

南京理工大学

毕业设计说明书(论文)

作者：全香保 准考证号：014910253064

教学点：南京信息职业技术学院

专业：电子工程

题目：数字化语音存储与回放系统

指导者：李玲 副教授/高工
(姓 名) (专业技术职务)

评阅者：
(姓 名) (专业技术职务)

2012 年 4 月

毕业设计说明书（论文）中文摘要

本文介绍的数字化语音存储与回放系统基本实现传统的磁带语音录放系统功能，同时拥有较大的提升空间。其基本原理是对语音的录音与放音的数字化控制。为了更全面的掌握知识，本设计采用多个模块共同实现这一设计要求。采用单片机作为控制器件，控制模数转换和数模转换，并将模数转换后的数字信号存储于外部存储器中，通过按键控制其录放过程。为了增加语音存储时间，提高存储器的利用率，也可以采用了非失真压缩算法对语音信号进行压缩后再存储，而在回放时再进行解压缩。

关键词 数字化存储 回放 模/数转换 数/模转换

毕业设计说明书（论文）外文摘要

Title Digit-Voice Recorder and Playback System

Abstract

The digit-voice recorder and playback system that this paper introduces is the basic realization of the traditional tape voice record function. Its basic principle is the digital control for the recorder and playback of voice. In order to grasp and apply the knowledge learned more fully, this design uses multiple modules together to achieve the design requirements. This design using the microcontroller as a control device, control of analog to digital conversion, digital to analog conversion, store the digital signal after the analog to digital conversion in external memory, using the keys to control the playback process. For increase, to increase pronunciation store time and raise utilization ratio of the memory, it adopts non-distorted to compress algorithm before storing to pronunciation signal and to decompress in the playback.

Keywords Digital store Playback A/D Convert D/A Convert

目录

1 引言	1
2 总体方案设计	2
3 各模块硬件设计	5
3.1 话筒前置放大电路	5
3.2 带通滤波器设计	6
3.3 AT89S52 单片机基本电路	7
3.4 模数转换模块	13
3.5 数模转换模块	15
3.6 外部存储模块	17
3.7 功放电路设计	19
3.8 按键与显示模块	21
3.8.1 ZLG7289B1 芯片介绍	21
3.8.2 ZLG7289B 与其它部分连接图	23
3.9 供电电路模块	24
3 软件设计	26
结论	28
致谢	29
参考文献	30
附录	31
附录一 程序源代码	31
附录二 原理图及实物图	44

1 引言

传统的磁带语音录放系统因其体积大、使用不便,在电子与信息处理的使用中受到许多限制。本文提出的体积小,功耗低的数字化语音存储与回放系统将完全可以替代它。数字化语音存储与回放系统的基本原理是对语音的录音与放音的数字化控制。其中,关键技术在于,为了增加语音存储时间,提高存储器的利用率,采用了非失真压缩算法对语音信号进行压缩后再存储,而在回放时再进行解压缩,同时,对输入语音信号进行数字滤波以抑制杂音和干扰,从而确保了语音回放的可靠质量。

本系统能够对语音信号分别进行数据的采集直存直取,欠抽样采样和自相似增量调制等三种方法,完成了对语音信号的存储与回放。前置放大、滤波以及电平移位电路将语音信号控制在 A/D 转换器采样控制范围内以保证语音信号采样不失真。带通滤波器合理的通带范围有效的滤除了带外噪声,减小了混叠失真。后置带通滤波器用于滤除 D/A 转换产生的高频噪声以保证回放时音质清晰,无明显失真。

本系统设计主要分为以下几个模块:声音采集模块、带通滤波模块、A/D 转换模块、数据存储模块、D/A 转换模块、按键选择模块、放大器模块。声音采集模块用于外部语音信号,带通滤波模块作用是将声音转换后的电信号进行滤波,数据存储模块用于存储数字化处理后声音信号的数据,D/A 转换模块将数字信号转换为模拟信号输出,音频放大模块则是将采集的信号最终进行回放以检验系统整体性能,按键选择模块则是对录、放音、数据分段存取等功能进行选择。

2 总体方案设计

数字化的语音存储与回放系统完成将语音信号转化为电信号，经放大、滤波处理后通过 A/D 转换器转化为数字信号，再将数字化得语音信号存放在大容量的外部存储器之中。回放时，将数字化的语音信号经过 D/A 转换器转化为模拟信号，经过滤波放大后驱动扬声器产生声音。此过程主要由两个按键控制，即录音与放音按键，同时用数码管显示录放信息。系统整体框图如图 2 所示。

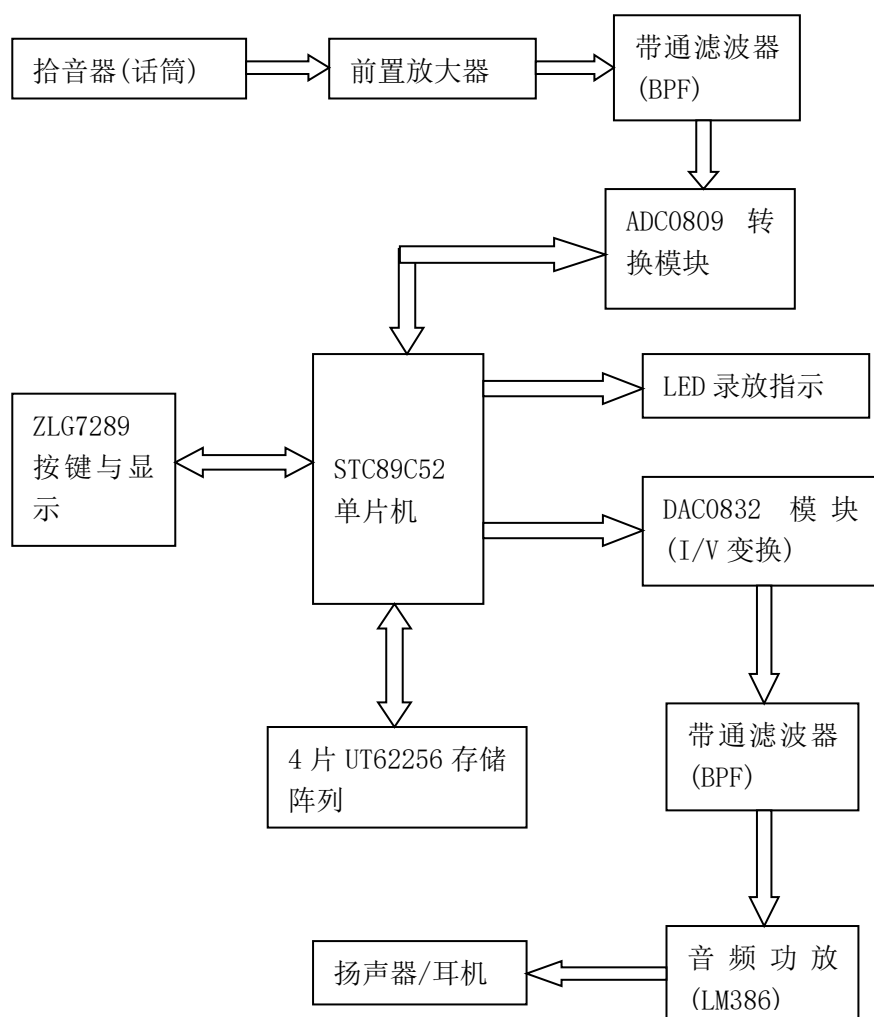


图 2 系统整体框图

从以上思路可以看出数字化语音存储与回放系统在硬件电路主要由运放电路、滤波电路、A/D 转换电路、微处理电路、大容量存储器电路、D/A 转换电路和运放电路等部分组成。

(1) 微处理器选择。本设计选用 PDIP 封装 AT89S52 单片机来控制数字语音存储与回放。AT89S52 单片机是一种低功耗、高性能 CMOS 8 位微控制器，具有 8K 在系统可编程 Flash 存储器。使用 Atmel 公司高密度非易失性存储器技术制造，与工业 80C51 产品指令和引脚完全兼容。片上 Flash 允许程序存储器在系统可编程，亦适于常规编程器。在单芯片上，拥有灵巧的 8 位 CPU 和在系统可编程 Flash，使得 AT89S52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

(2) A/D 转换模块选择。根据题目要求采样频率 $f_s=8\text{KHZ}$ ，字长=8 位，可选择转换时间不超过 125 μs 的八位 A/D 转换芯片。目前常用的 A/D 转换实现的方法有多种，鉴于转换速度的要求，我们采用 A/D 转换芯片 ADC0809。ADC0809 是带有 8 位 A/D 转换器、8 路多路开关以及微处理机兼容的控制逻辑的 CMOS 组件。它是逐次逼近式 A/D 转换器，可以和单片机直接接口。

(3) D/A 转换模块选择。D/A 转换芯片的作用是将存储的数字语音信号转换为模拟语音信号，由于一般的模拟转换器都能达到 $1\mu\text{s}$ 的转换速率，足够满足题目的要求，故我们在此选用了通用 D/A 转换器 DAC0832。

(4) 数据存储器选择。当采样频率 $f_s=8\text{KHZ}$ ，字长为 8 位时，一秒钟的语音需要 8K 字节的存储空间，本设计要求存储 20 秒的语音信号，则存储器至少需要有 $8\text{k}\times 20$ 容量，即 160K 容量。在这里我们选用 4 片 32K 的低功耗静态 RAM 存储器 62256，总的存储容量为 128K。可以基本满足存储需要，当然若在存储时采用数据压缩算法其可存储语音时间将更长。此设计中没有采取此算法。

(5) 话筒前置放大电路器件选择。拾音器输出的毫伏信号实测其范围约为 $20\sim 25\text{mV}$ ，此电信号太小不能够进行采样，后级 A/D 转换输入信号的动态范围为 $0\sim 5\text{V}$ ，语音信号的范围与采样范围的比较得出放大器的放大倍数应为 200 倍左右，所以为了将从拾音器获得的微弱语音信号放大，本系统中才用两个相同拾音器进行信号输入，将它们背对背安装，前置放大电路由一级差分放大电路和一级增益可调反相放大电路组成。采用低噪声双运算放大器 NE5532。

(6) 带通滤波器运放选择。声音信号经动圈拾音器转有源滤波器换成电压信号，通过前级放大，在对其进行数据采集之前，有必要经过带通滤波器除带外杂波，选定该滤波器的通带范围为 $300\text{Hz}\sim 3.4\text{KHz}$ ，其作用是：

- a) 保证300—3400Hz的语音信号不失真的通过滤波器；
- b) 滤除通带外的低频信号，以减少带外功频等分量的干扰，大大减少噪声影响；
- c) 该下限频率可下延到270Hz左右；便于滤除通带外的高次谐波，以减少因8kHz采样率而引起的混叠失真，根据实际情况，该上限频率可在2700Hz左右，带通滤波器按品质因数Q的大小为窄带滤波器（ $Q > 10$ ）和带通滤波器（ $Q < 10$ ）两种，本题中，上限频率 $f_h=3400\text{Hz}$ ，通带滤波器中心频率 f_0 与品质因数Q分别为：

$$f_0 = \sqrt{f_h f_l} = \sqrt{3400 * 300} \text{HZ} = 1010 \text{HZ}$$

$$Q = f_0 / BW = f_0 / (f_h - f_l) = 1010 / (3400 - 300) = 0.326$$

显然， $Q < 10$ ，故该带通滤波器为宽带带通滤波器。宽带带通滤波器由高通和低通滤波器级联构成。本设计采用TL084四运放放大器，该TL084 是四输入运算放大器与高速结型场效应管的结合良好的匹配，高电压的J - FET 和双极晶体管电路在一个单片tegrated 回路范围。

（7）功放模块选择。本设计选用 2025 立体声功放模块。2025 是一个完整的十六脚双排塑料封装音频放大器，它是为轻便的盒式录音机播放器和收音机而设计的。

（8）电源模块选择。本设计使用的单片机，运放电路分别使用+5V、0V 和+12V、-12V 电压，因此电源模块需要提供这些电压，选用 LM7805 三端 1.5A 正电源稳压电路实现+5V 电压供给，选用 LM7812 提供+12V 电压，选用 LM7912 提供-12V 电压。

（9）显示及按键电路器件的选择。显示用两位共阴极数码管即可，按键可用 4 个独立按键分别控制录音、放音、数据上传和下传操作。因为外围器件较多，单片机 I/O 口可用数量有限，为了节省 I/O 口，我们选用广州周立功单片机发展有限公司自行设计的数码管显示驱动及键盘扫描管理芯片 ZLG7289B1。

3 各模块硬件设计

3.1 话筒前置放大电路

拾音器是一种声传感器，声传感器是把外界声场中的声信号转换成电信号的传感器。拾音器包括拾音头和音臂等附件，其换能装置主要有压电式、电磁式、电容式以及半导体等。唱针耦合在线圈上的称动圈式，耦合在磁钢上的称动磁式。此外，也有将唱针耦合在衔铁上的称为动铁式，也称可变磁阻式。在本设计中采用动圈式拾音器。

拾音器输出的毫伏信号实测其范围约为 20~25mV，此电信号太小不能够进行采样，后级 A/D 转换输入信号的动态范围为 0~5V，语音信号的范围与采样范围的比较得出放大器的放大倍数应为 200 倍左右，本系统中采用两个相同拾音器进行信号输入，将它们背对背安装，前置放大电路由一级差分放大电路和一级增益可调反相放大电路组成。本设计采用低噪声双运算放大器 NE5532。差分放大电路的增益为 $A_1 = -R_6 / R_3 \times (v_{s1} - v_{s2})$ 。反相放大器的增益为 $A_2 = -PR_1 / R_7$ 。

声音的拾取选用两个特性基本相同的话筒，将它们背对背的安装，假设声源到达两拾音器的距离分别为 L1 和 L2，背景声音（噪声）到达两个拾音器的距离分别为 L3 和 L4。由于生源离话筒的距离相对较近， $L1 \neq L2$ ，生源在话筒上产生语音信号属于差分信号，通过差分电路得到放大；而背景声离话筒的距离相对较远，可以认为 $L3 \approx L4$ ，因此，背景声在话筒上产生的信号对差分放大电路来说相当于共模信号，从而被有效地抑制。

差分放大电路如图 3.1.1 所示。

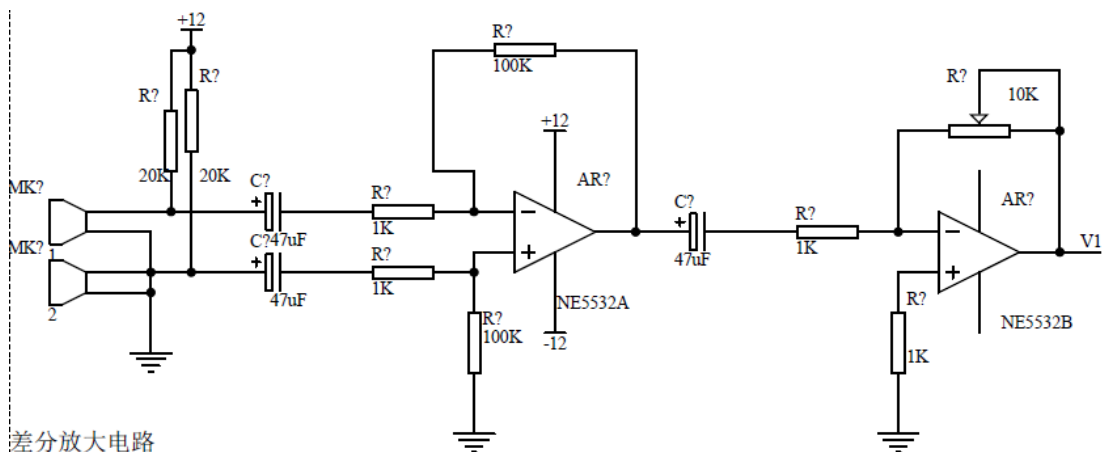


图 3.1.1 前置差分放大电路

3.2 带通滤波器设计

此处设计的带通滤波器使用于信号输入端和信号输出端。

前文已经讲过本设计采用 TL084 四运放放大器作为带通滤波器的主要器件，且经计算的 $Q < 10$ ，故该带通滤波器为宽带带通滤波器。宽带带通滤波器由高通和低通滤波器级联构成。

(1) 4 阶低通滤波器的原理图如图 2.2.1 所示，由两级 2 阶多重反馈低通滤波器级联而成。主要指标：通带增益 $A_o=1$ ，截止频率 $f_c=f_h=3.4\text{Hz}$ ，选择 $Q_1=0.541$ ， $Q_2=1.306$ 。主要计算如下：

选基准电容 C_o 为 2200pF ，则

$$\text{基准电阻 } R_o = 1 / (2\pi f_c C_o) = 21.29 \text{ K}\Omega$$

$$C_4 = 4Q_1^2(1 + A_o)C_o = 5151\text{pF}，\text{取标称值 } 5100\text{pF}，$$

$$C_5 = C_o = 2200\text{pF}，$$

$$R_9 = R_o / (2Q_1 A_o) = 19.67\text{k}\Omega，\text{取标称值 } 20\text{k}\Omega，$$

$$R_{10} = A_o R_9 = 19.67\text{k}\Omega，\text{取标称值 } 20\text{k}\Omega，$$

$$R_{11} = R_o / [2Q_1(1 + A_o)] = 9.83\text{k}\Omega，\text{取标称值 } 10\text{k}\Omega，$$

同样有：

$$C_6 = 4Q_2^2(1 + A_o)C_o = 0.0313\mu\text{F}，\text{取标称值 } 0.033\mu\text{F}，$$

$$C_7 = C_o = 2200\text{pF}，$$

$$R_{12} = R_o / (2Q_2 A_o) = 8.15\text{k}\Omega，\text{取标称值 } 8.2\text{k}\Omega，$$

$$R_{13} = A_o R_{12} = 8.2\text{k}\Omega，\text{取标称值 } 8.2\text{k}\Omega，$$

$$R_{14} = R_o / [2Q_2(1 + A_o)] = 4.07\text{k}\Omega，\text{取标称值 } 3.9\text{k}\Omega。$$

低通滤波器电路如图 3.2.1 所示

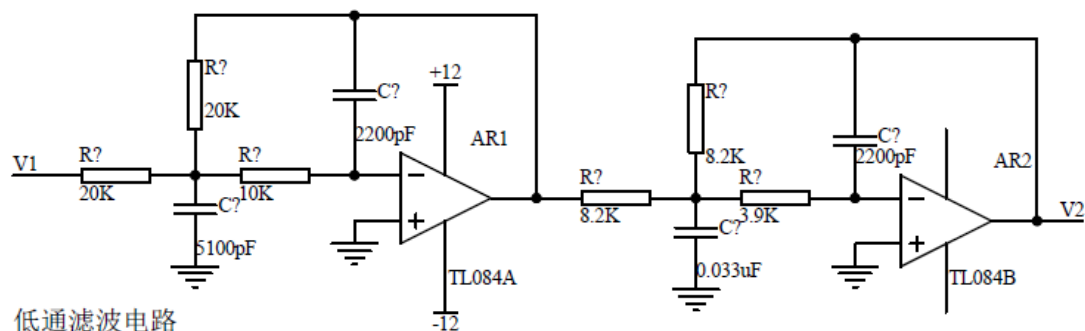


图 3.2.1 低通滤波器电路图

(2) 高通滤波器原理图如图 2.2.2 所示，有 TL082 构成 4 阶高通滤波器，通带增益 $A_0 = 1$ ，截止频率 $f_c = f_L = 300\text{Hz}$ ，第三极的 Q_3 为 0.451，第四级 Q_4 为 1.306。主要参数计算如下：

电容 $C_8 \sim C_{13} = C_0 = 0.033\mu\text{F}$ ， $R_0 = 1 / (2\pi f_c C_0) = 16.08\text{k}\Omega$ ，

$R_{15} = R_0 / [Q_3(2 + 1/A_0)] = 11.88\text{k}\Omega$ ，取标称值 $12\text{k}\Omega$ ，

$R_{16} = R_0 / [Q_3(1 + 2A_0)] = 21.75\text{k}\Omega$ ，取标称值 $22\text{k}\Omega$ ，

$R_{17} = R_0 / [Q_4(2 + 1/A_0)] = 4.10\text{k}\Omega$ ，取标称值 $3.9\text{k}\Omega$ ，

$R_{18} = R_0 / [Q_4(1 + 2A_0)] = 62.98\text{k}\Omega$ ，取标称值 $3.9\text{k}\Omega$ ，

高通滤波器图如图 3.2.2

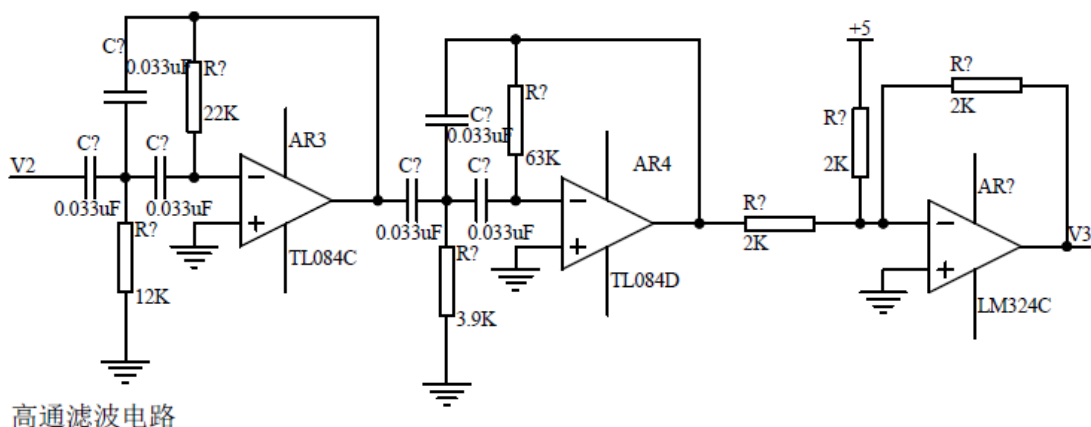


图 3.2.2 高通滤波器电路图

3.3 AT89S52 单片机基本电路

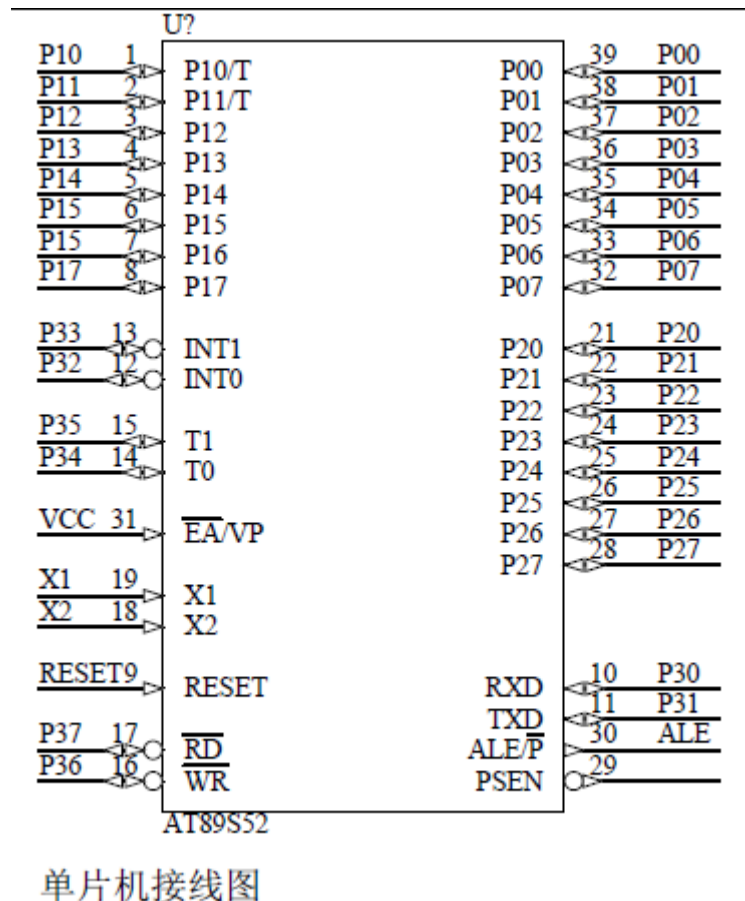
AT89S52 是一种低功耗、高性能 CMOS 8 位微控制器，具有 8K 在系统可编程 Flash 存储器。使用 Atmel 公司高密度非易失性存储器技术制造，与工业 80C51

产品指令和引脚完全兼容。片上 Flash 允许程序存储器在系统可编程，亦适于常规编程器。在单芯片上，拥有灵巧的 8 位 CPU 和在系统可编程 Flash，使得 AT89S52 为众多嵌入式控制应用系统提供高灵活、超有效的解决方案。

AT89S52 具有以下标准功能：8k 字节 Flash，256 字节 RAM，32 位 I/O 口线，看门狗定时器，2 个数据指针，三个 16 位定时器/计数器，一个 6 向量 2 级中断结构，全双工串行口，片内晶振及时钟电路。另外，AT89S52 可降至 0Hz 静态逻辑操作，支持 2 种软件可选择节电模式。空闲模式下，CPU 停止工作，允许 RAM、定时器/计数器、串口、中断继续工作。掉电保护方式下，RAM 内容被保存，振荡器被冻结，单片机一切工作停止，直到下一个中断或硬件复位为止。

（1）管脚介绍

AT89S52 引脚及管脚连线定义如下图所示



8 个 TTL 逻辑电平。对 P0 端口写“1”时，引脚用作高阻抗输入。当访问外部程序和数据存储器时，P0 口也被作为低 8 位地址/数据复用。在这种模式下，P0 具有内部上拉电阻。在 flash 编程时，P0 口也用来接收指令字节；在程序校验时，输出指令字节。程序校验时，需要外部上拉电阻。

P1 口：P1 口是一个具有内部上拉电阻的 8 位双向 I/O 口，p1 输出缓冲器能驱动 4 个 TTL 逻辑电平。对 P1 端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流（IIL）。此外，P1.0 和 P1.2 分别作定时器/计数器 2 的外部计数输入（P1.0/T2）和时器/计数器 2 的触发输入（P1.1/T2EX），具体如下表所示。在 flash 编程和校验时，P1 口接收低 8 位地址字节。

引脚号	第二功能
P1.0	T2（定时器/计数器 T2 的外部计数输入），时钟输出
P1.1	T2EX（定时器/计数器 T2 的捕捉/重载触发信号和方向控制）
P1.5	MOSI（在系统编程用）
P1.6	MISO（在系统编程用）
P1.7	SCK（在系统编程用）

表 3.3.1 P1 口第二功能

P2 口：P2 口是一个具有内部上拉电阻的 8 位双向 I/O 口，P2 输出缓冲器能驱动 4 个 TTL 逻辑电平。对 P2 端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流（IIL）。在访问外部程序存储器或用 16 位地址读取外部数据存储器（例如执行 MOVX @DPTR）时，P2 口送出高八位地址。在这种应用中，P2 口使用很强的内部上拉发送 1。在使用 8 位地址（如 MOVX @RI）访问外部数据存储器时，P2 口输出 P2 锁存器的内容。在 flash 编程和校验时，P2 口也接收高 8 位地址字节和一些控制信号。

P3 口：P3 口是一个具有内部上拉电阻的 8 位双向 I/O 口，p2 输出缓冲器能驱动 4 个 TTL 逻辑电平。对 P3 端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。作为输入使用时，被外部拉低的引脚由于内部电阻的原因，将输出电流（IIL）。P3 口亦作为 AT89S52 特殊功能（第二功能）使用，

如下表所示。在 flash 编程和校验时，P3 口也接收一些控制信号。

引脚号	第二功能
P3.0	RXD（串行输入）
P3.1	TXD（串行输出）
P3.2	INT0（外部中断 0）
P3.3	INT0（外部中断 0）
P3.4	T0（定时器 0 外部输入）
P3.5	T1（定时器 1 外部输入）
P3.6	WR（外部数据存储器写选通）
P3.7	RD（外部数据存储器写选通）

表 3.3.2 P3 口第二功能

RST：复位输入。晶振工作时，RST 脚持续 2 个机器周期高电平将使单片机复位。看门狗计时完成后，RST 脚输出 96 个晶振周期的高电平。特殊寄存器 AUXR(地址 8EH)上的 DISRTO 位可以使此功能无效。DISRTO 默认状态下，复位高电平有效。

ALE/PROG：地址锁存控制信号（ALE）是访问外部程序存储器时，锁存低 8 位地址的输出脉冲。在 flash 编程时，此引脚（PROG）也用作编程输入脉冲。在一般情况下，ALE 以晶振六分之一的固定频率输出脉冲，可用来作为外部定时器或时钟使用。然而，特别强调，在每次访问外部数据存储器时，ALE 脉冲将会跳过。如果需要，通过将地址为 8EH 的 SFR 的第 0 位置“1”，ALE 操作将无效。这一位置“1”，ALE 仅在执行 MOVX 或 MOVC 指令时有效。否则，ALE 将被微弱拉高。这个 ALE 使能标志位（地址为 8EH 的 SFR 的第 0 位）的设置对微控制器处于外部执行模式下无效。

PSEN：外部程序存储器选通信号（PSEN）是外部程序存储器选通信号。当 AT89S52 从外部程序存储器执行外部代码时，PSEN 在每个机器周期被激活两次，而在访问外部数据存储器时，PSEN 将不被激活。

EA/VPP：访问外部程序存储器控制信号。为使能从 0000H 到 FFFFH 的外部程序存储器读取指令，EA 必须接 GND。为了执行内部程序指令，EA 应该接 VCC。在 flash 编程期间，EA 也接收 12 伏 VPP 电压。

XTAL1:振荡器反相放大器和内部时钟发生电路的输入端。

XTAL2:振荡器反相放大器的输出端。

（2）主要性能指标

- 与 MCS-51 兼容
- 4K 字节可编程闪烁存储器
- 三级程序存储器锁定
- 128*8 位内部 RAM
- 32 可编程 I/O 线
- 两个 16 位可编程定时器/计数器
- 5 个中断源
- 低功耗的闲置和掉电模式
- 片内振荡器和时钟电路, 时钟频率 1.2—12MHz; 可有时钟输出
- 有强的位寻址\位处理能力

（3）CPU时钟电路

AT89S52 单片机有一个用于构成内部振荡器的反相放大器, XTAL1 和 XTAL2 分别是放大器的输入、输出端。石英晶体和陶瓷谐振器都可以用来一起构成自激振荡器, 在 XTAL1 和 XTAL2 两端跨接晶体或陶瓷振荡器, 就构成了稳定的自激振荡器, 其发出的脉冲直接送入内部时钟发生器, 见图 3.3.2 所示:

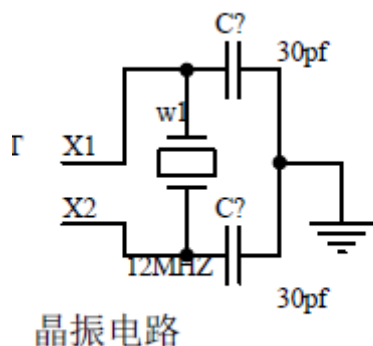


图3.3.2 AT89S52的时钟电路

外接晶振时, C1、C2 值通常选择为 30pF 左右; 外接陶瓷振荡器时, C1、C2 约为 47pF。C1、C2 对频率有微调作用, 震荡频率范围是 1.2—12MHz。本设计采用 12MHz 的外部晶振。

（4）复位电路

AT89S52 通常采用上电自动复位和按键手动复位两种方式。上电复位电路在通电瞬间，在 RC 电路充电过程中，RST 端出现正脉冲，从而使单片机复位。按键手动复位又分为按键电平复位和按键脉冲复位，按键电平复位是将复位端通过电阻与 Vcc 相连，按键脉冲复位是利用 RC 微分电路产生正脉冲来达到复位的目的。

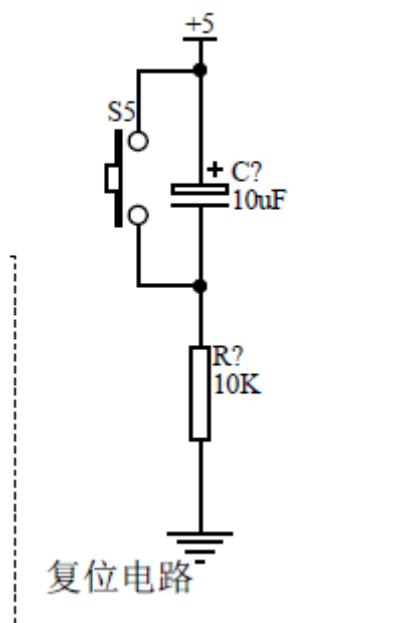


图3. 3. 3 单片机复位电路

（5）ISP下载口

AT89S52单片机只能用ISP下载口下载，图为

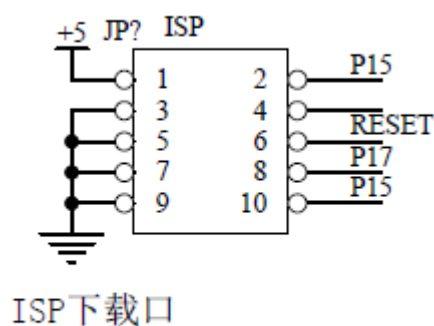


图3. 3. 4 ISP下载扣

3.4 模数转换模块

ADC0809 是美国 National Semiconductor 公司推出的一款中低速廉价 8 位逐次逼近型 AD 转换芯片，具有八路输入通道，可轮流对外部八路模拟信号进行采样转换；其典型转换时间为 100us，最大能达到 11KSPS 的采样率；总的不可调误差为 $\pm 1\text{LSB}$ ，单一 5V 电源供电，低功耗——15mW，广泛用于中低速采样系统当中。由于 ADC0809 芯片内有三态输出缓冲电路，因而可直接与单片机的数据总线相连，无须附加逻辑接口电路。另外，由于 ADC0809 与 CMOS 和 TTL 兼容，因而可构成简单的数据采集最小系统。

（1）管脚介绍

ADC0809 为 28 脚双列直插式封装，其引脚分布如图 3.4.1 所示。各主要引脚功能如下：

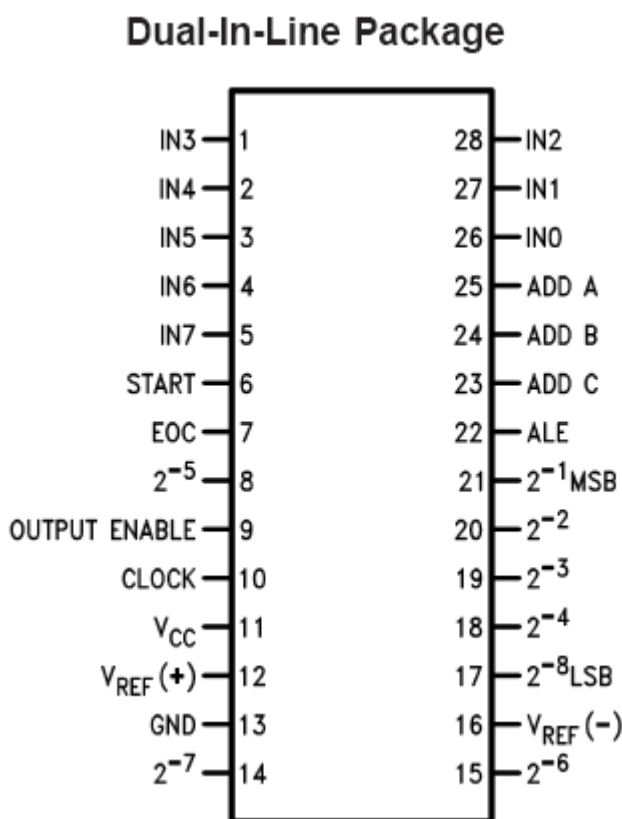


图 3.4.1 ADC0809 管脚图

V_{CC} ：芯片工作的电源电压端，电压范围为 4.5~5.5V，推荐使用的典型电压值为 5V。

GND：模拟和数字接地端。

REF+和 REF-：正和负的基准电压输入端，范围为 $0 \sim V_{CC}$ 。ADC0809 内部不含电压基准电路，需要外部提供。REF+的值要大于 REF-的值，应用时常用的方法是直接将 REF-接模拟地。

IN0~IN7：模拟电压信号输入端，电压范围为 $0 \sim (V_{REF+}) - (V_{REF-})$ 。通过可编程操作模拟多路开关选择输入的有效通道。

START：转换启动信号输入端。当开端出现一个 100ns 的脉冲时启动 AD 转换。

ADDA、ADDB、ADDC：通道地址输入端。八个模拟信号输入通道的输入选通地址由这三个引脚确定。

ALE：输入地址锁存端。锁存 ADDA、ADDB、ADDC 输入的地址码。

EOC：转换结束信号输出端。当内部进行完一次 AD 转换时该引脚输出高电平，此引脚也可作为中断信号触发单片机中断，是单片机进行相应处理。

OE：数据输出使能端。当进行完一次 AD 转换时，置位该引脚则数据经三态门输出至 8 位数据输出端，此时单片机可以读取 AD 转换结果。

CLOCK：外部时钟输入端。ADC0809 没有内部时钟振荡电路，需外加时钟信号方能正常工作，典型值为 640KHz。本系统加的是 500KHz。

LSB~MSB：8 位数据输出端。AD 转换的结果最终经三态门输出至这 8 个引脚。

(2) ADC0809与单片机连接图

ADC0809与AT89S52单片机连接图如图3.4.2所示

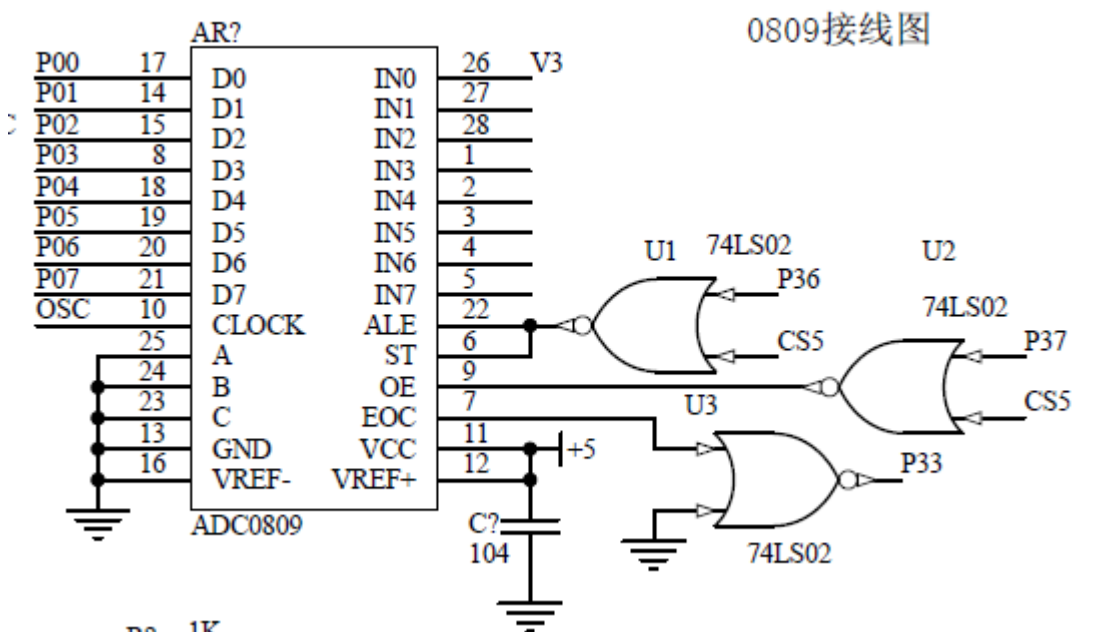


图 3.4.2 ADC0809 与 AT89S52 单片机连接图

3.5 数模转换模块

DAC0832 是采用 CMOS 工艺制成的单片直流输出型 8 位数/模转换器。旨在直接与 8080, 8048, 8085, Z80 及其他通用的微型处理器进行相接。存储的硅铬 R-2R 电阻梯形网络将参考电流分开, 并为电路提供合适的温度处理特性 (全范围最大线性温度误差的 0.05%)。电路利用 CMOS 电流开关和控制逻辑来取得最少的电能损耗和最小的输出泄露电流误差。特殊的电路也能提供 TTL 逻辑输入电压的水平兼容。

(1) DAC0832 管脚介绍

DAC0832 管脚图如 3.5.1 所示

D0~D7: 8 位数据输入线, TTL 电平, 有效时间应大于 90ns (否则锁存器的数据会出错);

ILE: 数据锁存允许控制信号输入线, 高电平有效;

CS: 片选信号输入线 (选通数据锁存器), 低电平有效;

WR1: 数据锁存器写选通输入线, 负脉冲 (脉宽应大于 500ns) 有效。由 ILE、CS、WR1 的逻辑组合产生 LE1, 当 LE1 为高电平时, 数据锁存器状态随输入数据线变换, LE1 的负跳变时将输入数据锁存;

XFER: 数据传输控制信号输入线, 低电平有效, 负脉冲 (脉宽应大于 500ns) 有效;

WR2: DAC 寄存器选通输入线, 负脉冲 (脉宽应大于 500ns) 有效。由 WR2、XFER 的逻辑组合产生 LE2, 当 LE2 为高电平时, DAC 寄存器的输出随寄存器的输入而变化, LE2 的负跳变时将数据锁存器的内容打入 DAC 寄存器并开始 D/A 转换。

IOUT1: 电流输出端 1, 其值随 DAC 寄存器的内容线性变化

IOUT2: 电流输出端 2, 其值与 IOUT1 值之和为一常数

Rfb: 反馈信号输入线, 改变 Rfb 端外接电阻值可调整转换满量程精度

Vcc: 电源输入端, Vcc 的范围为 +5V~+15V

VREF: 基准电压输入线, VREF 的范围为 -10V~+10V

AGND: 模拟信号地

DGND: 数字信号地

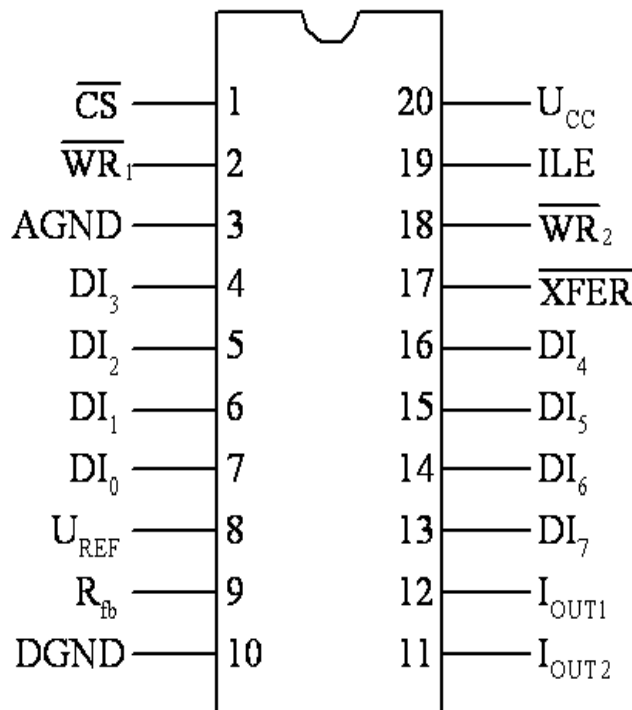


图 3.5.1 DAC0832 管脚图

(2) 主要性能指标

- 电流设置时间：1 μ s
- 分辨率：8位
- 线性度：8, 9或者10位（保证温度）
- 低功耗：20 mW
- 单电源提供：直流5-15V

(1) DAC0832 与单片机连接图

DAC0832 与 AT89S52 单片机连接图如图 3.5.2 所示

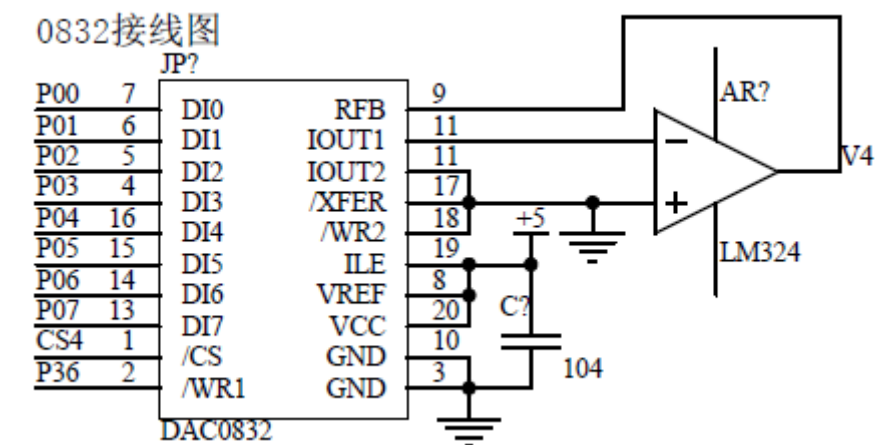


图 3.5.2 DAC0832 与 AT89S52 单片机连接图

3.6 外部存储模块

在数字化语音存储与回放的设计中可用 4 片 UT62256 采用分页存储技术存储器来存储语音信号。UT62256 具有在线可擦写、读写速度快、信息保存可靠、存储容量大等优点，每片的容量为 32K 字节。由于 89C52 一般能寻址 64K 字节。所以需要利用端口进行地址译码，本系统中另加四根线（P1.0、P1.1、P1.2、P1.3）控制 74HC138 进行地址译码输出，使寻址空间扩展到 128K 字节。UT62256 读取时间仅为 70ns，单一+5V 电源，低功耗：待机为 1uA（LL 系列），启动工作为 30~40mA。输入输出全兼容 CMOS 和 TTL 电路。但是 UT62256 是 SRAM 存储器，掉电后数据会丢失，为了保持数据可以附加后备电池供电（UT62256 只需 2V 电压即可维持片内数据），如果不加后备电源只有系统上电后再重新录制。

（1）UT62256 管脚介绍

UT62256 引脚分布如图 3.6.1 所示：

A0 - A14：地址总线 (Address)
D0 /D7：输入/输出口 (input/output)
CS：端口选择 (Chip select)
WE：输入始能 (Write enable)
OE：输出始能 (Output enable)
VCC：电源始能 (Power supply)
VSS：接地 (Ground)

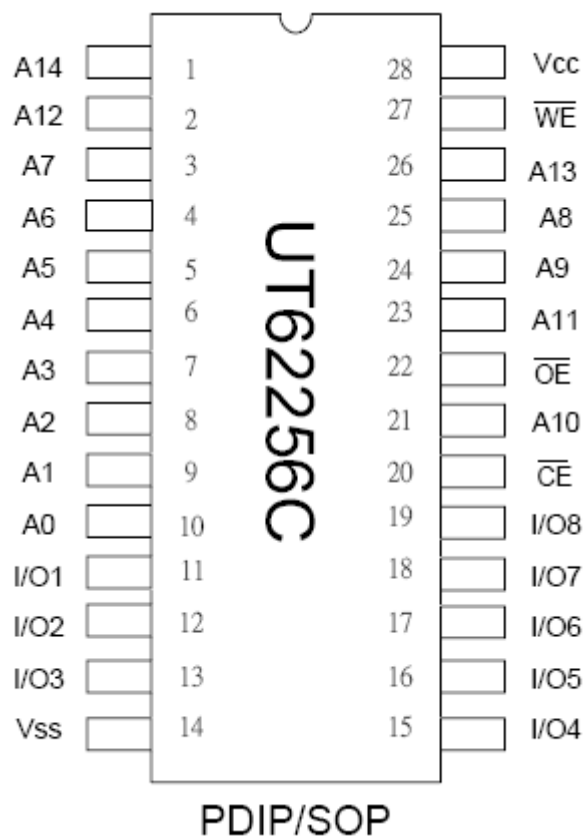


图 3. 6. 1

(2) UT62256C 与单片机连接图

UT62256C 与 AT89S52 单片机连接电路如图 3. 6. 2 所示

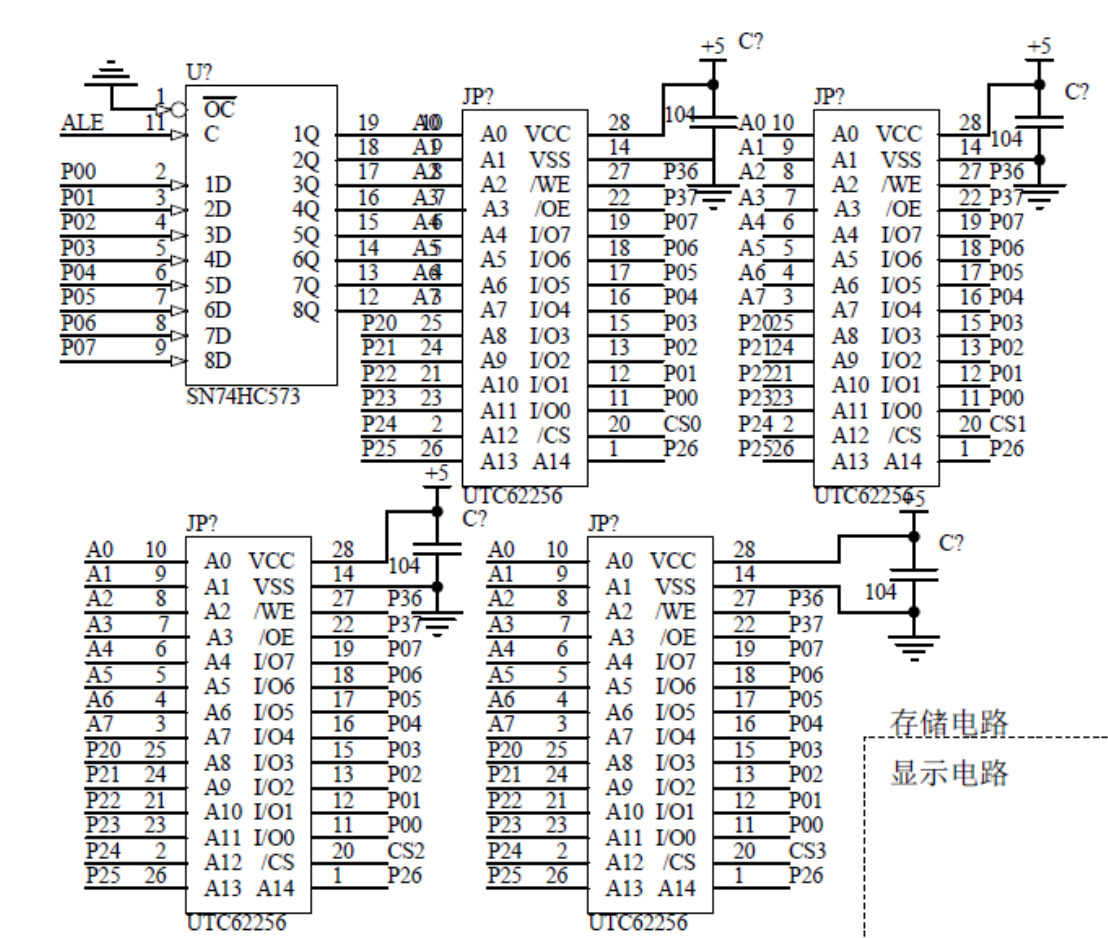


图 3.6.2 62256 接线图

3.7 功放电路设计

TEA2025 是欧洲生产的双声道功率放大集成电路，该电路具有声道分离度高、电源接通时冲击噪声小、外接元件少，最大电压增益可由外接电阻调节等特点，应用于袖珍式或便携式立体声音响系统中作功率放大。

(1) TEA2025 管脚介绍

TEA2025 引脚图如 3.7.1 所示

BRIDGE: BTL 辅助输出，双声道时悬空

OUT1/OUT2: 功放电路 1/2 信号输出端

BOOT1/BOOT2: 功放电路 1/2 自举端

GND: 功放电路接地线

FEEDBACK: 功放电路负反馈元件端

IN1/IN2: 功放电路 1/2 信号输入端

SVR: 纹波滤波元件连接端

+VS: 工作电源电压输入端

GND: 接地线端

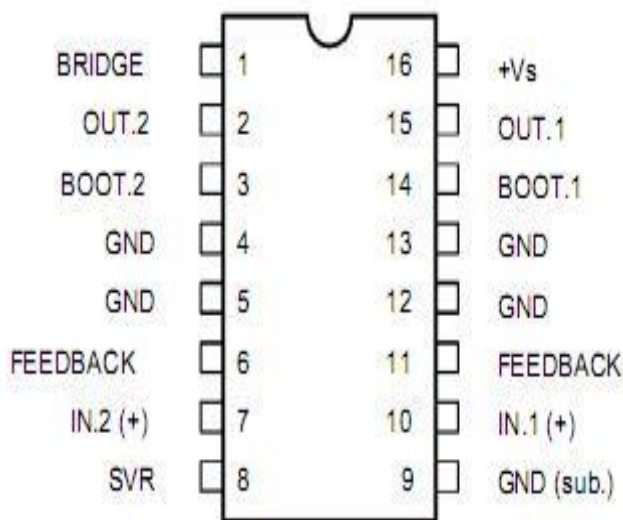


图 3.7.1 TEA2025 管脚图

（2）主要性能指标

- 工作电源电压范围为 3~12V。
- 其输出功率由使用电源电压和负载而定
- $P=0.1W \times 2$ ($V_{CC}=3V$, $R_L=4\Omega$)
- $P=1W \times 2$ ($V_{CC}=6V$, $R_L=2\Omega$)
- $P=2.3W \times 2$ ($V_{CC}=9V$, $R_L=4\Omega$)。
- 适用于单声道桥式(BTL)或立体声线路两种工作状态。
- 它还具有外接元件少, 声道分离度高, 爆破噪声少, 电压增益可由外接电阻调节等特点。
- 采用双列直插 16 脚塑料封装(DIP16)。

（2） TEA2025 功放电路图

TEA2025 功放电路图如图 3.7.2 所示

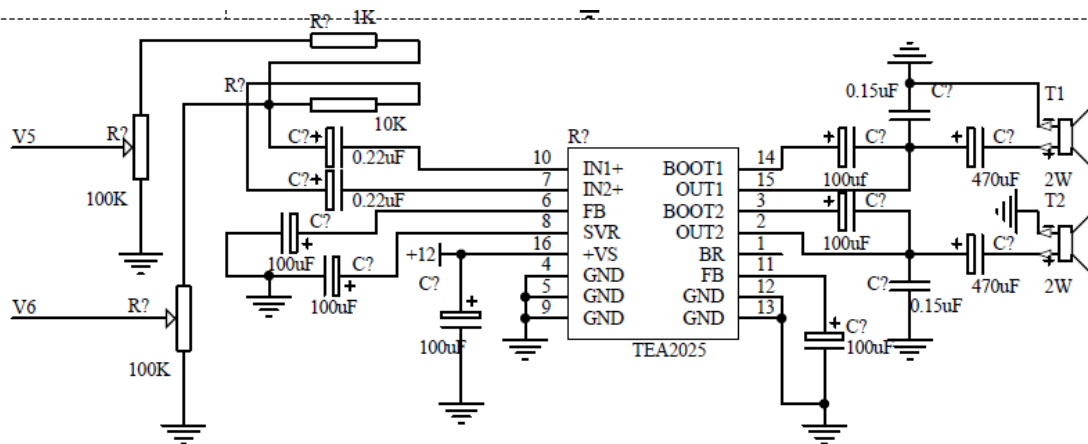


图 3.7.2 TEA2025 功放电路图

3.8 按键与显示模块

本设计按键扫描及数码管显示驱动采用广州周立功单片机发展有限公司自行设计的数码管显示驱动及键盘扫描管理芯片 ZLG7289B1。显示采用两位共阴极数码管，按键采用四个独立按键。

3.8.1 ZLG7289B1 芯片介绍

ZLG7289B 是广州周立功单片机发展有限公司自行设计的数码管显示驱动及键盘扫描管理芯片，可直接驱动 8 位共阴式数码管（或 64 只独立 LED），同时还可以扫描管理多达 64 只按键。ZLG7289B 内部含有显示译码器，可直接接受 BCD 码或 16 进制码，并同时具有 2 种译码方式。此外，还具有多种控制指令，如消隐、闪烁、左移、右移、段寻址等。ZLG7289B 采用 SPI 串行总线与微控制器接口，仅占用少数几根 I/O 口线。利用片选信号，多片 ZLG7289B 还可以并接在一起使用，能够方便地实现多于 8 位的显示或多于 64 只按键的应用。

（1）ZLG7289B 的主要特性

- 很宽的工作电压范围：+2.7~6V；
- 直接驱动 8 位共阴式数码管（1 英寸以下）或 64 只独立的 LED；
- 能够管理多达 64 只按键，自动消除抖动；
- 段电流可达 15mA 以上，字电流可达 100mA；
- 利用功率电路可以方便地驱动 1 英寸以上的大型数码管；
- 具有左移、右移、闪烁、消隐、段点亮等强大功能；
- 要显示的数据提供有两种不同的译码方式（也可以选择不译码）；
- 不接数码管而仅使用键盘管理功能时，工作电流可降至 3mA；

- 与微控制器之间采用SPI 串行总线接口，操作方便，占用I/O 资源少；
- 工作温度范围：-40~+85℃；
- 封装：SOP-28，DIP-28。

(3) ZLG7289B 管脚介绍

ZLG7289 管脚图如图 3.8.1 所示

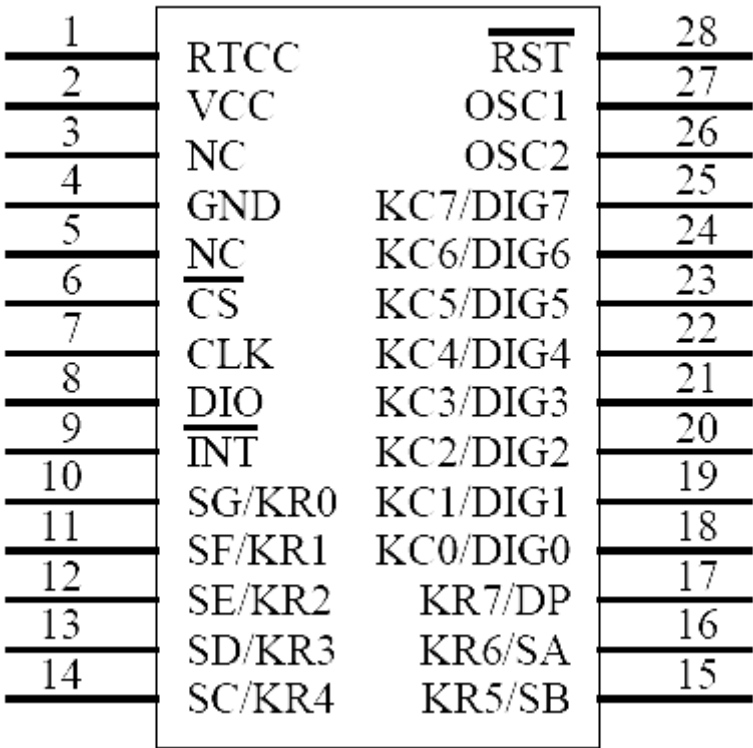


图 3.8.1 ZLG7289 管脚图

引脚名称及功能描述

RTCC：接电源

Vcc：电源，+2.7~6V

NC：悬空

GND：接地

NC：悬空

CS SPI：总线片选信号，低电平有效

CLK SPI：总线时钟输入信号，上升沿有效

DIO SPI：总线数据信号，双向

INT：键盘中断请求信号，低电平（下降沿）有效

SG/KR0：数码管g 段 / 键盘行信号0

SF/KR1：数码管f 段 / 键盘行信号1

SE/KR2：数码管e 段 / 键盘行信号2

SD/KR3：数码管d 段 / 键盘行信号3

SC/KR4：数码管c 段 / 键盘行信号4

SB/KR5：数码管b 段 / 键盘行信号5

SA/KR6：数码管a 段 / 键盘行信号6

DP/KR7：数码管dp 段 / 键盘行信号7

DIG0/KC0~DIG7/KC7:数码管字选信号0 / 键盘列信号0~数码管字选信号7 / 键盘列信号7

OSC1/OSC2：晶振输入信号/晶振输出信号

RST：复位信号，低电平有效

3.8.2 ZLG7289B 与其它部分连接图

在本语音系统中只用到了ZLG7289的显示和按键读取功能，它与单片机、数码管和按键的连接图如图3.8.2所示

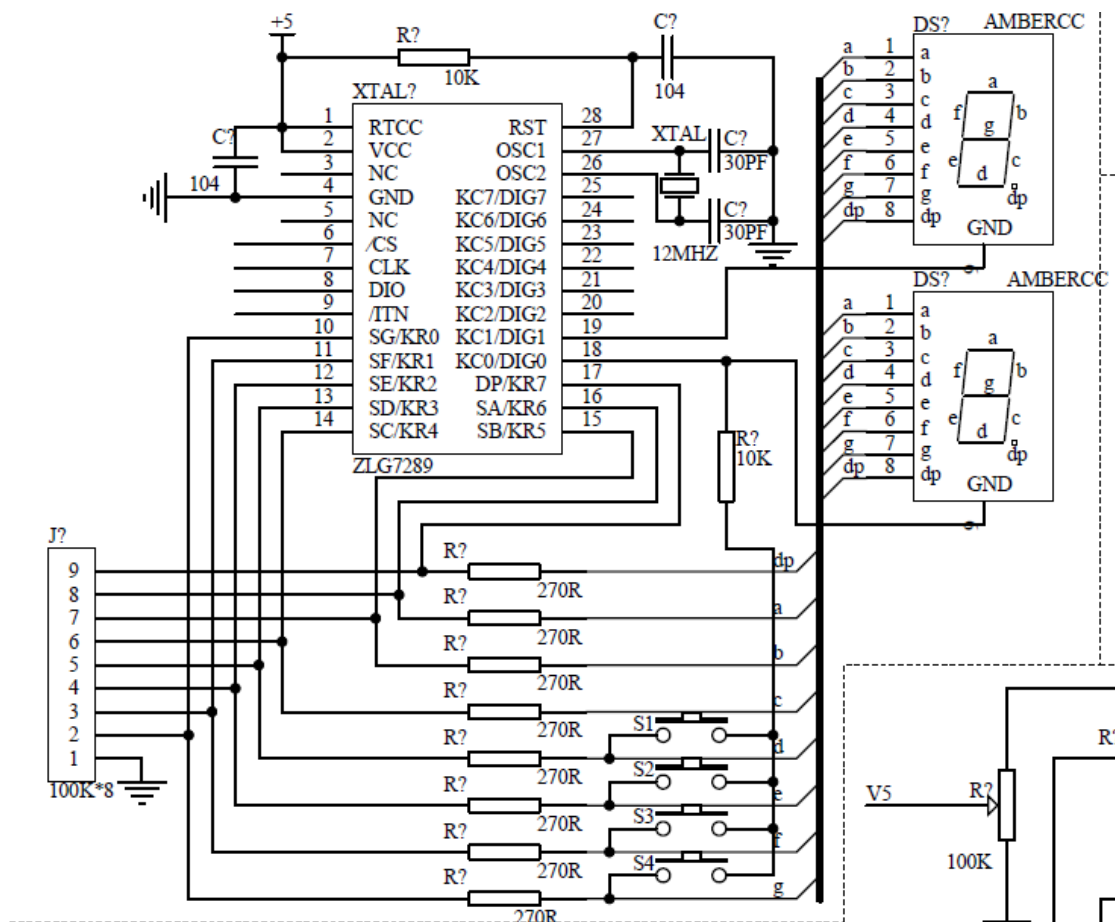


图3.8.2 ZLG7289B与其它器件连接图

3.9 供电电路模块

本设计使用的单片机及运放电路分别使用+5V、0V 和+12V、-12V 电压，选用 LM7805 三端 1.5A 正电源稳压电路实现+5V 电压供给，选用 LM7812 提供+12V 电压，选用 LM7912 提供-12V 电压。因为本设计中所用的芯片较多，单片机、存储芯片，数码管驱动芯片，数模转换及模数转换芯片都需用+5V 电压供电，所用所需电流较大，设计图中选用两片 LM7805 稳压管产生电流较大的+5V 电压。

(1) +5V 电压产生电路

+5V 电压产生电路如图 3.9.1 所示：

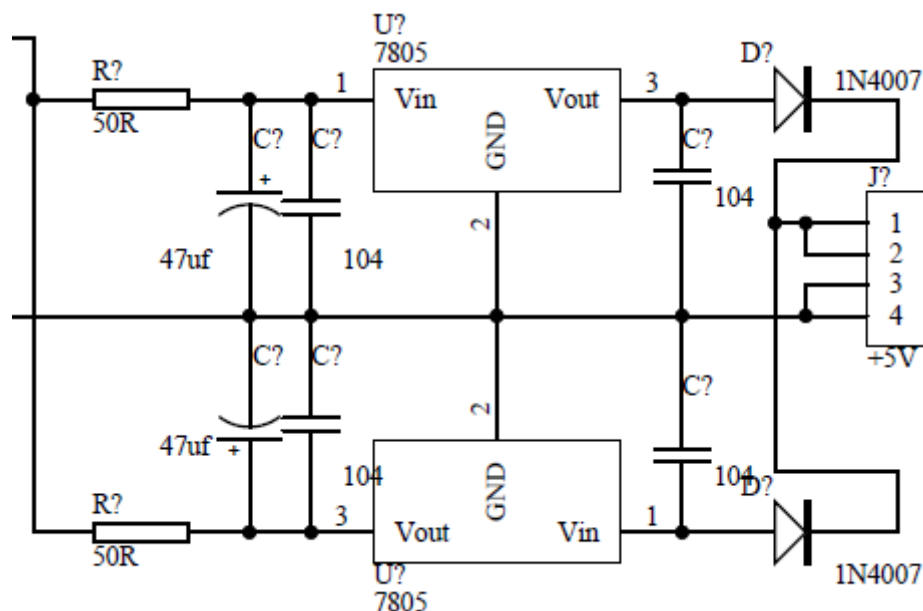


图 3.9.1 +5V 电压产生电路

(2) +12V 及-12V 电压产生电路

+12V 及-12V 电压产生电路如图 3.9.2 所示

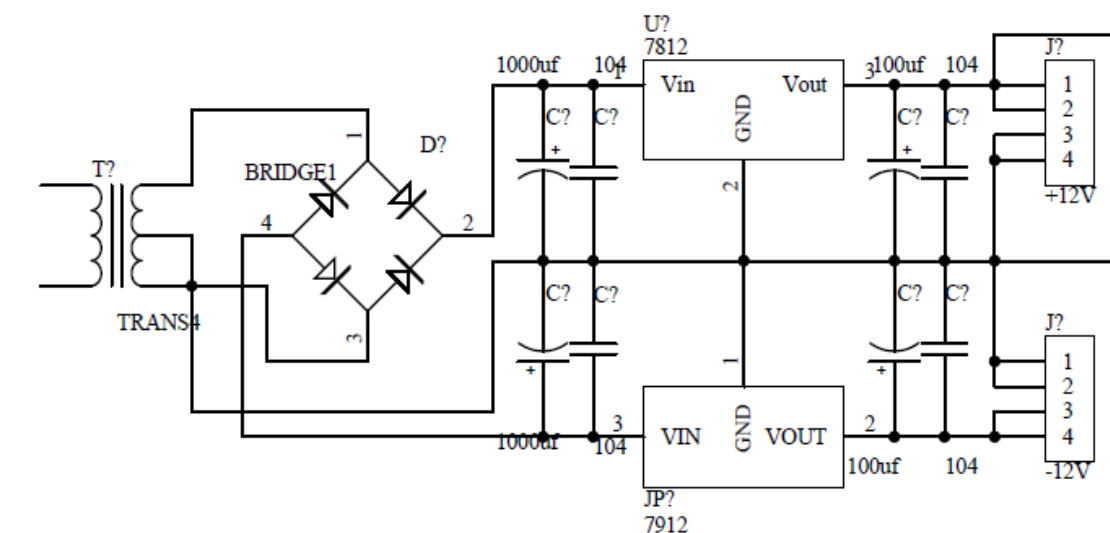


图3. 9. 2 +12V及-12V电压产生电路

3 软件设计

为了充分利用单片机的片内资源和硬件特点，本系统的电路设计采用总线控制的方式实现对 AD、DA、SRAM 等器件的读写操作。由于 ZLG7289 采用的是 SPI 的接口，所以对它的操作需用软件来模拟 SPI 通信时序。采用总线控制方式最大的优点就是利用单片机自身硬件产生的读写时序实现对外部器件的操作，通过对外接器件进行统一编址，使得总线得以分时复用。经过此次学习，对比了以往软件模拟的编程方式和总线控制的方式，深深的体会到了总线控制方式的优越性。当然，简便归简便，在编程调试过程中还是遇到了各种各样平常没有遇到的问题、情况、特殊现象；有些问题是我们平常认为很自然的不会发生的，也有些问题是我们平时根本就没有想到过的。

总之，动手了肯定会有收获和认识。在实践中运用知识可以深化对知识的理解和认识，可以更好的巩固和应用所学知识。做事一定要敢于尝试，不能局限于用一种方法，有时候要善于“偷懒”充分利用片内资源。

系统总体实现流程如图 3 所示：

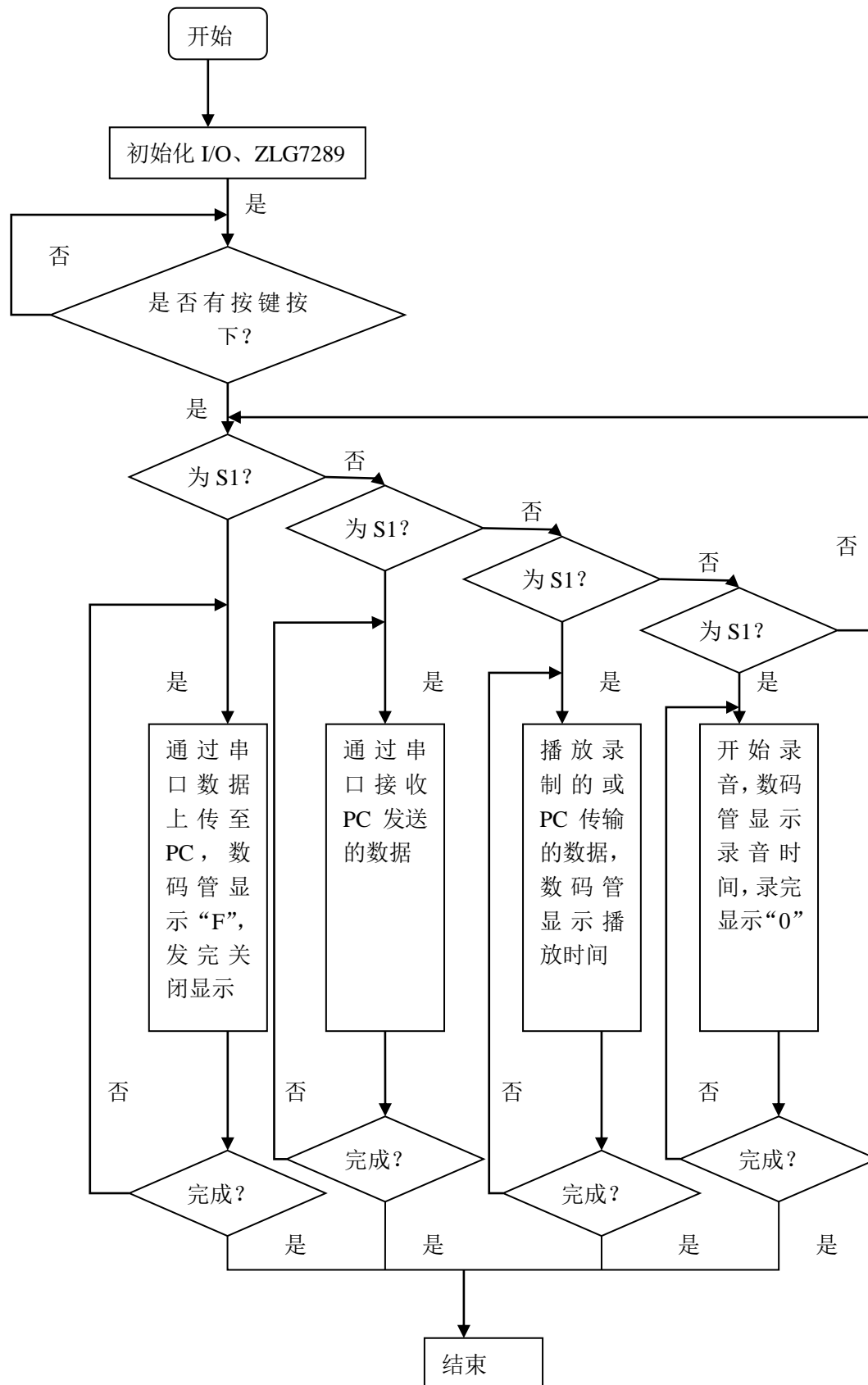


图3 程序流程图

结论

经过几个月的学习、思考、资料查阅、制作调试，终于完成了这个课题。调试结果和预期的吻合，不加压缩算法存储时间 16s 左右，加压缩算法压缩后存储时间达 32s，且无明显失真。电路板改过后存储时间 24s，回放无明显失真；数据可以通过串口上传和下载，即：录制的声音信息可以传到电脑上查看，也可以把电脑里声音信号的文件载入 SRAM 进行播放。实现了单片机和计算机之间的大容量数据交换，打破了只能模拟输入的局限。

掌握单片机的应用开发需要一个过程。首先必须掌握数字电路和模拟电路方面的知识，其次还必须学习单片机原理、硬件结构、扩展接口和编程语言。初次开发时由于没有经验，可能要经过多次反复才能完成项目。这时，会得到较大收获和积累，主要表现在硬件设计的积累，软件设计方面的积累和设计经验方面的积累。

要热衷于学习和使用新技术，善于挑战自我。另外，还要细心和认真，因为硬件设计上的一个小疏忽往往会造成非常大的经济损失；只有细心和认真的检查，负责任的测试，不懈的学习和积累，才能使硬件设计水平持续不断的提高。

致谢

在课题完成之际，我首先要向指导老师李玲老师表示深深的感谢。在课题制作期间，我经历过无从下手的窘境，都是靠杨老师在百忙之中抽出时间来对我悉心指导，给我提出了各方面的指导意见，给我提供各方面的重要资料，她是我学习工作进程中的中坚力量。感谢老师给了我一次锻炼的机会，同时也要感谢我的搭档郑路同学给予我的帮助。

参考文献

- [1]杨刚. 电子系统设计与实践. 电子工业出版社, 2009. 3.
- [2]何立民. 单片机应用系统设计. 北京: 航天航空大学出版社.
- [3]李广弟. 单片机基础, 北京: 北京航空航天大学出版社, 2001.
- [4]何希才. 新型实用电子电路400例. 电子工业出版社, 2000.
- [5]张毅刚. 新编MCS-51单片机应用设计. 哈尔滨工业大学出版社, 2003.
- [6]贾立新, 王涌. 电子系统设计与实践. 北京:清华大学出版社, 2007.
- [7]黄恩, 魏炜, 邹搜涛, 等. 数字化语音存储与回放系统硬件设计. 中国仪器仪表, 2003.
- [8]谭浩强. C程序设计. 北京:清华大学出版社, 1999.
- [9]李华. MCS 系列单片机实用接口技术. 北京: 北京航空航天大学出版社, 1993.
- [10]潘新民. 单片机实用系统设计. 北京: 人民邮电出版社, 1992.
- [11]康华光. 电子技术基础. 北京: 高等教育出版社, 1999.
- [12]郭天祥. 51单片机C语言教程. 电子工业出版社, 2009.
- [13]高海春, 任开达, 孔德峰. 数字化语音存储与回放系统. 华东船舶工业学院学报, 2000.
- [14]程佩青. 数字信号处理教程. 北京: 清华大学出版社, 1995.
- [15]大江. 电子元器件识别及检修实用技术. 北京: 中国戏剧出版社, 2009.

附录

附录一 程序源代码

```
/******  
//总体电路设计采用总线控制方式  
//对外围器件的访问统一采用地址编码方式  
*****/  
  
#include<reg51.h>  
#include<intrins.h>  
#include<math.h>  
#include<absacc.h> //包含绝对地址定义的头文件  
  
#define ADC_XBYTE[0x7FF0] //ADC 0 通道地址  
#define ADC_CTRL 0xFD //ADC 控制信号  
#define DAC_XBYTE[0x6FFF] //DAC 地址  
#define DAC_CTRL 0xFC //DAC 控制信号  
  
//RAM 地址范围为 0x8000~0xffff （共 4 页）  
//ADC 的地址范围为 0x7ff0~0x7ff7 （共 8 个通道）  
//DAC 的地址范围为 0x6fff （只要低 2 位为 ff 即可）  
  
static unsigned int RAM_address; //RAM 地址变量，起始地址 0x8000  
unsigned char count, m, n=1, key = 0xff;  
  
//显示字符编码  
  
unsigned char code zifu[]={0x7e, 0x30, 0x6d, 0x79, 0x33, 0x5b, 0x5f,  
0x70, 0x7f, 0x7b, 0x0e, 0x47, 0x00};  
  
//ZLG7289 接口定义  
  
sbit ZLG7289CS = P1^5;  
sbit ZLG7289CLK = P1^6;  
sbit ZLG7289DIO = P1^7;
```

```
sbit  ZLG7289INT  = P3^2;
sbit  ADCEOC      = P3^3;

void delay_us(unsigned char n)  //执行一次约为 14us
{
    for(;n--;n>0)
    {
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
    }
}

void nNop(unsigned char i) //短延时
{
    for(;i>0;i--) ;
}

void LDelay(unsigned int i) //长延时
{
    unsigned int j;
    for(;i>0;i--)
        { for(j=1000;j>0;j--);}
}

//函数功能：ZLG7289 复位指令
#define ZLG7289_Reset()          ZLG7289_Instruc(0xa4)

//函数：SPI_WriteOneByte()
//功能：向 SPI 总线写数据
//说明：写入的数据长度为 1 字节

void SPI_WriteOneByte(unsigned char Wdata)
```

```
{  
    unsigned char i;  
    for(i=0;i<8;i++)  
    {  
        ZLG7289DIO = (bit) (Wdata&0x80);  
        ZLG7289CLK = 1;  
        Wdata <<= 1;  
        nNop(1);  
        ZLG7289CLK = 0;  
        nNop(1);  
    }  
}  
  
//函数: SPI_ReadOneByte()  
//功能: 从 SPI 总线读数据  
//说明: 返回 1 字节数据  
unsigned char SPI_ReadOneByte()  
{  
    unsigned char i,temp;  
    ZLG7289DIO = 1;                //将 I/O 切换到输入状态  
    for(i=0;i<8;i++)  
    {  
        ZLG7289CLK = 1;  
        nNop(5);  
        temp <<= 1;  
        if (ZLG7289DIO) temp++;  
        ZLG7289CLK=0;  
        nNop(5);  
    }  
    return temp;  
}
```

```
}  
  
//函数: ZLG7289_Instruc()  
//功能: 执行 ZLG7289 纯指令  
//说明: 指令长度为 1 字节  
void ZLG7289_Instruc(unsigned char Instruc)  
{  
    unsigned char i;  
    i = IE;  
    IE &= 0xfa;  
    //禁止外部中断;  
    ZLG7289CS = 0;  
    nNop(5);  
    SPI_WriteOneByte(Instruc);  
    ZLG7289CS = 1;  
    nNop(5);  
    IE = i;  
}  
  
//函数: ZLG7289_Instruc_Data  
//功能: 执行 ZLG7289 带数据指令  
//说明: 指令长度以及数据长度均为 1 字节  
void ZLG7289_Instruc_Data(unsigned char Instruc,unsigned char Data)  
{  
    unsigned char i;  
    i = IE;  
    IE &= 0xfa;  
    ZLG7289CS = 0;  
    nNop(5);  
    SPI_WriteOneByte(Instruc);  
    nNop(5);
```

```
SPI_WriteOneByte(Data);

ZLG7289CS = 1;

nNop(5);

IE = i;

}

//函数: ZLG7289_ReadKey()
//功能: 执行 ZLG7289 读键盘指令
//说明: 返回 1 字节的按键号
//按键号的范围为: 0 ~ 63 (即 0x0 ~ 0x3f)
//若没有按按键被按下了, 返回 255 (即 0xff)
unsigned char ZLG7289_ReadKey()
{
    unsigned char Key;

    ZLG7289CS = 0;

    nNop(10);

    SPI_WriteOneByte(0x15);

    nNop(5);

    Key = SPI_ReadOneByte();

    nNop(1);

    ZLG7289CS = 1;

    nNop(5);

    return Key;
}

//函数: ZLG7289_Download
//功能: 下载数据
//说明: 将要显示的数据下载到 ZLG7289 内, 并译码显示
//参数: 数码管编号 x, 要显示的数据 dat
void ZLG7289_Download(unsigned char x,unsigned char dat)
{
```

```
x |= 0x90;          //以自己编码的方式进行译码 1001 0xxx
ZLG7289_Instruc_Data(x, dat);
}

//ZLG7289 初始化函数
void ZLG7289_Init()
{
    LDelay(20);          //延时一定时间,以使系统电源稳定
    //ZLG7289 I/O 口初始化
    ZLG7289CS = 1;
    ZLG7289CLK = 0;
    ZLG7289DIO = 1;
    ZLG7289INT = 1;
}

void Record()//普通录音函数
{
    unsigned char temp, page, page_ctrl;
    unsigned int i;
    for(page=0;page<4;page++)          //分 4 页存储
    {
        switch(page)
        {
            case 0:page_ctrl=0xF8;break;//RAM 第一页控制信
号
            case 1:page_ctrl=0xF9;break;//RAM 第二页控制信
号
            case 2:page_ctrl=0xFA;break;//RAM 第三页控制信
号
            case 3:page_ctrl=0xFB;break;//RAM 第四页控制信
号
```



```
        default:break;

    }

    RAM_address=0x8000;           //RAM 地址复位
    for(i=0;i<32768;i++)          //每页 32k 字
    {

        if(count >= 18)
        {

            count = 0;

            if(m == 10)
            {

                m = 0;

                ZLG7289_Download(1,zifu[n]);

                n += 1;

                if(n==10)n=1;

            }

            ZLG7289_Download(0,zifu[m]);

            m += 1;

        }

        P1=ADC_CTRL;    //ADC 控制有效
        ADC=0x00;        //启动通道 0 的 ADC 转换
        delay_us(7);     //等待转换结束 11.0592MHz    7
        temp = ADC;      //读 ADC 的值
        P1=page_ctrl;    //RAM 分页控制有效
        XBYTE[RAM_address++] = temp;
        delay_us(12);

    }

}

m = 0;

n = 1;
```

```
ZLG7289_Download(0, zifu[0]);           //录音完毕显示“0”
ZLG7289_Download(1, zifu[12]);          //录音完毕显示“无显
示”
}
void Play()//普通放音函数
{
    unsigned char temp, page, page_ctrl;
    unsigned int i;
    for(page=0; page<4; page++)           //分4页读取
    {
        switch(page)
        {
            case 0: page_ctrl=0xF8; break;
            case 1: page_ctrl=0xF9; break;
            case 2: page_ctrl=0xFA; break;
            case 3: page_ctrl=0xFB; break;
            default: break;
        }
        RAM_address=0x8000;
        for(i=0; i<32768; i++)
        {
            if(count >= 18)
            {
                count = 0;
                if(m == 10)
                {
                    m = 0;
                    ZLG7289_Download(1, zifu[n]);
                    n += 1;
                }
            }
        }
    }
}
```

```
        if (n==10)n=1;

    }

    ZLG7289_Download(0,zifu[m]);

    m += 1;

}

P1=page_ctrl;

temp = XBYTE[RAM_address++];

delay_us(8);

P1=DAC_CTRL;

DAC = temp;

}

}

m = 0;

n = 1;

ZLG7289_Download(0,zifu[0]);           //放音完毕显示“0”

ZLG7289_Download(1,zifu[12]);          //放音完毕显示“无显

示”

}

void UART_init()//串口初始化函数

{

    SCON = 0X50;           //串口工作于模式 1

    TMOD = 0x20;           //T1 工作于定时模式 2

    PCON |= 0x80;          //电源管理寄存器（此时单片机正常工作）

    TL1 = TH1=253;

    TH1 = TL1 = 255;

    TR1=1;                 //启动定时器 T1

    ES = 0;                //关串口中断，采用查询方式收发数据

}

void UARTSend()//串口发送函数
```

```
{

    unsigned char temp, page, page_ctrl;

    unsigned int i;

    ZLG7289_Download(0, zifu[11]);          //开始发送显示“F”

    for(page=0; page<4; page++)              //分 4 页读取

    {

        switch(page)

        {

            case 0: page_ctrl=0xF8; break;

            case 1: page_ctrl=0xF9; break;

            case 2: page_ctrl=0xFA; break;

            case 3: page_ctrl=0xFB; break;

            default: break;

        }

        RAM_address=0x8000;                  //RAM 地址复位

        for(i=0; i<32768; i++)                //每页 32k 字

        {

            P1=page_ctrl;                    //RAM 分页控制有效

            temp = XBYTE[RAM_address++];      //读取 RAM 中的数据

            SBUF=temp;                        //将 RAM 中的数据发送出去

            while(!TI);                      //等待数据发送完毕

            TI=0;                            //为下次发送数据做准备

        }

    }

    ZLG7289_Download(0, zifu[12]);          //发送完毕显示“无显

示”

}

void UARTReceive() //串口接收函数

{
```

```
    unsigned char temp, page, page_ctrl;
    unsigned int i;
    for(page=0;page<4;page++)                //分 4 页存储
    {
        switch(page)
        {
            case 0:page_ctrl=0xF8;break;//RAM 第一页控制信
号
            case 1:page_ctrl=0xF9;break;//RAM 第二页控制信
号
            case 2:page_ctrl=0xFA;break;//RAM 第三页控制信
号
            case 3:page_ctrl=0xFB;break;//RAM 第四页控制信
号
            default:break;
        }
        RAM_address=0x8000;                //RAM 地址复位
        for(i=0;i<32768;i++)                //每页 32k 字
        {
            while(!RI);                    //等待接收数据
            RI=0;                          //为下次接收做准备
            temp = SBUF;                    //读串口发送来的数据
            P1=page_ctrl;                   //RAM 分页控制有效
            XBYTE[RAM_address++] = temp;
        }
    }
}

void main()
{
```

```
unsigned char k;

ZLG7289_Init();           //初始化 ZLG7289

ZLG7289_Reset();          //复位 ZLG7289

UART_init();              //串口初始化

EA = 0;

TMOD |= 0x01;             //T0 工作于 16 位定时模式

TH0 = (65536-50000)/256;

TL0 = (65536-50000)%256;

ET0 = 1;                  //使能定时/计数器 T0 中断

TR0 = 1;                  //启动定时/计数器 T0

ITO = 1;                  //外部中断 0 选择边沿触发方式

EXO = 1;                  //打开外部中断 0

EA = 1;                   //开总中断

while(1)

{

    k = key;

    key = 0xff;

    switch(k)

    {

        case 0:

            Record();break;

        case 1:

            Play();break;

        case 2:

            UARTReceive();break;

        case 3:

            UARTSend();break;

        default:

            break;
```

```
        }  
    }  
}  
  
//外部中断 0 服务函数  
void Int0(void) interrupt 0  
{  
    key = ZLG7289_ReadKey();  
}  
  
//定时计数器 0 中断服务函数  
void T0int() interrupt 1  
{  
    TH0 = (65536-48000)/256;  
    TL0 = (65536-48000)%256;  
    count += 1;  
}
```

说明：任一按键按下后必须等到该按键所要执行的功能结束才能响应其它按键；但 S2 按键（允许串口接收数据按键）按下，接收从 PC 机发送来的数据完毕后并不会自动退出（没有编写更为完善的控制程序），此时要按下复位键使系统复位后才能进行其它按键功能操作。

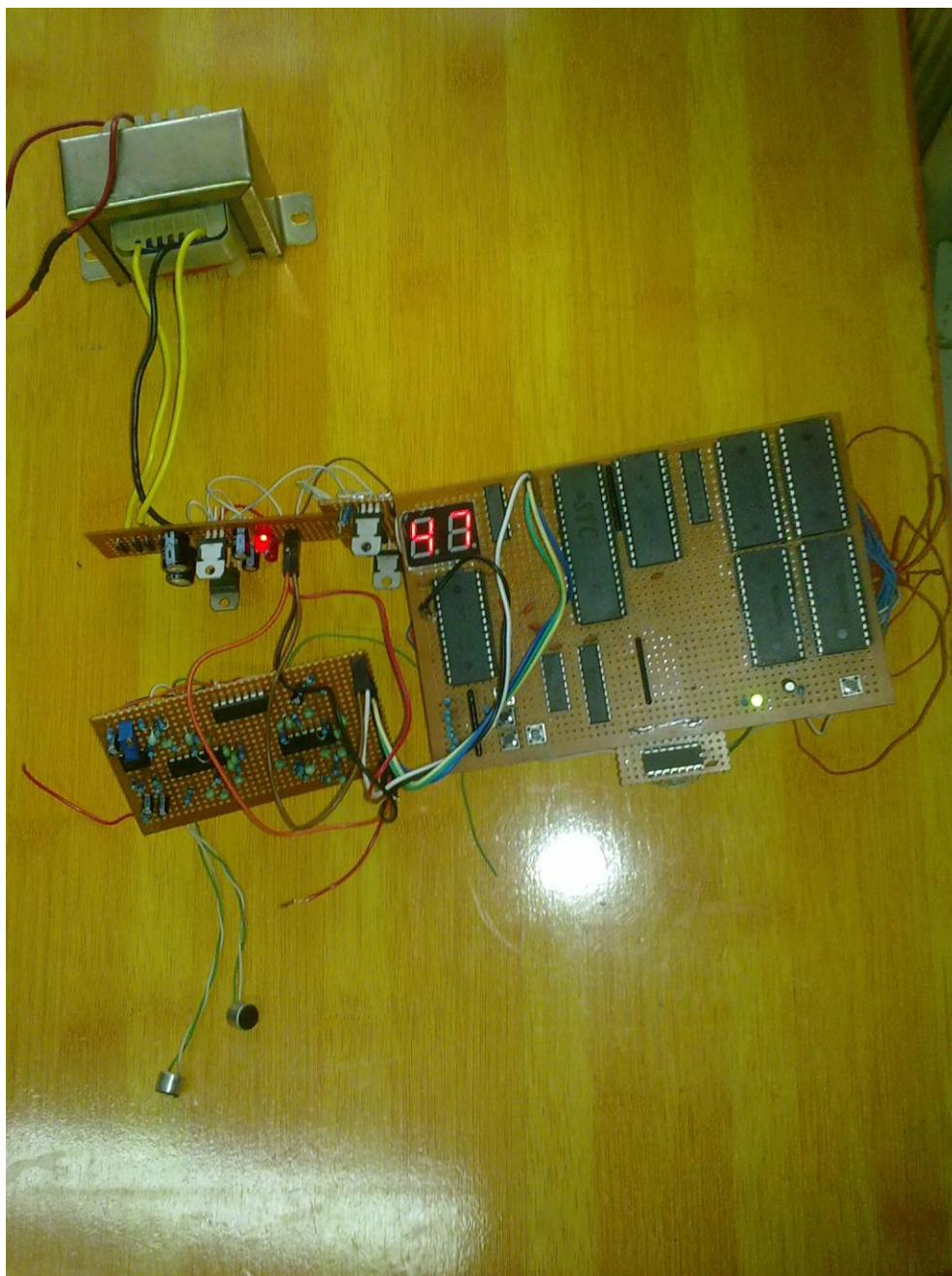


图 B 实物图