实时示教功能对于系统的音视频信息实时处理能力提出了更高的要求，需要对系统进行优化处理，才能确保实时性。

2.4 嵌入式视频生产示教系统的软硬件架构

2.4.1 嵌入式视频生产示教系统的硬件架构

针对目前生产示教系统在工厂车间的应用所表现出来的不足，并根据嵌入式系统的设计理念，在系统总体模式和思想的指导下，本文提出了系统的硬件架构，如图 2.3 所示^[63]：

本系统采用嵌入式Linux进行软件的研发，GUI设计采用Qt/Embedded，该装置以AMD公司推出的高性能、低功耗、高集成度的片上系统MIPS32 Au1200 SOC为核心进行设计。外围扩展存储器模块、触摸屏接口、LCD显示模块接口、音频输入输出接口、双以太网接口、用于连接键盘、鼠标以及摄像头的USB接口、通用I/O（输入/输出）接口、GPIO口以及电源模块等，并且，系统支持NORFLASH和NANDFLAH两种启动模式。

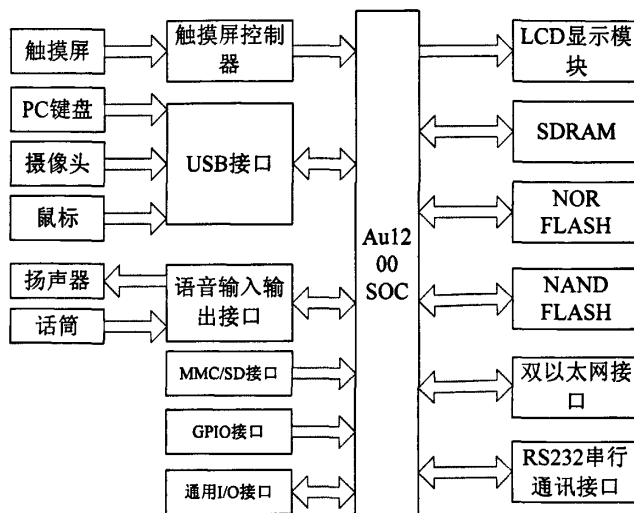


图 2.3 嵌入式视频生产示教系统的硬件结构框图

Fig 2.3 Hardware architecture of the system of teaching in video manufacturing based on embedded system

2.4.2 嵌入式视频生产示教系统的软件架构

在上面硬件模型的设计完成之后，就可以根据具体的硬件环境和应用功能的需求，进行软件系统的功能和结构设计。

根据本论文前面提出的功能需求，结合整个系统的软件复杂性，装置采用了层次式的软件体系结构，嵌入式系统应用的特制性，要求提供系统源码层次上的支持，而嵌入式Linux正适应了系统的这一需求，嵌入式Linux操作系统不仅具有支持多线程、多进程、支持多种平台等特点，而且相对于其他一些商用的实时操作系统，它具有源码开发、易移植、模块化、资源丰富等优势。本装置包含硬件架构的一体化层次框架如图 2.4 所示^[64]。

① 系统应用软件层

系统应用软件层是系统的上层应用程序层，它包含了系统各种功能实现的应用程序模块，主要包括了在系统需求分析中所提到的音视频播放的应用程序（例如视频播放的上层控制程序）、图片处理的应用程序（例如图片显示设置程序）、文字处理的应用程序（例如滚动字幕的间隔显示控制程序）、实时示教信息处理等。

② 系统资源层

这一层主要包含了基本函数库、各种通用协议栈、音视频的编解码库、系统 GUI（Qt/Embedded）库、嵌入式数据库等系统资源。这些库是系统运行过程中所必不可少的，它们为系统的应用提供了支持，一个完善的系统资源层对于提高系统的开发效率和系统运行的稳定性有着很好的辅助作用。

③ 嵌入式 Linux 操作系统层

这一层是整个系统的核心层，它所包括的内容也是最多的，它首先包含了各种接口，这些接口主要完成存储管理的调度、网络接口的实现、各种系统进程的调度、以及文件系统的管理等等，另外这一层中包含了在程序开发过程中，十分重要的驱动程序，通过驱动程序，就可以实现上层应用程序和底层硬件平台之间的交互，它是一个系统与底层平台的翻译层，通过它，就可以把上层程序中用高级语言所编写的应用程序，转化为能被硬件层所能理解的内容进行实现。对于 Au1200 而言，MAE 是一个十分具有特色的板载区间，系统专门为它提供了驱动程序。系统除了处理这些管理操作之外，还负责 I/O(输入/输出)的管理。

④ 引导程序装载层

在系统软件架构中，最接近硬件平台的是引导程序装载层，它是系统运行过程中，首先运行的一层，它主要是完成生产示教系统运行之前的各种硬件初始化

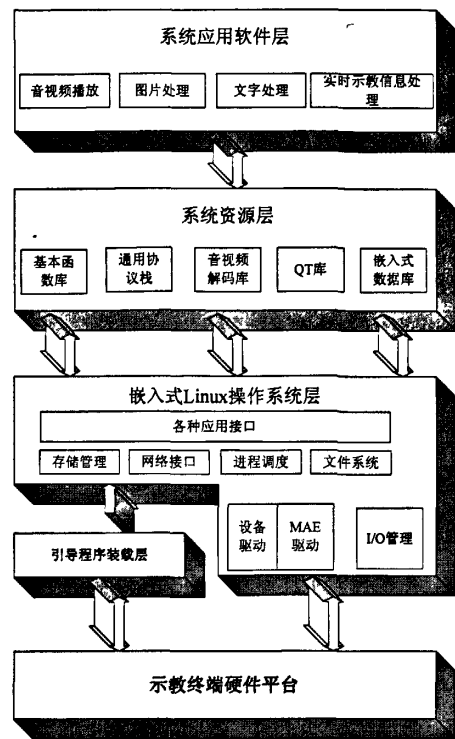


图 2.4 嵌入式视频生产示教系统的软件结构框图

Fig 2.4 Software architecture of the system of teaching in video manufacturing based on embedded system

操作、嵌入式 Linux 内核引导以及系统串口调试等过程。

2.5 本章小结

本章首先分析了车间层中嵌入式视频生产示教系统所需要处理的多媒体种类有哪些，阐明了车间层的总体需求和系统的总体设计原则，接着在此基础上，提出了嵌入式视频生产示教系统的硬件架构和软件架构。

3 嵌入式视频生产示教系统的关键技术研究

3.1 引言

通过对系统总体硬件架构和软件架构的设计和介绍,同时结合系统开发过程中遇到的实际困难,嵌入式视频生产示教系统的关键技术问题主要包括了以下几个方面:

① 基于 Au1200 的音视频编解码技术研究

Au1200 的音视频编解码处理能力相比同类型的其他芯片要强,这主要是 Au1200 具有比较特别的 MAE(媒体加速引擎)解码芯片,MAE 是 Au1200 上面的一个硬件设备,它肩负着视频解码,比例缩放,颜色空间转换以及数据过滤等任务,正是由于 MAE 肩负了这么多的视频编解码任务,所以才能使 MIPS32 核心的处理器专心的处理系统一体化界面的各种功能处理以及音频的处理等等。因为本系统是基于 Au1200 芯片进行研发的,所以就有必要对基于 Au1200 的音视频编解码技术进行研究。

② 实时示教功能的编解码器原理

系统的整体需求分析中,包含对视频实时示教功能的支持,这就牵扯到系统实时示教过程中所应用到的编解码器选取的问题,编解码器的选取要求尽量满足实时视频传输的要求,H.264 做为一种专门为网络活动图像传输制订的标准,适合做为实时示教功能视频编解码器所采用的标准,实现实时示教功能所需要编解码器的结构及原理将在本章进行研究。

③ 基于 Au1200 的视频实时示教功能的优化方法研究

在生产示教系统的实时示教功能的研发过程中,由于所选用的 H.264 视频编解码器的实时视频处理能力有限,所以造成系统的实时示教功能实现效果不理想,但是可以采用一些有针对性的优化措施进行弥补,所以有必要对视频实时示教功能的优化方法进行研究。

④ QT 技术在播放器界面开发中的应用研究

具有一个丰富的,可集成并且可以裁剪的系统应用界面,是嵌入式系统相对于其他低层次系统的优势之一,本系统在开发过程中,需要专门针对系统所支持的各种多媒体信息进行系统界面的开发,要求开发出的系统界面具有人性化的特点,而且要方便操作示教系统的员工进行相关的播放控制和其他一些控制操作,所以开发出的生产示教系统界面,就要求有丰富的控制功能按钮,做为 GUI 技术中,具有诸多优势的 QT 技术,能够满足系统界面开发过程中的这些需求,所以有必要对系统开发过程中所应用到的 QT 技术进行研究。

3.2 基于 Au1200 的音视频编解码技术研究

3.2.1 Au1200 的总体结构及特性

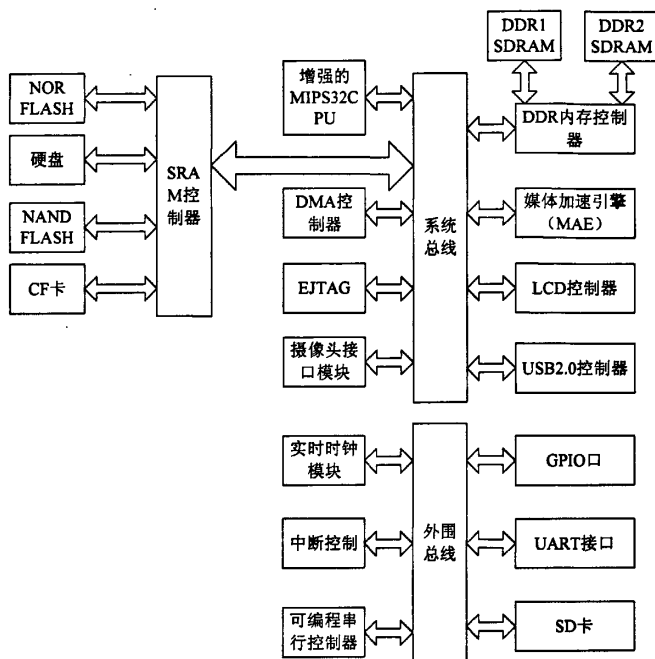


图 3.1 Au1200 功能结构框图

Fig 3.1 Functional Architecture of Au1200 Processor

AMD 公司（现为 RMI 公司）推出的 Alchemy 系列中的 Au1200 芯片，是一款低功耗、高性能的处理器，具有十分好的性能和特性，它的优点主要包括以下几个方面：功耗比较低，小于 400mW；支持 DDR1 和 DDR2 内存；支持 NAND 型 FLASH 启动；集成了媒体加速引擎(Media Acceleration Engine, 简称 MAE)；不需要外部 DSP 就可以实现视频文件的直接解码播放，从而可以简化编程环境并减少组件；USB2.0 控制器支持 HOST 模式和 DEVICE 模式，从而支持更高的下载速度和更加灵活的连接；具有叠加和混合功能的 LCD 控制器；操作系统支持 WindowsCE5.0 以及 Linux2.6^[46]。图 3.1 为 Au1200 内部功能框图^[43-44]。

3.2.2 MAE 的组件功能

MAE 可被划分为前端处理硬件和后端处理硬件，每一部分被设计用于完成特定的任务：MAE 前端能够自动完成复杂度和密集度最高的视频解码任务，包括反量化、反余弦变换、运动补偿等；MAE 后端完成执行图像横向和纵向比例放大/缩小，可编程的颜色空间处理和滤波功能。按照对图像数据的处理功能可将 MAE 分为以下几项功能^[43-45]：

① MAE 解码支持块图像数据

MAE 是为对基于块格式的视频数据进行解码而设计的, 通过软件进行变长度解码, 然后把宏块数据连同运动矢量等其他数据一同传给 MAE, 这样 Au1200 能够处理多种块格式图像了。由此, 宏块和 8×8 块中的图像数据是由 MAE 中专门设计的硬件模块自动完成的。

② MAE 解码支持颜色空间转换

由于 Au1200 处理器支持当前最流行的标准视频格式, 因此 MAE 能够处理输入的 YCbCr 4:2:2 和 YCbCr 4:2:0 格式, 并输出为 aRGB8888、RGB888、RGB565 或 RGB555 格式。

③ MAE 解码支持运动补偿

媒体加速引擎提供了一组多重可配置寄存器, 这使得它能够支持所有基于 MPEG 和 WMV9 格式的运动补偿处理, 支持到 1, 1/2 或 1/4 像素精度。当从 I 帧或 P 帧中取参考数据时, MAE 略过参考帧内不在所取数据块边界的数据部分, 系统构架这种策略使得所取数据总是处以可能的最高效的形式。

④ MAE 支持反量化和 IDCT 变换

反量化、IDCT 变换和运动补偿是通过 MAE 用硬件方式实现的, 从而减轻了处理器核心的负担, MAE 为这些任务进行了专门优化。由于反量化和 IDCT 变换相似且相互依赖, 所以为这两种处理专门设计了一条共享的 MAE 数据通道, 这不仅显著降低了片上的存储空间需求, 还附带降低了 BOM 成本。反量化和 IDCT 变换都是通过一系列可配置的寄存器和混合数学模块完成的, 在其中便能够完成 MPEG-1, MPEG-2, MPEG-4 等视频格式的 IDCT 变换和反量化, 并可完成 WMV9 的反量化 (8×8 , 8×4 , 4×8 和 4×4)。反量化和 IDCT 变换输出的是以列排序的 YCbCr 格式的 8×8 大小的平面块数据, 用于 MAE 的后端处理。

⑤ 缩放和滤波

同 MAE 的前端视频数据的处理方式完全不同, MAE 后端是一个独立的硬件组件。MAE 后端读入一帧图像数据并排成列, 一个宏块为一列。水平方向和垂直方向的缩放相互独立, 并在一次遍历内便可完成, 这种数据处理方式大大减小了处理器的尺寸和降低了成本。缩放功能本身是可配置的, 每一阶滤波都可由一个用户可编程的 32 位的系数参数进行配置, 这大大增强了对任何滤波过程进行处理的灵活性。

3.2.3 MAE 的编解码过程研究

随着视频编码标准和技术的全面提升, 高质量、高分辨率的视频解码, 如果单纯用软件, 甚至在一些强大的 CPU 上都难以实现。Au1200 的媒体加速引擎 (MAE) 是用硬件完成视频解码过程中复杂度最高的部分, 特别适用于 MPEG1、

MPEG2、MPEG4 和 WMV9 的视频解码。媒体加速引擎具有强大的解码功能，内核将上层软件解析出来视频宏块写回给 DDR 存储器，然后通过 DMA 控制 MAE 从 DDR 存储器取出视频宏块数据，将其解压缩成 RGB 数据格式，并将 RGB 显示数据再写回到 DDR 存储器，最后由 LCD 控制器从 DDR 存储器中取出数据输出到 FrameBuffer 上显示^[47]。

在 MAE 的视频解码过程中，非常依赖能使大量视频数据方便存储的压缩技术，它是一块板载硬件设备，一种能解码流行的基于块的视频格式的硬件加速单元，它主要负责对上层软件解码出来的可变长度编码进行解码，经过 MAE 处理之后，所播放的视频画面质量可达到 DVD 画质(720 ×480 的分辨率)，并可以在不破坏影像质量的情况下拉伸至 1024 ×768。

下面的图 3.2 和图 3.3 分别表明了视频的编码过程以及 MAE 的视频解码过程^[43-45]；



图3.2 视频编码过程

Fig 3.2 Video Encoding Process Overview

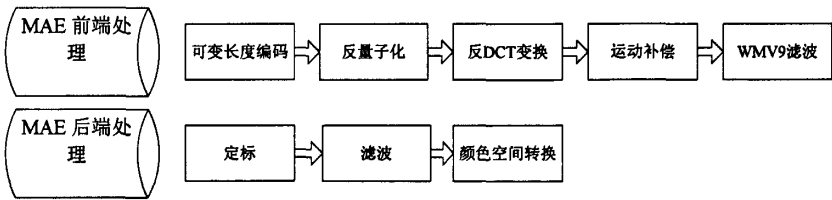


图 3.3 MAE 视频解码过程

Fig 3.3 MAE Video Decoding Overview

音视频播放模块整体架构如表 3.1 所示。其中 GUI 层提供和用户之间的交互并负责向下层程序发送用户命令，控制整个播放器的播放。而文件读取解析这一层主要负责读取源文件，接收上层 GUI 发过来的用户命令，控制音视频文件的播放。而再下一层的各类解码库提供了音视频的解码，其中音频解码库直接将音频数据解析出来，发送给硬件，视频解码库则只是将视频的宏块解析出来，而至于比较耗费时间和资源的反量化、IDCT、运动补偿等解码工作以及图像横向和纵向

比例放大/缩小, 可编程的颜色空间处理和滤波功能等显示任务则交给 MAE 处理。这样通过将视频解码的耗费时间和资源的工作交与硬件处理, 节省了系统资源, 解放了 CPU, 使得 CPU 可以被释放出来处理音频模块以及其它工作。音视频播放的整体架构和 workflows 如图 3.4 所示^[44-45]。

表 3.1 播放模块软件架构

Table 3.1 Software structure of player module

GUI		
文件读取解析		
GUI 库	音频解码库	视频解码库 (MAE 驱动)
内核		
硬件		

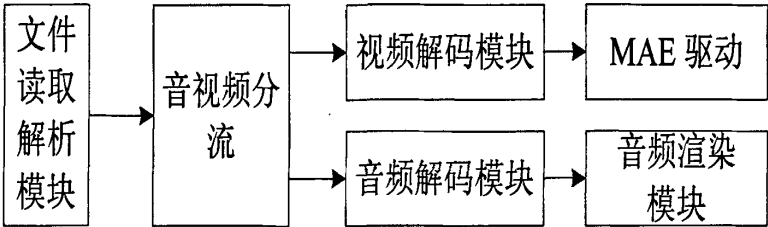


图 3.4 音视频播放模块整体工作流程图

Fig 3.4 Flow chart of the whole process of audio and video player module

3.2.4 I 帧、P 帧、B 帧对视频播放效果的影响

MPEG 压缩的基本思想是：帧内压缩和帧间压缩。采用分组的压缩方法, 把几帧图像分为一组(VOP), 为防止运动变化, 帧数不宜取多, 帧的分类方式为^[44-46]:

- ① 定义帧：将每组内各帧图像定义为三种类型, 即 I 帧、B 帧和 P 帧;
- ② 预测帧：以 I 帧做为基准帧, 以 I 帧预测 P 帧, 再由 I 帧和 P 帧预测 B 帧;
- ③ 数据传输：最后将 I 帧数据与预测的差值信息进行存储和传输。

以下是三种帧的特点介绍:

- ① 编码帧。I 帧具有如下特点:
 - 1) 它是一个全帧压缩编码帧。它将全帧图像信息进行 JPEG 压缩编码及传输;
 - 2) 解码时仅用 I 帧的数据就可重构完整图像;
 - 3) I 帧描述了图像背景和运动主体的详情;
 - 4) I 帧不需要参考其他画面而生成;
 - 5) I 帧是 P 帧和 B 帧的参考帧(其质量直接影响到同组中以后各帧的质量);
 - 6) I 帧是帧组 VOP 的基础帧(第一帧),在一组中只有一个 I 帧;
 - 7) I 帧不需要考虑运动矢量;

② P 帧是前向预测编码帧。

P 帧是以 I 帧为参考帧，在 I 帧中找出 P 帧“某点”的预测值和运动矢量，取预测差值和运动矢量一起传送。在接收端根据运动矢量从 I 帧中找出 P 帧“某点”的预测值并与差值相加以得到 P 帧“某点”样值，从而可得到完整的 P 帧。P 帧特点：

- 1) P 帧是 I 帧后面相隔 1-2 帧的编码帧；
- 2) P 帧采用运动补偿的方法传送它与前面的 I 或 P 帧的差值及运动矢量(预测误差)；
- 3) 解码时必须将 I 帧中的预测值与预测误差求和后才能重构完整的 P 帧图像；
- 4) P 帧属于前向预测的帧间编码。它只参考前面最靠近它的 I 帧或 P 帧；
- 5) P 帧可以是其后面 P 帧的参考帧，也可以是其前后的 B 帧的参考帧；
- 6) 由于 P 帧是参考帧，它可能造成解码错误的扩散；
- 7) 由于是差值传送，P 帧的压缩比较高。

③ B 帧：双向预测内插编码帧。

B 帧以前面的 I 或 P 帧和后面的 P 帧为参考帧，“找出”B 帧“某点”的预测值和两个运动矢量，并取预测差值和运动矢量传送。接收端根据运动矢量在两个参考帧中“找出(算出)”预测值并与差值求和，得到 B 帧“某点”样值，从而可得到完整的 B 帧。B 帧有如下特点：

- 1) B 帧是由前面的 I 或 P 帧和后面的 P 帧来进行预测的；
- 2) B 帧传送的是它与前面的 I 或 P 帧和后面的 P 帧之间的预测误差及运动矢量；

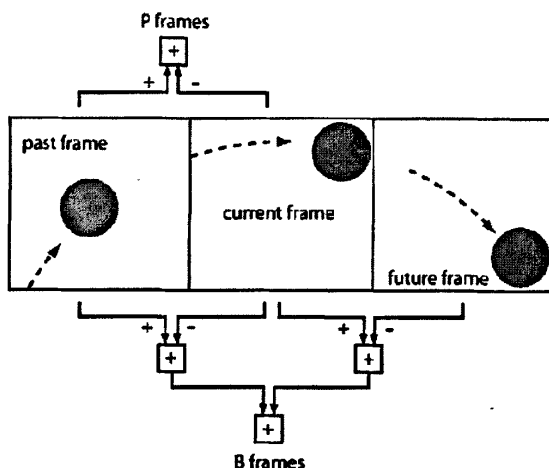


图 3.5 运动估计中 P 帧和 B 帧的比较

Fig 3.5 P and B Frame Comparisons for Motion Estimation

3) B 帧是双向预测编码帧;

4) B 帧压缩比最高,因为它只反映两参考帧间运动主体的变化情况,预测比较准确;

P 帧和 B 帧在运动估计中的对比,如图 3.5 所示^[44-46],一个球在静态的背景中运动,体现了一个帧到另一个帧间只有很小的变化。运动估计允许只描述球的位置变化(用两个 x/y 值这样的向量就可以了)而不是整个帧的数据。

在系统视频播放功能的研发过程中,如果正在播放的视频分辨率较高,系统的播放器会出现卡壳的情况,画面播放模糊不清,不流畅,甚至失真的情况,这主要是由于系统 CPU 核心的视频解码能力不足所造成的,究其根本原因,就是和前端处理过程运动补偿时 I 帧、P 帧和 B 帧在运动估计中的处理方式不同有关,如果 B 帧在视频文件中所占的比例较高,就有可能使资源有限的 Au1200 处理器难以满足视频解码的需求,进而影响到系统音视频的正常输出。

所开发的嵌入式视频生产示教系统进行音视频信息处理,主要是依靠 MAE 的解码能力去完成,所以在这种情况下不能像 DSP 技术那样进行相关的硬件优化,但是却可以对播放的视频文件进行一些软件处理,目的是要在保证视频播放清晰度的基础之上,减少损耗 CPU 比较厉害的 B 帧比例,适当增加 I 帧和 P 帧的比例,在系统的研发过程中,这种方法的应用,对于一些高分辨率的视频还是比较有效果的:不但使系统的整体播放更加稳定,而且还使系统的资源占用率更低,从而使系统更加的适合车间环境中的长时间示教播放。

3.3 基于 Au1200 的视频实时示教功能的优化方法研究

3.3.1 视频实时示教功能优化处理的必要性

在系统开发过程中,尽管嵌入式视频生产示教系统已经能和其他的终端设备连接,进行音视频信息的实时交互,但是视频显示效果却不理想,虽然解码可以达到 25fps 以上,但是图像的显示却只有 5-7fps,进而造成音视频不同步问题,同时终端的 CPU 负载也高达 90%以上,容易发生系统崩溃。

究其原因,是因为编解码器的运动补偿过程相当耗时,对编码的实时性影响很大,因此要对其进行优化处理,而且优化的时候要特别强调快速算法。

对编解码器的优化策略主要可分为三类:

① 算法方面的优化:对于比较费时的 DCT/IDCT 变换和运动估计,在图像质量得到保证的前提下,尽量采用快速算法;

② C/C++语言的通用优化,主要有:简化表达式、数据流优化、删除公共子表达式、展开循环、冗余分配和优化跳转等;

③ 硬件平台优化,包括:高效利用寄存器、自动增量寄存器寻址、使用块重

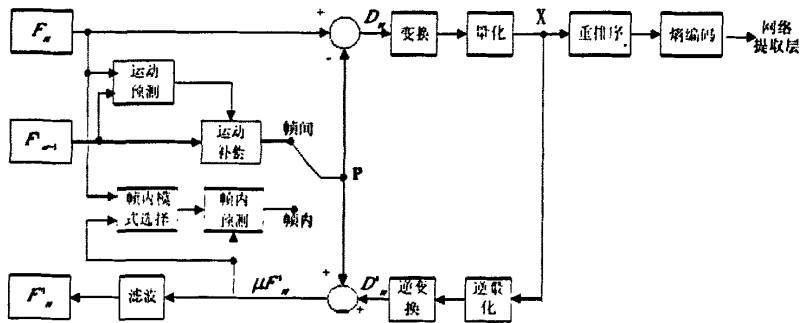
复、并行指令和乘积累加器,另外一些 CPU 还为复杂的 DSP 处理提供了专用的硬件 DSP 单元和相应的增强指令集。

本文采用前两种方法对视频播放器解码进行优化。

3.3.2 视频实时示教功能编解码器的原理

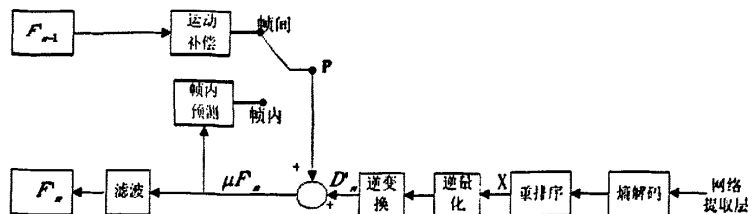
H.264视频编码标准在编码质量和压缩比上比原有的视频编码标准都有了明显的提高^[56]。在相同的视觉感知质量上,编码效率比H.263, MPEG-2和MPEG-4提高了50%左右^[57],并且有更好的网络友好性^[58-59]。H.264以其卓越的压缩性能在电视、高清晰度电视、卫星电视、存储媒体、无线多媒体应用等方面显示出了巨大的应用潜力。

与早期的视频压缩标准相同,H.264标准没有明确的定义编解码器,但着重定义了编码码流的语法以及对码流解码的方法^[58-59]。实际上,一个兼容的编码器和解码器大致包括的功能模块如图3.6和图3.7所示。除了去块滤波器以外,大部分基本的功能模块(预测、变换、量化、熵编码)在以前的标准中都采用了同样的原理。但H.264标准的重要变化体现在各个功能模块实现的细节上。



3.6 H.264编码器结构

Fig 3.6 The encoder structure of H.264



3.7 H.264解码器结构

Fig 3.7 The decoder structure of H.264

H.264编码器主要包含两个数据流路径:前向路径和重构路径。

在前向路径中,待编码的输入帧以宏块为单位进行帧内或者帧间模式编码。每个被编码的当前宏块与当前宏块的预测宏块P相减得到当前宏块的残差Dn^[59-62]。Dn再通过变换和量化形成量化变换系数X。X与解码该宏块所需要的其他信息(例如宏块的预测模式、量化步长、运动矢量信息等)一起形成压缩比特流进行存储或者通过网络抽象层(NAL)进行传送。而当前宏块的预测宏块P是基于重构帧得到的。在帧内模式时,P由当前帧中已经编码宏块N的重构宏块uF'n得到(此时由未经滤波的采样值形成P);在帧间模式时,P由一个或者多个参考图像通过运动补偿预测得到。而参考图像为已经编码重构的视频帧。

在重构路径中,量化变换系数X经过与前向路径的反过程重构成Dn的一个失真版本D'n^[59-62](这主要是因为量化过程引入了失真)。然后预测宏块P加上D'n产生当前编码宏块的重构宏块uF'n。最后一帧图像的所有重构宏块经过块效应滤波后组成当前帧的参考图像。而H.264解码器完全按照编码器重构路径对从NAL接收到的视频数据进行解码。这样就保证了编码端和解码端数据的一致性。

3.3.3 算法优化方法的研究

基于上文对于I帧、B帧、P帧的VOP思想阐述,本文采用的是梯度下降最优算法^[55-62],对视频文件编解码过程中的运动估计过程进行优化,具体的算法迭代优化过程将在下一章中进行介绍,以下是梯度下降算法的像素点处理过程研究:

设 $f(x, y)$ 表示输入的原始图像, $f(x', y')$ 表示输入的参考图像,为了一般化的目的,假定全局运动估计是以任意形状的VOP为输入图像的,同时参考图像也为任意形状,这样形状为矩形的VOP就是一种特殊。一个掩码 w_i 被用来指明像素点 (x, y) 和像素点 (x', y') 是否同时出现在输入的VOP和参考的VOP内,若同时出现则 $w_i = 1$,否则 $w_i = 0$ 。全局运动估计的目标是最小化同时出现在相邻两帧图像中所有像素的亮度信号的绝对残差和,即所有像素点 (x, y) 与它本身在预测图像中位于 (x', y') 误差之和。用 $I_1(x, y)$ 表示为某点 (x, y) 在当前输入图像的亮度, $I_0(x_i', y_i')$ 表示为点 (x, y) 在前一帧图像中的预测图像的亮度,用 $P = [a, b, c, d, e, f, g, h]$ 表示所求的全局运动参数矢量,定义目标函数为式:

$$E = f(P) = \sum_i w_i [I_0(x_i', y_i') - I_1(x_i, y_i)]^2 = \sum_i w_i e_i^2 \quad (3.1)$$

其中每个像素的误差:

$$e_i = I_0(x_i', y_i') - I_1(x_i, y_i) \quad (3.2)$$

一般而言,通过透射模型变换方程式得到的 (x', y') 不会落在整数位置,双线性插值被用来取得对应像素值。迭代的非线性Levenberg-Marquadet^[55]最优化方法用来最小化目标函数即可求得运动参数P, Levenberg-Marquadet最优化方法要求计算每个像素的误差 e_i 对各个运动参数的偏导数:

$$\frac{\partial e_i}{\partial a} = \frac{x_i}{d_i} \frac{\partial f'}{\partial x'} \quad (3.3)$$

$$\frac{\partial e_i}{\partial b} = \frac{y_i}{D_i} \frac{\partial f'}{\partial x'} \quad (3.4)$$

$$\frac{\partial e_i}{\partial c} = \frac{1}{D_i} \frac{\partial f'}{\partial x'} \quad (3.5)$$

$$\frac{\partial e_i}{\partial d} = \frac{x_i}{D_i} \frac{\partial f'}{\partial y'} \quad (3.6)$$

$$\frac{\partial e_i}{\partial d} = \frac{y_i}{D_i} \frac{\partial f'}{\partial y'} \quad (3.7)$$

$$\frac{\partial e_i}{\partial d} = \frac{1}{D_i} \frac{\partial f'}{\partial y'} \quad (3.8)$$

$$\frac{\partial e_i}{\partial h} = -\frac{x_i}{D_i} \left(x_i' \frac{\partial f'}{\partial x'} + y_i' \frac{\partial f'}{\partial y'} \right) \quad (3.9)$$

$$\frac{\partial e_i}{\partial g} = -\frac{x_i}{D_i} \left(x_i' \frac{\partial f'}{\partial x'} + y_i' \frac{\partial f'}{\partial y'} \right) \quad (3.10)$$

其中, $D_i = hx + gy + 1$ 。

利用上述偏导数, Levenberg-Marquadet 最优化方法^[55]计算近似的海深矩阵 H 和加权的梯度向量 b , 计算式子如下: 其中海深矩阵 H 为 8×8 矩阵, 梯度向量 B 为 8 维向量。

对于海深矩阵 H :

第一行元素:

$$a_{11} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial a}; a_{12} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial b}; a_{13} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial c}; a_{14} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial d};$$

$$a_{15} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial e}; a_{16} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial f}; a_{17} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial g}; a_{18} = \sum_i \frac{\partial e_i}{\partial a} \frac{\partial e_i}{\partial h};$$

第二行元素:

$$a_{21} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial a}; a_{22} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial b}; a_{23} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial c}; a_{24} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial d};$$

$$a_{25} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial e}; a_{26} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial f}; a_{27} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial g}; a_{28} = \sum_i \frac{\partial e_i}{\partial b} \frac{\partial e_i}{\partial h};$$

第三行元素:

$$a_{31} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial a}; a_{32} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial b}; a_{33} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial c}; a_{34} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial d};$$

$$a_{35} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial e}; a_{36} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial f}; a_{37} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial g}; a_{38} = \sum_i \frac{\partial e_i}{\partial c} \frac{\partial e_i}{\partial h};$$

第四行元素:

$$a_{41} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial a}; a_{42} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial b}; a_{43} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial c}; a_{44} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial d};$$

$$a_{45} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial e}; a_{46} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial f}; a_{47} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial g}; a_{48} = \sum_i \frac{\partial e_i}{\partial d} \frac{\partial e_i}{\partial h};$$

第五行元素:

$$a_{51} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial a}; a_{52} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial b}; a_{53} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial c}; a_{54} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial d};$$

$$a_{55} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial e}; a_{56} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial f}; a_{57} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial g}; a_{58} = \sum_i \frac{\partial e_i}{\partial e} \frac{\partial e_i}{\partial h};$$

第六行元素:

$$a_{61} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial a}; a_{62} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial b}; a_{63} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial c}; a_{64} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial d};$$

$$a_{65} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial e}; a_{66} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial f}; a_{67} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial g}; a_{68} = \sum_i \frac{\partial e_i}{\partial f} \frac{\partial e_i}{\partial h};$$

第七行元素:

$$a_{71} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial a}; a_{72} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial b}; a_{73} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial c}; a_{74} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial d};$$

$$a_{75} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial e}; a_{76} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial f}; a_{77} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial g}; a_{78} = \sum_i \frac{\partial e_i}{\partial g} \frac{\partial e_i}{\partial h};$$

第八行元素:

$$a_{81} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial a}; a_{82} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial b}; a_{83} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial c}; a_{84} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial d};$$

$$a_{85} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial e}; a_{86} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial f}; a_{87} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial g}; a_{88} = \sum_i \frac{\partial e_i}{\partial h} \frac{\partial e_i}{\partial h};$$

对于梯度向量 b 各个分量分别为:

$$b_1 = \sum_i e_i \frac{\partial e_i}{\partial a}; b_2 = \sum_i e_i \frac{\partial e_i}{\partial b}; b_3 = \sum_i e_i \frac{\partial e_i}{\partial c}; b_4 = \sum_i e_i \frac{\partial e_i}{\partial d};$$

$$b_5 = \sum_i e_i \frac{\partial e_i}{\partial e}; b_6 = \sum_i e_i \frac{\partial e_i}{\partial f}; b_7 = \sum_i e_i \frac{\partial e_i}{\partial g}; b_8 = \sum_i e_i \frac{\partial e_i}{\partial h};$$

基于鲁棒 M-估计器^[55]的全局运动估计的方法的目标函数必须是统计上鲁棒的, 本文改进的算法的目标函数式:

$$f(p) = \sum_i \rho(w_i(I_0(x'_i, y'_i) - I_1(x_i, y_i)), \sigma^2) \quad (3.11)$$

本文使用的 ρ 函数为 German-Mclurc 函数^[56-57], 其函数解析式如下:

$$\rho(e_i, \sigma^2) = \log(1 + e_i^2 / (2\sigma^2)) \quad (3.12)$$

其中, σ^2 为给定图像区域 R 的均方误差, 由下式给出:

$$\sigma^2 = \frac{1}{N} \sum_i [I_0(x'_i, y'_i) - I_1(x_i, y_i)]^2 \quad (3.13)$$

w_i 为图像像素残差的权重函数:

$$w_i = \begin{cases} 0 & e_i^2 > c\sigma^2 \\ 1 & e_i^2 \leq c\sigma^2 \end{cases} \quad (3.14)$$

权重函数被简化为二值函数, 上式中 $1 \leq c \leq 3$, 用于调整权重函数的判别条件。利用高斯-牛顿迭代最优化方法^[58], 设第 k 步得到的解 $p^{(k)}$ 。在 $p^{(k)}$ 处的梯度下降方向 $\sigma^{(k)}$ 由下式给出:

$$\sigma^{(k)} = -H^{-1}(p^{(k)})g(p^{(k)}) \quad (3.15)$$

其中 $H(p^{(k)})$ 是 ρ 函数在 $p^{(k)}$ 处的海深矩阵, $g(p^{(k)})$ 是 ρ 函数在 $p^{(k)}$ 处的梯度。 $H^{-1}(p^{(k)})$ 是 $H(p^{(k)})$ 的逆矩阵。

最优化迭代, 过程如下式:

$$p^{(k+1)} = p^{(k)} + a_k \delta^{(k)}, a_k > 0 \quad (3.16)$$

a_k 由 Armijo 型线性搜索^[58]确定。式 (3.18) 为 ρ 函数的海深矩阵, 式 (3.19) 为 ρ 函数的梯度。

$$H_{kl} = \sum_i \frac{\partial^2 \rho}{\partial e_i^2} \frac{\partial e_i}{\partial p_k} \frac{\partial e_i}{\partial p_l} \quad (3.18)$$

$$g_k = \sum_i \frac{\partial \rho}{\partial e_i} \frac{\partial e_i}{\partial p_k} \quad (3.19)$$

其中, p_k 为参数 p 的第 k 个分量, 即 $\frac{\partial e_i}{\partial p_k} = \frac{\partial e_i}{\partial a}$ 。对于八参数透射模型, $k=8$, $l=8$, 对于六参数模型, $k=6$, $l=6$, 由式 (3.16) 两边左乘 H 得:

$$\sum_i H_{kl} \delta_i = -g_k \quad (3.20)$$

该式为线性方程组, 由高斯消元法^[58]即可求得梯度下降方向 $\sigma^{(k)}$

梯度下降算法的迭代过程:

在改进目标函数下, 令 $H = H_{kl}$, $b = a_k \sigma^{(k)}$ 。

运动估计参数通过更新如下式子:

$$\Delta p = H^{-1}b \quad (3.21)$$

$$p(k+1) = p(k) + \Delta p \quad (3.22)$$

3.4 QT 技术在播放器界面开发中的应用研究

3.4.1 QT 相关技术简介

Qt/Embedded 应用在嵌入式系统开发领域有诸多优势, 本文选用它进行系统界面功能的研发。

Qt/Embedded 是一个完整的自包含 GUI 和基于 Linux 的嵌入式平台开发工具, 大范围的 Qt/Embedded API 可用于多种开发项目。Qt/Embedded 可以开发市场上多种类型的产品和设备, 从消费电器 (移动电话、联网板和机顶盒) 到工业控制设

备（如医学成像设备、移动信息系统等）[40-43]。

Qtopia 是基于 QT/E 的图形界面应用程序集,通过 Qtopia 真正地将图形界面展现在用户的眼前。如果说 QT/E 是灵魂,那么 Qtopia 就是肉体,所以 Qtopia 是一个面向嵌入式 Linux 的全方位应用程序开发平台,同时也是为基于 Linux 的 PDA,智能电话和其他移动设备设计的一个全面的用户界面[47-51]。

Qt设计器是一个用Qt编写的可视化界面设计工具和代码编写器,应用程序可以使用代码编写,也可以使用Qt设计器来加快系统窗口界面的开发。Qt图形设计器和tmake（或qmake）编译器是完全集成在一起的。

用Qt设计器设计一个窗体是个简单的过程,开发者点击一个工具箱按钮即可加入一个想要的窗口部件,然后在窗体上点击一下即可定位这个部件,部件的属性可以通过属性编辑器修改。部件的精确位置和大小并不重要,开发者可以选择部件并且上面应用布局。Qt设计器去掉了界面设计中费时的“编译、连接与运行”的循环,改正或改变设计变得容易起来。

Qt设计器支持一种基于工程的程序设计方法。一个工程表现为一个.pro文件,可以用tmake来根据这种文件生成Makefile。开发者可以先创建一个工程,然后加入所需的窗体和源代码。开发者可以通过子类化来实现用户界面与底层功能的完全分离,也可以通过在Qt设计器里面直接编写代码来将代码与窗体放在一起。程序中用到的图标和其他图象自动地被同一工程中的所有窗体共享,这样可以减小可执行文件的体积并且能加快载入的速度。窗体设计被保存成XML格式的.ui文件并且被uic(用户界面编译器)转换成为头文件和源文件。tmake构建工具在它生成的makefile中自动地包含了uic的规则,因此开发者不需要自己去加入uic。

3.4.2 信号与槽机制

信号和槽机制是用于对象之间的通讯。信号/槽机制是 Qt 的一个中心特征并且也是 Qt 技术与其它具有类似功能工具包相比,比较不相同的部分。

在图形用户界面编程中,我们经常希望一个窗口部件的一个变化被通知给另一个窗口部件。更一般地,我们希望任何一类的对象可以和其它对象进行通讯。

较老的工具包使用一种被称作回调的通讯方式来实现同一目的。回调是指一个函数的指针,所以如果你希望一个处理函数通知你一些事件,你可以把另一个函数(回调)的指针传递给处理函数。处理函数在适当的时候调用回调。回调有两个主要缺点。首先他们不是类型安全的。我们从来都不能确定处理函数使用了正确的参数来调用回调。其次回调和处理函数是非常强有力地联系在一起的,因为处理函数必须知道要调用哪个回调[47-52]。

在 QT 技术中,是应用信号和槽机制代替回调函数完成对象间的通讯。

从 QObject 类^[53]一个子类(比如 QWidget 类)继承的所有类可以包含信号和

槽。当对象改变它们的状态的时候，信号将会被发送，在这个时候信号不知道是否会被接收。

槽用来接收信号，它们是正常的成员函数。一个槽不知道它是否被任意信号连接。此外，对象不知道关于这种通讯机制和能够被用作一个真正的软件组件。

可以把许多信号和所希望的单一槽相连，并且一个信号也可以和所期望的许多槽相连。把一个信号和另一个信号直接相连也是可以的。

信号和槽的连接原理如图 3.8 所示^[52]：

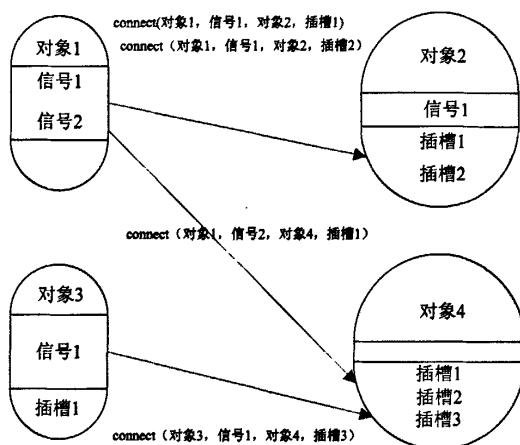


图 3.8 信号和槽的连接

Fig 3.8 The connection between signals and slots

信号与槽的具体应用方法如下：

① 首先建立一个类，并设置类的访问权限：

```
class Entity
{
public: //设置访问权限
    Entity();
    int value() const { return val; }
    void setValue( int );
private:
    int val;
};
```

② 定义信号和槽：

```
class Entity : public QObject
```

```

{
    Q_OBJECT
public:
    Entity();
    int value() const { return val; }
public slots:
    void setValue( int ); //定义槽
signals:
    void valueChanged( int ); //定义信号
private:
    int val;
};
void Entity::setValue( int v )
{
    if ( v != val ) {
        val = v;
        emit valueChanged(v); //通过它来发射信号
    }
}

```

③ 信号与槽的连接方法:

```

Entity a, b;
connect(&a, SIGNAL(valueChanged(int)), &b, SLOT(setValue(int)));
b.setValue( 4 );
a.setValue( 7 );
b.value();

```

当调用 `a.setValue( 7 )` 的时候, 会使 `a` 发射一个信号 `valuechanged()`, `b` 将会在它的 `setValue` 槽中接收这个信号, 接下来, `b` 会发射同样的 `valuechanged()` 信号, 但是因为没有槽去连接 `b` 的 `valuechanged()` 信号, 所以没有任何事情发生。

通过上面对信号/槽机制阐述可以看出, 信号与槽机制在系统的图形化界面开发过程中发挥着重要的作用。利用这项技术, 可以很方便的对系统视频播放器的控制功能进行增减。在系统多媒体播放界面开发的初始阶段, 视频播放不能通过触摸屏和鼠标进行直接的控制, 这给生产示教系统在车间层的实际应用过程中带来许多不便, 像播放器界面中的 Play (播放) 键、Pause (暂停) 键、stop (停止) 键、Next 键 (上一个)、Pre 键 (下一个)、音量调节键等, 如果能够通过信号和槽的

机制，把需要完成的播放器控制功能添加到 QT 所设计的播放器图形界面当中，将会给车间员工的实际应用带来许多方便，具体的实现过程将在下一张中进行阐述。

3.5 本章小结

本章重点研究了嵌入式视频生产示教系统开发过程中所涉及到的关键技术，这些关键技术包括基于 Au1200 的音视频编解码技术的研究，QT 技术在播放器界面功能开发中的应用技术研究，视频实时示教功能的编解码器的原理，视频实时示教功能优化方法的研究。关键技术的具体实现及优化过程将在下一章中进行详细阐述。

4 嵌入式视频生产示教系统的实现及优化

4.1 嵌入式视频生产示教系统软件开发环境的实现

4.1.1 系统的开发工具和开发流程

一个健壮、高效、兼容性好的开发环境是我们所需要的。同时考虑到开发的效率及通用性等因素，本系统主要采用了以下开发工具：

① GNU 编译器 GCC

编译器通常包括编译，汇编和链接工具。编译器的工作是通过编译和汇编工具将源代码(通常使用高级语言编写)翻译成目标文件(通常是低级的目标代码或者机器语言)，由于目标文件的执行还可能还需要其他系统库等目标文件的支持，所以需要链接工具将它们链接成可执行文件。在现代编译器的实现中，这个工作一般是分为两个阶段来实现的：第一阶段，编译器的前端接受输入的源代码，经过词法，语法和语义分析等等得到源程序的某种中间表示方式(中间代码)。第二阶段，编译器的后端将前端处理生成的中间代码进行一些优化，并最终生成在目标机器上可运行的代码(包括链接过程)。

GCC 是在 UNIX 以及类 UNIX 平台上广泛使用的编译器集合，它能够支持多种语言前端，包括 C、C++、Objective. C、Fortran 和 Java 等。GCC 具有一定的硬件无关性，可以生成高质量的可执行代码，利用“Makefile”文件可以方便的控制 GCC 进行编译。

② Makefile 文件

Makefile 文件在嵌入式开发中作用重大，关系到了整个工程的编译规则。一个工程中的源文件不计数，其按类型、功能、模块分别放在若干个目录中，Makefile 定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作，因为 Makefile 就像一个 Shell 脚本一样，其中也可以执行操作系统的命令。Makefile 文件包括目标，依赖，规则三要素。

③ 调试器 GDB

GNU 的调试器称为 gdb，该程序是一个交互式工具。gdb 功能强大，可完成包括设置断点、监视程序变量的值、程序的单步执行、修改变量的值等调试任务。

由于嵌入式系统资源有限，通常情况下只能运行嵌入式操作系统和功能比较简单的嵌入式软件。因此，嵌入式开发环境一般运行在 PC 机或者工作站上，这种用于开发嵌入式系统的计算机称为宿主机，嵌入式软件的运行环境则称为目标机。本课题宿主机采用的是 RedHat-Linux 9.0。开发人员可以通过 TELNET 协议远

程登录到服务器和目标板上进行远程开发。由于软件开发环境是在 X86 体系结构下，软件运行环境是在 MIPS 体系结构下，这里就涉及到如何在 X86 环境下生成能够在 MIPS 环境下运行的可执行代码的问题。交叉编译就是在一个平台上生成另一个平台上的可执行代码。

本课题中，交叉编译就是在主机平台（X86 PC 机）上生成目标机器（MIPS 终端）的可执行代码。在这里主机就是装有 Linux 操作系统的机器，上面安装有交叉编译环境，进行应用软件的开发和编译。目标机 MIPS32 平台，上面具有引导程序（bootloader）、嵌入式操作系统内核、根文件系统，它们都被烧写进 MIPS32 平台的 Flash 中的特定位置。由目标板上启动代码引导嵌入式系统上的操作系统内核，运行起嵌入式系统。目标板与主机通过网线或串口线相连，最终在主机上把应用程序编译成 MIPS32 格式的二进制文件通过网线或串口传输到目标板上运行。

嵌入式软件的编写、开发调试的部分工作在主机上完成，另一部分工作在目标板上完成，其主要流程为：编写代码→交叉编译→交叉链接→下载调试，如图 4.1 所示^[63]，具体过程描述如下：

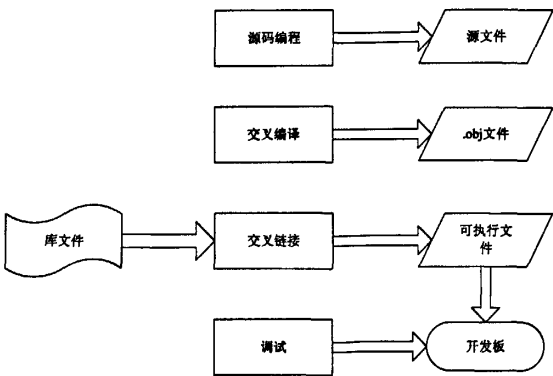


图 4.1 嵌入式视频生产示教系统的软件开发流程

Fig 4.1 Software development process of the system of teaching in video manufacturing based on embedded system

首先，在宿主机上完成编程工作。本课题中的嵌入式系统主要由 C 或 C++语言结合编写，编程工作完成了软件源代码文件。

其次，在主机上建立的交叉编译环境中，将每个源代码文件用交叉编译器 mipsel-linux-gcc 或 mipsel-linux-g++生成相应的.obj 文件，接着根据这些 obj 文件用途不同将其链接成相应的可执行文件或库文件，主要完成处于不同.obj 文件内的变量和函数的重定位（relocation），这个过程主要使用的工具是交叉链接器：mipsel-linux-ld，mipsel-linux-ar 等。

最后,通过远程下载方法,将生成的可执行文件和库文件下载到目标板的内存或烧写到其 Flash 上运行。软件的调试对软件的正确、可靠、稳定运行非常重要,简单程序的调试工作可直接在开发板上用 gdb 进行调试,底层软件和复杂的上层软件则需要使用 BDI2000 协助进行软硬件联合调试或远程调试,应用的主要工具是: mipsel-linux-gdb。

4.1.2 Qt/Embedded 的移植

QT 是 Trolltech 公司开发的一个多平台的 C++图形用户界面应用程序框架。它提供给应用程序开发者建立图形用户界面所需的功能。QT 是完全面向对象的很容易扩展,并且允许真正地组件编程。QT/E 是 QT 在 Embedded 平台,即带有帧缓冲(FrameBuffer)支持的 Linux 平台下的版本。通过交叉编译实现 QTE 在嵌入式信息交互终端上运行。图 4.2 为 Qt/Embedded 应用开发的一般流程。

交叉编译 Qt/Embedded 首先要编译 Qt-X11 版本,编程出 uic 工具和 qvfb,这是两个在 Qt/Embedded 上的开发工具。然后配置 Qt/Embedded 的编译环境,交叉编译 Qt/Embedded。

① qt-x11:(主要是编译获得 uic)

```
tar xzf qt-x11-2.3.2.tar.gz
```

```
mv qt-2.3.2 qt-x11
```

```
cd qt-x11
```

设置环境变量

```
export QTDIR=$PWD
```

```
export PATH=$QTDIR/bin:$PATH
```

```
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

```
./configure -no-xft
```

```
make
```

② 该终端采用了 Qt/Embedded 2.3.7 作为目标板 Linux 图形界面库。在交叉编译前需要先修改 Qt/Embedded 的配置文件。接着是设置环境变量,如下:

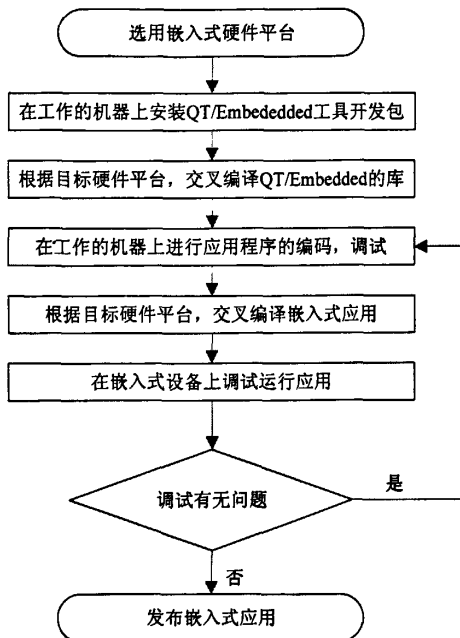


图 4.2 QT/Embedded 应用开发的一般流程

Fig 4.2 The ordinary flow of QT/Embedded application development

```
tar xfz qt-embedded-2.3.7.tar.gz
```

```
mv qt-2.3.7 qt
```

```
cd qt
```

```
export QTDIR=$PWD
```

```
export PATH=$QTDIR/bin:$PATH
```

```
export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
```

然后将 Qt/Embedded 配置为动态链接库，并支持 JPEG, PNG, GIF 图像格式，像素位数：

```
./configure -depths 16 -shared -xplatform linux-mips-g++ -gif -system-libpng  
-system-jpeg -qvf -thread -qconfig
```

交叉编译后，在/qt-2.3.7/lib 路径下的生成 Embedded Qt 的动态链接库，主要有 libqte.so.2, libqte.so.2.3, libqte.so.2.3.7, libjpeg.so.62, libpng12.so.0, libpthread.so.0 等，将这些库文件拷贝到目标板文件系统中。这样当开发板启动以后，就可以在嵌入式系统下运行基于 Qt/Embedded 的应用程序了。

4.2 多媒体播放界面开发的应用实现

播放器界面开发过程中，应用到的程序文档如表 4.1 所示：

表 4.1 AMD 媒体播放器软件应用程序

Table 4.1 The AMD Media Player software application

缩写	定义
MAI	Media Application Interface
DRM	Digital Rights Management
MPEG	Audio/Video Compress Standard
MP3	MPEG Layer 3 Audio Compression Standard

4.2.1 视频播放功能的丰富和实现

① 信号与槽的关联

通过调用 QObject 对象的 connect 函数来将某个对象的信号与另外一个对象的槽函数相关联，当发射者发射信号时，接收者的槽函数将被调用。该函数的定义如下：

```
bool QObject::connect ( const QObject * sender, const char * signal, const
QObject * receiver, const char * member )
```

这个函数的作用就是将发射者 sender 对象中的信号 signal 与接收者 receiver 中的 member 槽函数联系起来。当指定信号 signal 时必须使用 QT 的宏 SIGNAL(), 当指定槽函数时必须使用宏 SLOT()。如果发射者与接收者属于同一个对象的话，那么在 connect 调用中接收者参数可以省略。

② 视频播放控制按钮添加的过程

1) 视频播放控制按钮的创建：

视频播放控制包括播放、暂停、快进、快退等，本程序列出了这些控制功能中的几种：

```
pushButton_play = new QPushButton( this, "play" ); //播放器控制按钮的创建
pushButton_play->setGeometry( QRect( 100 , 650, 60, 30 )); //播放器按钮大小
的设置
```

```
pushButton_pause = new QPushButton( this, "pause" );
pushButton_pause->setGeometry( QRect( 200 , 650, 60, 30 ));
volume_slider = new QSlider(0,10,1,0,Qt::Horizontal,this);
volume_slider->setGeometry(560, 650, 100, 10);
```

2) 建立信号和槽的链接：

当视频播放控制按钮创建好之后，就可以通过所设计的 intchange 选项值和系统的硬件层衔接在一起。

connect(pushButton_play, SIGNAL(clicked()), this, SLOT(Btnplay_Click())); // 建立信号与槽之间的链接

connect(pushButton_pause, SIGNAL(clicked()), this, SLOT(Btnstop_Click()));

3) 设置控制按钮上的字符:

通过下面的代码可以对按钮上的字符进行设置。

pushButton_play->setText(tr("play")); //设置按钮上面的字体

pushButton_pause->setText(tr("pause"));

4) 播放器功能的硬件层实现:

上面的步骤只是完成了视频播放控制按钮的创建, 如果要使这些控制功能在硬件层得以实现, 就需要把触摸屏上的点击事件和硬件层衔接在一起, 通过下面的代码完成:

```
switch(int_change){
    case 1:{
        Player_ExecuteEngineCommand(pPlayer, M_TEXT("play"), (unsigned int
*)M_NULL); //执行播放命令
        iodata.subcmd = AU1200_LCD_SET_WINDOW;
        iodata.window.flags = WIN_POSITION;
        iodata.window.xpos = xpos_maiplayer;
        iodata.window.ypos = ypos_maiplayer;
        issue_ioctl();
        int_change = 0;
        break;
    }
    case 2:{
        Player_ExecuteEngineCommand(pPlayer, M_TEXT("quit"), (unsigned int
*)M_NULL); //执行播放退出命令
        int_change = 10;
        pPlayer->m_bExit = M_TRUE;
        break;
    }
    case 3:{
        Player_ExecuteEngineCommand(pPlayer, M_TEXT("pause"), (unsigned int
*)M_NULL); //执行播放器暂停命令
        int_change = 0;
```

```

        break;
    }
    case 7:{
        QString qstr_value=QString::number(slider_value,10);
        QString slider_pos="volume" + qstr_value;
        Player_ExecuteEngineCommand(pPlayer, M_TEXT((char
        *) (slider_pos.ascii())), (unsigned int *)M_NULL); //音量控制
        break;
    }
}

```

4.2.2 视频播放器界面大小的设定与实现

① 在 Au1200 的媒体应用程序接口（简称 MAI）文档下面的参数设置程序 mai_param.h 中，对视频位置参数进行定义：

```

m_u32 uiTvmxsize;
m_u32 uiTvmysize;
m_u32 uiTvmxpos;
m_u32 uiTvmypos;

```

② 在库文件中 maividrend 文件夹下的 video.c 视频输出控制程序中，把定义的视频位置参数赋值给视频播放器的位置参数以及 MAE 里面的位置参数：

```

Tvm_xpos = pVideoDisplay->uiTvmxpos;
Tvm_ypos = pVideoDisplay->uiTvmypos;
Tvm_xsize = pVideoDisplay->uiTvmxsize;
Tvm_ysize = pVideoDisplay->uiTvmysize;
iodata.window.xpos = pVideoDisplay->uiDstXpos = Tvm_xpos;
iodata.window.ypos = pVideoDisplay->uiDstXpos = Tvm_ypos;
iodata.window.xsize = Tvm_xsize;
iodata.window.ysize = Tvm_ysize;
MAE_disp.display_linesize = Tvm_xsize;
MAE_disp.display_height = Tvm_ysize;

```

③ 视频播放窗口设置的硬件层实现：

```

    case 4:{
        if(fullscreen == 1){ //需要全屏播放
            Player_ExecuteEngineCommand(pPlayer, M_TEXT("setvideodisplay
            0,0,640,480,0"), (unsigned int *)M_NULL); //全屏播放的硬件实现
        }
    }
}

```

```

ypos_maiplayer = 5;
xpos_maiplayer = 5;
xsize_maiplayer = 640;
ysize_maiplayer = 480;
fullscreen = 0;
}
xpos_maiplayer = iodata.window.xpos;
ypos_maiplayer = iodata.window.ypos;
QString qstr_playclip_ypos=QString::number( ypos_maiplayer, 10 );
QString qstr_playclip_xpos=QString::number( xpos_maiplayer, 10 );
QString qstr_playclip_xsize=QString::number( xsize_maiplayer, 10 );
QString qstr_playclip_ysize=QString::number( ysize_maiplayer, 10 );
QString size_pos="setvideodisplay " + qstr_playclip_xpos + "," +
qstr_playclip_ypos + "," + qstr_playclip_xsize + "," + qstr_playclip_ysize + "," + "0";
Player_ExecuteEngineCommand(pPlayer, M_TEXT((char
*)(size_pos.asci))), (unsigned int *)M_NULL); //非全屏播放的硬件实现
issue_ioctl();
int_change = 0;
break;
}

```

播放器上层程序中的 *maiengine* 是通过设置 *pVideoDisplay* → *uiDstXpos* , *pVideoDisplay* → *uiDstYpos* 参数的值, 并且将他们传递给: *iodata.window.xpos* , *iodata.window.ypos* 来设置视频窗口的位置, 通过设置 *pVideoDisplay* → *uiDstXpos* 和 *pVideoDisplay* → *uiDstYpos* , 并将这两个值传递给参数 *iodata.window.xpos* 和 *iodata.window.ypos* 来设置外围 *framebuffer* 窗口的大小, 而如果想要正确的显示改变后视频图象的话, 还需要再设置一下 *MAE* (媒体加速引擎) 的具体参数 *MAE_disp.display_linesize = xsize* ; *MAE_disp.height_linesize = ysize* ; 具体修改 *MAE* 的位置是在 *MAE-drive* 文件目录的下面进行。

4.2.3 多媒体信息一体化播放界面的编译

tmake 是 Trolitech 公司创建的, 它是用来为不同的平台和编译器书写 *Makefile* 文件的工具。手写 *Makefile* 文件是比较困难并且比较容易出错的, 尤其是需要给不同的平台和编译器分别写几个不同的 *Makefile* 文件的时候。使用 *tmake* 工具, 开发者可以通过创建一个简单的 “.pro” 文件, 并且运行 *tmake* 工具, 就可以自动生成适当的 *Makefile* 文件。

tmake 会自动识别编译器, tmake 的利用可以为开发者节约出更多的时间和精力来编写其他更加重要的代码。Trolltech 公司使用 tmake 作为 Qt 库和 Qt 所提供的工具中的主要连接和编译工具。tmake 也注意了 Qt 的特殊需求, 它可以自动的包含 moc 和 uic 的连接和编译规则。

tmake 在生成 makefile 时自动包含变量定义: 编译器, 工具及选项, 文件指示: 输出地址、源文件、目标文件, 隐式规则, 安装规则, 子库, 编译。

以下是 maiplayer.pro 文件, 有如下信息:

```
SOURCES += main.cpp
unix{
    UI_DIR = .ui
    MOC_DIR = .moc
    OBJECTS_DIR = .obj
}
FORMS    = maiplayer.ui/    // 用户界面窗口
IMAGES   = images/filenew/
TEMPLATE = app //生成模板的类型
CONFIG += qt warn_on_release
LANGUAGE = C
```

转到保存这些文件的目录下, 接着运行 tmake 命令如下: tmake -o Makefile.pro, 就可以自动生成 Makefile 文件。再运行 make 命令, 根据已经生成的 Makefile 文件进行编译, 如果不出现运行错误的话, 在该目录下将生成一个新的可执行文件 maiplayer, 接下来, 就可以在通过终端输入 ./maiplayer 或者直接点击 maiplayer 图标, 运行已经生成的可执行程序。

4.3 基于 Au1200 视频实时示教功能的设计与实现

4.3.1 视频编解码器的算法优化

由前一章的阐述可以看出, 在视频解码的过程中, B 帧相对于 I 帧和 P 帧来说, 消耗的 CPU 资源更多, 如果想要顺畅的播放一些分辨率比较大, B 帧较多的视频, 就需要对视频编解码器以及播放器的实现代码进行算法优化^[56-62]。

运动估计是播放器算法的核心过程之一。该算法被频繁的调用, 在整个解码过程中占用很大比重, 在优化时, 本系统采用了前文介绍的梯度下降最优算法, 梯度下降最优算法是一个迭代的过程, 为了减少噪声的影响, 在第一次迭代时, 计算一个 e_i 的直方图, 在直方图中 e_i 值最大的前 10% 的像素点被剔除, 不参与迭代计算。详细的迭代过程描述如下:

① 每个位于 (x_i, y_i) 的像素点 i ，根据透射变换式与初始运动参数，计算相应的位置 (x', y') 。

② 算像素点 i 的误差值 e_i 。

③ 算误差值 e_i 关于各个运动参数的偏导数。

④ 像素点 f 的贡献加到海深矩阵 H 和梯度向量 b 中，然后判断是否处理完需要处理的像素，若未完成，则跳回(1)继续计算；否则下一步。

⑤ 解线性方程组： $H\Delta p = b$ ，得到 Δp ，更新运动参数： $p(k+1) = p(k) + \Delta p$ 。

⑥ 断是否满足收敛条件。若不满足，则跳到(1)继续迭代，否则结束算法。

收敛条件为满足以下三个条件之一：

1) 数达到32步；

2) 次迭代运动参数中平移分量变化小于预定阈值0.001；

3) 量外，相邻两次迭代运动参数中其他分量变化小于预定阈值0.00001。

由于传统基于梯度下降的分层实现算法的优点^[55]，本文采用的快速全局运动算法仍然采用这样思想，不同之处在于使用了更为鲁棒的M-估计器作为目标函数以减少噪声的影响，并增加了空间域特征点和时域特征点选取模块，这两个模块选取贡献大的特征点参与全局运动估计的计算，一方面减少了参加计算的像素数目，另一方面，所选取的特征点更能很好地体现全局运动趋势。由于这些改进，在保证准确度的同时，大大地加速了运算时间。

4.3.2 C 代码优化

在程序编辑过程当中，应该注意以下问题。通过这些编程方法的应用，可以起到优化系统资源配置的目的：

① 尽量少使用全局变量，因为全局变量要一直在内存中保留，这样一方面占用了内存的空间，另一方面，数据在内存中存取和运算要比在寄存器中进行运算慢很多。

② 在函数前面使用`_inline`关键字，这个关键字是表示让编译器直接调用函数体替代函数。这样虽然增加了程序长度，但是执行效率会提高很大。

③ 数组改成指针形式，有利于加快速度。

④ 尽量精简条件判断。

⑤ 使用`a++`替代`a=a+1`，`a+=6`替代`a=a+b`，`a[i++]`替代`a[i]`，`i++`。

4.3.3 视频编解码器优化前后的实时示教效果对比

图4.4是本论文所设计生产示教系统的播放器与Qt自带的嵌入式播放器在分别播放视频时，对CPU占用率的比较，为了验证系统的实时性，在1.5Mbps的码率和25fps的帧频下，对三种不同复杂度的视频格式进行了解码，测试结果如表所示。从图表中可以看出，本论文所设计的系统，即使对于相对复杂的码流，也可以达

到实时显示, 稳定播放的要求。因为本系统的资源占用率较低, 所以整个系统的稳定性相对于没有优化之前, 得到了有效提升, 从而能够使系统更加适合车间层的应用。

表 4.2 系统占用资源对比

Table 4.2 The comparison of system resources in using			
视频格式	MPEG-1	MPEG-2	MPEG-4
QT自带播放器CPU占用率	63%	80%	82%
本系统设计的播放器CPU占用率	22%	35%	50%

4.4 嵌入式视频生产示教系统的实现效果

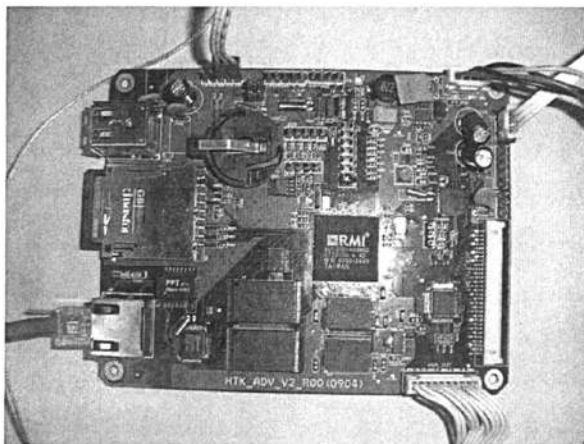


图 4.3 嵌入式视频生产示教系统的硬件实现

Fig 4.3 The hardware application of the system of teaching in video manufacturing based on embedded system

图 4.3 是本文所设计的嵌入式视频生产示教系统的硬件层实现。

本文所设计的嵌入式视频生产示教系统, 在经历了长时间, 不间断的运行测试之后, 表现出以下的特点:

关键的音视频播放功能运行流畅, JPEG、BMP、PNG、GIF 等图片格式的显示效果良好, 滚动字幕能够平滑的进行播放, 整个系统的画面协调、美观, 具有良好的外观和易用性, 同时, 系统的资源占用量较低, 系统的整体运行效果稳定。

在系统运行之前, 需要对文件系统中的多媒体播放控制文档进行编辑, 包括视频播放参数、图片播放参数以及文字播放参数的设定:

① 视频播放的参数配置:

[Option]

FTPADDR= //利用远程文件系统进行播放时所需的 IP

SIZE=26572820 //视频文件大小

Filename=/program/amd_lancel.mpg //视频文件目录名称

Frequency=10% //视频文件播放的频率

Md5Code=6cb53ac2d4cc3c9cbe088e0343f102e7 //特征识别码

② 图片播放的参数配置:

[Option]

SIZE=254357 //图片文件总体大小

Filename=/image/1.bmp //图片文件目录名称

Frequency=30% // 图片文件播放频率

Md5Code=29494d505e38f84ec6b63bed26adcccf //特征识别码

③ 文字播放的参数配置:

[Option]

FONTSIZE=10 //字体大小设置

FONTCOLOR=clWindowText //字体格式设置

BACKCOLOR=clCream //背景颜色设置

SPEED=26 //播放速度控制

MESSAGE= //滚动信息内容

对系统的多媒体播放参数进行上述的设定之后,接下来,就可以对系统的多媒体播放效果进行测试,具体的实现效果如图 4.4 和 4.5 所示:

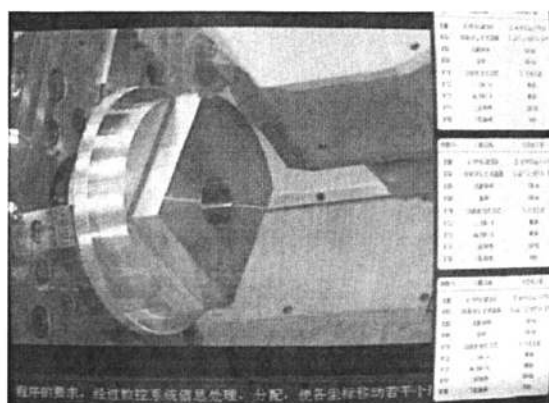


图 4.4 嵌入式视频生产示教系统的运行界面

Fig 4.4 The running interface of the system of teaching in video manufacturing based on embedded system

① 嵌入式视频生产示教系统多媒体一体化播放界面的实现如图 4.4 所示：图中左边是数控车床的零件加工演示视频，右边是零件加工的各项工艺参数，下方是数控加工过程中应当注意的一些技术规范信息。

② 嵌入式视频生产示教系统播放控制功能的实现如图 4.5 所示：

车间员工可以通过添加了播放控制按钮的 GUI 界面，对示教视频的播放进行控制，这能够让本文所开发的嵌入式视频生产示教系统产生更佳的车间层应用效果。



图 4.5 视频播放控制

Fig 4.5 The control of video display

4.5 本章小结

随着信息社会以及计算机软硬件技术的进步，嵌入式信息产品的设计和应用得到了迅速的发展，需要把Au1200等处理器芯片应用于嵌入式Linux下进行嵌入式视频生产示教系统开发的需求也会变得越来越普遍，本章在阐述了一些系统开发过程中所遇到的关键技术问题解决方案之后，重点针对系统实时示教功能开发过程中，所遇到的资源占用率高，播放卡壳，音视频不同步的问题，进行了算法优化和C代码优化，取得了良好的优化效果。最后本章给出了系统的实现效果。

5 结论

本论文针对目前车间层对于多媒体信息示教的需求,研究了嵌入式系统、多媒体技术、GUI 技术的研究现状以及生产示教系统的技术应用研究现状,完成了嵌入式视频生产示教系统整体架构的设计,研究了嵌入式视频生产示教系统在开发过程中所遇到的一些关键技术问题,在最后实现了这些关键技术问题,并重点对嵌入式视频生产示教系统的实时示教功能进行了有针对性的优化处理。取得如下成果:

① 提出了基于 Au1200 的嵌入式视频生产示教系统的软件架构,系统能够支持包括音视频在内的多种信息载体,而且运行情况良好,这为车间层的生产示教提供了有效的解决方案。

② 在系统实时示教功能研发的过程中,针对系统资源占用高的问题,从算法优化和 C 代码优化的角度,进行了处理,通过这些优化处理,能够提高本系统在车间层应用中的稳定性,从而有效提高系统的整体性能。

③ 常见绘图软件对 Linux 的支持较少,本文在对系统实时示教功能优化的基础之上,对 QT 相关技术进行了验证和实现,方便了车间层员工对本系统的应用。

系统试运行结果表明,本系统在播放视频等多媒体文件的过程中,资源占用较少,系统在长时间运行的过程中,性能比较稳定,从而证明了本文所设计的系统是可行的,具有较高的应用价值。

致 谢

在硕士论文完成之际，我由衷地感谢我的导师，朱文老师、刘飞老师和鄢萍老师，本文的研究工作自始至终都得到了老师们热情、耐心和细致的指导与关怀，论文的字里行间都饱含了老师们的心血和汗水，在此谨向他们表示最衷心的感谢并致以最崇高的敬意！回首近三年的求学生涯，老师严谨的治学态度、忘我的工作作风、一丝不苟的求实精神、清廉节俭的生活作风给予了我许多深刻的教诲。老师惜时如金，但在指导学生时却从不吝分惜秒，以他们那博大的胸怀、高尚的品格、渊博的学识、系统工程的思想、高瞻远瞩和创造性的思维方式培育学生做人和成才，老师不仅指导我们如何学习和工作，更教会了我们如何做人处事，这将使我们终生受益。

衷心感谢重庆海特克系统集成有限公司总经理易润忠先生在课题的研究过程中所给予的细心指导和帮助。

衷心感谢重庆大学制造工程研究所三年来给我提供的良好科研环境，使我能够顺利地完科研课题，从而能够撰写出本论文，同时使我的学术和技术水平得到了提高。

衷心感谢课题组贺晓辉、胡新元、任凡、童亮、王东强等博士研究生，孟祥涛、向丹丹、洪亮、张巍、吴文娟、韩燕丽、齐维龙、张朝霞、刘辉亮、林茂疆、林志雄等硕士研究生在我论文写作期间给予我的支持和帮助，使我的论文和课题能顺利地完，在此向他们表示衷心的感谢与美好的祝福！

衷心的感谢王院长，邓院长和陈老师对我生活上以及学业上的关心与帮助，我将在以后的工作和学习当中，更加积极努力，为我可敬又可爱的母校争光。

感谢远在家乡、每天节衣缩食、一心一意供我读书、白发丹心的父母亲，你们的含辛茹苦与谆谆教诲，做儿子的会永记心间。

最后，衷心感谢在百忙之中评阅论文和参加答辩的各位专家、教授！

秦大鹏

二〇〇九年四月 于重庆

参考文献

- [1] 徐艳, 张承伟等. 大东区制造业发展战略研究[D]. 大连理工大学硕士学位论文, 2007.
- [2] 王仕林. 青岛制造业技术创新发展模式研究[D]. 青岛科技大学硕士学位论文, 2006.
- [3] 张旭梅, 但斌, 刘飞. 企业信息化工程[M]. 北京: 科学出版社, 2003.
- [4] 刘飞. 制造自动化的广义内涵、研究现状和发展趋势[J]. 机械工程学报, 1999, 35(1): 1-5
- [5] 丁伯慧, 张付英. 先进制造技术与模式的研究综述[J]. 机电一体化, 2003.
- [6] 罗振壁, 汪劲松, 周兆英等. 敏捷制造-21 世纪制造企业的战略[J]. 机械工程学报, 1994, 3(4): 21-25
- [7] 姚锡凡, 张毅, 田春光. 制造业信息化内涵及其在我国发展的现状[J]. 机电工程技术, 2003, 32(3): 21-25
- [8] 张曙. 制造业信息化的内涵和策略[J]. 中国工程科学, 2001, 3(7): 7-11
- [9] John Parker, Mechanical Manufacturing Technology in the 21st Century [J]. The Proceedings of the First International Conference on Mechanical Engineering. China, 2000.
- [10] 于海斌, 朱云龙. 可集成的制造执行系统[J]. 计算机集成制造系统-CIMS, 2000, 6(6): 1-6.
- [11] Liu Fei, Yin Chao, Liu Sheng. Regional networked manufacturing system. Chinese Journal of Mechanical Engineering. 2000, 13(Supplement): 97-103
- [12] 刘飞, 雷琦, 宋豫川. 网络化制造的内涵及研究发展趋势[J]. 机械工程学报. 2003, 39(8): 1-6
- [13] 张大波. 嵌入式系统原理. 设计与应用[M]. 北京: 机械工业出版社, 2004. 11
- [14] 周炜. 基于嵌入式 Linux 无线流媒体播放器的设计与实现[D]. 合肥工业大学硕士学位论文, 2008
- [15] ITU-T Rec. H.264/ISO/IEC 11496-10. "Advanced Video Coding"[M]. Final Committee Draft, Document JVTE022, September 2002
- [16] 程璞, 李奇, 倪超. 基于嵌入式系统的家庭智能网络终端设计[J]. 微型电脑应用, 2005, 21(10): 50-52.
- [17] 舒冬梅, 王芬芬. 远程测控终端的设计[J]. 仪器仪表学报, 2003, 24(4): 82-84.
- [18] 孙宏. HUST-B 远程终端装置[J]. 信息技术, 2003, 27(6): 80-81.
- [19] 杜亚娟, 陆新, 蔡惠明等. 手持信息终端的研究[J]. 计算机工程, 2002, 28(9): 21-23.
- [20] 王文生, 贾志科, 唐昆, 崔慧娟. 基于优化 SPIHT 的实时视频编码算法[J]. 清华大学学报(自然科学版). 2003, 43(1): 27-30.
- [21] 赵安邦, 秦波, 唐昆. 嵌入式多媒体通信终端设计及实现[J]. 微计算机信息, 2005, 21(7): 1-3.

- [22] 陈云鹤, 徐晶. 基于 QT/Embedded 的媒体播放器的设计与实现[J]. 微型电脑应用. 2006, 22 (6) : 33-34, 52.
- [23] 洗俊峰, 赵小侠, 钟玉琢. 多媒体技术的应用现状与发展趋势[J]. 广西广播电视大学学报: 2002, 13(3) : 16-21.
- [24] K. Nadehara, H.J. Stolberg, M. Ikekawa, E. Murata, I. Kuroda. Real-time software MPEG-1 video decoder design for low-cost, low-power applications[J]. IEEE Workshop on VLSI Signal Processing. 1996, 45: 438-447.
- [25] 程清钧. 多媒体技术与应用[M]. 北京: 高等教育出版社, 2003: 43-59.
- [26] 杨素敏, 张西红. VBR MPEG 视频传输模型[J]. 计算机与网络, 2003(12): 16-19.
- [27] Information technology- Generic coding of moving pictures and associated audio information-part2: Video, ISO/IEC 13818-2, 2000: 33-36.
- [28] ISO/IEC. IS 11172: Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 MBit/s(MPEG-1) [S], 1993.
- [29] ISO/IEC. IS 13818-2: Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video(MPEG-2 Video) [S], 1995.
- [30] ISO/IEC IS 14496-2: Information Technology—Generic Coding of Audio-visual Objects part2: Visual(MPEG-4 Video) [S], 1999.
- [31] 数字音视频编解码技术标准工作组. 信息技术先进音视频编码第二部分[M]. 视频, 2005.
- [32] 周立功等. ARM 嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2005.
- [33] 百度贴吧: <http://tieba.baidu.com/f?kz=139720370>
- [34] 百度百科: <http://baike.baidu.com/view/3226.htm>
- [35] 王云博等. 一种轻量级嵌入式 GUI 系统及其实现[J]. 计算机应用. 2006(9): 2244-2246, 2249.
- [36] 孙少华, 徐立中. 面向嵌入式 Linux 系统的图形用户界面[J]. 微机发展, 2005, 15 (10): 123-125.
- [37] 丁丁, 习勇, 魏急波. 三种主流嵌入式图形用户界面的移植及性能比较[J]. 电子产品世界, 2004, (09): 73 ~74, 79.
- [38] 夏琳, 张光显, 张丽娟. 基于 Linux 操作系统的嵌入式 GUI 及其设计改进[J]. 现代电子技术, 2004, (21): 103-105.
- [39] Qt 参考文档. <http://www.qtopia.org.cn/doc/qiliang.net/qt/index.html> [Z]. 2007-05-04
- [40] Troltech. Qt Reference Documentation Qt / Embedded version 2. 3. 7[Z]. Norway: Troltech 公司, 2001.
- [41] 任善全, 吕强, 钱培德, 杨季文. 一个基于 Qt / Embedded 的嵌入式 Linux 应用程序的实现[J]. 计算机应用与软件, 2006, 23 (2): 105-107.

- [42] 万长胜等. 用 Qt 进行面向对象软件的设计与开发[J]. 核电子学与探测技术, 2004,24(3):311-313.
- [43] 李全虎. 交互界面开发工具-Qt[J]. 中国科技信息, 2005,(5):33.
- [44] AMD.AMD Technologies Inc. AMD Alchemy™ Au1200™ Processor Data Book[Z]. 2005-07.
- [45] AMD.AMD Technologies Inc. AMD Alchemy Au1200 Mae White Paper[Z]. 2005-07.
- [46] AMD.AMD Technologies Inc. AMD Alchemy™ Au1200™ Processor Media Application Interface Software Development Kit[Z]. 2005-07.
- [47] 方明. 基于 AMD 嵌入式处理器的广告媒体播放器研究与实现[D]. 西安电子科技大学硕士学位论文. 2006.
- [48] 李玉星. 装备嵌入式信息交互终端人机界面设计方法的研究与应用[D]. 重庆大学硕士学位论文. 2007.05
- [49] 许建. 基于 QT 的嵌入式浏览器和 GUI 的实现[D]. 西安电子科技大学硕士学位论文, 2008.
- [50] 张春艳. 基于 Qt 的嵌入式图形用户界面研究与实现[D]. 大连海事大学硕士学位论文, 2008.
- [51] 陈云鹤. 基于 Qt 的嵌入式媒体播放器的设计与实现[D]. 华中科技大学硕士学位论文. 2006.
- [52] QT 参考文档: <http://www.qiliang.net/qt/signalsandslots.html>
- [53] QT 参考文档: <http://www.qiliang.net/qt/qobject.html>
- [54] QT 参考文档: <http://www.qiliang.net/qt/geometry.html>
- [55] 袁亚湘, 孙文瑜. 最优化理论和方法[M]. 北京: 科学出版社, 1997.
- [56] ITU-T REC.H.264 ISO/IEC 14496-10 AVC, Document JVT G050, 7th Meeting: Pattaya, Thailand, March, 2003
- [57] Thomas Wiegand, Gary J. Sullivan, Gisle Bjntegaard, and Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Syst. Video Technol., Vol.13(7), p.657-673, July 2003.
- [58] Ralf Schafer, Thomas Wiegand, Heiko Schwarz, "The emerging H.264/AVC Standard" EBU Technical Review, Jan. 2003
- [59] Ye ui Warig, Miska M. Hannuksela, Viktor Varsa, "The Error Concealment Feature in the H.26L Test Model" Proc. IEEE International Conference on Image Processing (ICIP'02), Vol. II, pp. 729-732, Sept. 2002.
- [60] S. enger, T. Stockhammer, "An Overview on the H.264 NAL Concept" Doc. JVT-B028, Geneva, Switzerland, January 2002

- [61] Thomas Wiegand, Gary J. Sullivan Et Al. Overview Of The H.264/AVC Video Coding Standard. IEEE Transactions On Circuits And Systems For Video Technology, 2003.
- [62] Atul Puria, Xuemin Chen, Ajay Luthra. Video Coding Using the H.264/MPEG-4 AVC Compression Standard. Signal Processing: Image Communication, 2004.
- [63] 王勇, 盖江南, 阎文丽. 嵌入式系统编程源代码解析[M]. 北京: 电子工业出版社, 2008.
- [64] 陈翌, 田捷, 王金刚. 嵌入式软件开发技术[M]. 北京: 国防工业出版社, 2003.

附录

A. 攻读硕士学位期间发表的论文

- [1] 秦大鹏, 贺晓辉. 一种基于 Au1200 的嵌入式高清媒体播放器的视频处理优化. 机电一体化 (已录用).

B. 攻读硕士学位期间参加的科研课题

- [1] 作为项目主研人员参加了国家 863 重点项目“面向离散制造的可配置 MES 产品及行业解决方案 (重点)” (项目编号: 2007AA040701);
- [2] 作为项目主研人员参加了重庆市科技攻关项目“车间制造过程信息化支持平台开发与应用示范” (项目编号: CSTC2007AA2013);
- [3] 作为项目主研人员参加了国家创新基金课题“32 位嵌入式智能仪器仪表开发平台及应用” (项目编号: 07C26215111917)。