



电子科技大学

University of Electronic Science and Technology of China

学 士 学 位 论 文

BACHELOR DISSERTATION

论文题目 基于RTOS的智能USB转接器

学生姓名 _____ 孙启民

学 号 _____ 2011042040022

专 业 _____ 电子信息科学与技术

学 院 _____ 物理电子学院

指导教师 _____ 刘艺

指导单位 _____ 电子科技大学

2015年5月25日

摘 要

本课题设计了一个基于RTOS智能USB 转接件。使用RT-Thread 作为系统的软件平台，两个STM32单片机作为硬件平台。通过一个自制的语法解释器，将用户可以随时自行编写的内部文件解释为键盘的过滤器，进而实现键盘的改键、键盘宏、宏录制等功能，进而增强用户对键盘的掌控能力，可以轻易实现复杂的键盘操作。本课题可看做windows 应用程序“按键精灵”的简易硬件实现。

关键词：RT-Thread，USB，单片机，按键精灵

ABSTRACT

This paper design a RTOS-based intelligent USB adapter. Use RT-Thread as the software platform, two STM32 MCU as the hardware platform. through a DIY grammar interpreter, the user can always write your own internal documents interpreted as a keyboard filter, thus achieving change keyboard keys, keyboard macros, macro recording and other functions, and thus enhance the user's ability to control the keyboard, you can easily implement complex keyboard. This system can be seen as a hardware implementation of windows application "QuickMacro".

Keywords: RT-Thread, USB, MCU, QuickMacro

目 录

| | |
|---------------------------------|---|
| 第1章 引言 | 1 |
| 1.1 USB转接器的背景与意义 | 1 |
| 1.2 USB转接器的设计内容 | 2 |
| 1.2.1 主控芯片选型 | 2 |
| 1.2.2 PCB设计 | 3 |
| 1.2.3 在主控芯片上安装RT-Thread系统 | 3 |
| 1.2.4 编写基于RTT系统的相关模块驱动 | 3 |
| 1.2.5 编写应用层程序 | 3 |
| 1.3 功能简介 | 3 |
| 1.3.1 改键功能 | 3 |
| 1.3.2 键盘宏 | 4 |
| 1.3.3 蓝牙KVM | 4 |
| 1.3.4 语音识别 | 4 |
| 1.3.5 鼠标手势识别 | 4 |
| 1.4 特色与创新 | 4 |
| 1.4.1 嵌入式操作系统 | 4 |
| 1.4.2 在线编程 | 5 |
| 1.4.3 跨平台 | 5 |
| 第2章 引言 | 6 |
| 2.1 USB转接器的背景与意义 | 6 |
| 2.2 USB转接器的设计内容 | 7 |
| 2.2.1 主控芯片选型 | 7 |
| 2.2.2 PCB设计 | 8 |
| 2.2.3 在主控芯片上安装RT-Thread系统 | 8 |
| 2.2.4 编写基于RTT系统的相关模块驱动 | 8 |
| 2.2.5 编写应用层程序 | 8 |
| 2.3 功能简介 | 8 |
| 2.3.1 改键功能 | 8 |

| | |
|------------------------------------|-----------|
| 2.3.2 键盘宏 | 9 |
| 2.3.3 蓝牙KVM | 9 |
| 2.3.4 语音识别 | 9 |
| 2.3.5 鼠标手势识别 | 9 |
| 第3章 时域积分方程数值方法研究 | 10 |
| 3.1 时域积分方程时间步进算法的阻抗元素精确计算 | 10 |
| 3.2 时域积分方程时间步进算法阻抗矩阵的存储 | 10 |
| 3.2.1 时域积分方程时间步进算法产生的阻抗矩阵的特征 | 10 |
| 3.2.2 数值算例与分析 | 10 |
| 3.3 时域积分方程时间步进算法矩阵方程的求解 | 11 |
| 3.4 本章小结 | 12 |
| 第4章 全文总结与展望 | 13 |
| 4.1 全文总结 | 13 |
| 4.2 后续工作展望 | 13 |
| 参考文献 | 14 |
| 致 谢 | 14 |
| 外文资料原文 | 15 |
| 外文资料译文 | 17 |

缩略词表

主要符号表

第1章 引言

1.1 USB转接器的背景与意义

键盘是最常见的计算机输入设备，它广泛应用于微型计算机和各种终端设备上，计算机操作者通过键盘向计算机输入各种指令、数据，指挥计算机的工作。计算机的运行情况输出到显示器，操作者可以很方便地利用键盘和显示器与计算机对话，对程序进行修改、编辑，控制和观察计算机的运行。

键盘的接口有AT接口、PS/2接口和最新的USB接口，台式机曾多采用PS/2接口，大多数主板都提供PS/2 键盘接口。而较老的主板常常提供AT 接口也被称为“大口”，已经不常见了。USB作为新型的接口，一些公司迅速推出了USB接口的键盘。由于USB 接口具有热插拔、功能多、速度快等特点，现代电脑普遍使用USB外接键盘。

USB，是英文Universal Serial Bus（通用串行总线）的缩写，而其中文简称为“通串线”，是一个外部总线标准，用于规范电脑与外部设备的连接和通讯。是应用在PC领域的接口技术。USB接口支持设备的即插即用和热插拔功能。USB是在1994年底由英特尔、康柏、IBM、Microsoft等多家公司联合提出的。

现代键盘普遍使用固件功能的芯片作为USB键盘的主控芯片。这种主控芯片的优点是成本低，稳定性高。缺点则是可扩展性差，无法完全对键盘进行控制。

要实现对键盘的完全控制，有软件和硬件两种方案。

硬件方案：制作USB转接器，将键盘的USB信息进行处理，再转发给主机，目前没有相关产品上市。

软件方案：AutoHotKey、按键精灵等。

硬件方案和软件相比，具有跨平台（软件和硬件平台）、无安装过程、不占用CPU等特点，缺点则是需要购置费用。

现代常用的通用操作系统有windows、Linux、安卓、MAC、iOS 等。由于系统底层接口差别很大，AutoHotKey与按键精灵这类软件几乎不可能实现跨平台使用，因此硬件方案在某些时候是唯一的选择，因此具有重要意义。

本课题在硬件上将实现类似“按键精灵”的功能，使用精简的AHK语法，给键盘使用者带来跨平台的相同使用体验。

1.2 USB转接器的设计内容

本课题制作了一个基于RTOS的智能USB转接器。其结构图如下：

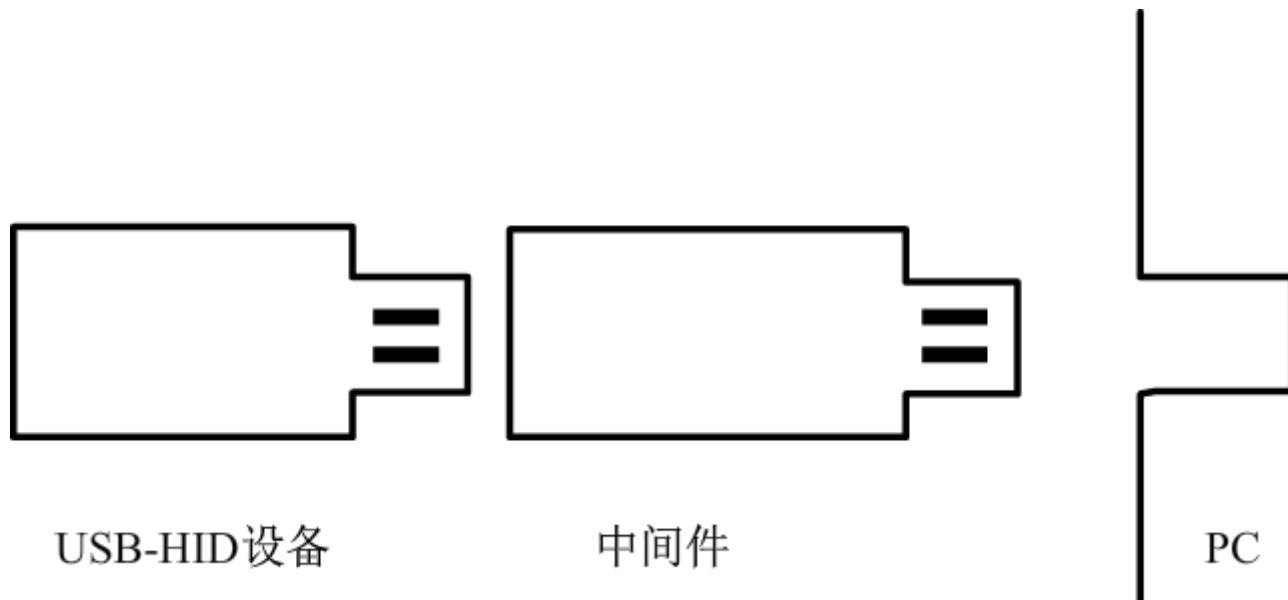


图 1-1 转接器示意图

如图2-1所示，本课题制作了一个USB转接器，同时作为USBHost与USBdev，置于USB-HID设备与USB主机之间，将HID设备的信息读取后进行处理再发给USB主机。

1.2.1 主控芯片选型

本课题由STM32单片机、OLED显示模块、USB接口等模块组成，各个模块之间通过单片机组成了一个有机的整体。由于本课题使用了USBhost接口和USBdev接口，所以需要选择带有USBhost（或OTG）和USBdev接口的单片机。目前不带MMU的单片机没有同时带两个USB口的，因此需要使用两个单片机进行协作来实现功能。由于作为USBhost的单片机需要实现较多的功能，因此应该选择主频较高的单片机，调查市面上的单片机型号之后，最终确定为STM32F407VG。作为USBdev的单片机只需要实现USBdev功能即可，所以可以选择价格较低的单片机。考虑到开发工具的统一，最终选定的作为USBdev的单片机是STM32F103C8。

1.2.2 PCB设计

由于市面上没有相关产品，本课题需要自行设计PCB并进行焊接。具体的电路设计将在下文中进行介绍。考虑到焊接难度，主控芯片选择的是较容易焊接的QFP封装。

1.2.3 在主控芯片上安装RT-Thread系统

RT-Thread是新兴的国产实时操作系统（RTOS），在核心板上移植RT-Thread，可以让RT-Thread运行在本课题的硬件上，从而使用RT-Thread带有的系统功能，方便本系统的开发，也能增强系统能力。

1.2.4 编写基于RTT系统的相关模块驱动

由于RT-Thread是嵌入式操作系统，所以操作底层硬件需要编写相应的驱动程序，便于应用层进行调用。

1.2.5 编写应用层程序

应用层程序即功能型程序，就是将本课题需要实现的功能通过编写线程进行实现。

1.3 功能简介

本系统由于是为了应用而开发，所以应用层的工作量比例较大，开发出了多种非常具有实用价值的功能。

1.3.1 改键功能

例如：通过修改文件系统中的KEY_T键盘映射文件，可以修改键盘键位映射，语法形式为：“RALT=RCTRL”。这个语句表示右边的ALT键将映射为右边的CTRL键。各个语句之间并不互相影响，即“RCTRL=RALT RCTRL=RCTRL”之后，相当于没有改键。

1.3.2 键盘宏

键盘宏，顾名思义就是将快捷键定义为其事件。比如将左边的CTRL+P定义为输出银行卡号码，将右边的ALT+N定义为向下箭头（仿emacs快捷键）。语法形式（仿AHK）为：“>~p::{up}”，意为将右边的ctrl键定义为向上箭头。通过使用键盘宏，可以极大地增强用户对键盘的控制能力，减少关键信息的重复输入（手机号码等）。通过仿emacs定义键位，可以实现全局emacs键位，可以极大地提高用户的输入速度。

1.3.3 蓝牙KVM

KVM：就是Keyboard Video Mouse的缩写，即可以将鼠标键盘视频模拟连接到其他电脑上。本系统只实现了键盘和鼠标的连接。本系统的硬件电路上集成有一个蓝牙模块，可以和其他的相同系统进行蓝牙通信。通过蓝牙，可以使用一套鼠标键盘控制多个硬件，极大地减少了跨平台工作者的操作复杂度。

1.3.4 语音识别

本系统硬件上集成了一个国产语音识别芯片LD3320，可以通过设置拼音进行识别。通过语音识别，可以方便地对电脑进行控制，比如在没有空余手的时候进行语音翻页。

1.3.5 鼠标手势识别

系统应用层通过算法对鼠标的手势进行识别，可以通过模拟键盘实现一些功能（比如最小化窗口，打开计算器等）。

1.4 特色与创新

本系统基于本人需求进行开发，市面上并没有相同的产品出售，因此系统的绝大部分功能都是自主创新。

1.4.1 嵌入式操作系统

本系统基于RT-Thread开发，可扩展性强，可定制性强，且具有多进程处理能力。传统小型嵌入式设备一般直接进行裸机编程，而本系统为了增强系统的能力，基于RT-Thread进行开发，充分利用了RTOS的优势。

1.4.2 在线编程

使用自制脚本分析器，可以进行在线编程，无需仿真器或设备。使得普通用户也能通过修改文件系统内的文件进行复杂的操作，大大增强了系统的可定制程度。

1.4.3 跨平台

通过实现标准USB-HID设备，实现真正的跨平台，Android、MAC、Linux、Windows都通过了测试。

第2章 引言

2.1 USB转接器的背景与意义

键盘是最常见的计算机输入设备，它广泛应用于微型计算机和各种终端设备上，计算机操作者通过键盘向计算机输入各种指令、数据，指挥计算机的工作。计算机的运行情况输出到显示器，操作者可以很方便地利用键盘和显示器与计算机对话，对程序进行修改、编辑，控制和观察计算机的运行。

键盘的接口有AT接口、PS/2接口和最新的USB接口，台式机曾多采用PS/2接口，大多数主板都提供PS/2 键盘接口。而较老的主板常常提供AT 接口也被称为“大口”，已经不常见了。USB作为新型的接口，一些公司迅速推出了USB接口的键盘。由于USB 接口具有热插拔、功能多、速度快等特点，现代电脑普遍使用USB外接键盘。

USB，是英文Universal Serial Bus（通用串行总线）的缩写，而其中文简称为“通串线”，是一个外部总线标准，用于规范电脑与外部设备的连接和通讯。是应用在PC领域的接口技术。USB接口支持设备的即插即用和热插拔功能。USB是在1994年底由英特尔、康柏、IBM、Microsoft等多家公司联合提出的。

现代键盘普遍使用固件功能的芯片作为USB键盘的主控芯片。这种主控芯片的优点是成本低，稳定性高。缺点则是可扩展性差，无法完全对键盘进行控制。

要实现对键盘的完全控制，有软件和硬件两种方案。

硬件方案：制作USB转接器，将键盘的USB信息进行处理，再转发给主机，目前没有相关产品上市。

软件方案：AutoHotKey、按键精灵等。

硬件方案和软件相比，具有跨平台（软件和硬件平台）、无安装过程、不占用CPU等特点，缺点则是需要购置费用。

现代常用的通用操作系统有windows、Linux、安卓、MAC、iOS 等。由于系统底层接口差别很大，AutoHotKey与按键精灵这类软件几乎不可能实现跨平台使用，因此硬件方案在某些时候是唯一的选择，因此具有重要意义。

本课题在硬件上将实现类似“按键精灵”的功能，使用精简的AHK语法，给键盘使用者带来跨平台的相同使用体验。

2.2 USB转接器的设计内容

本课题制作了一个基于RTOS的智能USB转接器。其结构图如下：

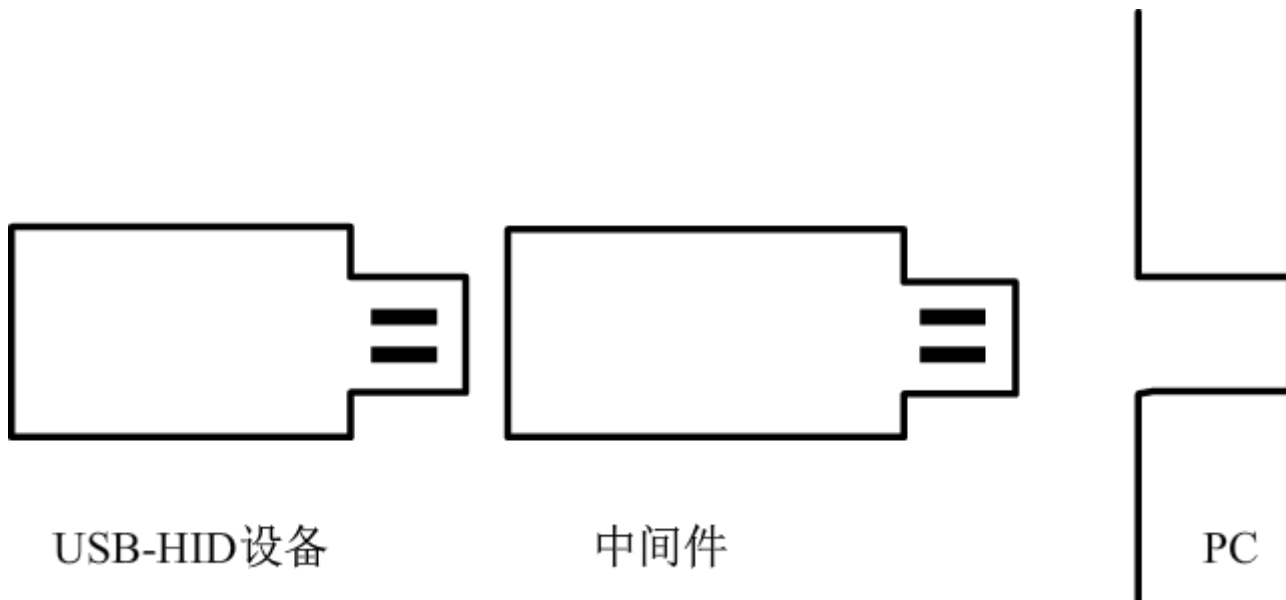


图 2-1 转接器示意图

如图2-1所示，本课题制作了一个USB转接器，同时作为USBHost与USBdev，置于USB-HID设备与USB主机之间，将HID 设备的信息读取后进行处理再发给USB主机。

2.2.1 主控芯片选型

本课题由STM32单片机、OLED显示模块、USB接口等模块组成，各个模块之间通过单片机组成了一个有机的整体。由于本课题使用了USBhost 接口和USBdev接口，所以需要选择带有USBhost（或OTG）和USBdev接口的单片机。目前不带MMU的单片机没有同时带两个USB口的，因此需要使用两个单片机进行协作来实现功能。由于作为USBhost的单片机需要实现较多的功能，因此应该选择主频较高的单片机，调查市面上的单片机型号之后，最终确定为STM32F407VG。作为USBdev的单片机只需要实现USBdev功能即可，所以可以选择价格较低的单片机。考虑到开发工具的统一，最终选定的作为USBdev的单片机是STM32F103C8。

2.2.2 PCB设计

由于市面上没有相关产品，本课题需要自行设计PCB并进行焊接。具体的电路设计将在下文中进行介绍。考虑到焊接难度，主控芯片选择的是较容易焊接的QFP封装。

2.2.3 在主控芯片上安装RT-Thread系统

RT-Thread是新兴的国产实时操作系统（RTOS），在核心板上移植RT-Thread，可以让RT-Thread运行在本课题的硬件上，从而使用RT-Thread带有的系统功能，方便本系统的开发，也能增强系统能力。

2.2.4 编写基于RTT系统的相关模块驱动

由于RT-Thread是嵌入式操作系统，所以操作底层硬件需要编写相应的驱动程序，便于应用层进行调用。

2.2.5 编写应用层程序

应用层程序即功能型程序，就是将本课题需要实现的功能通过编写线程进行实现。

2.3 功能简介

本系统由于是为了应用而开发，所以应用层的工作量比例较大，开发出了多种非常具有实用价值的功能。

2.3.1 改键功能

例如：通过修改文件系统中的KEY_T键盘映射文件，可以修改键盘键位映射，语法形式为：“RALT=RCTRL”。这个语句表示右边的ALT键将映射为右边的CTRL键。各个语句之间并不互相影响，即“RCTRL=RALT RCTRL=RCTRL”之后，相当于没有改键。

2.3.2 键盘宏

键盘宏，顾名思义就是将快捷键定义为其事件。比如将左边的CTRL+P定义为输出银行卡号码，将右边的ALT+N定义为向下箭头（仿emacs 快捷键）。语法形式（仿AHK）为：“>~p::{up}”，意为将右边的ctrl键定义为向上箭头。通过使用键盘宏，可以极大地增强用户对键盘的控制能力，减少关键信息的重复输入（手机号码等）。通过仿emacs定义键位，可以实现全局emacs键位，可以极大地提高用户的输入速度。

2.3.3 蓝牙KVM

KVM：就是Keyboard Video Mouse的缩写，即可以将鼠标键盘视频模拟连接到其他电脑上。本系统只实现了键盘和鼠标的连接。本系统的硬件电路上集成有一个蓝牙模块，可以和其他的相同系统进行蓝牙通信。通过蓝牙，可以使用一套鼠标键盘控制多个硬件，极大地减少了跨平台工作者的操作复杂度。

2.3.4 语音识别

本系统硬件上集成了一个国产语音识别芯片LD3320，可以通过设置拼音进行识别。通过语音识别，可以方便地对电脑进行控制，比如在没有空余手的时候进行语音翻页。

2.3.5 鼠标手势识别

系统应用层通过算法对鼠标的手势进行识别，可以通过模拟键盘实现一些功能（比如最小化窗口，打开计算器等）。

第3章 时域积分方程数值方法研究

3.1 时域积分方程时间步进算法的阻抗元素精确计算

时域积分方程时间步进算法的阻抗元素直接影响算法的后时稳定性，因此阻抗元素的计算是算法的关键之一，采用精度高效的方法计算时域阻抗元素是时域积分方程时间步进算法研究的重点之一。

.....

3.2 时域积分方程时间步进算法阻抗矩阵的存储

时域阻抗元素的存储技术也是时间步进算法并行化的关键技术之一^[7]，采用合适的阻抗元素存储方式可以很大的提高并行时间步进算法的计算效率。

3.2.1 时域积分方程时间步进算法产生的阻抗矩阵的特征

.....

由于时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组合，因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积分方程时间步进算法的阻抗矩阵特征相同。

3.2.2 数值算例与分析

.....。如表3-1所示给出了时间步长分别取0.4ns、0.5ns、0.6ns时的三种存储方式的存储量大小。.....。

如图3-1(a)所示给出了时间步长选取为0.5ns时采用三种不同存储方式计算的平板中心处 x 方向的感应电流值与IDFT方法计算结果的比较，.....。如图3-1(b)所示给出了存储方式为基权函数压缩存储方式，时间步长分别取0.4ns、0.5ns、0.6ns时平板中心处 x 方向的感应电流计算结果，从图中可以看出不同时间步长的计算结果基本相同。

由于时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组合，因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积分方程时间步进算法的阻抗矩阵特征相同。

表 3-1 计算 $2m \times 2m$ 理想导体平板时域感应电流采用的三种存储方式的存储量比较

| 存储方式 时间步长 | 非压缩存储方式 | 完全 压缩存储方式 | 基元函数 压缩存储方式 |
|--------------|----------|--------------|----------------|
| 0.4ns | 11.96 MB | 5.59 MB | 6.78 MB |
| 0.5ns | 10.17 MB | 5.06 MB | 5.58 MB |
| 0.6ns | 8.38 MB | 4.65 MB | 4.98 MB |

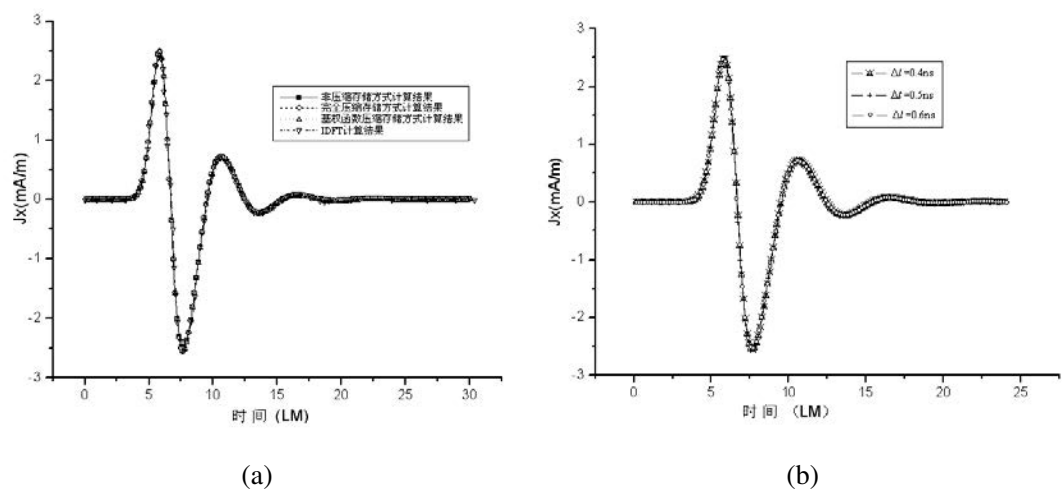


图 3-1 $2m \times 2m$ 的理想导体平板中心处感应电流 x 分量随时间的变化关系
(a)不同存储方式的计算结果与IDFT方法的结果比较；(b)不同时间步长的计算结果比较

3.3 时域积分方程时间步进算法矩阵方程的求解

.....

定理 3.3.1 如果时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组合.....

证明 首先, 由于.....

.....

根据....., 结论得证。 ■

3.4 本章小结

本章首先研究了时域积分方程时间步进算法的阻抗元素精确计算技术，分别采用DUFFY 变换法与卷积积分精度计算法计算时域阻抗元素，通过算例验证了计算方法的高精度。……

第4章 全文总结与展望

4.1 全文总结

本文以时域积分方程方法为研究背景，主要对求解时域积分方程的时间步进算法以及两层平面波快速算法进行了研究。

.....

4.2 后续工作展望

时域积分方程方法的研究近几年发展迅速，在本文研究工作的基础上，仍有以下方向值得进一步研究：

.....

致 谢

在攻读博士学位期间，首先衷心感谢我的导师 XXX 教授，……
.....

The Name of the Game

1.1 xxx

1.1.1 xxx

1.1.1.1 xxxx

1.2 xxx

1.2.1 xxx

1.2.1.1 xxxx

English words like ‘technology’ stem from a Greek root beginning with the letters $\tau\epsilon\chi\ldots$; and this same Greek word means *art* as well as technology. Hence the name $\text{T}_{\text{E}}\text{X}$, which is an uppcase form of $\tau\epsilon\chi$.TeX (actually $\text{T}_{\text{E}}\text{X}$), meaning of $\tau\epsilon\chi$

Insiders pronounce the χ of $\text{T}_{\text{E}}\text{X}$ as a Greek chi, not as an ‘x’, so that $\text{T}_{\text{E}}\text{X}$ rhymes with the word blecchhh. It’s the ‘ch’ sound in Scottish words like *loch* or German words like *ach*; it’s a Spanish ‘j’ and a Russian ‘kh’. When you say it correctly to your computer, the terminal may become slightly moist.

The purpose of this pronunciation exercise is to remind you that $\text{T}_{\text{E}}\text{X}$ is primarily concerned with high-quality technical manuscripts: Its emphasis is on art and technology, as in the underlying Greek word. If you merely want to produce a passably good document—something acceptable and basically readable but not really beautiful—a simpler system will usually suffice. With $\text{T}_{\text{E}}\text{X}$ the goal is to produce the *finest* quality; this requires more attention to detail, but you will not find it much harder to go the extra distance, and you’ll be able to take special pride in the finished product.

On the other hand, it’s important to notice another thing about $\text{T}_{\text{E}}\text{X}$ ’s name: The ‘E’ is out of kilter. This logo displaced ‘E’ is a reminder that $\text{T}_{\text{E}}\text{X}$ is about typesetting, and it distinguishes $\text{T}_{\text{E}}\text{X}$ from other system names. In fact, TEX (pronounced *tecks*) is

the admirable *Text EXecutive* processor developed by Honeywell Information Systems. Since these two system names are Bemers, Robert, see TEX, ASCII pronounced quite differently, they should also be spelled differently. The correct way to refer to T_EX in a computer file, or when using some other medium that doesn't allow lowering of the 'E', is to type '—TeX—'. Then there will be no confusion with similar names, and people will be primed to pronounce everything properly.

此名有诗意

1.1 xxx

1.1.1 xxx

1.1.1.1 xxxx

1.2 xxx

1.2.1 xxx

1.2.1.1 xxxx

英语单词“technology”来源于以字母 $\tau\epsilon\chi$...开头的希腊词根；并且这个希腊单词除了 technology 的意思外也有 art 的意思。因此，名称 TEX 是 $\tau\epsilon\chi$ 的大写格式。

在发音时， $\text{T}_{\text{E}}\text{X}$ 的 χ 的发音与希腊的 chi 一样，而不是“x”，所以 $\text{T}_{\text{E}}\text{X}$ 与 blecchhh 押韵。“ch”听起来象苏格兰单词中的 loch 或者德语单词中的 ach；它在西班牙语中是“j”，在俄语中是“kh”。当你对着计算机正确读出时，终端屏幕上可能有点雾。

这个发音练习是提醒你， $\text{T}_{\text{E}}\text{X}$ 主要处理的是高质量的专业书稿：它的重点在艺术和专业方面，就象希腊单词的含义一样。如果你仅仅想得到一个过得去——可读下去但不那么漂亮——的文书，那么简单的系统一般就够用了。使用 $\text{T}_{\text{E}}\text{X}$ 的目的是得到最好的质量；这就要在细节上花功夫，但是你不会认为它难到哪里去，并且你会为所完成的作品感到特别骄傲。

另一方面重要的是要注意到与 $\text{T}_{\text{E}}\text{X}$ 名称有关的另一件事：“E”是错位的。这个偏移“E”的标识提醒人们， $\text{T}_{\text{E}}\text{X}$ 与排版有关，并且把 $\text{T}_{\text{E}}\text{X}$ 从其它系统的名称区别开来。实际上，TEX(读音为 tecks)是 Honeywell Information Systems 的极好的 Text EXecutive 处理器。因为这两个系统的名称读音差别很大，所以它们的拼写也不同。在计算机中表明 $\text{T}_{\text{E}}\text{X}$ 文件的正确方法，或者当所用的方式无法降低“E”时，就要写作“TeX”。这样，就与类似的名称不会产生混淆，并且为人们可以正确发音提供了条件。