

# 基于电话线的图文声信号传输技术和系统的研制

## 摘要

本系统主要针对普通的固定电话交流形方式单一的不足，在原来只传输语音的基础上再加入手写图文信号的输入与传输，是对多媒体通讯设备的开发。这些新加的功能，可以有效地解除由于方言间不通造成的表述上的误会。还可以满足在电话通讯过程中对图形、路线、设计的探讨。更重要的是，可以极大地降低对使用者的限制。在某些特殊情况下，还具有保密功能；而听力和语言能力有障碍的聋哑人也能自如使用。

本文围绕在普通电话机添加手写显示部分，获取图文信号，并传输语音信号的基础上两路信号同时传输的软硬件实现，开展了以下工作：

1 设计了一个基于电话线传输图文声信号的功能结构方案，该方案以数字信号处理器 TMS320C6713 为核心，具有快速处理数据，且扩展功能强大的优势。

2 应用 DSP 芯片进行对信号的处理、控制，完成 DSP 与手写板、液晶显示模块的连接，实现手写输入信号在液晶显示屏上的显示，验证了软硬件设计的正确性。

3 采用电话线作为模拟语音信号和数据传输的共同信道，选择二进制频移键控调制解调方式将数据信号搬移到较高频段，通过频分多路复用的方法实现两路信号同时传输，并以应用程序实现。

4 针对公共电话网的现实情况，提出了基于 DSP 的系统在入网后可能遇到的问题，并给出了相应的解决方法。编写的程序和设计的电路，对其他类似的应用都可以起到一定的启迪和参考作用。

**关键字：** 多媒体固定电话 手写输入 DSP 频分复用 公共电话网

# Technology and System Development of Transmitting Voice and Text

## Signal Based on the Telephone Line

### Abstract

To enhance the function of the ordinary telephone, by which the user can communicate with others in voice only, this system adds handwriting-test signal inputting and transmitting, besides the simultaneous voice. It is a type of multimedia communication equipment. When the two talkers being from different areas speak in different accents, there always must be a condition that the two cannot know each other. While, using these functions, the users will avoid the embarrassing situation and misunderstanding. Additionally, if they want to discuss the figure, route, design on line, it also can meet their demands. What is more, it has canceled the restriction on user effectively. In some special cases, it can be used to maintain secrecy. And the deaf-mutes, with the obstacle of the hearing and pronouncing, can use it freely.

As requested, that is adding in the ordinary telephone hand-writing partly and revealing some, and transmitting two signals of pronunciation and picture and text, in order to realize software and hardware at the same time, the following works are finished:

1 Based on the telephone line facsimile transmission function of the structure of acoustic signals program, which TMS320C6713 digital signal processor as the core, with fast data processing, and the expansion of powerful advantages.

2 Application of DSP chips for signal processing, control, complete DSP with the tablet, liquid crystal display module connected to realize the handwriting input signal at the display on LCD screen

3 Using telephone lines as the common channel, for analog voice signal and data transmission. And use the binary frequency shift keying modulation removal demodulation of data signals to higher frequencies . Simultaneously achieve two-way signal transmission by frequency division multiplexing method.

4 Through the public telephone network related knowledge study, DSP-based system in the network after the possible problems encountered and to give the corresponding solution. Preparation procedures and the design of circuits, to other similar applications can play a role of inspiration and reference.

**Key words:** Multimedia fixed-line, DSP, handwriting input , OFDM , public telephone network

## 原创性声明

本人郑重声明：所呈交的学位论文，是本人在指导教师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名: 厚琳 日期: 2009.5.24

## 关于学位论文使用权的说明

本人完全了解中北大学有关保管、使用学位论文的规定，其中包括：  
①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签 名: 厚琳 日期: 2009.5.29

导师签名: 孙海英 日期: \_\_\_\_\_

## 1 绪 论

### 1.1 本课题的目的和意义

从“摇把子”的磁石电话机<sup>[1]</sup>到如今的程控电话，固定电话通信已拥有庞大的用户群，产生了巨大的社会和经济效益。截至 2008 年 3 月底，我国固定电话用户已达 3.62 亿户，固定电话普及率达 27.8 部每百人。然而，普通电话机单一的语音传输形式使其在人们的日常交流和使用中，逐渐表现出了不足。

本课题的目的在于开发一种新型的多媒体电话机。这种电话机不仅能满足使用者的日常要求，传输语音信号。并且，在原有座机、听筒的结构基础上，加入手写输入的设备和显示设备，达到在普通电话机上扩展手写输入并显示功能的要求。

使用者通过这种电话机可以在语音交流的同时，使用扩展的手写输入设备辅助语言同受话者探讨问题。使用手写设备作为输入工具可以满足对多种信息传递的要求。除了可以实现发送文字类的信息外，手写输入信息可以根据使用者的构想更有效地录入各种线条、图形，从而大大提升实用性，更广泛地适用于各类人群。

首先，由于不同地域、国家语言间的不互通经常容易造成交流上的误会。但是无论身在何处，图形和线条都是对事物说明的最佳媒介。其次，在电话交谈过程中常常会遇到关于图形、路线及设计的探讨，要想单纯地用语言表示清楚可谓难上加难。最后，就是，本系统消除了对使用者的各种限制。不管是在要求保密除通话双方外不许有第三方了解通话内容的某些特殊情况下，还是听力和语言能力有障碍的聋哑人，使用这种多媒体电话机都能自由地与外界沟通。

### 1.2 固定电话多媒体化在国内外的发展现状

据统计，在人类的通信中，信息的有效性，有 55%~60% 是通过视觉实现的；33%~38% 是通过语音实现的；有 7% 是通过内容实现的<sup>[1]</sup>。传统的通信工具，包括电话、传真等，由于功能单一，严重地制约了信息有效性的充分发挥。而固定电话多媒体通信

则是集语音、文字、图像为一体，具有交互能力和多种功能的通信方式，能够消除现有通信工具的局限性，最大限度地实现信息的有效性，将会实现真正意义上的面对面沟通和电话机上网，其突出特点是传统的通信工具和其他多媒体通信方式所无法替代的。因此，无论发达国家和地区，还是发展中国家和地区，或是贫困落后的国家和地区，固话用户都有改变固话单一功能，向多媒体通信转变的愿望和需求。

### 1.2.1 多媒体固定电话的发展

作为与 Internet、移动通信网并称三大网络的固定通话网，正在向多媒体通信的方向发展。作为通信终端设备的固定电话机，也不断融入更多样化的功能。

目前，在许多国家，用户已可在家中用固定电话机收发文字信息，下载铃声，从手机上的通信录中接收数据，发送彩色图片信息；欧洲一些国家的固定电话运营商开始大规模更新网络，在 2004 年推出与最新手机功能同步的新款家用多媒体固定电话机；2004 年 4 月，英国电信推出彩显家用固定电话机，用户只要插入有联系信息的手机智能卡，就可把这些信息输入到家用固定电话机上使用；法国电信公司正在巴黎、里昂和图卢兹 3 城市进行彩屏手机与固定电话机间的图像通信试验，计划推出能与最新款式 3G 手机实现完全交互的可视固定电话机。英国电信认为，从用户登记使用新业务的情况看，发展前景大有希望，现在使用多媒体通信电话机的用户通常大约每月发出 12 条文字信息，这一数字还在上升。正如意大利电信所提出的“凡是手机能提供的服务，固定电话都能做到，谁说固定电话乏味？它将与手机一样受人欢迎。”<sup>[1]</sup>

我国非常重视下一代固定电话的发展，对多媒体智能电话进行试验。上海首先推出了一种名叫“家家 E 短消息”电话，这种电话比普通电话多了一种收发短消息的功能，人们通过短消息互通业务，就可以用固定电话传递信息；从 2002 年开始，我国陆续在广州北京、青岛和苏州等城市安装多媒体信息公用电话，它是一种集话音通话、网络信息服务与广告发布于一体的新一代公用数字通信设备。2004 年初，苏州电信的新一代智能公用电话亮相街头，截至 2 月底，全市增加了 5000 多部，成为公用电话的热点业务。

如北京平治东方公司研发的一款多媒体通讯终端产品-多媒体电话。它集智能通信、计算机网络、媒体咨询于一体，融合多种技术和应用，它是一款固网增值应用产品，但

数据的传输主要还是依靠互联网实现。威艾德通讯技术(深圳)有限公司开发的多媒体电话具有曲目编排播放功能，但不具有双向多媒体交互功能等等。

因此，当务之急是研发先进的固定电话终端设备，使其与现代网络技术的发展同步，既要满足传统业务的需求，又要适用于新业务的发展和不同层次用户的需要，以解决固话发展的瓶颈问题。

### 1.2.2 基于公共电话网开展的多媒体通信业务

公共电话网在我国乃至全球是覆盖面最广且最为经济的公用通信网，因而多媒体技术在电话网上的应用仍是人们关注的领域之一。在不断更新、开发更实用更便捷的电话终端的同时，还为在电话网上开展多媒体通信业务而研制的各种新技术。在当代社会对通信要求日益提高的形势下，这些新技术也逐步融入到我们的日常生活中。

#### 1. HDSL (High bit-rate Digital Subscriber Line, 高比特率数字用户环路) 技术

HDSL 是美国 Bellcore 研究室于 1988 年底提出的，并于 1991 年初研制出了第一代高比特率数字用户线(HDSL)样机。随着集成芯片价格的下降及 HDSL 技术进一步的成熟，HDSL 已于 90 年代初商品化，并大量进入市场。HDSL 能够在现有的铜线电缆上以高达 2Mbit/s 的速率传输，为我们提供了一种颇有竞争力的用户接入途径，并以低价格、高质量的方法满足了用户对多媒体通信业务的需求。

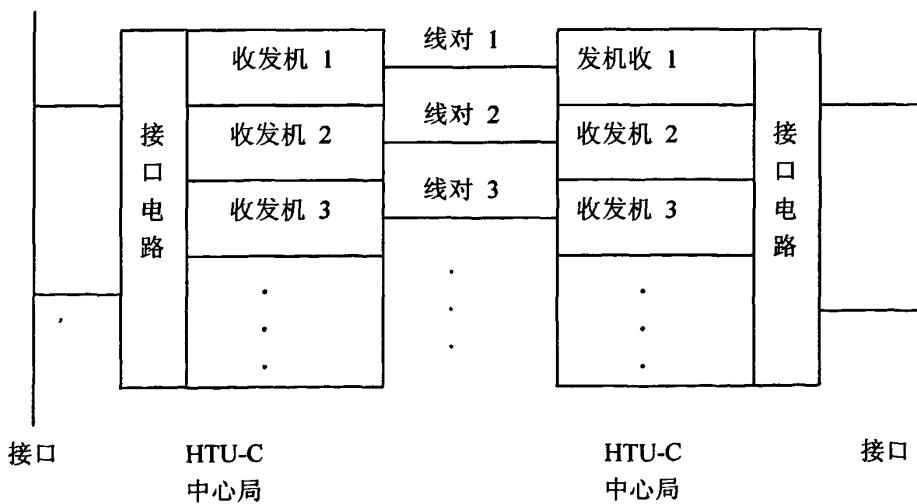


图 1.1 HDSL 系统配置示意图（两线对例）

图 1.1 是 HDSL 系统的主要组成部分示意图。它主要包括两台设备，一台装于中心局，另一台装在用户端。如果用于专用网时，则在线路两端各装一台设备即可，两台设备之间由对绞铜线线对相连。设备主要包括接口电路和收发机（收发机=发送器+接收器+混合网络+回声抵消器）。使用的线对数则根据传输距离需要可选择两线或三线对。传输时，信号通过混合网络分成两部分或三部分，再通过收发机发送至线路上，接收端再将分开的信号复接成一路满足接口要求的信号。

HDSL 系统的收发机中采用了高速自适应数字滤波器、信号处理器，以及回声抵消器等 HDSL 技术，从而使高比特率、低噪声和双工传输得以实现。

在实际应用中，通常 HDSL 利用两条标准电话线路实现 1.544Mbit/s T1 或 2.048Mbit/s E1 的数字业务传输，传输距离可达 3~5km。三线对方式可以延长无中继器传输距离，与两线方式相比，一般能增加 10% 左右的传输距离。当然通过降低传输速率的方法也可延长传输距离。

目前，HDSL 技术主要用于数字中继、高速数据业务、会议电视等视频业务、用户线扩容、光纤到路边的最后一公里。应用 HDSL 技术，可在一条电话线路上实现上百路数字电话(每路 9.6kbit/s)。HDSL 技术也可以用于交换式兆级数据业务(SMDS)、帧中继(FR)系统以及远程教学、家庭娱乐等多媒体业务。

## 2. DSVD (Digital Simultaneous Voice and Data, 数字同步话音和数据) 技术

DSVD 是一种新的调制解调技术，它能够在一条单一的电话线路上实现话音和数据的同步传输。1994 年 9 月制定了 DSVD 协议规范，调制解调器制造商把这些协议添加到了 V.34 调制解调器中，并于 1995 年 5 月首次推出了支持 DSVD 技术的 Modem。

DSVD Modem 可把模拟话音转换成数字信号，并可对它进行压缩，压缩后的信号量仅仅是模拟信号量的 1/8。DSVD Modem 把数字话音信号流与来自通信应用程序的数据流混合在一起。这种组合信号在公用电话网上进行传输。在链路的接收端，DSVD Modem 又把数字话音与数据进行分离，然后对分离出来的数字话音进行解压缩，使它重新转换成模拟语音信号。DSVD 系统的网络结构图如图 1.2 所示。

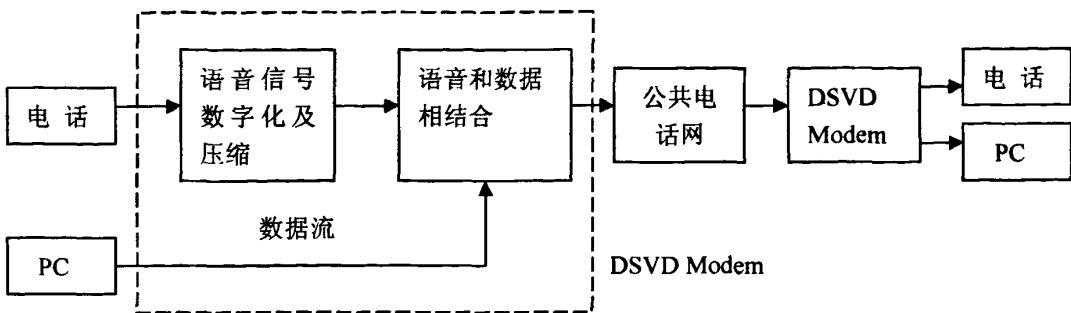


图 1.2 DSVD 系统的网络结构图

DSVD 技术最普遍的用途是给用户提供更好的技术支持和帮助用户进行协同工作。如遇到困难的用户可以请求技术人员拨入他的 PC。然后，他们就可以在电话中讨论遇到的问题。与此同时，为用户提供帮助的技术人员还能传输系统文件，对系统文件进行检查、编辑，并把修改后的版本传递给远程用户的计算机。

DSVD 是一种传输技术，且这种技术与具体的应用程序无关，所以技术人员可使用过去运行的何程序，而不必作任何修改。这意味着技术人员可以运行现行的任何远程控制或文件传输实用软件。DSVD 对于协同工作也同样有用。另外，DSVD 技术还可用于可视电话、个人会议、家庭购物、多用户电脑游戏等多媒体通信业务。

### 3. ADSL (Asymmetrical Digital Subscriber Line, 非对称数字环路)

ADSL 是于 1995 年定稿的 ANSI 标准。为了与 ETSI 标准等效，在 ANSI 标准中加了一个附件，如果形成通用的 ADSL 国际标准。到了 1996 年下半年，美国等有关厂家已将 6Mbit/s 下行和 640kbit/s 上行双向能力的 ADSL 技术产品推向了市场。

ADSL 是在具有混合线规的现有用户双绞线上同时传送 MTS(消息电信业务)单工和双工数字信道的传输技术，也就是说该技术可以从端局或远程终端为用户提供 6.144Mbit/s 和 640kbit/s 双向信道，同时保留着普通的老电话业务。传输距离达 3.7~6km。其中下行的 6.144Mbit/s 信道可以分成 3 个 2.048Mbit/s，以便同时传输 3 个电影信道，或分成 2 个 3Mbit/s 信道供数字广播电视用。上行的 640kbit/s 双工信道则可分成 384kbit/s 供电视会议用，160kbit/s 基本速率 ISDN 和剩余空间供每个信道的信令用。

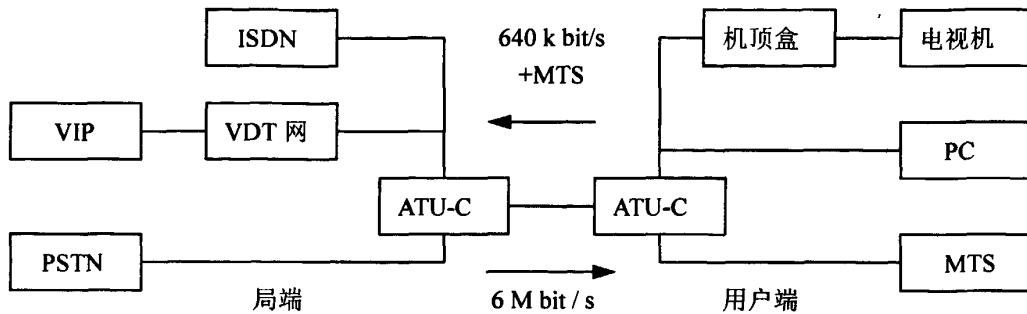


图 1.3 ADSL 系统的网络结构图

ADSL 系统的网络结构如图 1.3 所示。在用户一侧,把进入用户的信号分接成两部分:一部分是 MTS, 来自 PSDN; 另一部分为高比特率信号和低比特率控制信道, 来自多媒体网络。MTS 可以接入一普通电话, 而高比特率信号和低比特率控制信道则可以传送某一业务模块, 至于哪种模块,要取决于具体的应用。比如, 这些业务模块可以是一个 MPEG 译码器, 一个多媒体工作站, 或者是一台 PC 等。在中心局, ADSL 系统将执行分接上行的 MTS 信号和控制信道, 同时还要复接下行的 MTS 和下行的高比特率信号, 通过这种网络结构可以把点播图像、交互式多媒体通信、远程教学和其他视频业务发送到千家万户。目前, 英国电信公司和美国贝尔电话公司均已利用该技术在 PSTN 上成功实现了 VOD 业务, 另外这一技术也同样适用于数字电视广播。

ADSL 技术的主要应用领域有娱乐, 包括电影点播、音乐点播、交互式 TV; 教育/社会事业, 包括远程教学、线上书刊和手册、医疗健康会诊; 远程交换, 包括在家工作、电视会议; 小型商业, 包括电视会议、信用卡图像/签名核实; 交互式游戏等。

多媒体通信技术已广泛应用于社会的各个行业、各个领域,影响着人类的生活方式和生活质量。

(1) 视频会议系统是通过视、音频压缩和多媒体通信技术实现的, 支持人们远距离进行实时信息交流与共享。开展协同工作的应用系统, 视频会议通过传送协作成员的视、音频信息, 极大地方便了协作成员之间真实、直观的交流; 同时, 视频会议还利用多媒体支持协同工作中各种信息的处理, 从而可营造一个多人共享的工作空间。视频会议系统经历了几个发展阶段, 网络介质从 PSTN、ISDN、ATM、LAN 发展到因特网。ITU-T 发布了一系列的视频会议标准, 主要有: H.320、H.323、H.324、H.310

(2) 视频点播系统 (VOD, Video On Demand) 是计算机技术、网络通信技术、多媒体技术、电视技术和数据压缩技术等多学科、多领域融合交叉的产物，是可对视频节目内容进行自由选择的交互式电视点播系统。VOD 技术在社会各领域（如服务、教育、金融等）的应用十分广泛，显示出了强大的生命力。

(3) 现代远程教育是随着现代信息技术发展而产生的一种新型的开放式教育网络，它借助先进的通讯技术传递信息，可以实现不同地点实时地、交互地或者有选择地进行学习。利用 WEB 强有力的交互性，创建一种交互式的学习环境。

(4) 远程医疗是利用远程通信技术、计算机多媒体技术及信息技术来传输医学信息以进行诊断、治疗和教育的一门应用学科，这里所描述的医学信息包括医学图像、实时的音频和视频、患者的病历、医学设备输出的数据等。

### 1.3 课题的来源

本课题来源于山西省太原市科技兴市项目《带手写板的聋哑人电话机》。该项目是一款专为聋哑人设计，通过基于电话线的图文信号传输，用以实现聋哑人之间、聋哑人与正常人之间的日常交流的通讯工具。也可用于语言不通或特殊情况下的，正常人间的交流。

随着社会的不断发展，人们对电话的使用越来越要求多媒体化的实现。本课题的设立正是为了顺应消费者多样的使用需求。基于电话线的图文声信号传输系统，是为了方便电话机的使用者在进行语音交流的同时进行图像文字的交流而研制的一款多媒体的通信设备。增加的图文传输功能可以允许使用者在远距离对话交流的同时加入文字和图画作为补充，在描述物体、指引道路、讨论设计等情况下，使说明更加生动具体，便于理解交流。此外，该功能还可以满足使用者在不说话的条件下完成信息的交互，从而达到保密的目的。这种电话最大的特点是，在通话的同时双方可借助液晶显示屏传递文字和图像，具有同步交流的功能。因此，它也可以被称为“无纸化传真机”。<sup>[1]</sup>

本课题研制开发的多媒体电话机，是一种可以记录图象文字的通信工具。它能把用语言难以表达的文字、图形等编成电信号传送出去，到对方后在复原成与发送文件一样的信息。这样一来就解决了电话口说无凭，只能由受话方单方面猜测想象的尴尬，。

#### 1.4 本文的主要工作内容安排

本项目属于应用型电器的研制，主要研究固定电话多媒体化的开发。为拓展现有电话线的多媒体应用奠定基础，该项目的研究成功可促进我国固定电话向多功能多媒体方向的进一步发展，为电话线网络资源高效合理地利用提供科学依据。

本文的主要工作内容如下：

- 1 设计一个基于电话线传输图文声信号系统的功能结构方案；
- 2 应用 DSP 芯片对信号进行处理、控制，实现对获取到的手写数据信号的处理和显示；
- 3 采用电话线作为模拟语音信号和数据传输的共同信道，并要实现两路信号同时传输且不相互影响；
- 4 提出了基于 DSP 的系统在接入公共电话网后可能遇到的问题，并给出相应的解决方法。

## 2 系统的总体设计方案

### 2.1 系统各功能模块的划分

本多媒体电话机系统是在普通电话机的基础上来设计的。在外部形态上来说大致与普通电话机相同，同时也具有普通电话机的功能。最主要的特点加入了与电话机连接的手写板和显示屏。人们在手写板上写字，依靠手写板来传递信息，通过电话线的传输，在受信人的显示屏上显示，以达到交流的目的。

系统的工作的主要过程如下：

发话人通过手写板，输入要发送的信息，经由 DSP 显示到液晶显示模块；发话人确认信息可以传送，DSP 对图文的数据信号进行调制，转化为适合于电话线传输的信号；再送入公用电话网。

当有呼叫信号传送至话机，由振铃系统向受话人发出提示信号。受话人接收信号，经过解调，将信息送入 DSP 处理，把信息传送到液晶显示屏幕上显示。

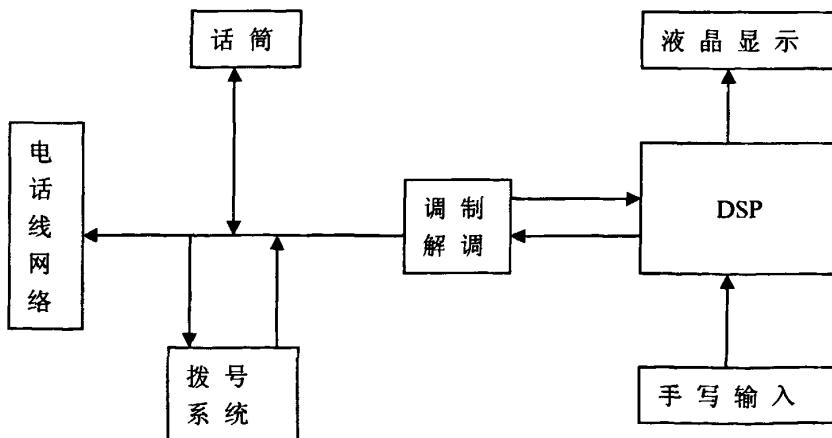


图 2.1 系统总体结构图

### 2.2 系统的结构

## 2.2.1 系统的发射部分

在原有电话的基础增加新的功能，把单一的由语音信号传输增加为可选择的由手写板采集的图文信号传输或两种信号同时传输。并在保持原文字、图形被正确识别的情况下，以电话线为传输信道，在其频率范围内利用语音信号的高频区域进行传输。

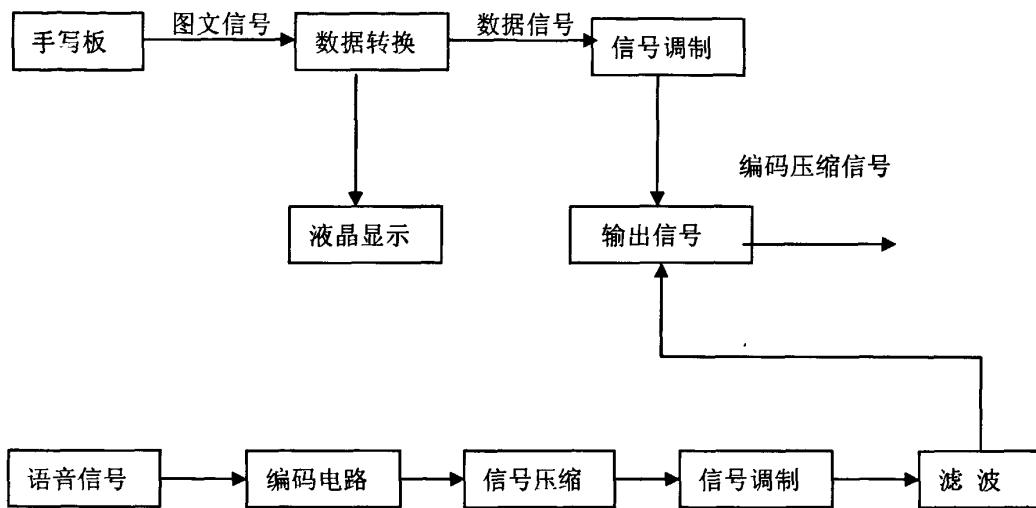


图 2.2 发射部分原理框图

### 图文信号传输部分：

(1) 手写板 把手写板作为信息输入工具，使用者可以自由输入图形和文字，并且通过扫描拾取获得有用的点阵图文信号。

(2) 数据转换 通过手写板输入的文字或图形是有一系列的坐标信息表述的，但是与我们常遇到的二维坐标不同，将需要数据信息并转化为一系列二维点阵坐标，再由点阵形式转换成二进制码表示的数据信号。

(3) 信号调制 对数据信号的调制实质上就是对信号的压缩编码，其作用是将丰富的宽频谱数据信号转换为能适合信道传输的窄带数据信号。

(4) 液晶显示 液晶显示主要起到人机交互的作用。让发话人对自己所写的内容，要表达的信息也有所见。

### 语音信号传输部分

话音传输部分的电路依然以现在使用的普通电话机的电路为主，只是在信号传输的前端增加了一个滤波电路。

## 2.2.2 系统的接受部分

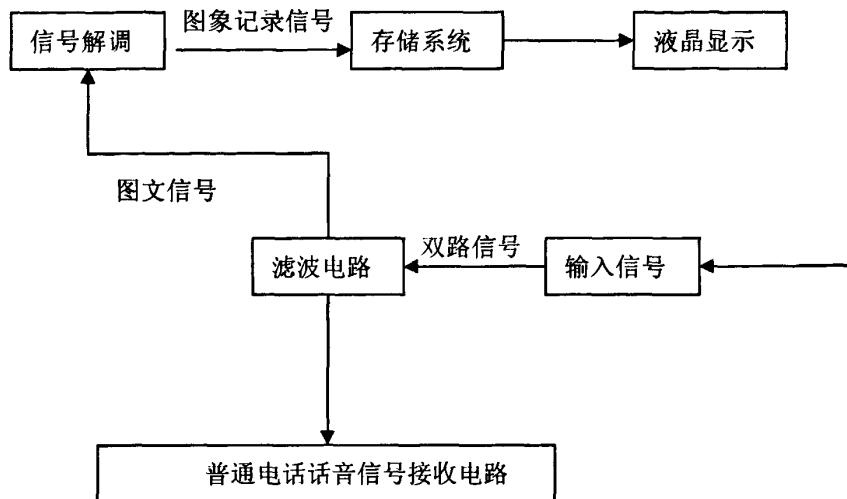


图 2.3 接收系统原理框图

在接收信号前端增加滤波电路是为了把较低频段的话音信号和较高频段的图文信号区分出来，然后分别送入话音和图文的接收部分进行处理。

图文信号接收部分：

- (1) 信号解调   解调与发送系统的调制刚好相反，它是将从电话信道接收的窄带压缩信号解码还原为数字信号。
- (2) 存储系统   存储系统的主要功能是将解码电路传送过来的信息缓存，待送入显示设备以供接收方阅读。
- (3) 液晶显示器   液晶显示器是本话机与外界传递信息的工具。用来显示传输的图像和文字信息。

话音信号接收部分

话音信号的接收与普通的电话机相同，都是将来自电话线的电信号通过检测、放大后，转化为推动扬声器发声的音频信号。

### 2.3 模拟信号与数据信号同时传输的方案

作为信号传输部分，系统设计了一种 ASVD (Simultaneous Voice and Data)，即模拟式语音数据同传的方案。

ASVD 是通过线路带宽划分来实现语音/数据同时传输的。也就是一部分线路带宽固定传输数据，余下的另一部分线路带宽固定传输模拟语音信号。其优点是利用带宽固定的一路信道中同时传送两路不同的语音、数据信号。作为两路信号在窄带的电话线中同时传输的关键技术，系统采用频分多路复用方式<sup>[1, 4]</sup>，

在电话通信中，清晰度（即发出的声音（无连贯意义的音节），在收听的地方可能听懂的正确程度的百分数）是一个很重要的评判标准<sup>[1]</sup>。保证清晰度，则要求电话传输系统必须有恰当的传输频带。我们知道，人类语音的频率范围是 80~8000Hz。话音信号的高频部分对清晰度是比较重要的，影响话音清晰度的率在 500~20000Hz 之间。其中，1000~2000Hz 是保证清晰度的必传部分<sup>[1]</sup>。而话音频谱中的高频部分的作用则主要是对音色的表现。在电话系统中，话音信号低频部分包含的能量较多，若除去，虽然清晰度可以符合要求，但话音信号的能量却不够，必会影响受话程度。所以电话频带首先应从兼顾清晰度和能量两方面着手，其次要考虑音色，即逼真度<sup>[1]</sup>。

我国对电话线传输频带规定是 300~3400Hz。由于话音信号的清晰度和能量主要分布在 300~2000Hz 的范围内，所以，我们将本系统话音信号的传输频带拟定为 300~2600Hz，图文信号在 2700~3400Hz 内传输，中间以 100Hz 的防护带<sup>[4]</sup>作为隔离，防止两路信号相互干扰。

### 2.4 系统硬件的选择

如今，消费者对通信和多媒体应用提出了更高的要求，诸如更高的性能，更多的功能、更低的功效和更小的尺寸。技术的融合和应用的多样化使数字信号处理器（DSP, Digital Signal Processor），越来越多地深入到人们的生活中，其应用领域几乎涵盖了所

有的行业。

数字信号处理器(DSP)最突出的两大特色是强大的数据处理能力和高运行速度，加上具有可编程性，实时运行速度可达每秒数以千万条复杂指令程序，并且对外接口灵活，使得它在许多电子信息产品的研制、开发与应用中，发挥着越来越重要的作用<sup>[10]</sup>。

#### 2.4.1 TMS320C6713 的简介

TMS320C6713（以下简称为 C6713）为高性能 32 位浮点 DSP，其主频最高可达 300MHz，处理速度高达 2400MIPS /1800MFLOPS。其采用具有 8 个 32 位功能单元的超长指令字（VLIW）体系结构的 CPU 内核，所以在每个周期能够执行 8 条 32 位字长的指令<sup>[8]</sup>。

之所以选择 C6713 作为系统的核心控制，不仅仅是因为它突出浮点运算的性能和高速处理能力，更重要的是它具有优异的片内外资源。

首先，C6713 具有丰富的片内外设资源，其中包括 2 个 McASP、2 个 McBSP、2 组 IIC 总线、1 组 GPIO、2 个 32 位通用定时器、1 个 16 位主机接口 HPI。此外，C6713 还有 32 位的 EMIF 总线，4 个空间，每个空间均可与 SDRAM、SBSRAM 和异步外设实现无缝接口。这些丰富的资源为完成系统硬件部分的搭建提供了十分有利的条件。

其次，C6713 的 CPU 内核为哈佛结构，取指和取数通路分离，并可同时进行。有 2 个数据通路，分割数据通路包含 4 个功能单元和 16 个 32 位寄存器堆。功能单元 D1、M1、S1、L1、（数据通路 A）和 D2、M2、S2、L2（数据通路 B）用于完成算术、逻辑运算和数据存/取等操作，而寄存器堆则用于暂存操作数<sup>[8]</sup>。

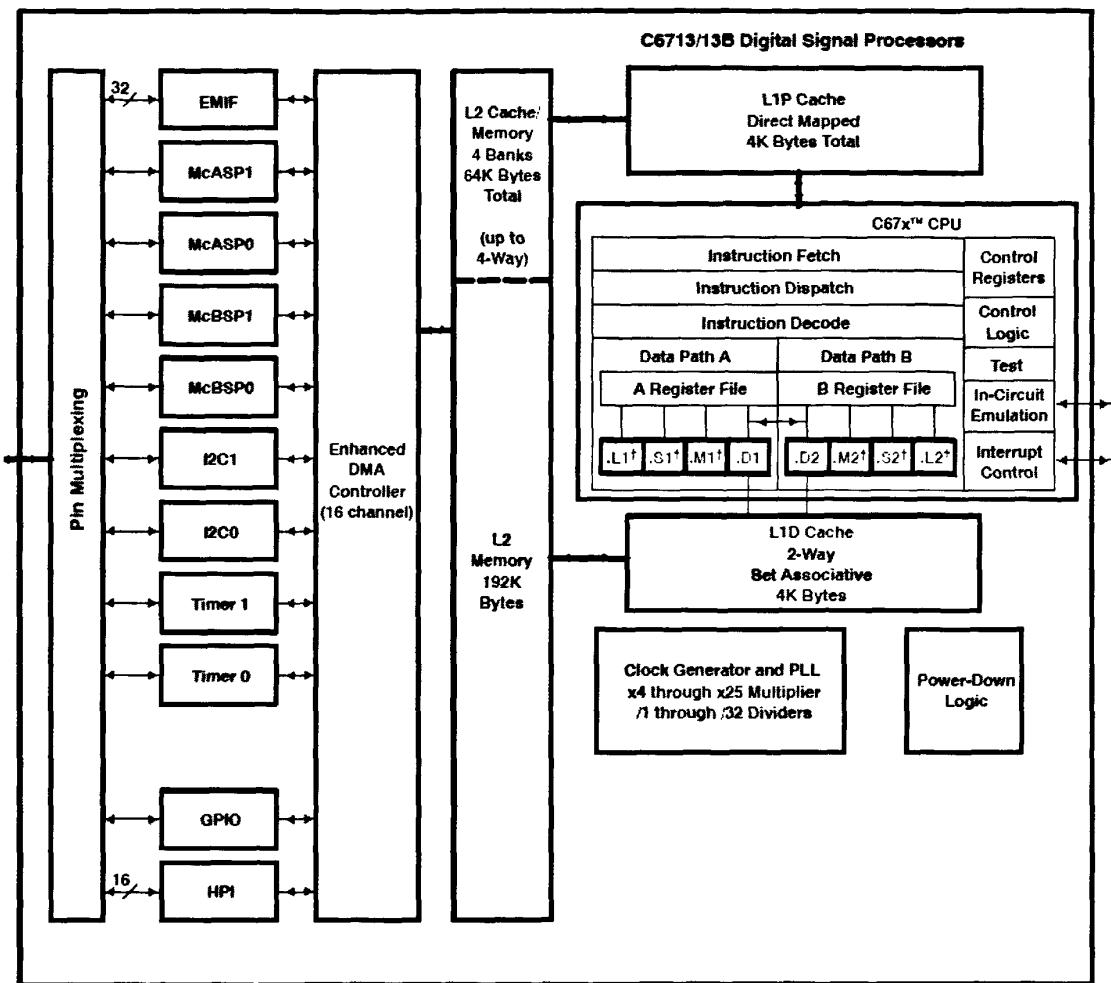


图 2.1 TMS320C6713 的内部结构

#### 2. 4. 1. 1 CCS 集成开发环境及应用软件开发流程

TI 公司的 DSP 集成开发软件环境 CCS，是一个基于 Windows 操作系统上的 DSP 开发平台，可以加速和提高程序员创建和测试实时嵌入式信号处理系统的开发过程，从而缩短将新产品推向市场所需要的时间。

CCS 集成开发环境具有功能：集成可视化编辑界面、集成代码生成工具、基本调试工具、断点工具、探针工具、分析工具、数据的图形显示工具、提供 DSP 工具，并支持 RTDX 实时数据交换技术等、开放式的技术，支持其它第三方的 ActiveX 插件等<sup>[11, 12]</sup>。

TI 公司的 DSP 芯片既可以使用汇编语言，也可以使用 C 语言进行软件开发。考虑到软件的可移植性问题，应尽量采用 C 语言进行开发。TI 公司提供了一个集成开发环境 CCS，使用 CCS 进行 DSP 应用软件开发的典型流程，如图 3.1 所示。

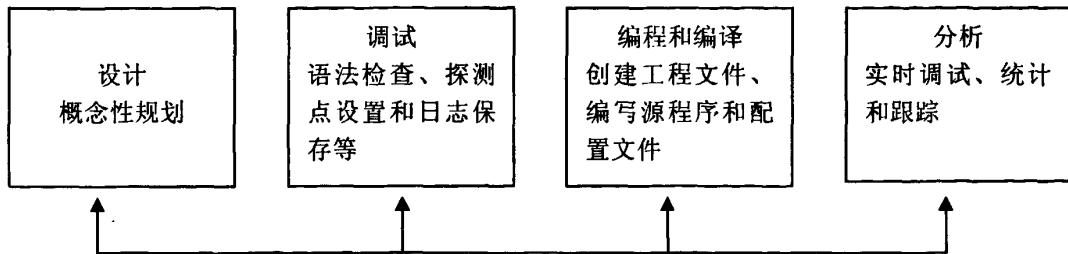


图 2.2 软件开发流程图

#### 2.4.1.2 TMS320C6713 在通信领域的应用

九十年代中 DSP 在数字 GSM 手机应用和无线基站应用中都获得了巨大的成功。与此同时，DSP 开始全面拓展到新兴应用，并在宽带通信、数字控制、数字音频、数字视频等众多市场上捷报频传。如今，通信仍然是 DSP 最大的应用市场。

为了向需要高精度的应用提供尽可能高级别的分辨率，对能够提供动态范围的浮点 DSP 需求正在逐步增加。自 1997 年推出 TMS320C67x DSP 系列产品以来，TI 已经以其极具吸引力的价位提供了一系列代码兼容型高性能浮点器件。由于它为设计人员提供了易用性，使客户能够充分使用浮点器件。对致力于增强新一代视频应用能力的设计人员而言，他们能够充分利用基于 0.13 微米铜处理技术开发的 C6713 DSP 器件所提供的高达 225MHz 的卓越性能。

最初 C6713 的应用在于专业数据通信和语音处理，各种专用调制解调器、声码器、数据加密机，初步获得市场。其后 C6713 应用扩展到广泛的民用产品，诸如通用调制解调器、数字答录机、无线通信终端。如今，鉴于其实时操作着色、传输、压缩和图像增强的能力以及处理专用算法和行业标准算法的能力（如 Dolby Digital、MPEG-2、MP3 和 DTS）C6713 被应用到各类有线和无线的可视通信系统中，通过一条单独接入家庭的宽

带线路传输大量的个性化数据、视频、语音。当然，在数字语音通信等诸多方面 C6713 的应用还有很多。

## 2. 4. 2 液晶显示模块

液晶显示模块(LCM, Liquid Crystal Display Module)，它是将液晶显示器件、驱动及控制电路以及温度补偿、驱动电源、背光等辅助电路组合在一起的一种相对独立的显示器件或设备。英文名称叫“LCD Module”简称“LCM”，中文一般称为“液晶显示模块”。  
[27]

根据显示信息种类和用途的差别，液晶显示模块可分为笔段型、字符型和图形、图像显示模块。

(1) 笔段型液晶显示模块，是将长条状显示像素组合起来以显示信息的液晶显示模块，主要用于显示数字、西文字符、某些专用符号或固定图形，也可用于显示一个汉字或一个汉字组，被广泛应用于各种数字仪器、计数器、计算器等应用场合；

(2) 字符型液晶显示模块，是一类专门用于显示字母、数字、符号等信息的点阵型液晶显示模块，由于字符型液晶显示模块都属于小规模液晶显示模块，其采用的液晶显示器件的电极图形大都是  $5 \times 8$  或  $5 \times 11$  的点阵块，采用的驱动控制器件相对统一，因此，不管具体显示屏尺寸如何，操作字符型液晶显示模块的控制命令和模块接口信号都具有较大的兼容性；

(3) 图形、图像液晶显示模块，根据显示信息量的区别，它可分为大规模、中规模、小规模点阵图形式液晶显示模块，相应所采用的点阵数、显示驱动控制器的方式、驱动控制器件等都有所不同，因此对点阵图形式液晶显示模块操作时的控制命令和模块接口信号都有较大差别。

由于作为人机交互的媒介，系统应用到的显示屏主要对手写的图形文字进行显示。相比之下，能够对图形、文字等自由显示的图形、图像液晶显示模块更能满足要求。

本系统选用的是 LCM240128ZK 是北京青云创新科技发展有限公司推出的  $240 \times 128$  点阵带中文字库中规模图形液晶显示模块，内含 7602 个简体中文字形。液晶显示模块留有一个接口与外部通信，LCM240128ZK 可以通过这个接口接收 DSP 写入的显示命

令和数据，并按照命令和数据的要求进行显示，外部电路也是通过这个接口读出显示模块的工作状态和显示数据。LCM240128ZK 提供 8/4 位数据并行方式传输。

LCM240128ZK 液晶显示模块主要由 1 片 LCD 控制器 Ra8820、2 片行驱动器、3 片列驱动器、1 片显示存储器和 1 块 240x128 点阵液晶显示屏组成。行驱动器是与 Ra8820 相配套的 68 路行驱动器，它将来自 Ra8820 的串行信号转换成并行信号，以驱动液晶显示屏上相应的行。此模块有 128 行点阵，因此用了 2 片行驱动器。列驱动器是与 Ra8820 配套的 80 路列驱动器，它将来自 Ra8820 的串行列信号转换成并行信号，以此驱动液晶显示屏上相应的列。

#### 2.4.3 手写输入设备

近年来，手写输入为机交互提供了一种新的手段。随着对手写板和识别软件的开发普及，实现音频、视频加手写的多选择交互功能将更加完善。目前，手写输入通常用于和 PC 机连接，或是通信设备。手写绘图输入设备对计算机来说是一种输入设备，最常见的是手写板，其作用和键盘类似。当然，基本上只局限于输入文字或者绘画，也带有一些鼠标的功能。

手写板一般是使用一只专门的笔，或者手指在特定的区域内书写文字。手写板通过某种方法将笔或者手指走过的轨迹记录下来，然后识别为文字。对于不喜欢使用键盘或者不习惯使用中文输入法的人来说是非常有用的，因为它不需要学习输入法。手写板还可以用于精确制图，例如可用于电路设计、CAD 设计、图形设计、自由绘画以及文本和数据的输入等。

在市面上手写板的品牌很多，但根据其工作原理手写板可以分为：电阻压力式、电容触控式和电磁压感式。

电阻压力式手写板是由一层可变形的电阻薄膜和一层固定的电阻薄膜构成，中间由空气相隔离。当用笔接触手写板时，上层电阻受压变形并与下层电阻接触，下层电阻薄膜就能感应出笔的位置。

电容触控式手写板的工作原理是通过人体的电容来感知手写笔的位置，即当使用者用笔接触到触控板的瞬间，就在板的表面产生了一个电容。当基板察觉到某一点发生了

电容的变化，便能计算出笔所在的位置，并向电脑报告。电脑根据笔的位置，便能将光标做相应的移动，或做其它的动作。由于笔必须接触到基板才能作用，因此，当您提起笔后，不论您如何移动，荧屏上的光标都不会移动。用起来比较不自然，反而有点像鼠标。

电磁压感式手写板是通过在手写板下方的布线电路通电后，在一定空间范围内形成电磁场，来感应手写笔的笔尖的位置进行工作。使用者可以用它进行流畅的书写，手感也很好，绘图很有用。在基板表面附着有一种传感矩阵，这种传感矩阵与一块特殊芯片一起，持续不断地跟踪着手写笔的“轨迹”，经过内部一系列的处理，从而能够每时每刻精确定位笔尖的位置（X、Y坐标），同时测量由于笔尖与板间距离（压力大小）形成的值的变化，确定Z坐标，最终完成X、Y、Z坐标值的确定<sup>[13]</sup>。

比较以上三种不同的手写板：由于制造时，电阻值与电容值很难控制到均匀与精确，所以手触板有一些先天上的限制。首先，如果做到超过约七，八厘米大小，其产品优良率就会变得很低，从而快速提高成本。其次，基板的分辨率比较低，笔的位置很难精确判断。最后是基板性能比较不稳定，使用一段时间后，也容易变得不好用。

系统选用的是，电磁压感式手写板。因为。这种手写板是在图形板方式下工作的，其X、Y坐标的精度可高达每毫米40点（即每英寸1000点），压感级达到1024级。手写笔也是电磁感应的，使用中会放出电磁波，由基板感应到后，计算出笔的位置，报告给电脑。然后，电脑再做出移动光标或其它相应的工作。此外，电磁式手写板所用的手写笔无需电源供给，特别适合于便携式产品

## 2.5 本章小结

本系统是在普通固定电话上增加液晶显示功能、图文传输功能、书写功能和语音图文转换功能而产生的新型通讯设备。主要利用DSP进行信息处理、程序控制，并实现各功能之间的转换。通过电话线实现模拟语音信号与数据的同时传输，并在保证话音质量的同时，使图文信号能够顺利地传输接收。

人们在写字板上写字，依靠写字板来传递信息，通过电话机的传输，在接受者的显示板上来显示，以达到交流的目的。语音和图文的传输是经过由DSP控制的电话系统

来输送的。所以写字板和电话机的连接及各种信号的转换和传输是设计的重点和难点。

为了在一条电话线同时传输话音和数据两路信号，使用频分复用技术，将现有电话线的带宽划分为不同的频率带使用。在使用时，话音信号占用电话线的一部分带宽传送，图文数据则是使用另外的频带传递。这样，就可以在电话线中实现系统设计的同时传输模拟语音信号和图文数据信号的方案。

除了对系统的整体设计做出方案，本章还对应用到的硬件做了选择。

对系统的核心器件——运算处理芯片，选择使用高精度、高性能的 TMS320C6713；采用电磁压感式的手写板作为手写的输入设备；点阵图形、图像液晶显示模块 LCM240128ZK 实现人机交互的显示功能。

### 3 手写输入信号的采集与处理

#### 3.1 对手写输入信号的采集

首先把手写板和计算机连接好，使其能通讯。然后从串行口中引出信号线，利用示波器对串口通信码值进行监测。

监测串形通讯口的九根信号线，通过多次计算机开关机操作发现，1号、5号、6号、8号、9号线一直保持零电平。4号线电压为6.0V，3号线电压为-6.0V，7号线电压由0跳变为6.0V后，2号线上有波形输出。断开1号、4号、6号、8号、9号线后手写板能正常工作。断开3号、7号线手写板不能工作。再改用3号线直接接-6.0V电压，7直接接+6.0V电压，手写板正常工作。

据此我们得出结论：手写板与计算机之间以单工方式通信，3号线为手写板提供负电平信号，7号线由低到高跳变时手写板发通信码启动工作。

##### 3.1.1 TMS320C6713 URAT 口与手写板 COM 口的连接

C6000系列的DSP是高性能的数字信号处理器，通常进行串口通信时，DSP与异步串口UART的通信可以通过三种方法实现。第一种方法是利用 McBSP（多通道缓冲串口）工作在通用的串口模式下；第二种方法利用 McBSP 工作在 GPIO（通用输入输出）模式；第三种方法利用总线和芯片扩展工作于异步串口模式<sup>[16]</sup>。

基于以上的实现方法，本系统采用第三种，由C6713的EMIF总线扩展UART芯片(TI公司的TL16C752BPT)，用硬件实现异步数据传输。这样可简化接口程序，让软件实现起来更简单。

TL16C752B采用8位异步平行存储器接口，采用+3.3V电源供电，可以与C6713的外部存储器接口(EMIF)直接接口。TMS320C6713与TL16C752B通用异步收发器UART连接，可以设置二个独立的通用异步串口，其接口标准可以配置为

RS232/RS422/RS485，本项目中，应用其与手写设备相连，进行串行通信。

此外，因为 DSP 的串口电压为 3.3V，手写板的串口电压经测量为 6.0V，为了实现 6.0V 和 3.3V 电压之间的数据传输，系统在外部接口和 UART 芯片 TL16C752BPT 间加入了串口电压转换芯片 MAX3160。

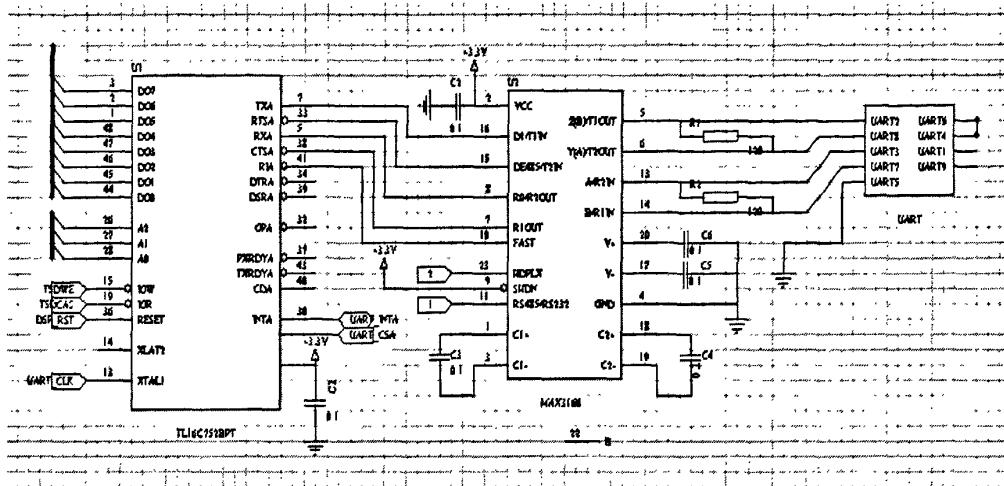


图 3.1 应用 TL16C752BPT 和 MAX3160 在 C6713 上扩展 UART 原理图

表 3.1 UART 口的定义<sup>[15]</sup>

1	2	3	4	5	6	7	8
TXD	NC	GND	CTS	RXD	RTS	Connect with 7 or 8	

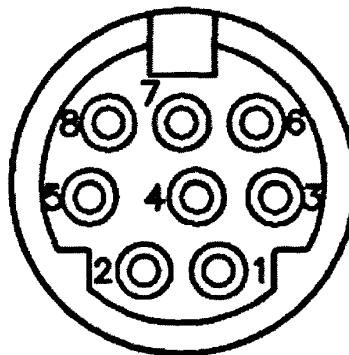


图 3.2 UART 口的管脚排列

本系统的图文手输入端选用的是北京汉王科技有限公司出品的串口手写板。使用串口手写板的优势在于实现简单，并且可以大大降低系统的经济成本和开发成本。

该手写板的串行接口是采用的 COM 口的母口，方便与 PC 机连接。其管脚的排列和定义<sup>[14]</sup>如下所示：

表 4.2 COM 口的定义（母口）

1	2	3	4
CD	TXD	RXD	DTR
Carrier Detect	Transmit Data	Receive Data	Data Terminal Ready
5	6	7	8
GND	DSR	RTS	CTS
System Ground	Data Set Ready	Request to Send	Clear to Send
			RI
			Ring Indicator

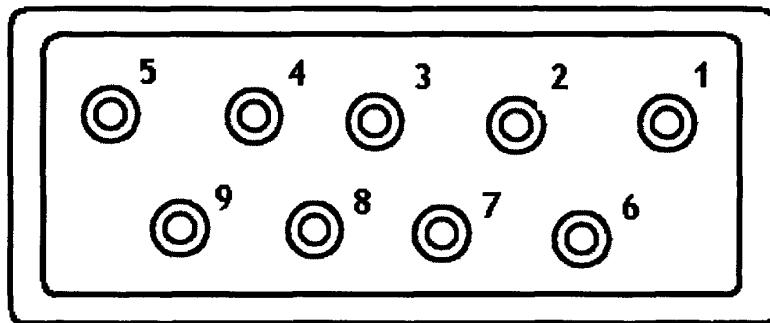


图 3.3 COM 口母口的管脚排列

在上述工作完成以后，可以简单的通过导线或电缆连接手写板的 COM 口和 DSP 的 UART。期间要特别注意两接口定义的不同，否则会导致数据传输的失败。

表 3.3 具体的接口对应关系

COM	UART	说 明
2 (TXD)	5 (RXD)	接收手写板传输的数据，送入 DSP 内处理
3 (RXD)	6 (RTS)	向手写板发送传输数据的请求
5 (GND)	3 (GND)	共地

由上面的表格可以清楚地说明二者连接后的通信过程。首先，在接口连接前先给手写板接入一个外部电压（+5.0V～+6.0V），作为工作电压；然后连接对应管脚，DSP 通过 UART 芯片由管脚 6(RTS)向手写板的数据接收管脚 3（RXD）发出负电压（-6.0V～-5.0V），请求手写板传送数据；手写板接到请求后，通过验证，允许发送请求，再由管脚 2(TXD)发送数据至 DSP，6(RTS)负责接收。

### 3.1.2 接口连接的软件编程

串行异步通信协议包括两个方面的内容：一是字符的传送格式的规定；二是数据传送速率的要求。串行异步通信的字符格式包括四个部分：起始位、数据位、校验位和停止位。起始位在一个字符中占 1 位，该位必须为 0，表示一个字符的开始。数据位至少 5 位，最多 8 位。数据排列方式是低位在前，高位在后。校验位占 1 位，根据需要可选，可不选。奇偶校验用于有限差错检测。如果选择奇校验，则组成数据位和校验位的逻辑 1 的个数必须是奇数，否则传送出错（奇校验错）；选择偶校验，则组成数据位和校验位的逻辑 1 的个数必须是偶数，否则传送出错（偶校验错）。停止位表示一个字符的结束，可以为 1 位、1 位半或 2 位<sup>[4]</sup>。

异步通信的数据传送速率是用波特率衡量的。波特率是指单位时间传送二进制数的位数，微机中常见的有 110、300、600、1200、2400、4800、9600、19200。观察示波器检测获取到的码型，发现最小码原宽度大约为 100μs，通过计算波特率为 9.6K 时最小码原宽度为 104μs，所以此通信波特率为 9.6K。此外，通过与北京汉王公司的技术人员沟通还得知，该公司出品的串口手写板所传输的数据是停止位为 1，且无奇偶校验位的。在接口连接的程序中，在设置串口接口参数时将会用到以上的数据。

要应用 DSP C6713 实现 UART 口的通信，首先要对其进行初始设置，格式如下：

```
UART_Setup( UInt32 channel, UInt16 UartBaud, UInt8 UartWordLen,  
            UInt8 UartStopBits,  
            UInt8 UartParity,  
            UInt8 UartFifoControl,  
            UInt8 UartLoopway);
```

系统参数本节开头所提到的，写入上面的设置，即可完成对 UART 的初始化（详细设置请见附录）。

除要对 UART 的参数设定外，还要对存储待处理数据的寄存器做初始设置，以及对 UART 的中断设置。由于采用 UART 芯片 TL16C752BPT 带有 64bit 的接收 FIFO，所以以 FIFO 的字长判断中断。

DSP 通过 UART 通信获取数据的流程图如图 4.3 所示(程序见附录)。

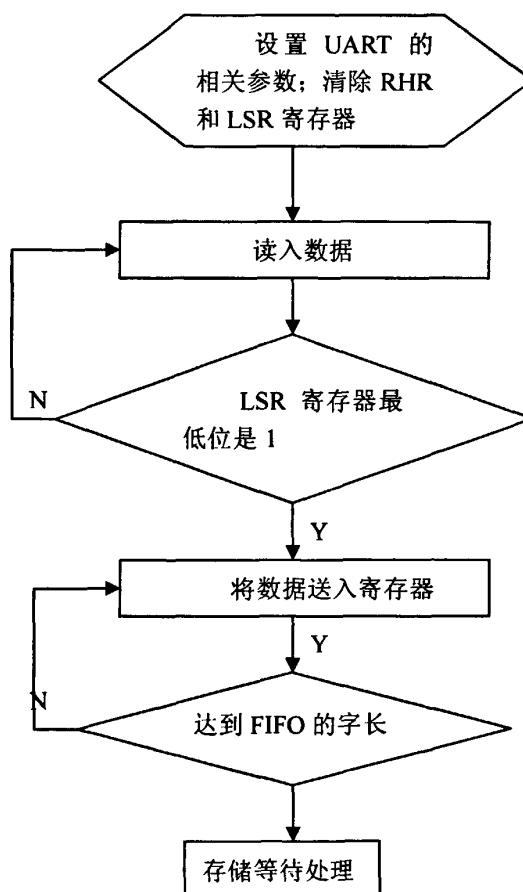


图 3.4 DSP 通过 UART 通信获取数据的流程图

### 3.2 对手写信号的分析和处理

把手写板与 DSP 连接，对手写笔的各种动作，DSP 接受并记录下来，最后分析出编码规则。

首先，将手写板的 COM 口的 2 脚与 DSP UART 口的 5 脚(即接收脚)通过一根导线相连，在不同的两种状态时，使 COM 的 2 脚分别与 UART 口的 5 脚连通，UART 口的 1 脚(发射脚)不用。要监视手写板与 UART 口的通讯数据，首先将开关打到使 COM 口 2 脚与 UART 口 5 脚连通的状态，使 UART 口与手写板通讯电缆连接好，这时监控的就是手写板往 UART 口发送的数据流，同样，将开关打到 COM 口 3 脚与 UART 口 6 脚连通的状态时，监控的就是手写板收到的 UART 口传送回来的数据。同时需要注意二者的共地连接。

通过对采集获取到大量数据分析发现，当手写笔在板上动做时，手写板由串口发送一组组码值，其中，获取到的每一组码包括 5 个字节。当手写笔由运动转为静止时最后一组码发送 10 个字节，后 5 个字节重复前面的发送；当手写笔静止时，不发送数据。通过具体的对手写板横纵边界、四个顶点及笔上按键的多次测定，发现每组数据的 5 个 字节的数据格式，如下表示：

表 3.4 手写板的数据格式

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Byte 0	1	1					Key L	Key R
Byte 1	0	X6	X5	X4	X3	X2	X1	X0
Byte 2	0	X13	X12	X11	X10	X9	X8	X7
Byte 3	0	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Byte 4	0	Y13	Y12	Y11	Y10	Y9	Y8	Y7

所有的 4 个字节中都只有 7bit 的有效位。X/Y 的低高位并不是简单的数位的拆分，而是对每一个点的坐标的粗微表述，类似于日常用到的游标卡尺的粗微调。低高位的表示方式就是把一块手写板在划分成若干个大的区域之后，再在各大区中细分得到若干小

区。其中，低位是对坐标的细微表述，可以看成是 40 point / mm 分辨率的数字表达。它的表述范围是 (0x00h ~ 0x7Fh)，即一个低位字节最多可以达到 128 点。高位字节则是该点坐标大区的定位。

由已知的结果看，有 5 个字节组成的一个数据组是一个完整的结构体，其中的各个字节都不能独立存在。所以根据这个特点，在对手写数据的处理程序中，应用结构体的方式对数据组定义：

```
struct uart
{
    unsigned int FB;
    unsigned int XL;
    unsigned int XH;
    unsigned int YL;
    unsigned int YH;
}s;
```

通常对于一组数据的处理会选择以数组的方式定义。但是，数组可以为操作同类型的数据提供了方便，却不能反映同一事物的各个方面，或者说不能直接反映同一事物的各个侧面间的有机联系。而现实世界中所有事物都是有其不同侧面的有机组合而体现的。比如：一个学生，可以用他的姓名、性别、年龄、专业、成绩等数据来描述；一个箱子可以用它的长、宽、高、材料、自重等数据来描述。这些数据的特点是，他们可能属于不同的数据类型，但是密切相关，因为他们是同一事物从不同侧面。所以，C 语言提供了一种新的数据集合类型，结构体(Struct)数据类型。定义结构体可以增强数据组的完整性，方便对这样以若干字节的方式出现的数据<sup>[31]</sup>。

获得的手写输入信号将由 DSP 进一步处理后，将被送入液晶显示模块显示，同时也将被送入信号传输模块等待调制，送入传输线。具体操作及流程将在第 5 部分和液晶模块的连接中详细介绍。

### 3.3 本章小结

系统选用的输入设备是电磁触控式的 COM 串口手写板。为实现手写板和 DSP 的连接，首先要了解手写板串口管脚传输的情况。由于汉王公司在设计产品时对 COM 口的管脚设定和通用标准有所不同，所以要与 PC 机相连通过示波器的监测，才能对管教的定义和传输电平的高低有所了解。在对手写板的个体情况弄清楚后，实现 COM 口和 UART 口的连接。最后，编写相关的程序，从 UART 口获取手写数据。

其实，在以后的工作中真正重要的是所获取到的数据，所以，对数据的分析和处理就是本阶段的重点。为了便于分析，我们用不同的方法使获取到的数据都有一定的规律可循，比如，取四顶角的点、与 X 轴或 Y 轴平行的直线。通过大量的数据分析，发现了其中的规律。并用结构体的方式把 5 个字节一组的数据组合在一起，以备以后的取用和处理。

## 4 液晶显示模块

随着信息时代的高速发展，人们需要有大量获取的信息和多种多样的显示方式出现在液晶显示屏幕上，从而实现人机交互。

本系统采用的液晶显示模块，选择点阵图形液晶显示模块。它具有显示信息丰富多样（可显示汉字、字符、曲线、图形），功耗低、体积小、重量轻、寿命长、不产生电磁辐射干扰等优点，是微处理器系统中的最理想显示器件，近年来被广泛应用于微处理控制的智能仪器仪表、工业控制领域、通信器材和家用电器中<sup>[27]</sup>。

当 DSP 与液晶显示模块此类读写周期较慢的设备连接访问时有两种方式，分别是直接访问和间接访问<sup>[27]</sup>。

（1）直接访问方式 直接访问方式是把液晶显示控制器作为存储器或 I/O 设备直接挂在 DSP 的总线上。控制器的数据线接在数字信号处理器的数据总线上，片选及控制寄存器选择信号线由 DSP 的地址总线译码产生，读和写操作由 DSP 的读写操作信号控制。

由于液晶显示控制器的微处理器接口时序 Intel 时序和 Motorola 时序两种，所以微处理器与液晶显示控制器的直接访问方式接口首先要涉及到两种时续的转换问题；另外在读/写操作液晶显示控制器时，要尤其注意 DSP 的读写建立时间与液晶显示控制器要求的数据设置和数据保持时间的匹配问题，根据需要可增加必要的硬件电路或在软件编程时予以协调。

（2）间接访问方式 间接访问方式是一种常用的访问方式，此种方式无需硬件扩展即可实现与任意时序慢速设备之间的时序匹配。间接访问方式是将液晶显示模块与 DSP 的并行 I/O 接口连接，DSP 通过对该 I/O 接口的操作，间接地实现对模块的控制。间接显示方式的特点是电路结构简单，接口电路与时序无关，控制时序完全靠软件编程实现，占用系统资源少。

因此，本系统采用间接访问的方式实现 DSP 与 LCD 的连接。在硬件电路上用一个 8 位并行接口与模块的数据线连接，作为数据总线，一个 3 位并行接口作为时序控制信

号线，模块的片选信号直接接高低电平作选通态。

#### 4.1 硬件接口电路的设计

C6713是3.3V电源器件，而LCD是5V电源供电。DSP芯片输出给5V的芯片，无需加任何缓冲电路，可以直接连接，但是DSP芯片不能输入5V的信号，它超过了DSP芯片的电源电压，将会造成损坏。而作为信号处理器C6713不单需要将数据写入控制器，也要从控制器中读数据，因此数据总线上的信号流是双向的。如果直接把两者相连，那么数据的写入时就可能会对C6713造成损害。

解决这个问题可以采用电平转换芯片74LVTH16245进行总线电平转换。它是16位的芯片，可以同时对16个引脚实现电平的转换。5V电平的数据经电平转换后，变成了3.3V电平的数据，与LCD控制器准备接收的数据相匹配。

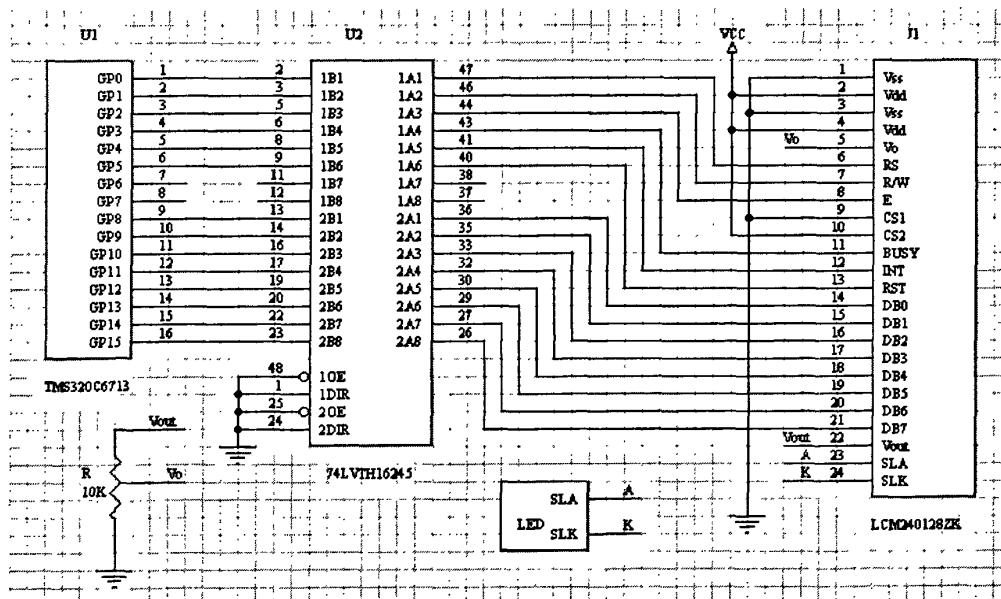


图4.1 C6713和LCM240128ZK的连接原理图

图4.1中，GP8~GP15通过电平转换与液晶模块的数据线DB0~DB7连接，GP0、GP1、GP2、GP3、GP4、GP5通过电平转换分别与液晶模块的RS、R/W、EN、BUSY、INT、RST相连。片选信号CS1和CS2分别接地和高电平，使液晶模块一直处于选通状态。V<sub>out</sub>、V<sub>o</sub>通过一个10K滑动变阻器与V<sub>ss</sub>和地相连。V<sub>out</sub>引脚是模块的负电压输出，V<sub>o</sub>为LCD屏

提供操作电压，通过调节 $V_o$ 的值可以调节LCD的显示对比度，在这里调整 $V_o$ 的电压值，使其与 $V_{dd}$  (+5V) 的差值为20V。LCM240128ZK模块上有两组 $V_{ss}$ 和 $V_{dd}$ 模拟/数字地和电源，但在使用时模拟地/电源和数字地/电源连接在一起，所以图5.1中只标注一组。SLA、SLK是LED背光源的正负极，分别接5V和地。

RS、R/W 两个控制口通过不同的组合可以实现读/写指令、读/写数据等。当 RS 被输入高电平时，会对模块的指令寄存器存取，否则，就对资料存储器进行操作。R/W 为读写控制引脚，当输入为高电平时，此时模块处在被读取状态，DSP 就不能够对其进行写动作，反之亦然。BUSY 为模块的忙碌标志引脚，当其为高电平时，DSP 不能够对模块进行任何操作，所以在存取寄存器前应先通过该引脚检查模块的忙状态。RST 为复位引脚，当其为低电平，模块被复位，模块要求它的低电平时间不少于 100ms。

## 4. 2 液晶显示编程软件的实现

### 4. 2. 1 对系统的初始化

对系统进行的初始化设置,包括对 C6713 和 LCM240128ZK 的初始化(源代码见附录)。

C671x 接口需要系统提供一个外部时钟源 (ECLKIN)。基于 ECLKIN 由内部产生 ECLKOUT 信号。与 DSP 接口的 LCD 的存储器需在 ECLKOUT (EMIF 时钟周期) 下工作。EMIF 和他所支持的存储器接口的控制通过 EMIF 中存储器映射的寄存器来维护。因此，对 DSP 的初始化主要完成对外部存储器 (EMIF) 的设置。

LCD 的初始化主要完成对 LCD 指令集的设定，包括文字模式、绘图模式、光标设置、工作及显示视窗、屏幕更新频率、中断与忙设定等<sup>[26]</sup>。

#### 4. 2. 1. 1 LCD 的初始化

LCM240128ZK 的指令主要是通过设置它内部的寄存器来实现的，它包含有 22 个寄存器，包含对电源、亮度、光标、工作视窗、字体、行距、中断等的设置。每个寄存器

都由写入的 8bit 数据，对其功能状态设定。

下面以部分寄存器为例，描述其具体功能：

表 4. 1 REG[00h] LCD Controller Register (LCR)

bit	功 能 描 述
7-6	<p>电源模式(Power Mode)</p> <p>11:正常模式(Normal Mode) 控制器的所有功能都可以使用(Available)</p> <p>10:等待模式(Standby Mode) 只有 DRAM 与 ROM 的存取功能被禁止，其他功能都可以使用，LCD 亦照常工作</p> <p>01:睡眠模式(Sleep Mode) 除了允许寄存器的读写外，其他 LCD 显示与 DDRAM、ROM 的存取将被禁止</p> <p>00:关闭模式(Off Mode) 除了唤醒(Wake-Up)电路工作处，其他功能都被禁止.当 Wake-Up 电路被触发，控制器将进入正常模式</p>
5	<p>软件重置</p> <p>所有寄存器回到初始值，但是 RAM 的内容不会被清除。</p> <p>1:重置所有寄存器 0:无作用</p>
4	<p>设定自动重置功能</p> <p>当此位元处于致能(设为 1)时，若控制器无法在 2 毫秒内得到一完整指令或资料时，则控制器会忽视这个指令或资料。</p> <p>1:致能自动重置功能 0:禁能自动重置功能</p>
3	<p>选择显示工作模式</p> <p>1:文字模式，写入的资料被视为是 GB/BIG/ASCII 等字码。 0:绘图模式，写入的资料被视为是 Bit-Map 的模式。</p>
2	设定萤幕显示为开启或关闭，此位元用来控制连接到 LCD 驱动器界面的'DISP_OFF'信号。

号

1:DISP\_OFF' 信号输出 High(荧幕显示开启)

0:DISP\_OFF' 信号输出 Low(荧幕显示关闭)

---

1 闪烁模式选择

1:整个荧幕闪烁，闪烁时间由寄存器[80h]BTR 来设定

0:正常显示

---

本系统在对LCD的初始设置中应用如下语句：

LCD\_CmdWrite(0x00); LCD\_CmdWrite(0xCD);

此语句表达的意思是：先读入写指令的寄存器地址 0x00h，然后再向此地址写入值 0xCDh。

0xCDh 转换成二进制即为 1100 1101，对应于该寄存器进行设置的结果为：

采用对控制器所有功能都可以使用的正常电源模式；保留对所有寄存器设置的值；若控制器无法在 2 毫秒内得到一完整指令或资料时，则控制器会忽视这个指令或资料；写入的资料被视为是 GB、BIG、ASCII 等字码；连接到 LCD 驱动器界面的'DISP\_OFF' 信号输出 Low；整个荧幕闪烁，而闪烁时间由寄存器[80h] BTR 来设定。

对于其他寄存器的设定都同以上的 REG[00h] LCR 的设定相仿，再次不一一列举，详细的设定及相应数值请见附录中的 LCD 初始程序。

#### 4.2.1.2 对 GPIO 寄存器的设置

C6713 的片上备有 16 个通用输入/输出外设(GPIO, the general-purpose input/output)，为外围设备提供了可配置为输入或输出的制定的通用管脚。但是，DSP 提供的 GPIO 管脚都是地址线、数据线、控制线等多路复用的，要使用这些管脚，就必须首先对其设置，将其驱动起来。设置为输出时，用户可以通过向内部寄存器写入数据来控制有输出管脚驱动的状态。设置为输入时，用户可以通过读出内部寄存器的数据来检测状态。

在使用 GPIO 管脚之前，首先要使之在 GPIO 使能寄存器内被激活才可以作为通用 I/O 口运行。用户可以通过 GPDIR 寄存器独立地把每个 GPIO 管脚设置为输入或者输出。当设置为输出时 (OUTPUT)，GPVAL 寄存器 GPxVAL 位内的数值将受到相应的 GPx

管脚的驱动，当设置为输入时（INPUT），输入状态能从 GPxVAL 位读取<sup>[8, 30]</sup>。

由以上的硬件接口电路结合液晶显示模块的管脚说明，可以对 C6713 的 GPIO 管脚作出如下的设置<sup>[30]</sup>：

```
Uint32 PinIDdata = GPIO_PIN15| GPIO_PIN14 | GPIO_PIN13 | GPIO_PIN12
|GPIO_PIN11 |GPIO_PIN10|GPIO_PIN9|GPIO_PIN8;
```

```
GPIO_pinEnable(hGpio,PinIDdata);
```

```
GPIO_pinDirection(hGpio,PinIDdata,GPIO_OUTPUT);
```

其中，GP3、GP4 分别与 LCM240128ZK 的 BUSY 和 INT 管脚连接，用于接收由 LCD 返回的忙检测信号和中断信号，因而设为输入。其余管脚则均设为输出，以实现 DSP 对 LCM240128ZK 内部的存储器写入数据，实现控制和显示。

#### 4. 2. 2 时序匹配

由于 DSP 处理速度要远远高于液晶模块的显示速度，因此若将 DSP 芯片直接与液晶模块连接，DSP 需要增加若干等待周期，这样就要求对时序的匹配特别注意<sup>[24, 26]</sup>。

LCM240128ZK 支持 6800/8080 时序，本文采用出厂时默认 6800 时序。6800 时序波形如图 4.2，其参数列表为表 4.2。

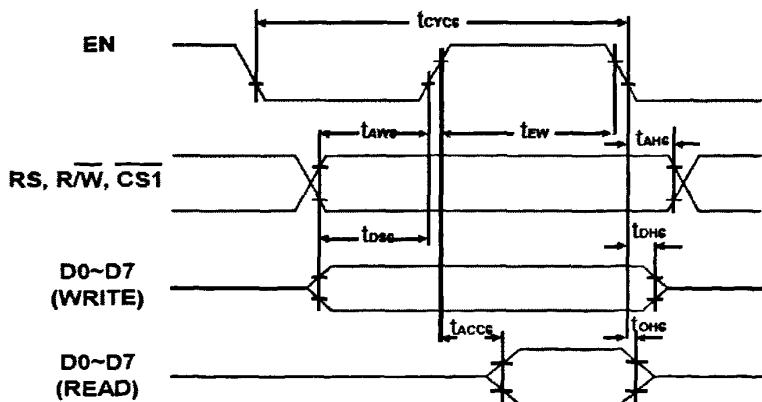


图 4.2 6800 系列时序波形

图 4.2 中的参数值参看下表：

表4.2 6800系列时序参数列表

Signal	Symbol	Parameter	Rating	Unit
			Min	Max
RS, R/W, CS1	$t_{AH6}$	Address hold time	10	-- ns
	$t_{Aw6}$	Address setup time	63	-- ns
	$t_{CYC6}$	System cycle time	800	-- ns
D0 to D7	$t_{DS6}$	Data setup time	63	-- ns
	$t_{DH6}$	Data hold time	10	-- ns
	$t_{ACC6}$	Access time	--	330 ns
	$t_{OH6}$	Output disable time	10	-- ns
EN	$t_{EW}$	Enable pulse width	400	-- ns

将 DSP C6713 的读写信号线与 LCM240128ZK 接口的读写信号线直接相连，时序由 DSP 的内部读写逻辑控制。但是由于 LCD 模块是 DSP 的一个慢速外设，要使两者的速度达到匹配，还必须进行一些时序方面的控制处理<sup>[26]</sup>。而对液晶的操作，主要是理解产生液晶控制芯片 Ra8820 的控制时序，下面结合 LCD\_DataWrite 子程序来分步说明其时序产生的具体过程。

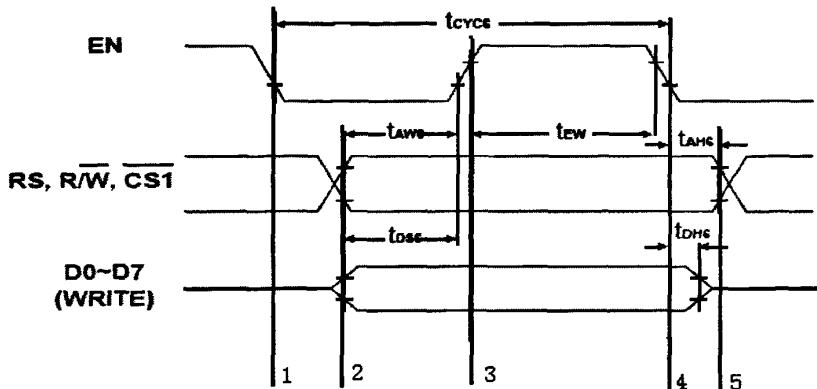


图4.3 Ra8820的控制时序分步图示

如上图 4.3 所示，将控制时序分为五步实现：

### (1) 第一步时序的产生

```

GPIO_pinWrite(hGpio,GPIO_PIN2,0); // EN = 0 ;
GPIO_pinWrite(hGpio,GPIO_PIN1,1); // R/W=1 read status
DelayEN0( ); // 400-63=337ns

```

DelayEN0( )会保持当前的电平，使 GP2 写入的 0 值一直保持到下一步时序的产生，也可以说是将数据线上的值锁存进来，直到下一指令的执行。

这时会产生图 4.3 中粗线 1 处的时序，即置 EN(enable)信号为 0；延迟 Delay 的数值由产生时序 1 和时序 2 的间隔来计算。

### (2) 第二步时序的产生

```
GPIO_pinWrite(hGpio,PinIDdata,Data_Data);  
GPIO_pinWrite(hGpio,GPIO_PIN2,0); // EN = 0;  
GPIO_pinWrite(hGpio,GPIO_PIN1,0); // R/W =0 write status  
GPIO_pinWrite(hGpio,GPIO_PIN0,0); // RS=0 To write to the Data register
```

```
DelayTAW6( );
```

这时需要把 Data\_Data 数据值写入资料寄存器。将 R/W、RS 的电平置低。在延迟时间的设置上，由于 6800 时序的参数  $t_{AW6}$  设定为最短不少于 63ns，且和  $t_{EW}$  之间没有时间间隔的设定，故为便于实现在此忽略二者间的空隙。

### (3) 第三步时序的产生

```
GPIO_pinWrite(hGpio,GPIO_PIN2,1); // EN = 1;  
DelayEN1( ); // Tew >=400 ns
```

在这步中，我们需要把 EN 置为高电平，在 6800 时序中 EN 信号高电平动作。

### (4) 第四步时序的产生

```
GPIO_pinWrite(hGpio,GPIO_PIN2,0); // EN = 0;  
DelayTDH6( );
```

这时需要把 EN 置成低电平，得到图 4.3 中粗线 4 处所示的时序。

### (5) 第五步时序的产生

```
GPIO_pinWrite(hGpio,GPIO_PIN1,1); // R/W =1 read status
```

此处得到图4.3中粗线5所示的时序。

到此为止，已经完成了一个完整的写数据的控制时序，实现了将数据写入资料寄存器的过程。读/写指令和读数据的过程与此过程相似，可以参考写出（源程序见附录）。



图 4.4 DSP 连接液晶模块显示结果

#### 4.3 由手写输入到液晶显示

在本文中，通过 3.2 节对手写板采集到的信号进行分析处理，已经得知，采集回的手写板数据由 5 个字节的一组值表示。其中的第一个字节为特征码，当手写板输入有效，获取的数据为坐标数据时，其具体表示为 0xC0h 或 0xC1h。所以在对手写数据的处理可以以第一字节作为要处理数据起始的判断标志。

由图 4.4 的流程可以清晰地阐述出本系统对手写数据的获取到显示在液晶模块上的处理思路。

首先，提取一个结构体（见 3.2 节），以其第一个字节的特征值作为评判，在程序中通过位运算（和 0xF0h 位与）判断，若获得的结果显示等于 0xC0h，并表示手写笔在板上动作有效，即此时手写板输出的是相应的坐标值。因此，保存数据下一步的处理。否则，读取下一个结构体，直至读到的是坐标信息为止。然后，由于液晶模块的显示数据是以二维坐标的形式作为基础的，所以要将手写结构体的 5 字节转换成二维的结构。但是，系统选用的液晶显示模块是  $240 \times 128$  的点阵液晶显示屏，也就是说，任一点的坐标只可能出现在  $(0x00h, 0x00h)$  到  $(0xEFh, 0x7Fh)$  的范围内。然而手写板的区域的大得多，经过多次的实验，手写板的坐标值最大可达  $(0x4Fh \times 0x7Fh)$ 。鉴于以上的数据关系，要把手写输入的信息显示到液晶显示屏上，就要经过一定的计算。

## 4.3.1 手写数据送入液晶显示前的处理

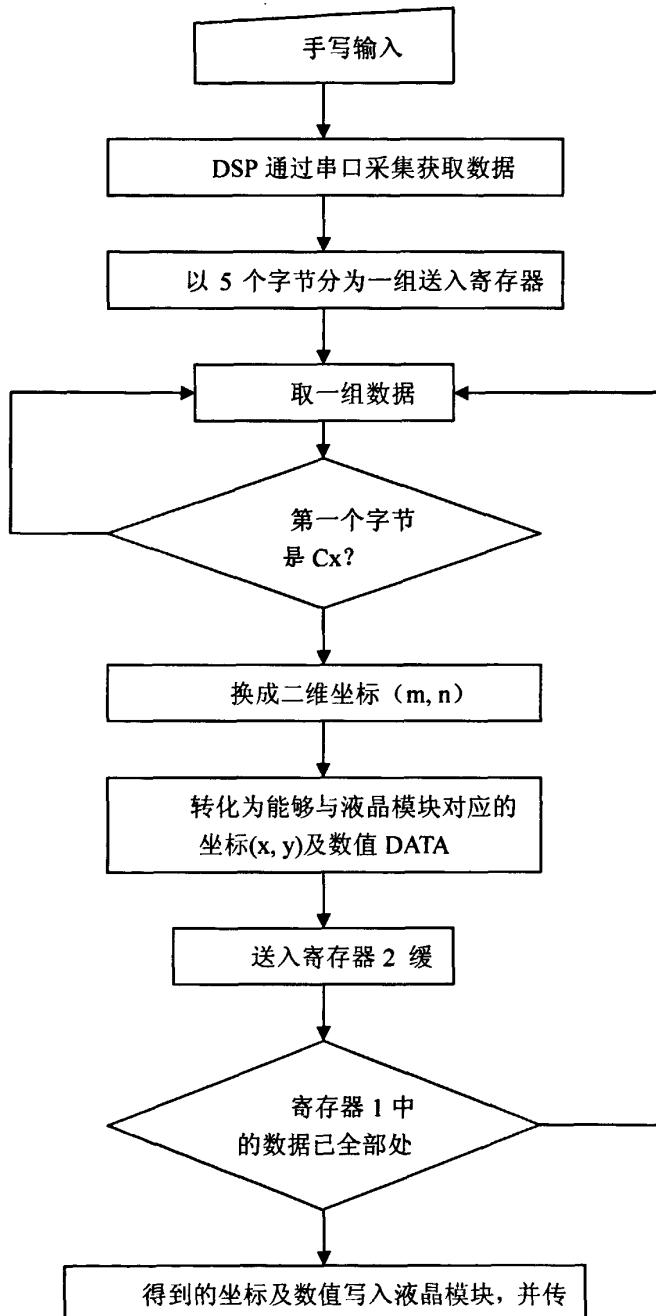


图 4.5 对手写数据处理的总体流程图

设手写信号的一个结构体 (o,p,q,r,s), 分别对应特征位、X 的低高位和 Y 的低高位。

在判定 o 的值并确定所取数据为有效的坐标数据后，按照以下步骤进行计算，就可以得到其在液晶显示屏上相应的坐标值。

(1) 把 X 的低高位和 Y 的低高位的 4 字节转化成常用二维坐标 (m, n)

系统使用的手写板规格为，书写范围  $120 \times 90$  (mm)，分辨率 40 point / mm。但在使用中不能完全以此作为基准，只能参考。

在 4.2 节已经得出在所有的 4 个字节中都只有 7bit 的有效位。X/Y 的低高位并不是简单的数位的拆分，而是对每一个点的坐标的粗、微表述，类似于日常用到的游标卡尺的粗微调。低高位的表示方式就是把一块手写板在划分成若干个大的区域之后，再在各大区中细分得到若干小区。其中，低位是对坐标的细微表述，可以看成是 40 point / mm 分辨率的数字表达。它的表达范围是 (0x00h~00x7Fh)，即一个低位字节最多可以达到 128 点。高位字节则是该点坐标大区的定位。清楚了手写板数据的定义，那么手写板上任一点的坐标值都可以通过下面的式子转化成常用的二维坐标 (m, n)

$$m = q \times 0x7Fh + p \quad (4.1)$$

$$n = s \times 0x7Fh + r \quad (4.2)$$

(2) 由手写板坐标得到对应液晶显示屏的坐标 (k, l)

LCM240128ZK 模块是  $240 \times 128$  点阵显示屏，要把手写板的较大范围的坐标放入相对小的液晶显示屏中显示，而且要求不失真。要实现这样的转化，就要对手写板的数据和液晶屏的坐标做多对一的映射。简单说就是，手写板的坐标重新划定区域，新的区域划分要和液晶显示屏的点阵相对应。

$$k = m / (0xd5h \times 0x02h) \quad (4.3)$$

$$l = n / 0x50h \quad (4.4)$$

(3) 用一种简单的新方法显示在液晶屏上

通常液晶屏的显示都会采用扫描式的一列一列地显示，对液晶显示屏上的显示数据时也会采用相应的扫描线的方式读取（直接对显示的黑白点阵用“1”“0” 表示）。但是本系统采用的显示方式是：先对液晶模块的光标进行定位，然后再写入一个字节的数值。

这是因为，根据使用者的一般书写习惯，通常会在纸张的中间开始书写。此外，本系统书写的多是比较简单的线条等，相对来说信息量不大。所以，扫描线式的输入读取

方法，会引入大量的无用信息增加冗余量。所以，采用先光标对定位，后写入一个字节的数值的方法，可以大大地减少数据量。

LCM240128ZK 的光标可以通过指令控制其位置，而且可以在初始液晶显示模块时改变 REG[10h] (CCR) 光标控制寄存器的值设定光标的宽度。系统设定光标固定为一个位元组的宽度 (8 个 Pixel)。对已转化得到的液晶显示坐标做一下处理，即：首先，对已得坐标 ( $k, l$ ) 以 8Pixel 为光标一次横向位移的值确定液晶屏光标的位置 ( $x = k / 0x08h, y = l - 0x01h$ ) 得到 ( $x, y$ )；然后，设置一个初始值 DATA=0x01h，由于  $k$  值是一个在 (0, 0xEFh) 区间内的任意值，可能在任何点上。把 0x01h 左移 ( $7-k \% 0x08h$ ) 位，再把移位后获得的值赋给 DATA。最后，根据 ( $x, y$ ) 和 DATA 三个数值可以唯一地确定在液晶显示上的一个点，并且这个点是与手写输入的点对应的。

比对用 0x00 填补空白的方法，用 3 个字节输入确定屏上的任意点节约了大量时间和存储空间。这样可以大大地减少数据量。

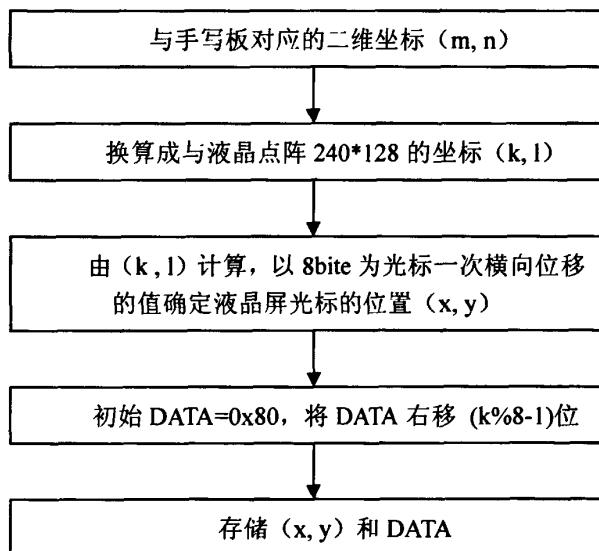


图 4.6 对手写输入点数据的处理

以上是对点坐标的处理，通常的输入不可能是单一的点，对于图形和文字我们把它看成是点的集合。集合中，点的关系有可能是同行或同列的，通过比较转化得到的光标 ( $x, y$ ) 就可以轻松地判断。由于液晶模块对光标在屏幕左上角时的定位是 (0,0)，因此把不同的  $y$  值，按升序排列就可以使点遵照液晶的规律由上到下按行写入；同理，把同

行得 x 值升序排列，就使点有做到有依次显示。

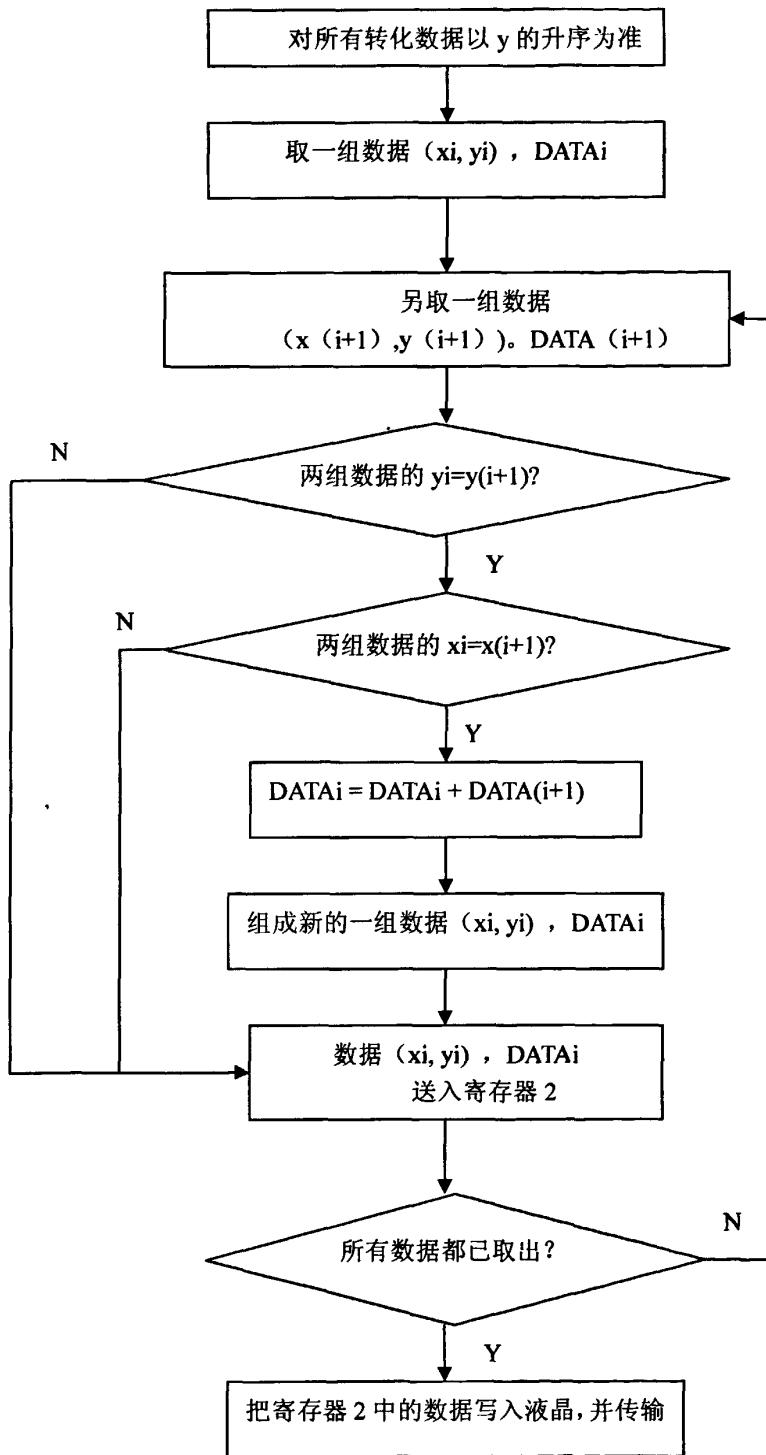


图 4.7 对手写线数据的处理

#### 4.3.2 对数据的修正

手写数据在送入液晶显示模块前，要经过多次的数据处理，而且要经过多个寄存器。在这个过程中难免会出现数据的突变，或者引入产生干扰的噪声值。所以，在送入液晶模块显示前对数据进行修正时十分必要的。

为了方便实现，系统采用的是均值差值法。

就是说，在连续采得一组值后，对多个值取平均值；然后计算所取得的数值和平均值的差。如果这个差在允许的范围内，则判定允许其对应的值输出；否则，进行修正。系统的液晶显示屏，光标的宽度设定的是 8bit，而列坐标则依然是“1bit”为一个移动单位。由此，设当行坐标差值大于 8，列坐标差值大于 4 时，其对应点将判决无效，不予显示。

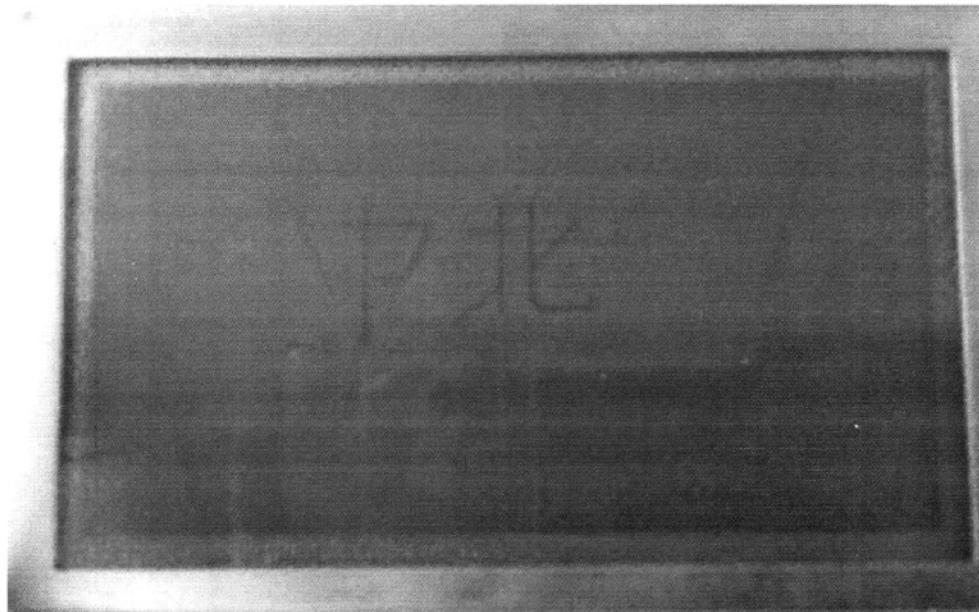


图 4.8 手写采集数据处理后液晶显示的照片

#### 4.4 本章小结

第一，在显示部分的连接控制时序的产生上。通常是以EN作为基本的参考。先对EN设定后，再完成对其他的参量设置。若加以改变，将控制时序的整体以时序图中要

实现的跳变作为参考点。在每一次跳变发生时，对涉及到的控制参数进行设定。这样可以更加清楚明了地对时序的产生进行控制，对各参数要求保持的延迟时间也可以更准确地予以控制。

同时，写入参量值时采用对各个参数分别使 1 位 2 进制数设定的方法，取代常用的写入 2 位 16 进制数同时对 8 个参量设定的方法。这样写入各引脚所对应的控制参数时，可以一目了然地检查参数的设定，避免可能出现的错误。更重要的是，可以节约 DSP 的使用资源。

第二，在手写数据显示的处理上。采用了一种简单的新方法：先对光标定位，然后再用一个字节的 16 进制数值写入 8bit 的点阵。

这是因为，根据使用者的一般书写习惯，通常会在纸张的中间开始书写。此外，本系统书写的多是比较简单的线条等，相对来说信息量不大。所以，通常液晶屏的显示都会采用扫描式的一列一列地显示，对液晶显示屏上的显示数据时也会采用相应的扫描线的方式读取。

通过这样的处理，可以使数据有序地写入液晶。同时减少数据量，有效地缩短数据写入的时间。而且对数据的正确性也有一定的保证。

另外，在数据的处理转换时，之前存储手写数据所采用的结构体方式显现出了它的优势。不仅在处理数据时取用方便，更在很大程度上保持了数据的完整性。

## 5 ASVD 方案的实现

公用电话网是传送语音信号的，语音本身是模拟信号，但公用电话网上的传输信道却上由模拟信道和数字信道混合组成的。所谓 SVD (Simultaneous Voice and Data, 数据/语音同传) 技术的主要特点是在进行数据通讯的同时利用普通电话机通话。根据具体实现方式的不同，SVD 有 ASVD 和 DSVD 两种解决方案。

ASVD 采用类似 Windows 操作系统的多任务处理机制和通讯带宽分割的方法，来实现数据/语音的同时传输。它采用固定的带宽来进行数据传输，而将剩下的带宽用于提供语音通道。也就是一部分线路带宽固定传输数据，余下的另一部分线路带宽固定传输模拟语音信号。其优点是利用带宽固定的一路信道中同时传送两路不同的语音、数据信号。即使不进行语音通话，这部分带宽也不会释放给数据传输之用。

作为两路信号在窄带的电话线中同时传输的关键技术，ASVD 可以采用频分多路复用方式<sup>[1-4]</sup>实现两路信号的同时传输，系统设计了一种模拟语音和数据同时传输的方案。

### 5.1 数据在模拟信道中的传输

数字通信是信息经编码变换处理后，以数字信号在信道上传输的，较之于模拟通信有很大的优点：抗噪声性能好，传输质量高，便于保密等，最突出的即抗干扰能力强，无噪声积累，只要不影响抽样判决的正确，即使在传输过程中波形有失真也不影响再生以后的波形；而模拟信号失真后，哪怕噪声很小，噪声的影响也是无法消除的。因此，数字通信得到迅速发展。

一般传输通道的频率特性总是有限的，即有上、下限频率，超过此界限就不能进行有效的传输。如果数字信号流的频率特性与传输通道的频率特性很不相同，那么信号中的很多能量就会失去，信噪比就会降低，使误码增加，而且还会给邻近信道带来很强的干扰。因此，在传输前要对数字信号进行某种处理，减少数字信号中的低频分量和高频分量，使能量向中频集中，或者通过某种调制过程进行频谱的搬移。这两种处理都可

以被看作是使信号的频谱特性与信道的频谱特性相匹配。

在模拟信道中传输数据信息有两种方式：基带传输和载波调制传输。

基带（Base-band）是指未经调制变换的信号所占的带宽，而数字基带传输就在传输距离不太长的情况下直接传输未经调制的基带信号，其波形所占据的频带一般从直流或较低的频率开始一直到很高的高频，频带很宽，由于这种信号的频谱是从大致的零频开始的，所以称之为数字基带信号<sup>[4]</sup>。所谓基带传输就是把信源生成的数字信号直接送入线路进行传输，如音频市话、计算机间的数据传输等。

数字信号的载波调制是信道编码的一部份。为了使数字信号在带通信道中传输，必须用数字信号对载波进行调制。例如，有线电视宽带综合网是基于模拟环境下的数字信号的传输，图象数字信号不是基带传输方式而是在射频通带中传输。

载波就是起“运载信息”作用的电波或光波，通常是正弦波或是周期性脉冲。“载波技术”是指利用载波传输信息的技术。主要的方法是把表示信息的信号加到载波上，使载波的频率、幅度或相位发生相应的变化（称为“调制”）；由于经调制后的信号中包含有原始信号的信息，传输到对方后，经解调、滤波等一系列过程后，可使原信号得以恢复<sup>[4]</sup>。例如传统的载波电话通信是把低频话音信号叠加到高频的载波上，使载波的幅度随着话音信号幅度的变化而变化。传输数字信号时也有三种基本调制方式：幅度键控，频移键控和相移键控，它们分别对应于用正弦波的幅度、频率和相位来传递数字基带信号。

由于普通电话线上只能传输声音信号，因此调制解调器要将描述图文的数字信号，转换为声音信号后经电话线传输。载波实际上也是一种声音信号，它携带着数字信息。可以这样理解，我们一般需要发送的数据的频率是低频的，如果按照本身的数据的频率来传输，不利于接收和同步。使用载波传输，我们可以将数据的信号加载到载波的信号上，接收方按照载波的频率来接收数据信号，有意义的信号波的波幅与无意义的信号的波幅是不同的，将这些信号提取出来就是我们需要的数据信号。故本系统采用载波传输。

## 5.2 频分多路复用

为了能够利用一条电话线同时进行话音和数据的传输，本系统使用了频率分割复用技术，将现有电话线的带宽划分为不同的频率带供不同的应用使用。当我们打电话时，话音信号只会占用电话线的一部分带宽传送，而电话线网络中的数据则是使用另外的空闲频率带传递图文信号。这样，我们就可以在使用电话线网络的同时使用电话。

频分复用<sup>[32]</sup>(FDM, Frequency Division Multiplexing)就是将用于传输信道的总带宽划分成若干个子频带(或称子信道)，每一个子信道传输 1 路信号。频分复用要求总频率宽度大于各个子信道频率之和，同时为了保证各子信道中所传输的信号互不干扰，应在各子信道之间设立隔离带，这样就保证了各路信号互不干扰(条件之一)。频分复用技术的特点是所有子信道传输的信号以并行的方式工作，每一路信号传输时可不考虑传输时延，因而频分复用技术取得了非常广泛的应用。频分复用技术除传统意义上的频分复用(FDM)外，还有一种是正交频分复用(OFDM)。

传统的频分复用典型的应用莫过于广电 HFC 网络电视信号的传输了，不管是模拟电视信号还是数字电视信号都是如此，因为对于数字电视信号而言，尽管在每一个频道(8MHz)以内是时分复用传输的，但各个频道之间仍然是以频分复用的方式传输的。

正交频分复用 OFDM<sup>[32]</sup>(Orthogonal Frequency Division Multiplexing)实际是一种多载波数字调制技术。OFDM 全部载波频率有相等的频率间隔，它们是一个基本振荡频率的整数倍，正交指各个载波的信号频谱是正交的。

OFDM 系统比 FDM 系统要求的带宽要小得多。由于 OFDM 使用无干扰正交载波技术，单个载波间无需保护频带，这样使得可用频谱的使用效率更高。另外，OFDM 技术可动态分配在子信道中的数据，为获得最大的数据吞吐量，多载波调制器可以智能地分配更多的数据到噪声小的子信道上。目前 OFDM 技术已被广泛应用于广播式的音频和视频领域以及民用通信系统中，主要的应用包括：非对称的数字用户环线(ADSL)、数字视频广播(DVB)、高清晰度电视(HDTV)、无线局域网(WLAN)和第 4 代(4G)移动通信系统等。

在电话通信中，并不要求接受设备把发送设备前的声音很准确的再发出来，因为要求毫不失真的重复产生原声波的波形，就需要将整个频带通过电话设备传过来，这样电

话通话的服务质量虽然提高了，但相应的对电话设备的质量要求也大为提高了，因而很不经济，因此在实际上，只要求能听懂所发送的语言，也就是通过电话设备只传送语言频带范围内最主要的一段频带就可以了，传送哪一段频带最好，可以用清晰度来评定<sup>[8]</sup>。

在电话通信中，清晰度（即发出的声音（无连贯意义的音节），在收听的地方可以能听懂的正确程度的百分数）是一个很重要的评判标准<sup>[3]</sup>。保证清晰度，则要求电话传输系统必须有恰当的传输频带。我们知道，人类语音的频率范围是 80~8000Hz。话音信号的高频部分对清晰度是比较重要的，影响话音清晰度的率在 500~20000Hz 之间。其中，1000~2000Hz 是保证清晰度的必传部分<sup>[37]</sup>。而话音频谱中的高频部分的作用则主要是对音色的表现。通过实验知道，清晰度和话音声波的能量与频率有关，分析结果是，高频部分对清晰度比较重要。因此要使电话听的清楚，必须传送 100HZ 以上的频率，不过要保留 2500HZ 以上的频率，意义就不大了。其次，在电话系统中，话音信号低频部分包含的能量较多，若除去，虽然清晰度可以符合要求，但话音信号的能量却不够，必会影响受话程度。所以电话频带首先应从兼顾清晰度和能量两方面着手，其次要考虑音色，即逼真度<sup>[3]</sup>。

我国对电话线传输频带规定是 300~3400Hz<sup>[37][38]</sup>。由于话音信号的清晰度和能量主要分布在 300~2000Hz 的范围内。所以，根据上述推理，我们将本系统话音信号的传输频带拟定为 300~2400Hz，图文信号在 2500~3400Hz 内传输，中间以 100Hz 的防护带<sup>[4]</sup>作为隔离，防止两路信号相互干扰。

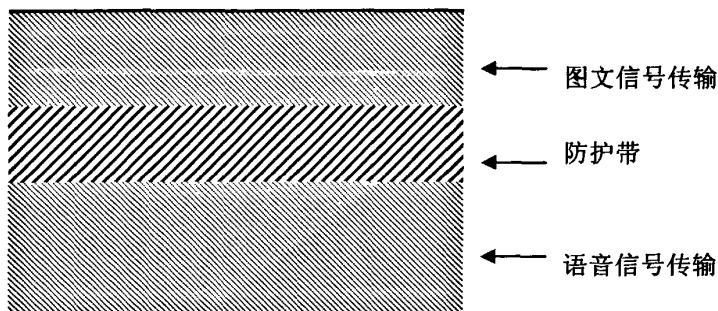


图 5.1 系统频分复用示意图

### 5.3 基于 DSP 实现的二进制频移键控（2FSK）调制解调方式

利用载波技术传输数字信号时有三种基本调制方式：幅度键控，频移键控和相移键控，它们分别对应于用正弦波的幅度、频率和相位来传递数字基带信号<sup>[4]</sup>。

FSK (Frequency Shift Keying)，又称频移键控法。FSK 是信息传输中使用得较早的一种调制方式，它的主要优点是：实现起来较容易，抗噪声与抗衰减的性能较好<sup>[4]</sup>。在中低速数据传输中得到了广泛的应用。它是利用基带数字信号离散取值特点去键控载波频率以传递信息的一种数字调制技术。

所谓 FSK 就是用数字信号去调制载波的频率。最常见的是用两个频率承载二进制 1 和 0 的双频 FSK 系统。具体说就是利用两个不同频率  $F_1$  和  $F_2$  的振荡源来代表信号 1 和 0，用数字信号的 1 和 0 去控制两个独立的振荡源交替输出。对二进制的频移键控调制方式，其有效带宽为  $B=2xF+2Fb$ ,  $xF$  是二进制基带信号的带宽也是 FSK 信号的最大频偏，由于数字信号的带宽即  $Fb$  值大，所以二进制频移键控（2FSK）的信号带宽  $B$  较大。

2FSK 信号的常用解调方法可采用非相干检测法和相干检测法，其中抽样判决器是判定那一个输入样值大，此时可以不专门设置门限电平。此外，2FSK 信号还有其他解调方法，比如鉴频法、过零检测法及差分检波法等。

#### 5.3.1 二进制频移键控（2FSK）调制

二进制频移键控（2FSK）中，当传送“1”码时对应于载波频率  $f_1$ ，传送“0”码时对应于载波频率  $f_0$

$$D_n = \begin{cases} 1, & \text{出现概率为 } P, \text{ 对应于 } \cos(\omega_1 t + \theta_n) \\ 0, & \text{出现概率为 } (1 - P), \text{ 对应于 } \cos(\omega_0 t + \varphi_n) \end{cases} \quad (5.1)$$

其中  $\omega_1 = 2\pi f_1$ ， $\omega_0 = 2\pi f_0$ ， $\theta_n$  为频率为  $f_1$  的载波的初始相位， $\varphi_n$  为频率为  $f_0$  的载波的初始相位。

由于在 2FSK 中， $\omega_0$  与  $\omega_1$  之间的改变是瞬间完成的。所以到容易想到，用一个矩形脉冲序列对一个载波进行调频可以获得 2FSK 信号。这正是频率键控通信方式早起采用的实现方法，也是利用模拟调频法实现数字调频的方法。2FSK 信号的另一种产生方法是采用键控法，即利用受矩形脉冲序列控制的开关电路对两个不同的独立频率源进行选通。

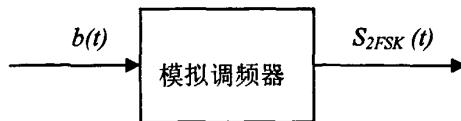


图 5.2 (a) 利用矩形脉冲产生 2FSK 信号

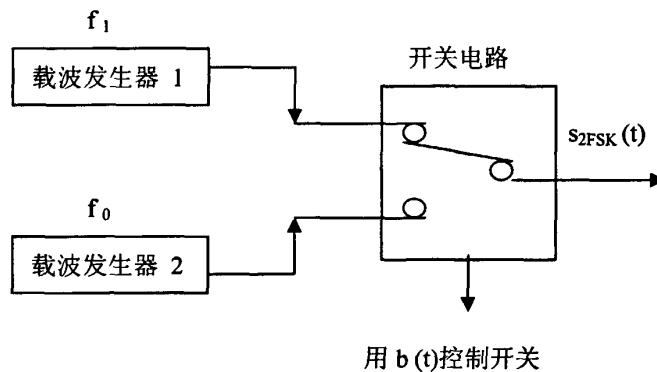


图 5.2 (b) 键控法产生 2FSK 信号

令  $\overline{D_n}$  为  $D_n$  的反码，即

$$\overline{D_n} = 1 - D_n \quad (5.2)$$

则有：当  $D_n = 1$  时， $\overline{D_n} = 0$ ；当  $D_n = 0$  时， $\overline{D_n} = 1$ 。

则 2FSK 信号可表示为：

$$s_{2FSK}(t) = \left[ \sum_{n=-\infty}^{\infty} D_n g(t - nT_s) \right] \cos(\omega_1 t + \theta_n) + \left[ \sum_{n=-\infty}^{\infty} \overline{D_n} g(t - nT_s) \right] \cos(\omega_0 t + \varphi_n) \quad (5.3)$$

其中，在分析中假设  $g(t)$  为单个矩形脉冲序列，脉宽为  $T_s$ ，其表达式为：

$$g(t) = \begin{cases} 1, & 0 \leq t \leq T_s \\ 0, & t \text{ 其他} \end{cases} \quad (5.4)$$

相位不连续的 2FSK 信号可以看成是两个 2ASK 调幅信号之和。

一般来说，键控法得到的  $\theta_n$  和  $\varphi_n$  是与序列无关的，反映在  $s_{2FSK}(t)$  上，仅表出  $\omega_1$  与  $\omega_2$  改变时其相位是不连续的；而用模拟调频法调制时，由于  $\omega_1$  与  $\omega_2$  改变时  $s_{2FSK}(t)$  相位是连续的，故  $\theta_n$ 、 $\varphi_n$  不仅与第 n 个信号码有关，而且  $\theta_n$  和  $\varphi_n$  之间也应保持一定的关系。

在求 2FSK 信号的功率谱密度时，可将 2FSK 信号表示成两个 2ASK 信号的和<sup>[44]</sup>，令：

$$b_1(t) = \sum_{n=-\infty}^{\infty} D_n g(t - nT_s) \quad (5.5)$$

$$b_2(t) = \sum_{n=-\infty}^{\infty} \overline{D}_n g(t - nT_s) \quad (5.6)$$

其中  $\overline{D}_n$  为  $D_n$  的反码，则相位不连续的 2FSK 信号可表示为

$$s_{2FSK}(t) = b_1(t) \cos(\omega_1 t + \theta_n) + b_2(t) \cos(\omega_0 t + \varphi_n) \quad (5.7)$$

根据 2ASK 信号的功率谱密度的表示式，可以的得到 2FSK 信号的功率谱密度的表示式：

$$\begin{aligned} P_{ss}(f) = & \frac{1}{4} f_s P (1-P) \left[ |G(f + f_1)|^2 + |G(f - f_1)|^2 \right] + \\ & \frac{1}{4} f_s P (1-P) \left[ |G(f + f_0)|^2 + |G(f - f_0)|^2 \right] + \\ & \frac{1}{4} f_s^2 (1-P)^2 |G(0)|^2 [\delta(f + f_1) + \delta(f - f_1)] + \\ & \frac{1}{4} f_s^2 P^2 |G(0)|^2 [\delta(f + f_0) + \delta(f - f_0)] \end{aligned} \quad (5.8)$$

从以上分析可以看到：第一，2FSK 信号的功率谱密度由连续谱和离散谱组成。其

中，连续谱有两个双边谱叠加而成，而离散谱出现在两个频率位置上；第二，若两个载频之差较小，比如小于  $f_s$ ，则连续谱出现单峰，若载频之差逐步增大，即  $f_0/f_1$  的距离增加，则连续谱将出现双峰；第三，由上面两特点看到，传输 2FSK 信号所需的第一零点带宽  $\Delta f$  约为

$$\Delta f = |f_1 - f_0| + 2f_s \quad (5.9)$$

式中， $f_s = 1/T_s$ 。

### 5.3.2 二进制频移键控（2FSK）的解调

二进制频移键控信号的解调分为相干解调和非相干解调其中，相干解调的抗干扰能力优于非相干解调，但相干解调要在接收端进行载波同步，因此实现起来比较复杂。相对于相干解调，非相干解调方式实现简单，无需恢复出同步的载波，因此二进制频移键控信号的解调一般都采用非相干解调。然而，当输入信号的信噪比很大时，两种解调方式的差别不很明显。一般而言，大信噪比时常采用包络检测法，小信噪比时可采用相干解调法<sup>[40]</sup>。非相干解调有包络检测法、鉴频法、过零检测法、差分检测法等。

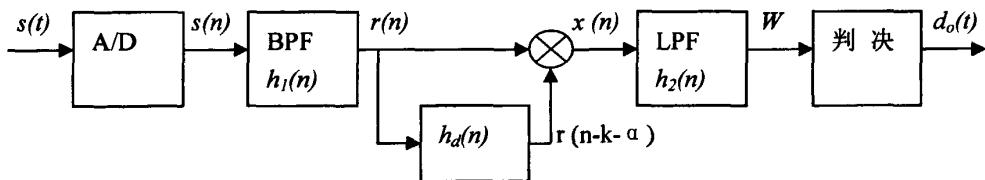


图 5.3 二进制频移键控(FSK)信号的非相干差分数字解调原理图

如图 5.3 所示为二进制频移键控(FSK)信号的非相干差分数字解调原理图。接收端的输入信号  $s(t)$  经过模数转换(A/D)变成数字信号  $s(n)$ ， $s(n)$  经过带通数字滤波器  $h_1(n)$  滤除带外的无用信号成分<sup>[37]</sup>。

在利用数字信号处理器(DSP)很容易地实现地数字信号的整数倍延迟的同时，可用一个单零点的有限冲激响应滤波器(FIR)来实现小数倍的延迟，其整体的解调性能完全取决于  $\pi/2$  的相位延迟精度。下面，本文介绍一种新的实现方法，可以比较简单的计算出 FIR 滤波器的系数。

其设计思想就是基于：要让过滤器的 FD (Fractional Delay, 分数延迟) 在要求的范围内 ( $0 \leq d \leq 1$ )，并且让每个系数都近似于  $d$  的多项式。

$$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^N \frac{D-k}{n-k} \quad (n = 0, 1, 2, \dots, N) \quad (5.10)$$

其中， $k = f_s / (4f_c)$  为滤波器系数。 $h(0) = 1 - D, h(1) = D$

可得滤波器  $h_D(n)$  的系统函数为：

$$H(z) = z^{-D} \quad (5.11)$$

则有，

$$H(e^{j\omega}) = e^{-j\omega D} = \cos(\omega D) - j \sin(\omega D) \quad (5.12)$$

本系统中由以上计算得到的参数如下： $D=5.7143$ , 在程序中将用到。

### 5.3.3 基于 DSP 实现的 2FSK 的结果

由以上工作为基础，应用 C 语言编程，通过 DSP 仿真得到下面的一组波形图：

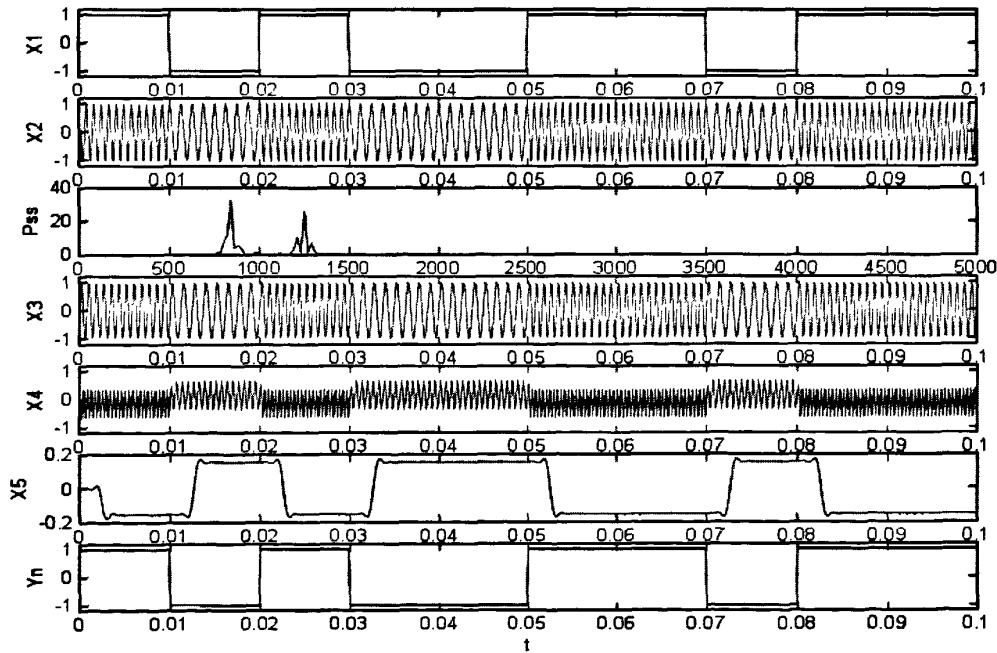


图 5.4 基于 DSP 实现的 2FSK 的调制解调结果

上图是以抽样频率  $f_s=9.6\text{kHz}$  为例的二进制频移键控信号调制与解调结果的波形图。X1 是手动输入的任意一组码字，引用的图中以 (1, 0, 1, 0, 0, 1, 1, 0, 1, 1) 为输入码字等待调制。将码字送入 DSP 经调制获得 X2，码字“0”和“1”分别对应频率为 800Hz、1400Hz 的载波，这一点也可以由波形 Pss 表示的功率谱示意图看出。从下面开始进入信号解调的阶段。X3 是信号在经过 FIR 滤波器  $h_D(n)$  处理后的信号，将其与 X2 波形相乘，得到 X4。经过低通滤波器滤波，得到如波形 X5 的非标准波，这样的波形基本可以分辨出高低的差值。但是鉴于对数字信号处理器的信号准确处理，键入判决器。

#### 5.4 本章小结

系统对两路信号同传采用的是 ASVD 方案，即模拟语音和数据同时传输。方案中要实现同传则需要应用频分多路复用技术。

频分复用技术使利用一条电话线同时进行话音和数据传输的想法可以实现。将我国对电话线规定的 300~3400Hz 传输频带划分为两个不同的频带供不同的应用使用。话音信号只占用电话线 300~2400Hz 的部分带宽传送，而手写的图文数据则是使用另外频率带 2500~3400Hz 内传递，中间以 100Hz 的防护带。这样，我们就可以在使用电话线网络的同时使用电话。

二进制频移键控(2FSK)是数字通信中常用的一种调制方式，其调制与解调的方法有很多。采用数字信号处理器 C6713 来实现二进制频移键控信号的数字调制与非相干数字解调，大大地降低了硬件电路的复杂程度，提高了系统的灵活性。

此外，系统在应用二进制频移键控信号非相干数字解调时，引入了一种新的计算 FIR 滤波器系数的方法。这种方法计算简单，但仍然可以达到较高的精度要求，在其他应用 FIR 滤波器的情况可以参考。

## 6 系统接入公话网的设计

### 6.1 公共电话网络

公共电话网<sup>[4]</sup>（PSTN，Public Switched Telephone Network）是传递电话信息的电信网，是可以进行交互型语音通信、开放电话业务的电信网。电话网包括本地电话网、长途电话网、国际电话网等多种类型。是业务量最大、服务面最广的电信网。电话网是开放电话精力为广大用户服务的通信网络。最早的电话通信形式只是两部电话机中间用导线连接起来便可通话，但当某一地区电话用户增多时要想使众多用户相互间都能两两通话，便需设一部电话交换机，由交换机完成任意两个用户的连接，这时便形成了一个以交换机为中心的单局制电话网。在某一地区(或城市)随着用户数继续增多，便需要建立多个电话局，然后由居间中继线路将各局连接起来，形成多局制电话网。

电话网经历了由模拟电话网向综合数字电话网的演变。除了电话业务，还可以兼容许多非电话业务。因此电话网可以说是电信网的基础。

数字电话网与模拟电话网相比，在通信质量、业务种类、为非话业务提供服务、实现维护、运行和管理自动化等方面都更具优越性。现在电话网正在向综合业务数字网、宽带综合业务数字网以及个人通信网的方向发展。

目前使用的程控交换机，直流馈电电压是 48V，由于铅蓄电池是浮充工作的，电话机挂机时实际电压约直流 50V。当电话机摘机后，话机直流回路接通，话机中有电流，这时电压降为 7~10V，视线路长短、话机不同而异。话机摘机后的电流不相同，受用户线路、电话机及局内交换设备影响。一般在 20~30mA。挂机时电流应该在 5 mA 左右。当铃响时，振铃电压是 75V、频率为 16~25Hz 的交流电压。

并且，如果是之后新装的线路中，大多地区已经使用数字模拟混合接入，即若你的电话为 06 年之后购买并符合标准的，则为数字信号，用载波模式装载到线路中传输，若为之前的或者局端设备还没有更新，那么则是模拟信号，用电流高低震荡的方式传送。

电话线路对电流有一定的限制，进入电缆前就安装了保险丝。交换设备供出一个个模

块出来时都有电，电话线是 48V 供电，线路上已串接了电阻，每一端大约 600 欧姆，负载稍微超过，交换机就会认为摘机，这样就会影响通讯。电话响铃的时候耗电量是最大的，线路设计时已经考虑到正常通话的电流，平均每户 48V 的直流电不超过 20 mA。在某个用户端长时间取电 100A 时就会被监测设备监测到，自动给该用户闭电<sup>[45]</sup>。

## 6.2 数据信号处理部分的入网连接

系统与公共电话网的连接绝不仅仅是接口与接口的连接。要是系统入网不仅仅要考虑电话线带宽、窄带信道的复用、信号调制解调的方式等由软件程序控制的理论问题，更要注重实际使用中会出现的应用问题。例如：

(1) 电话线中的电压值很高。即便是在电话接通之后，工作电压仍然会保持在 7~10V，振铃电压更是能高达 75V。如此高的电压，怎么能送入额定工作电压仅是 3.3V 的 DSP 中呢？

(2) 在实现语音信号和数据信号同时传输时，系统采用的是频分复用的方式。把信道分成了两部分：低频段的语音信号传输部分和高频段的数据信号传输部分。并且在两部分中间设立了隔离防护带。但实际信道中存在着各种各样的噪声，而且进过长距离的传输，各子频带中的信号可能发生偏移，严重的也可能会跨过防护带干扰到另一路信号。

### 6.2.1 DSP 与公共电话网的连接

第一个问题是一个至关重要的问题。首先，要保证 DSP 工作在正常的电压范围之内；其次要不影响电话的正常使用，即不能脱离电话机的工作环境——公用电话网。对于这个问题，我们将应用到一些外围电路。

具体方法是，用振铃过程的交流电压作为以下电路工作的起始标志。首先，应用一个延迟电路先记录系统 T 时刻的电压值；然后用一个电压比较器比较 T 和 T+△T。如果 T 小于 T+△T，表示电话处于挂机状态；如果 T 大于 T+△T，表示电话处于通话中，要对 DSP 上电工作。

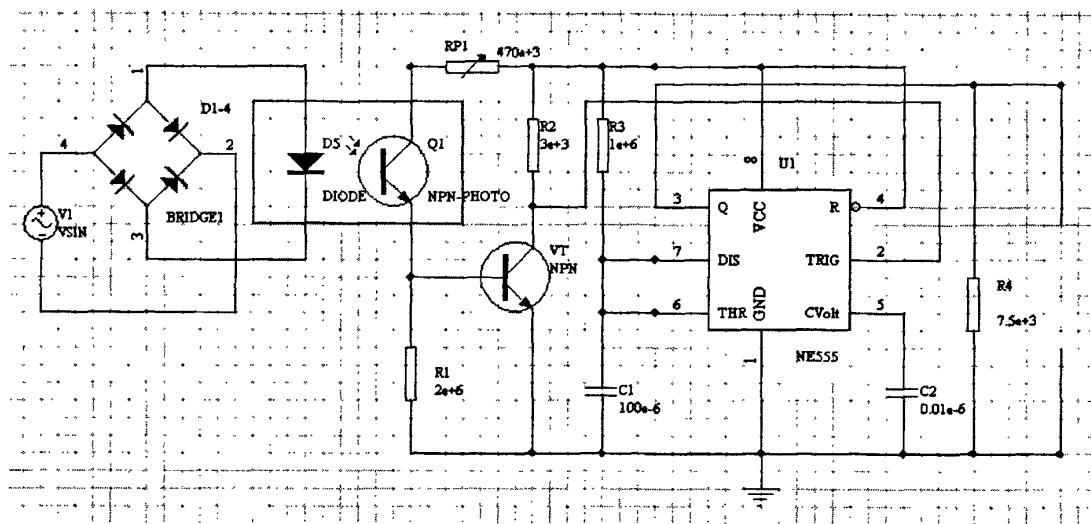


图 6.1 利用振铃电压设计的开关电路

NE555 与 R3, C1 组成单稳态延时电路, 延时时间为  $T = 1.1R3C1 = 10\text{ms}$ 。功能: 当来电话时, 由电话外线传来的电话铃音信号经过由 D1~D4 组成的桥式整流电路整流后, 变为直流信号电压。这一直流信号通过一只由 D6 和 O1 组成的光耦合器使 VT 导通。VT 导通后, 集电极电压降低, 这一低电压经 NE555 管脚 2 将单稳态电路触发翻转, 单稳态电路输出高电平, 通过 R4 进入下一电路。

### 6.2.2 对可能出现的噪声的滤波

为了应对第二个问题, 系统信号处理的前端将引入一个带阻滤波器。带阻滤波器是滤除某一个范围的频率, 而让其余频率通过的滤波器。如高通滤波器+低通滤波器可组成带阻滤波器。它大体分为模拟带阻滤波器和数字带阻滤波器。

模拟带阻滤波器一般是用电路元件(如电阻、电容、电感)来构成我们所需要的频率特性电路。模拟带阻滤波器的原理是通过对电容、电阻和电感参数的配置, 使得模拟滤波器对基波呈现很小的阻抗, 而对谐波呈现很大的阻抗, 这样当负载电流信号通过该模拟带阻滤波器的时候就可以把基波信号提取出来。目前, 有些有源滤波器利用模拟电路实现带阻滤波器检测负载电流的基波分量, 并且在实际中得到了应用。但是, 模拟带阻滤波器也有一些自身的缺点。这是由于模拟滤波器的中心频率对电路元件(如电容,

电阻，电感)的参数十分敏感，较难设计出合适的参数，而且电路元件的参数会随外界环境的干扰发生变化，这会导致中心频率的偏移，影响滤波结果的准确性。

数字带阻滤波器就是用软件来实现上面的滤波过程，可以很好地克服模拟滤波器的缺点，数字带阻滤波器的参数一旦确定，就不会发生变化，只要电网的波动频率在我们设计的范围之内，就可以比较好地提取出基波分量。数字滤波器根据其类型可以分为 IIR 型和 FIR 型。FIR 型只有零点，不容易像 IIR 型那样取得比较好的通带与阻带特性。所以，在一般的设计中选用 IIR 型。IIR 型又可以分成 Butterworth 型滤波器，Chebyshev I 型滤波器，Chbyshev II 型滤波器和椭圆型滤波器等。

系统中选用了 IIR 型滤波器，采用低通滤波器加高通滤波器的方法实现带阻滤波。这样不仅合理利用了 DSP 资源，而且省去了硬件搭建和调试的时间及花费。详细程序请见附录。

### 6.3 利用电话线上电压设计后备电源

在了解公共电话网络之后，特别是对线上电压电流的值了解之后，很自然地想到要利用电话线上已有的资源作为 DSP 的后备电源。因此，根据已知的 DSP 工作状态要求，和电话线上的电压电流值，设计得利用电话线上电压的微功率后备电源。

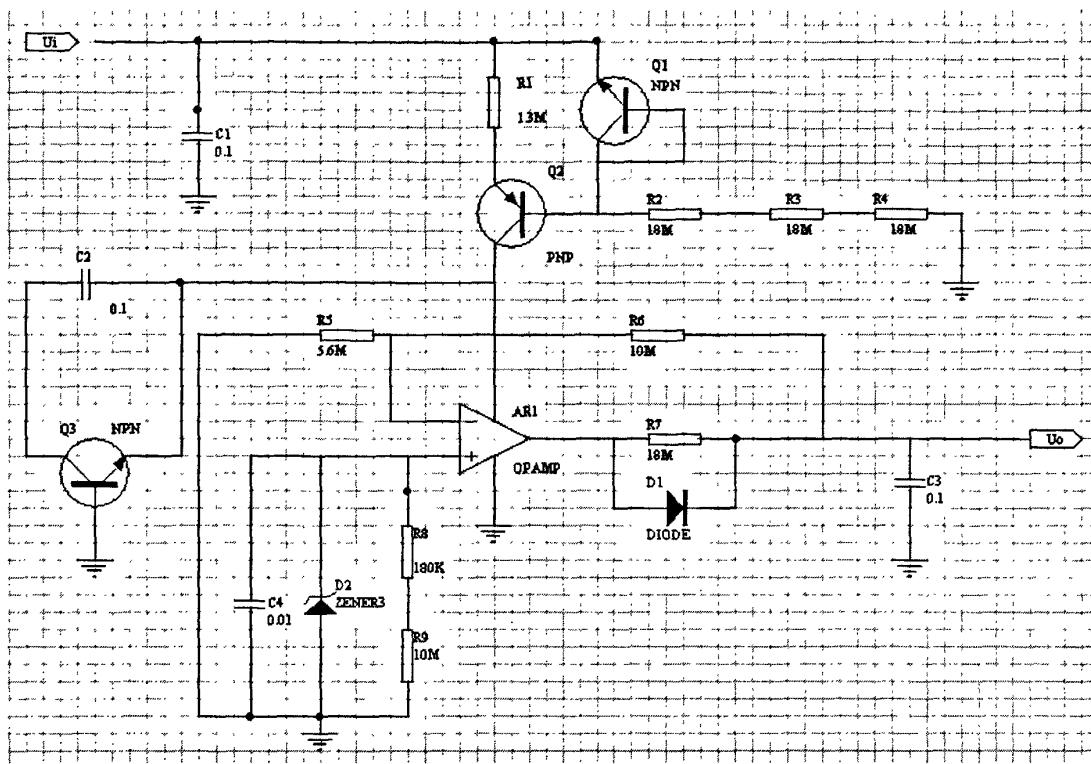


图 6.2 利用电话线电压设计的后备电源

如图是利用电话线上电压的微功率后备电源。微功率后备电源的输入电压  $U_i$  取自电话线上 7~10V 电压，输入电流低于  $4\mu\text{A}$ 。微功率后备电源的输出电压  $U$ ，为 3.3V，电流为  $1\mu\text{A}$ 。电路中， $VT_1$  和  $VT_2$  为电流源，为  $A_1$  提供约  $3\mu\text{A}$  电流。 $A_1$  等为参考电压电路和比较器电路，其静态消耗电流最大为  $0.6\mu\text{A}$ 。

#### 6.4 本章小结

本章的工作主要是对于系统接入公共电话网络的准备。

公共电话网无论是在待机状态还是在振铃、接听状态，其线上电压都远大于 DSP 的额定工作电压，所以如何避免损坏系统保护 DSP 芯片是入网连接要面对的首要问题。本文对电话接通前振铃状态的交流信号，设计了一个单稳延时电路，其目的即：当有电话打入，交流信号送入话机，将此交流信号作为单稳延时电路的输入，产生一个直流电

压。这个直流量可以对后面的电路起到参考作用。

然后，为实现两路信号同时入网传输，系统采用频分复用的方式，并在两路信号的频带中间加入了防护带隔离。但在真实信道中，噪声和任何可能发生的意外都可能干扰到信号的传输。所以，接受到调制信号后首先要经过带阻滤波器的再次分离。

最后，为了合理地利用资源，本文利用电话线上的电压设计了一个可以为 DSP 供电的微功率后备电源。

## 结 论

该课题经过调研、分析、从总体方案论证到相关软硬件设计，深入地探讨 DSP 结构的系统设计和公用电话网的多媒体应用；根据设计要求充分考虑系统的通用性，研制出了基于电话线的图文声信号传输的系统。

该系统在使用过程中帮助使用者利用图形、文字等阐述观点、表达感情，实现完整明了的交流沟通。手写板的自由输入以及液晶显示屏的实时显示，满足了不同层文化层次，不同年龄阶段的要求，对加强和方便交流能提供有力的支持。开发相应的外围补充电路和程序，随时根据实际需要应用在其他类似产品中。

硬件电路设计包括：

- (1) 以 DSP 模块核心，实现了系统设计要求中的手写数据信号的采集和显示的硬件系统平台；
- (2) 完成 DSP 与点阵图形式液晶显示模块的接口电路；
- (3) 完成 DSP 与电磁触控式手写板的连接电路；
- (4) 完成 DSP 与公用电话网络连接的外围电路设计；
- (5) 通过利用电话线上的电压为 DSP 提供微功耗电源的硬件电路设计。

软件实现部分包括：

- (1) 完成了对硬件各模块的测试软件设计；
- (2) 实现 DSP 对点阵图形式液晶显示模块的控制及驱动软件设计
- (3) 实现图文数据信号在模拟信道上应用 2FSK 调制解调的程序编写；
- (4) 实现与公用电话线网络通信连接应用的带阻滤波；

---

## 附录

### 附录 1 液晶显示程序

```
#include <csl.h>
#include <csl_irq.h>
#include "DEC6713.h"
#include "uartn.h"
#include <stdio.h>
#include <string.h>
/*****************/
#define LENGTH 160
Uint32 UART_Handle;
Uint32 UARTA_Handle;
Uint32 RevBuffer_A[160];
Uint16 i;
Uint8 int_flag = 1;
Uint8 TempData;
Uint16 BaudRate=baud_9k6;
Uint8 reg_data=0;
Uint8 Lth = 0;
Uint8 RcountA =0;
Uint8 ScountA =0;
Uint8 ReVal = 0;
#pragma DATA_SECTION(uart_ab_channel_select,".uart_var");
unsigned char uart_ab_channel_select = 0;
/*****************/
void init_uart();
extern far void vectors();
```

```
struct uart
{
    unsigned int FB;
    unsigned int XL;
    unsigned int XH;
    unsigned int YL;
    unsigned int YH;
}s[5];

struct lcd
{
    unsigned int x;
    unsigned int y;
}p[160];

struct lcd HW_to_LCD(struct lcd);

main()
{
    int n=0;
    /* Initialize the chip support library, must when using CSL */
    CSL_init();
    /*Initialize DEC6713 board.*/
    DEC6713_init();
    /*Initialize receive buffer.*/
    for(i=0;i<160;i++)
    {
        RevBuffer_A[i] = 0;
    }
}
```

```

/* Disable interrupt. */

IRQ_globalDisable();
IRQ_RSET(EXTPOL,0x0F);
IRQ_setVecs(vectors);
IRQ_map(IRQ_EVT_EXTINT6,6);
IRQ_disable(IRQ_EVT_EXTINT6)
IRQ_clear(IRQ_EVT_EXTINT6);

/*Open UART.*/
init_uart();

/* Enable interrupt */

IRQ_enable(IRQ_EVT_EXTINT6);
IRQ_globalEnable();
IRQ_nmiEnable();

for(n=0;n<160;n++)
{
    p[n]=HW_to_LCD(p[n]);
    printf("p[%d]=%x%x\n",n,p[n].x,p[n].y);
}

 *****
CHENGE THE DATA OF HANDWRITING TO FIT THE LCD
 *****
struct lcd HW_to_LCD(struct lcd r[160])
{
    int i=0,j;
    struct uart t[5];
    r[160]={ {0x00,0x00} };
    for(RcountA=0;RcountA<LENGTH;RcountA++)
    {

```

```

if((s[RcountA].FB&0xf0)==0xc0)      //根据起始位判断是不是有效的手写数据

{
    t[i]=s[RcountA];
    i++;
}

for(j=0;j<=i;j++)

{
    r[j].x=((t[j].XH-0x00)*0x80+t[j].XL)/(0xd5*0x02);

// 对 X,Y 的数据转换成与液晶 (240*128) 对应的坐标形式

    r[j].y=((t[j].YH-0x00)*0x80+t[j].YL)/0x50;

    printf("r[%d]=%0x %0x\n", j, r[j].x, r[j].y );

}

return r[];
}

*****\\

UART initialize routine

*****\\

void init_uart()

{
    /* Initialize uart_a */

    UARTA_Handle = UART_open(UART_A);

    BaudRate = baud_9k6;

    UART_setup(UARTA_Handle,BaudRate,
               data_w8,
               data_s1,
               0,
               0xc7,
               0x08);

    /* Clear RHR and LSR registers. */
}

```

```

TempData =UART_rget(UARTA_Handle,UART_RHR);

TempData = UART_rget(UARTA_Handle,UART_LSR);

/* Setup UARTA interrupt. */ UART_IntSetup(UARTA_Handle,uartint_rhr);

}

/*****************/
/* -UART interrupt handle function.
/*****************/

interrupt void
c_int6(void)
{
    /*Receive data and save in data_buffer.*/
    reg_data = DEC6713_cpld_rget(DEC6713_INTSTAT_REG);

    // 接收 UART 数据

    if((reg_data & 0x60) == 0x20)

    {

        TempData = UART_rget(UARTA_Handle,UART_LSR);

        if((TempData&0x01) == 0x01)

        {

            for(i=0;i<FIFO_Length;i++)

            {

                DEC6713_wait(BaudRate/baud_38k4 * 2026);

                RevBuffer_A[RcountA]=UART_receive_single(UARTA_Handle);

                s[RcountA].FB=UART_receive_single(UARTA_Handle);

                s[RcountA].XL=UART_receive_single(UARTA_Handle);

                s[RcountA].XH=UART_receive_single(UARTA_Handle);

                s[RcountA].YL=UART_receive_single(UARTA_Handle);

                s[RcountA].YH=UART_receive_single(UARTA_Handle);

            }

        }

    }

}

```

```
RcountA++;
if(RcountA == 160) RcountA =0;
}
else
{
    UART_rset(UARTA_Handle,UART_FCR,0x02);
}
int_flag = 0;
return;
}
```

## 附录 2 DSP 与 LCM 连接的程序

```
#include <csl.h>
#include <csl_gpio.h>
#include <csl_gpiohal.h>
#include <DEC6713.h>
#include <stdio.h>
#include <string.h>
/*****************/
static GPIO_Handle hGpio;
extern far void vectors();
/*****************/

void printlcd(void);
void LCD_Reset(void);
void LCD_Initial(void);
void LCD_Display_On(void);
void LCD_Display_Off(void);
void LCD_CursorX(unsigned);
void LCD_CursorY(unsigned);
void LCD_Clear(void);
void LCD_CmdWrite(unsigned);
void Windows(unsigned);
void LCD_DataWrite(unsigned);
unsigned LCD_CmdRead(unsigned);
unsigned LCD_DataRead(void);
void LCD_ChkBusy(void);
void printpoint(void);

void DelayXms(int);
```

```
void DelayEN1(void);
void DelayINI(void);
void DelayTAW6(void);
void DelayTDH6(void);
void DelayEN0(void);
void DelayTACC6(void);
void DelayEN1plusTACC6(void);
void DelayTOH6(void);

#define N 5

unsigned REG_READ;
unsigned DATA_READ;

Uint32 PinIDdata = GPIO_PIN0|GPIO_PIN2|GPIO_PIN3|GPIO_PIN8|GPIO_PIN9
|GPIO_PIN10|GPIO_PIN11|GPIO_PIN12;
Uint32 PinIDcontrol = GPIO_PIN1|GPIO_PIN15|GPIO_PIN14|GPIO_PIN13;

/*****************/
/*Main program area */
/*****************/

main()
{
    /* Initialize CSL,must when using CSL.*/
    CSL_init();

    /* Initialize DEC6713 board.*/
    DEC6713_init();
    IRQ_setVecs(vectors);

    /* point to the IRQ vector table */
    IRQ_globalEnable();
```

```

/* Globally enable interrupts

IRQ_nmiEnable();

/* Enable NMI interrupt

/* Set GPIO. */

hGpio = GPIO_open(GPIO_DEV0,GPIO_OPEN_RESET);

GPIO_reset(hGpio);

GPIO_pinEnable(hGpio,PinIDcontrol);

GPIO_pinDirection(hGpio,GPIO_PIN1,GPIO_OUTPUT);           // RS

GPIO_pinDirection(hGpio,GPIO_PIN15,GPIO_OUTPUT);          // R/W

GPIO_pinDirection(hGpio,GPIO_PIN14,GPIO_OUTPUT);          // EN

GPIO_pinDirection(hGpio,GPIO_PIN13,GPIO_OUTPUT);          // RST

GPIO_pinEnable(hGpio,PinIDdata);

LCD_Reset();           // LCD 复位

LCD_Initial();        // LCD 初始化设置

LCD_CmdWrite(0x00); LCD_CmdWrite(0xc5);

//REG[00] Bit3=0 为绘图模式，写入的资料会被视为 Bit-Map 模式

LCD_CursorX(0x00); LCD_CursorY(0x00);

while(1)

{

LCD_Clear();

LCD_CursorX(0x00);

//设定光标的位置

LCD_CursorY(0x00);

printpoint();

DelayXms(1000);

}

}

/*****************************************/

```

```

/*Sub program area */
*****  

struct point
{
    unsigned x;
    unsigned y;
    unsigned Data;
};

struct point LCD_point(struct point);

void printpoint(void)
{
    struct point p[N]={ {0,0,0x00} };

    //    int h;
    int i,j,k;
    struct point templ,temp2;

    for(i=0;i<N;i++)
    {
        p[i]=LCD_point(p[i]);
        printf("p[%d]= %d %d %x\n\n",i,p[i].x,p[i].y,p[i].Data);
    }

    for (i=0;i<N-1;i++)
        for (j=i+1;j<N;j++)
        {
            if ( p[i].y>p[j].y )
            {
                templ=p[i];
                p[i]=p[j];
                p[j]=templ;
            }
        }
}

```

```

    p[j]=temp1 ;
}

for (i=0;i<N;i++)
{
    printf("y:p[%d]= %d %d %0x ;",i,p[i].x,p[i].y,p[i].Data);
}
}

for (j=0;j<N-1;j++)
{
    for (k=j+1;k<N;k++)
    {
        if ( p[j].x>p[k].x )
        {
            temp2=p[j];
            p[j]=p[k];
            p[k]=temp2;
        }
        else if(p[j].x==p[k].x)
        {
            p[j].Data|=p[k].Data;
        }
    }
}

for (i=0;i<N;i++)
{
    LCD_DataWrite(p[i].Data);
}
}

```

```

{
    int k;
    int a=0,b=0;
    input(a,b);

    s.x=a/8;
    s.y=b-1;
    k=3-a%4;

    LCD_CursorX(s.x);LCD_CursorY(s.y);
    s.Data=0x01<<k;
    printf("DATA=%x\n",s.Data);
    return(s);
}

/*****************/
/*LCD Reset Subroutine */
/*****************/

void LCD_Reset(void)
{
    GPIO_pinWrite(hGpio,GPIO_PIN13,1); //RST = 1; 使复位位无效
    DelayXms(200);
    GPIO_pinWrite(hGpio,GPIO_PIN13,0); //RST = 0;
    DelayXms(350);

/* 当 RST 设置为低电平，且持续时间不小 LCM240128 规定的最长时间 100ms 时，
LCM240128 复位*/

    GPIO_pinWrite(hGpio,GPIO_PIN13,1); //RST = 1; 使复位位无效
    DelayXms(2);
}

/*****************/

```

```
/*LCD Function Initail Subroutine */

/***********************/

void LCD_Initial(void)

{
    LCD_CmdWrite(0x00);LCD_CmdWrite(0xCD);

    LCD_CmdWrite(0x08);LCD_CmdWrite(0x73);

    LCD_CmdWrite(0x10);LCD_CmdWrite(0x2E);

    LCD_CmdWrite(0x18);LCD_CmdWrite(0x80);

    LCD_CmdWrite(0x20);LCD_CmdWrite(0x1D);

    LCD_CmdWrite(0x30);LCD_CmdWrite(0x7F);

    LCD_CmdWrite(0x40);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0x50);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0x28);LCD_CmdWrite(0x1D);

    LCD_CmdWrite(0x38);LCD_CmdWrite(0x7F);

    LCD_CmdWrite(0x48);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0x58);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0x60);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0x70);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0x80);LCD_CmdWrite(0x33);

    LCD_CmdWrite(0x90);LCD_CmdWrite(0x1E);

    LCD_CmdWrite(0xB0);LCD_CmdWrite(0x27);

    LCD_CmdWrite(0xB8);LCD_CmdWrite(0xEF);

    LCD_CmdWrite(0xA0);LCD_CmdWrite(0x08);

    LCD_CmdWrite(0xC0);LCD_CmdWrite(0xF0);

    LCD_CmdWrite(0xD0);LCD_CmdWrite(0x20);

    LCD_CmdWrite(0xE0);LCD_CmdWrite(0x00);

    LCD_CmdWrite(0xF0);LCD_CmdWrite(0xA0);

}

/***********************/

/*LCD Cursor Set Subroutine */
```

```
*****  
void LCD_CursorX(unsigned Cursor)  
{  
    LCD_CmdWrite(0x60); LCD_CmdWrite(Cursor);  
}  
void LCD_CursorY(unsigned Cursor)  
{  
    LCD_CmdWrite(0x70); LCD_CmdWrite(Cursor);  
}  
*****  
/*LCD Windows Set Subroutine */  
*****  
  
void Windows(unsigned X)  
{  
    LCD_CmdWrite(0x10);LCD_CmdWrite(0xCE);  
    LCD_CursorX(X);LCD_CursorY(X);  
    LCD_CmdWrite(0x20);LCD_CmdWrite(0x1D);  
    LCD_CmdWrite(0x30);LCD_CmdWrite(0x7F);  
    LCD_CmdWrite(0x40);LCD_CmdWrite(0x00);  
    LCD_CmdWrite(0x50);LCD_CmdWrite(0x00);  
    LCD_CmdWrite(0x08);LCD_CmdWrite(0x73);  
    printlcd();  
    DelayXms(500);  
    LCD_CmdWrite(0x10);LCD_CmdWrite(0xEE);  
    LCD_CmdWrite(0x20);LCD_CmdWrite(0x27);  
    LCD_CmdWrite(0x30);LCD_CmdWrite(0xEF);  
    LCD_CmdWrite(0x40);LCD_CmdWrite(0x00);  
    LCD_CmdWrite(0x50);LCD_CmdWrite(0x00);  
}
```

```

*****
/*LCD Clear Screen Subroutine */

*****
void LCD_Clear(void)
{
    unsigned char REG_TMP;

    LCD_CmdWrite(0xE0);LCD_CmdWrite(0x00); //0X00 表示空白的字库代码， 即清屏
    REG_TMP = LCD_CmdRead(0xF0);

    REG_TMP &= (0xF7);

    REG_TMP |= (0x08);      // 此时，REG_TMP 的 bit3 为 1
    LCD_CmdWrite(0xF0);LCD_CmdWrite(REG_TMP);

    //将 E0 寄存器的内容写入到 DDRAM 中
    // REG[F0] Bit3 位表示重复写入 REG[E0]的资料到 DDRAM 中 Bit3=1 表示开始写入
    // Bit3=0 表示没有动作

    DelayXms(1);
}

*****
/*LCD Command Write Subroutine */

*****
void LCD_CmdWrite(unsigned Cmd_Data)
{
    int D0,D1,D2,D3,D4,D5,D6,D7;

    D0=Cmd_Data&0x01;
    D1=Cmd_Data&0x02;
    D2=Cmd_Data&0x04;
    D3=Cmd_Data&0x08;
    D4=Cmd_Data&0x10;
    D5=Cmd_Data&0x20;
    D6=Cmd_Data&0x40;
}

```

```

D7=Cmd_Data&0x80;

LCD_ChkBusy();

//Call LCD_ChkBusy to Check Busy Bit

GPIO_pinEnable(hGpio,PinIDdata);

GPIO_pinDirection(hGpio,PinIDdata,GPIO_OUTPUT);

GPIO_pinWrite(hGpio,GPIO_PIN12,D0);

GPIO_pinWrite(hGpio,GPIO_PIN11,D1);

GPIO_pinWrite(hGpio,GPIO_PIN10,D2);

GPIO_pinWrite(hGpio,GPIO_PIN9,D3);

GPIO_pinWrite(hGpio,GPIO_PIN8,D4);

GPIO_pinWrite(hGpio,GPIO_PIN3,D5);

GPIO_pinWrite(hGpio,GPIO_PIN2,D6);

GPIO_pinWrite(hGpio,GPIO_PIN0,D7);

GPIO_pinWrite(hGpio,GPIO_PIN14,0); // EN = 0 ;

GPIO_pinWrite(hGpio,GPIO_PIN15,0); // R/W =0 write status

GPIO_pinWrite(hGpio,GPIO_PIN1,1); // RS=1 To write to the instructor register

DelayTAW6();

GPIO_pinWrite(hGpio,GPIO_PIN14,1); // EN = 1;

DelayINI(); // 343.2 ns

GPIO_pinWrite(hGpio,GPIO_PIN14,0); // EN = 0 ;

DelayINI(); // 343.2 nsA

GPIO_pinWrite(hGpio,GPIO_PIN15,1); // R/W =1 read status

}

/****************************************/

/*LCD Data Write Subroutine */

/****************************************/

void LCD_DataWrite(unsigned Data_Data)

{
    int D0,D1,D2,D3,D4,D5,D6,D7;

```

```
D0=Data_Data&0x01;  
D1=Data_Data&0x02;  
D2=Data_Data&0x04;  
D3=Data_Data&0x08;  
D4=Data_Data&0x10;  
D5=Data_Data&0x20;  
D6=Data_Data&0x40;  
D7=Data_Data&0x80;  
  
LCD_ChkBusy();           //Call LCD_ChkBusy to Check Busy Bit  
  
GPIO_pinEnable(hGpio,PinIDdata);  
GPIO_pinDirection(hGpio,PinIDdata,GPIO_OUTPUT);  
GPIO_pinWrite(hGpio,GPIO_PIN12,D0);  
GPIO_pinWrite(hGpio,GPIO_PIN11,D1);  
GPIO_pinWrite(hGpio,GPIO_PIN10,D2);  
GPIO_pinWrite(hGpio,GPIO_PIN9,D3);  
GPIO_pinWrite(hGpio,GPIO_PIN8,D4);  
GPIO_pinWrite(hGpio,GPIO_PIN3,D5);  
GPIO_pinWrite(hGpio,GPIO_PIN2,D6);  
GPIO_pinWrite(hGpio,GPIO_PIN0,D7);  
  
GPIO_pinWrite(hGpio,GPIO_PIN14,0);      // EN = 0 ;  
GPIO_pinWrite(hGpio,GPIO_PIN15,0);      // R/W =0  write status  
  
GPIO_pinWrite(hGpio,GPIO_PIN1,0); // RS=0 To write to the Data register  
  
DelayTAW6();  
  
GPIO_pinWrite(hGpio,GPIO_PIN14,1); //EN = 1;  
  
DelayEN1(); //Tew >=400 ns  
  
GPIO_pinWrite(hGpio,GPIO_PIN14,0); //EN = 0;  
  
DelayTDH6();  
  
GPIO_pinWrite(hGpio,GPIO_PIN15,1); // R/W =1  read status  
  
GPIO_pinWrite(hGpio,GPIO_PIN1,1); // RS=1 To write to t
```

```

/*****************/
/*LCD Cmd Read Subroutine */
/*****************/
unsigned LCD_CmdRead(unsigned REG_Addr) //读出 REG_Addr 为地址的寄存器的内容
{
    int D0,D1,D2,D3,D4,D5,D6,D7;
    unsigned char REG_READ;
    LCD_CmdWrite(REG_Addr);

    GPIO_pinWrite(hGpio,GPIO_PIN1,1); // R/W =1   read status
    GPIO_pinWrite(hGpio,GPIO_PIN14,1); // RS=1   To read the instructor register
    DelayTAW6();

    GPIO_pinWrite(hGpio,GPIO_PIN14,1); // EN = 1;
    DelayTACC6(); //88ns  Taac6<=330ns

    D0=GPIO_pinRead(hGpio,GPIO_PIN12);
    D1=GPIO_pinRead(hGpio,GPIO_PIN11);
    D2=GPIO_pinRead(hGpio,GPIO_PIN10);
    D3=GPIO_pinRead(hGpio,GPIO_PIN9);
    D4=GPIO_pinRead(hGpio,GPIO_PIN8);
    D5=GPIO_pinRead(hGpio,GPIO_PIN3);
    D6=GPIO_pinRead(hGpio,GPIO_PIN2);
    D7=GPIO_pinRead(hGpio,GPIO_PIN0);

    D0=D0&0x01;
    D1=(D1<<1)&0x02;
    D2=(D2<<2)&0x04;
    D3=(D3<<3)&0x08;
    D4=(D4<<4)&0x10;
}

```

```
D5=(D5<<5)&0x20;
```

```
D6=(D6<<6)&0x40;
```

```
D7=(D7<<7)&0x80;
```

```
REG_READ=D0|D1|D3|D4|D5|D6|D7;
```

```
REG_READ=REG_READ&0xFF;
```

```
DelayEN1plusTACC6(); //Tew >=400 ns
```

```
GPIO_pinWrite(hGpio,GPIO_PIN14,0); //EN = 0;
```

```
DelayTOH6();
```

```
GPIO_pinWrite(hGpio,GPIO_PIN15,1); // R/W =1 read status
```

```
GPIO_pinWrite(hGpio,GPIO_PIN1,1); // RS=1 To read the instructor register
```

```
return REG_READ;
```

```
}
```

```
/********************************************/
```

```
/*LCD Data Read Subroutine */
```

```
/********************************************/
```

```
unsigned LCD_DataRead(void)
```

```
{
```

```
unsigned DATA_READ;
```

```
LCD_ChkBusy();
```

```
GPIO_pinWrite(hGpio,GPIO_PIN15,1); // R/W =1 read status
```

```
GPIO_pinWrite(hGpio,GPIO_PIN1,0); // RS=0 To read the data register
```

```
DelayTAW6();
```

```
GPIO_pinWrite(hGpio,GPIO_PIN14,1); // EN = 1;
```

```
DelayTACC6(); //88ns Taac6<=330ns
```

```
DATA_READ=GPIO_pinRead(hGpio,PinIDdata); //DATA_READ = LCD_Data;
```

```
DelayEN1plusTACC6(); //Tew >=400 ns
```

```
GPIO_pinWrite(hGpio,GPIO_PIN14,0); //EN = 0;
```

```

DelayTOH6();

GPIO_pinWrite(hGpio,GPIO_PIN15,1); // R/W=1 read status

GPIO_pinWrite(hGpio,GPIO_PIN1,1); // RS=1 To read the instructor register

return DATA_READ;

}

/***********************/

/*LCD Check Busy Subroutine */

/***********************/

void LCD_ChkBusy(void)

{

    int i;

    for(i=0;i<10000;i++)

        asm("NOP");

}

/***********************/

/*Delay Subroutine */

/***********************/

void DelayXms(int count)

{

    int i,j;

    for(i=0; i<count; i++)

        for(j=0; j<13000; j++)

            asm(" NOP ");

}

void DelayEN1(void)      // 440 ns

{

    int i;

    for(i=0;i<800;i++)

```

```
asm(" NOP ");

}

void DelayINI(void)      // 343.2 ns    DelayINI + DelayTAW6 + DelayTDH6= 440 ns
EN=0 的时间段

{
    int i;
    for(i=0;i<624;i++)
        asm(" NOP ");
}

void DelayTAW6(void)      //79.2 ns
{
    int i;
    for(i=0;i<144;i++)
        asm(" NOP ");
}

void DelayTDH6(void)      //17.6 ns
{
    int i;
    for(i=0;i<32;i++)
        asm(" NOP ");
}

void DelayEN0(void)      //422.4 ns
{
    int i;
    for(i=0;i<768;i++)
        asm(" NOP ");
```

```
}

void DelayTACC6(void)      //88 ns
{
    int i;
    for(i=0;i<160;i++)
        asm(" NOP ");
}

void DelayEN1plusTACC6(void)      //88 ns
{
    int i;
    for(i=0;i<640;i++)
        asm(" NOP ");
}

void DelayTOH6(void)      //88 ns
{
    int i;
    for(i=0;i<32;i++)
        asm(" NOP ");
}
```

## 附录 3 2FSK 的调制解体子程序

```

interrupt void
c_int6(void)
{
    short X1[40],X2[40],X3[40],X4[8];

/*Receive data and save in data_buffer.*/
reg_data = DEC6713_cpld_rget(DEC6713_INTSTAT_REG);

// Receive UARTA data
if((reg_data & 0x60) == 0x20)

{
    TempData = UART_rget(UARTA_Handle,UART_LSR);
//LSR[0]判断 FIFO 是否接收字符
if((TempData&0x01) == 0x01)

{
    for(i=0;i<FIFO_Length;i++)
    {
        RevBuffer_A[RcountA]=UART_receive_single(UARTA_Handle);
    }
    RB=RevBuffer_A[RcountA];      //RB=0x85;
    for(m=0;m<8;m++)//8 位数据依次右移,实现数字调制;
    {
        if(RB&0x01==1) //若接收到的数据为 1;
        {
            for(j=0;j<40;j++)
            {
                sbuf[j]=cos_table[k];
                k+=48;   //步长为 48, 频率为 2400Hz;
            }
        }
    }
}
}

```

```

if(k>960)

k=k%960;//取模，保证相位连续；

X1[j]=FirFDFilter(sbuf[j]);//延迟滤波；

X2[j]=DataMult(sbuf[j],X1[j]);//数组相乘；

X3[j]=FirLPFilter(X2[j]);//FIR 滤波

}

}

else //若接收的数据为 0;

{

for(j=0;j<40;j++)

{

sbuf[j]=cos_table[k];

k+=36; //步长为 36，频率为 1800Hz;

if(k>960)

k=k%960;//取模，保证相位连续；

X1[j]=FirFDFilter(sbuf[j]);

X2[j]=DataMult(sbuf[j],X1[j]);

X3[j]=FirLPFilter(X2[j]);

}

}

X4[m]=X3[32]; //任选一个数判决；

RB=RB>>1; //总共移 8 位；以上部分为一个字节的调制部分；

}

DeRevBuffer_A[RcountA]=JudgeBits(X4);

RcountA++;

if(RcountA == 160) RcountA =0;

}

else

{

UART_rset(UARTA_Handle,UART_FCR,0x02);

```

```
    },  
}  
int_flag = 0;  
return;  
}
```

## 附录 4 IIR 滤波子程序

```
#include <csl.h>
#include <csl_mcasp.h>
#include <csl_i2c.h>
#include <DEC6713.h>
#include <codec.h>
#include <IIC.h>

#define SAMPLELONG 3

short DataBuffer[1024]={0};

short R_in[1024];      /* Input samples R_in[0] most recent, R_in[127] oldest */
short DDataBuffer[1024];

/***********************/

short IIR_filter (short);

void DEC6713_AIC23_Config();

extern far void vectors();

/***********************/

IIR_filter

/***********************/

short IIR_filter(short input)
{
    short a20=0xee40;      //pass band corner frequency(FP)=2000,stop band corner
    frequency(FS)=3000,Q13

    short a10=0x2f95;
    short b20=0x24;
    short b10=0x48;
    short b00=0x24;

    short a21=0xEc81;
```

```
short a11=0x3141;  
short b21=0x94;  
short b11=0x128;  
short b01=0x94;  
  
short a22=0xe8ef;  
short a12=0x34ab;  
short b22=0x6a;  
short b12=0xd4;  
short b02=0x6a;  
  
short a23=0xE373;  
short a13=0x39e9;  
short b23=0x34d5;  
short b13=0x69a9;  
short b03=0x34d5;  
  
static short d10=0, d20=0;  
static short d11=0, d21=0;  
static short d12=0, d22=0;  
static short d13=0, d23=0;  
short d00,d01,d02,d03;  
int p00,p01,p02,p03;  
  
short isf=0x22b; //Q13  
  
short output=0;  
int y0=0,y1=0,y2=0,y3=0;  
int xn;  
int prod1, prod2, prod3, prod4, prod5;
```

```
//xn = _mpy(input,isf)>>15;  
xn = _mpy(input,isf);  
prod1 = _mpy(d20,a20);  
prod2 = _mpy(d10,a10);  
p00 = (xn + prod1 + prod2)<<3;  
p00 &=0xffff0000;  
d00 = (short)(p00>>16);  
//d00 = (short)((xn + prod1 + prod2)<<3);//shift left 3bits  
//!dk0 &=0xffff0000;  
prod3 = _mpy(d20,b20);  
prod4 = _mpy(d10,b10);  
prod5 = _mpy(d00,b00);  
y0 = prod3+prod4+prod5;  
d20 = d10;  
d10 = d00;
```

## 参 考 文 献

- [1] 陈芳烈,章燕翼.现代电信百科.北京:电子工业出版社.2007
- [2] 包东智.固定电话的发展前景及应思考的问题.电信快报,2004,No.7:41-43.
- [3] 王爱珍,董有尔.基于电话线的图像传输系统的设计与实现.现代电子技术.2003,(15):85--87.
- [4] 樊昌信,张甫翊等.通信原理.第5版.北京:国防工业出版社, 2001.
- [5] 刘威,袁小平.基于电话线的远程报警器的设计.山东电子.2004,(3):38—39.
- [6] 王振红,张凯,郝承祥.基于公共电话网的智能家居系统.控制工程.Vol.9,No.3,2002:55-57.
- [7] 皇甫忻忻.基于 DSP 的数字电话系统.科学技术与工程.Vol.7 No.5 2007(3):852—855.
- [8] 汪安民,张松灿.TMS320C6000DSP 实用技术与开发案例.北京:人民邮电出版社.2008
- [9] TMS320C67X/C67+DSP CPU and Instruction Set Reference Guide <http://www.ti.com>  
Literature Number:SPRU733A Nombor 2006-10-20.
- [10] 张雄伟,陈亮,徐光辉.DSP 芯片的原理与开发应用.北京:电子工业出版社.2002
- [11] [美] Rulph chasssing.DSP 原理及其 C 编程开发技术.王华 张建等译.北京:电子工业出版社, 2005.
- [12] Martin P. Bates. CCS C Program Function Reference.*Programming 8-bit PIC Microcontrollers in C*, 2008: 243-255
- [13] 刘元盛,李维敏,李哲英.计算机手写板的设计.电子产品世界,2003 (4): 37—39.
- [14] 通信设备接口协议手册. 北京: 人民邮电出版社.2005.
- [15] 周修理,杨方,陈萍.常见的串行通讯接口标准.电测与仪表.Vol.38,No.424, 2001:33—35.
- [16] 杨剑,贾冲,杨吉斌.基于 DSP 同步串口的异步串行通信接口的设计与实现,电子技术应用,2006(1):112-117.
- [17] Wilfried Elmenreich, Wolfgang Haidinger, Hermann Kopetz. A standard for real-time

- smart transducer interface. Computer Standards & Interfaces, Vol.28, 2006(6): 613-624
- [18] 王言章,程德福,张宁.基于 DSP 的高精度数据采集设计与实现,中国仪器仪表,2004(8):30-32. 郑江,王鹏,董云凤等.DSP 应用系统设计实践.北京: 北京航空航天大学出版社.2006.
- [19] 高红亮,张国忠,周晟.一种基于 DSP+UART 架构的语音 IC 开发系统,现代电子技术,2006(2),127:31-33.
- [20] 季昱,林俊超,余本喜.DSP 嵌入式应用系统开发典型实例.北京: 中国电力出版社.2005.
- [21] [美]Texas Instruments Incorporated.TMS320C6000 系列 DSP 编程工具与指南.田黎育,何佩琨,朱梦宇编译.北京:清华大学出版社, 2006.
- [22] TMS320C62X/67X Programmer Ccs Guide <http://www.ti.com> Literature Number:SPRU198I Revised:March 2006.
- [23] 彭启琮.DSP 技术的发展与应用[M].北京:高等教育出版社,2002.
- [24] 莫莉,谢运祥,张天保.基于 DSP 的液晶显示器的应用与实现.外国电子元器件,2007,(9):3--6.
- [25] Chul-Woo Park, Jee-Youl Ryu .Development of a new automatic gamma control system for mobile LCD applications.Displays, Vol 29, Issue 4, October 2008: 393-400.
- [26] 胡汉梅,程妮,李海军.基于 DSP 图形液晶显示器接口及程序设计[J].液晶与显示,2007,22(5):623—628.
- [27] 李宏,张家田等.液晶显示器件应用技术.北京: 机械工业出版社, 2005.
- [28] Yusof, Z.M.; Manap, N.A.; Suleiman, I.; Saleh, S.H.S.; Aspar, Z.;Implementation of wavelet codec by using Texas Instruments DSP TMS320c6701 EVM board; Signal Processing and its Applications, Sixth International, Symposium on. 2001;Volume 2, 13-16 Aug. 2001 Page(s):615 - 618 vol.2.
- [29] 潘敏,焦生杰, ,翁寅生.基于 DSP 的和 ST7920 的液晶显示模块的实现.集成电路应用,2007, (6): 48--50.
- [30] Texas Instruments Incorporated. TMS320C6000 DSP General-Purpose Input / Output (GPIO) Reference Guide [EB/OL]. <http://focus.ti.com/lit/ug/spru584a/spru584a.pdf>, 2004.

- [34] Texas Instruments Incorporated. TMS320C6000 Chip Support Library API Reference Guide [EB/OL]. <http://focus.ti.com/lit/ug/spru401j/spru401j.pdf>, 2004.
- [31] 谭浩强.C 程序设计.北京: 清华大学出版社.1999
- [32] 张怡, 张航, 唐成凯. 基于连续相位调制的 OFDM 信号研究. 西北工业大学学报, Vol.25, No.6 :834-837.
- [33] 车小林, 何晨. 多用户多输入多输出-正交频分复用系统的一种子信道分配方案. 上海交通大学学报, 2007, Vol.41, No.11:1820-1824.
- [35] Duplacy J, Louveaux J, Vandendorpe L. Interference free multi-user MIMO-OFDM [C] // International Conference on Acoustics, Speech, and Signal Processing. Philadelphia: IEEE, 2005: 1149-1152.
- [36] Cimini L J, Sollenberger N R. Peak-to-Average Power Ratio by Optimum Combination of An OFDM Using Partial Tra-nsmit Sequences. IEEE Commun Letter, 2000, 4(3): 86-88
- [37] 赵顺珍. 基于 DSP 的二进制频移键控的实现方法. 微计算机信息 (嵌入式与 SOC) , 2008, Vol.24 , No.1-2 :183-134.
- [38] 杨英强. 基于 TMS320C5402 的 2FSK 设计与实现. 现代电子技术. 2005(14), 205 : 113-115
- [39] Tea ching DSP through the practical case study of an FSK Modem.
- [40] 李琦, 杨幸芳. 一种 2FSK 解调算法的 DSP 实现. 现代电子技术, 2004(3):84-86
- [41] 黄红兵, 曹敦, 胡双红. 基于差分检波的数字化快速解调算法. 通信技术, 2007, Vol.40, No.12: 95-96.
- [42] R. Punchalard , A. Lorsawatsiri , J. Koseeyaporn , P. Wardkein , A. Roeksabutr. Adaptive IIR notch filters based on new error criteria. Signal Processing, Vol. 88, Issue 3, March 2008: 685-703
- [43] 林仙立, 高小榕. 基于分数延迟的有限冲激响应型梳状滤波器. 清华大学学报 (自然科学版) , 2004, Vol.44, No.12 :1692-1695
- [44] Xiong xiong He, Gang Li, Chunru Wan, Tao Wu .On normal realizations of digital filters with minimum roundoff noise gain. Signal Processing, Vol. 89, Issue 2, February 2009: 226-231

- [45] 薛义生,朱雪龙.多媒体通信终端组成及其实现的若干问题.电视技术,1999(5):16-21.
- [46] 周立云,胡月芬.轻松学修多功能电话机、无绳电话机.北京:电子工业出版社.2006.
- [47] 张新德.现代通信电器使用与维修技巧.北京:中国农业出版社.2004.
- [48] 张秀娟,陈新华.EDA 设计与仿真实践.北京:机械工业出版社.2002.

## 攻读硕士期间发表的论文

- [1] 第一作者. 基于 TMS320C67B 的液晶显示模块接口的实现[J]. 液晶与显示, 已录用.
- [2] 山西省太原市科技兴市项目《带手写板的聋哑人电话机》, 项目承担人.

## 致 谢

在本论文完成之际，谨向在学习和生活中给予我关心和帮助的老师、同学和亲人朋友们表示衷心的感谢。

首先，要感谢我的导师——王明泉教授和李博副教授。导师在学业上对我的悉心指导、培养和无私的帮助，学生会终生难忘。导师严谨细致的工作作风，渊博的知识，孜孜不倦的教诲和丰富的实际经验为我所钦佩。导师的谆谆教导，将在今后的人生道路中不断地给我启迪和帮助。

同时，我要感谢给予我帮助的同学。感谢教研室的各位同窗两年来对我学习和生活上的无私帮助，谨以此文，表达我对他们的无尽感激。

感谢评审论文和答辩组的各位专家老师！

谢谢所有帮助过我的人！