



北京北阳电子技术有限公司
Beijing Sunnorth Electronic Technology Co., Ltd

数字化语言存储与回放系统说明书

Electron—competition Specification

北京北阳电子技术有限公司

数字化语言存储与回放系统	3
摘要	3
设计功能及要求	4
一、方案设计与论证	4
二、硬件电路设计	5
2.1 电路方框图及说明	5
2.2 各部分电路设计	6
2.2.1 61A 板电路原理图	6
2.2.2 麦克风输入电路	7
2.2.3 存储器的连接电路	8
2.2.4 DAC 输出放大电路	8
2.2.5 键盘电路	9
三、软件设计	10
3.1 主程序流程	10
3.2 中断服务流程	11
3.3 AD 转换实现	11
3.4 数字滤波	13
3.4.1 概述	13
3.4.2 μ^n SP 的乘-累加 (MAC) 功能	14
四、系统测试及整机指标	18
五、总结	18
六、参考资料	19

数字化语言存储与回放系统

摘要

系统主控部分采用 SPCE061A 单片机，存储器采用 SPR4096。系统采用两种方案进行设计，一种是基于 SPCE061A 的 MicIN 的录音，一种是 SPCE061A 的 LineIN 的录音，都得到了不错的效果。本文给出了详细的设计过程。

关键词： SPCE061A SPR4096 数字滤波

SPCE061A 单片机概述

SPCE061A 是继 μ 'nSP 系列产品 SPCE500A 等之后凌阳科技推出的又一个 16 位结构的微控制器。目前有两种封装形式：84 引脚的 PLCC84 封装和 80 引脚的 LQFP80 贴片封装。

主要性能如下：

- 16 位 μ 'nSP 微处理器；
- 工作电压：VDD 为 2.4~3.6V(cpu), VDDH 为 2.4~5.5V(I/O)；
- CPU 时钟：32768Hz~49.152MHz；
- 内置 2K 字 SRAM、内置 32K FLASH；
- 可编程音频处理；
- 32 位通用可编程输入/输出端口；
- 32768Hz 实时时钟，锁相环 PLL 振荡器提供系统时钟信号；
- 2 个 16 位可编程定时器/计数器(可自动预置初始计数值)；
- 2 个 10 位 DAC(数-模转换)输出通道；
- 7 通道 10 位电压模-数转换器(ADC)和单通道语音模-数转换器；
- 声音模-数转换器输入通道内置麦克风放大器自动增益控制(AGC)功能；
- 系统处于备用状态下(时钟处于停止状态)耗电小于 $2\mu\text{A}@3.6\text{V}$ ；
- 14 个中断源：定时器 A/B，2 个外部时钟源输入，时基，键唤醒等；
- 具备触键唤醒的功能；
- 使用凌阳音频编码 SACM_S240 方式(2.4K 位/秒)，能容纳 210 秒的语音数据；
- 具备异步、同步串行设备接口；
- 具有低电压复位(LVR)功能和低电压监测(LVD)功能；
- 内置在线仿真电路接口 ICE (In- Circuit Emulator)；
- 具有保密能力；
- 具有 WatchDog 功能（由具体型号决定）

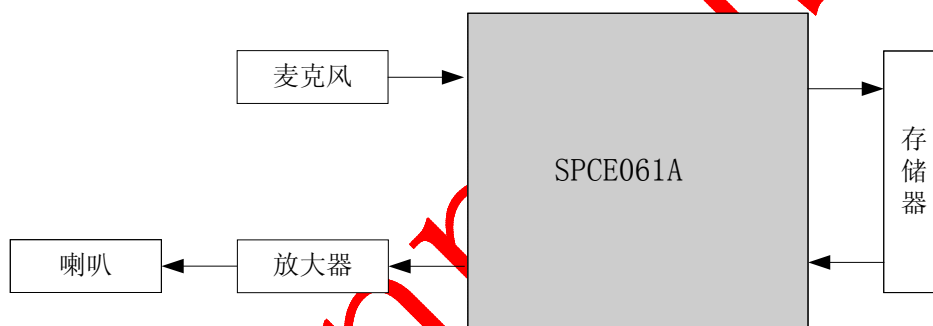
设计功能及要求

设计一个数字语音存储与回放系统。采样频率 $f_s = 8\text{kHz}$ ，语音存储时间 ≥ 10 秒，回放语音质量良好。

一、方案设计与论证

方案一：

使用 SPCE061A 单片机自带的 MicIN 输入，利用凌阳 SACM 库提供的 DVR 函数进行录放，录音资料存放到 SPR4096 中，通过 SPCE061A 自带的 DAC 进行播放。原理框图如图一。



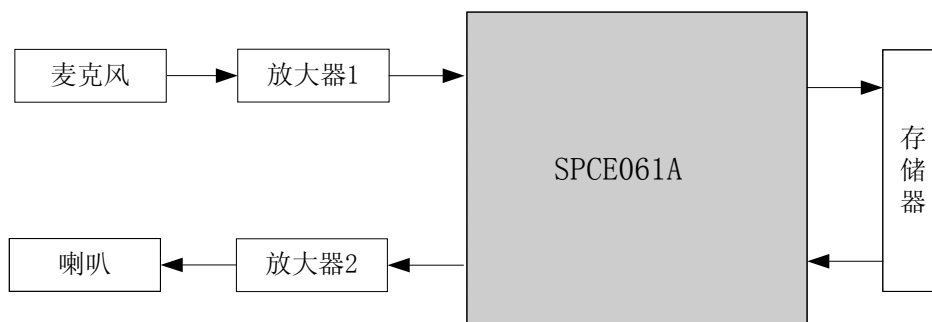
图一 MicIN 输入原理框图

方案二：

使用 SPCE061A 单片机自带的 MicIN 输入，不使用凌阳 SACM 库提供的 DVR 函数，直接将 ADC 的结果进行数字滤波，然后存放到 SPR4096 中，播放时从 SPR4096 中读取数据，直接送到 SPCE061A 自带的 DAC 进行播放。这个方案的硬件和方案一没有不同，两者的区别在软件方面。本方案的原理框图也如图一所示。

方案三：

使用 SPCE061A 单片机自带的 LineIN 输入，将 ADC 的结果进行数字滤波，然后存放到 SPR4096 中，播放时从 SPR4096 中读取数据，直接送到 SPCE061A 自带的 DAC 进行播放。本方案的原理框图如图二所示。



图二 LineIN 输入原理框图

方案比较:

因为 SPCE061A 的 MicIN 通道内置了带 AGC 的放大电路，而 LineIN 通道需要在芯片外围搭建这部分电路，使得方案三的硬件比方案一和方案二多，而且带 AGC 的放大电路也不是很好调的，所以，方案一和方案二比方案三更容易实现，效果也更好。

方案一和方案二比较，方案一利用凌阳 SACM 库提供的 DVR 函数进行录放，这很容易实现，效果也好。

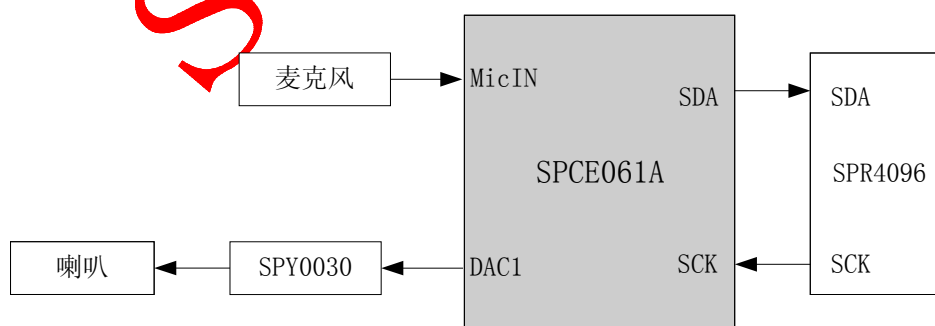
综上所述，方案一最优，方案二次之，方案三最差。

因为方案一已经有现成的例子（详见凌阳大学计划网站），所以我分别尝试了方案二和方案三。

二、硬件电路设计

2.1 电路方框图及说明

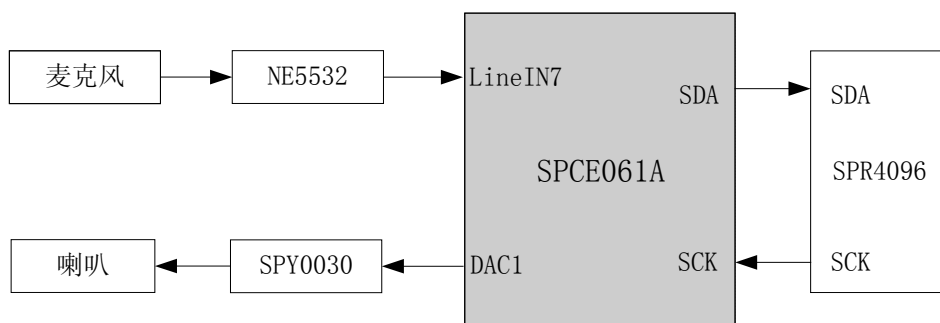
先看方案二，如图三。麦克风电路主要是将声音信号变为电信号。存储器采用 SPR4096，作用当然是用来存放录音数据了。放大器采用 SPY0030，将 DAC 输出的信号放大 20 倍，驱动喇叭。



图三 MicIN 输入电路框图

再看方案三，如图四所示。与方案二比较，本方案在麦克风和 SPCE061A 之间多了

一个放大器 NE5532，它的作用是将麦克风输入的微弱信号放大，同时进行电压偏移，将信号抬升到 SPCE061A 的 AD 测量范围（0~VDD）。

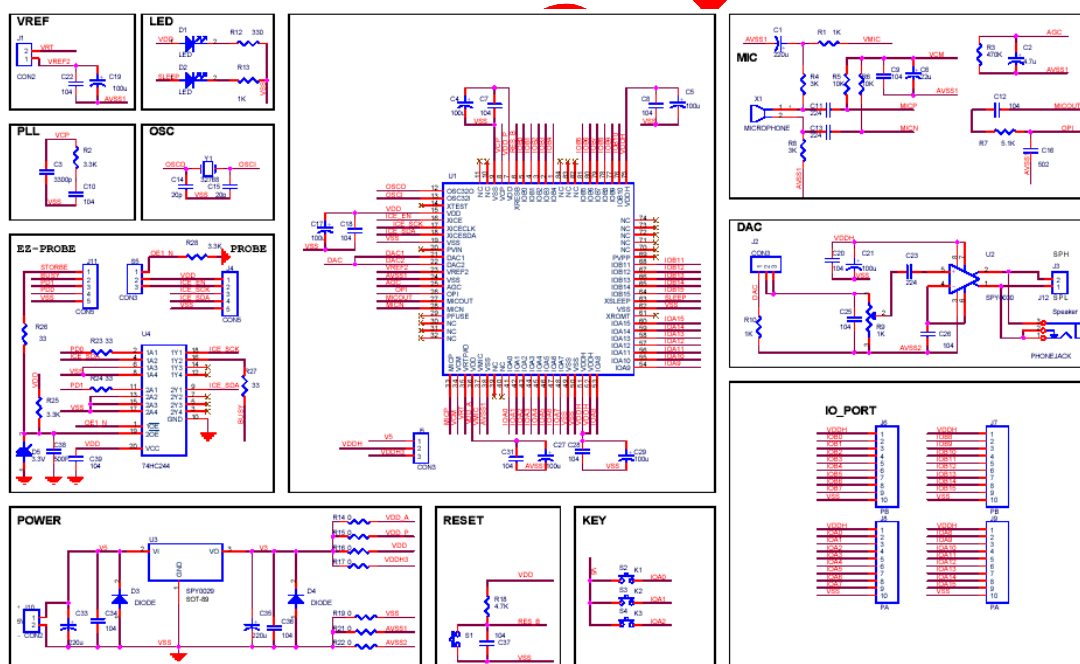


图四 LineIN 输入电路框图

2.2 各部分电路设计

2.2.1 61A 板电路原理图

本设计的硬件核心是 61A 板，它的原理图如图五所示。



图五 61A 板原理图

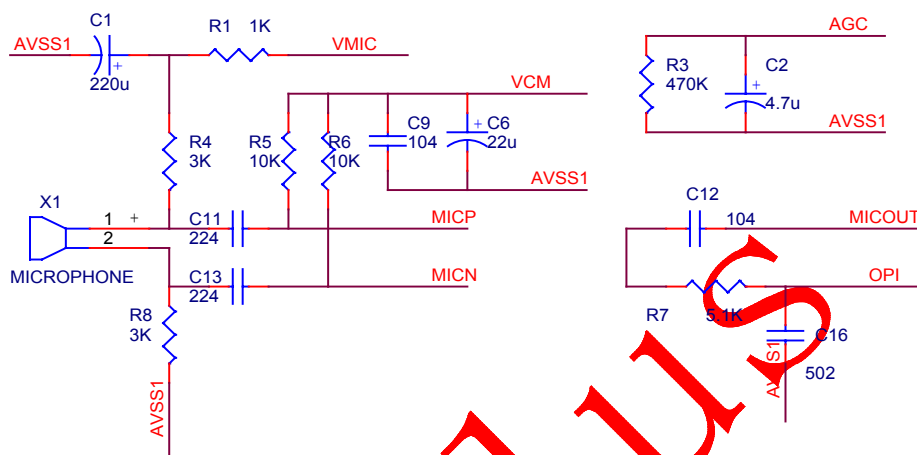
说明：

方案二需要使用 61A 板上的麦克风输入电路，DAC 输出放大电路，以及三个按键。

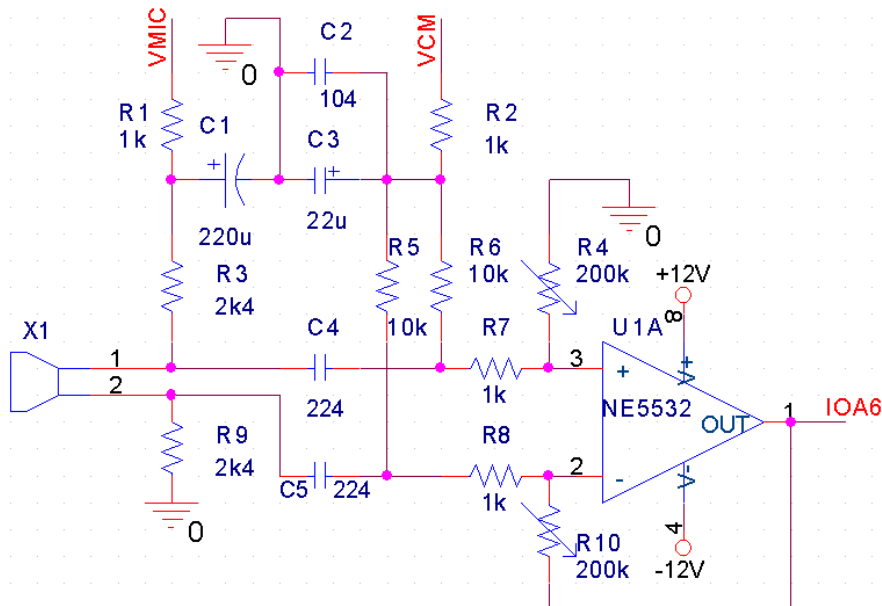
方案三需要使用 61A 板上的 DAC 输出放大电路，以及三个按键

2.2.2 麦克风输入电路

图六是方案二的麦克风输入电路，这部分在 61A 板上是现成的，不用外接。因为 SPCE061A 的 MicIN 通道内部做了 AGC 和滤波，所以外围电路特别简单。



图六 方案二的麦克风输入电路



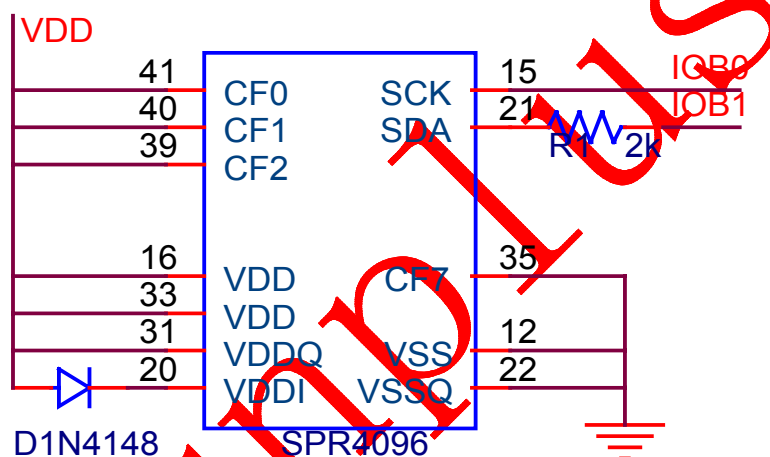
图七 方案三的麦克风输入放大电路

图七是方案三的麦克风输入放大电路，这部分需要外接。SPCE061A 的 VMIC 输出为 3.3V，正好作为麦克风的电源。VMIC 经 R1 和 R3 为麦克风提供 3.3V 的工作电

压,麦克风输出信号经 C4 和 C4 隔直后送到放大电路。放大电路的运放曾考虑过 LM324 和 NE5532, 最终选择了 NE5532, 原因是考虑到输入失调电压, LM324 的典型值为 1.5mV, NE5532 的典型值为 0.5mV, 采用 LM324 的话可能会导致输入失调电压把输入的信号“吃掉”。因为 SPCE061A 的 AD 测量范围是 0~VDD, 所以要把交流信号进行一个电压偏移, 使之不出现负电平, 电压偏移是在信号线加直流偏置电压实现的, 具体来说, 是把 VCM (这是 SPCE061A 的 ADC 参考电压输出脚, 系统正常工作时输出 1.6V) 经 R2 和 R5、R6 后接到信号线上。放大电路的增益为 R10 与 R8 的比值, 调整 R10 可以改变电路的增益。实验表明, 当 R10 调为 100k 时, 效果比较好。R4 是为了保证 NE5532 的 $R_p=R_n$ 而设的, 所以, $R_4=R_{10}$ 。

2.2.3 存储器的连接电路

图八是存储器的连接电路, 该电路同时适合方案二和方案三。



图八 存储器的连接电路

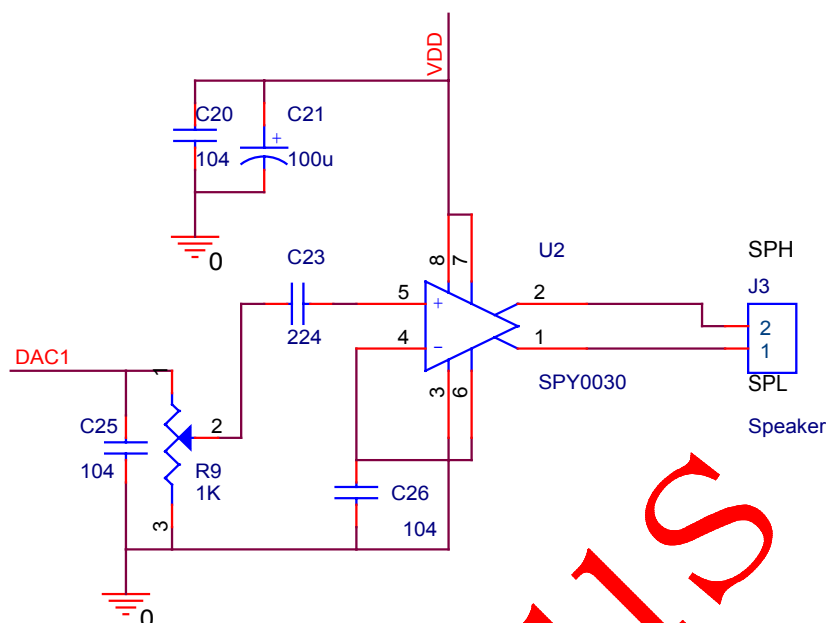
SPR4096 是凌阳公司提供的高性能 4M-bit (512K×8-bit) FLASH, 分为 256 个扇区 (Sector) 每个扇区为 2K-byte。SPR4096 还内置了一个 4K×8-bit 的 SRAM。

SPR4096 串行接口的工作频率可达 5MHz。SPR4096 有两个电源输入端 VDDI 和 VDDQ。VDDI 是给内部 FLASH 和控制逻辑供电的; VDDQ 是专门为 I/O 供电的。供电电压为 VDDQ: 2.25V ~ 3.6V, VDDI: 2.25V ~ 2.75V。

SPR4096 按串行接口模式工作, 要把 CF2~CF0 均接高电平。CF7 为低电平时选中 FLASH, 高电平时选中 SRAM。

2.2.4 DAC 输出放大电路

图九是 DAC 输出放大电路, 它在 61A 上也是现成的, 不需要外接。这个电路同时适用于方案二和方案三。



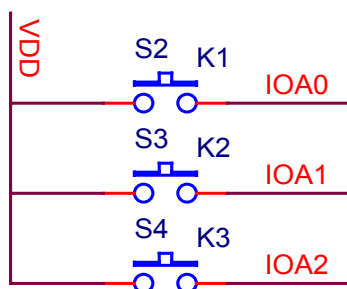
图九 DAC 输出放大电路

电路中 SPY0030 的放大倍数被固定为 20 倍。音量的大小可以通过电位器 R9 调整。

SPY0030 也是凌阳公司的产品。它的详细说明可以参见凌阳公司的 SPY0030 data sheet。和 LM386 相比，SPY0030 还是比较有优势的，比如 LM386 工作电压需在 4V 以上，SPY0030 仅需 2.4V 即可工作(两颗电池即可工作)；LM386 输出功率 100mW 以下，SPY0030 约 700mW。

2.2.5 键盘电路

图十是键盘电路，该电路在 61A 板上也是现成的，方案二和方案三都要用到该电路。按下 S2 开始录音，按下 S3 停止录音或者停止播放，按下 S4 开始播放。

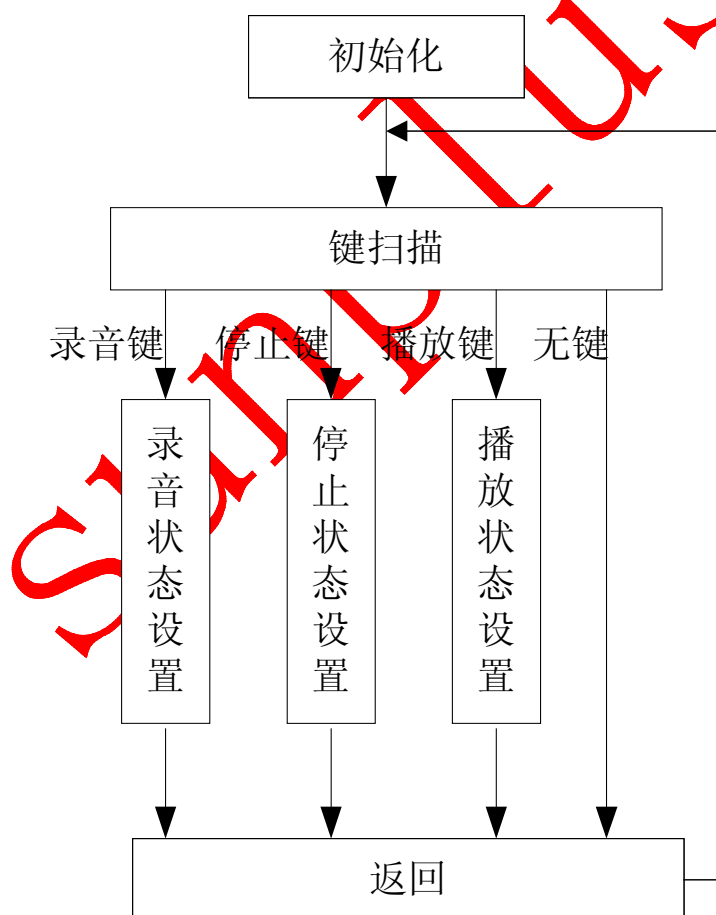


图十 键盘电路

三、软件设计

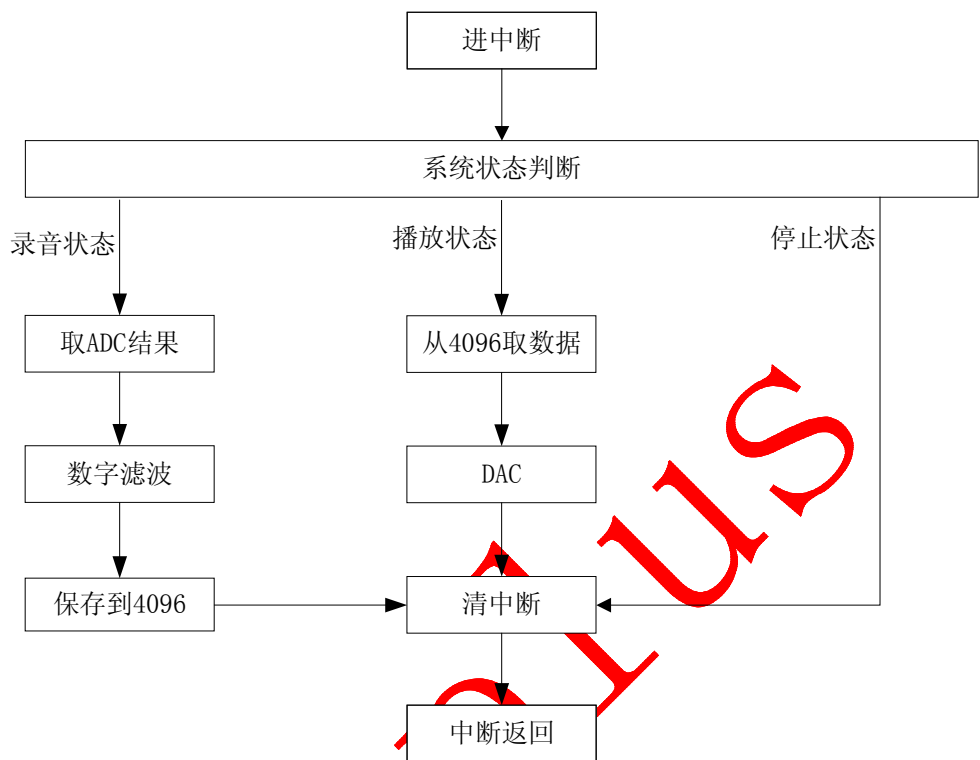
总体思路把系统分为录音、停止和放音三种状态，状态的改变用按键控制。录放音都放在中断服务程序里处理，主程序只处理键盘事件。主程序先对系统进行初始化，然后只处理键盘事件，判断按键值，并据此设置相应的系统状态。中断服务程序是通过定时器中断进入的。在中断服务程序中，先判断系统当前状态，根据系统状态进行相应操作。如果是录音状态，就去取 ADC 结果进行数字滤波，把滤波结果保存到 SPR4096 中，最后清中断标志退出中断服务；如果是放音状态，就从 SPR4096 中读出数据，进行 DAC，最后清中断标志退出中断服务；如果是停止状态，就不进行其它操作，直接清中断标志就退出中断服务程序。

3.1 主程序流程



图十一 主程序流程

3.2 中断服务流程



图十二 中断服务程序流程

3.3 AD 转换实现

我们先了解一下与 SPCE061A 的 AD 相关的寄存器。

ADC 的控制端口：P_ADC_Ctrl(读/写)(\$7015H)

ADC 多通道控制单元：P_ADC_MUX_Ctrl (读/写)(702BH)

模拟电压输入的存储单元：P_ADC_MUX_Data(读) (\$702CH)

P_ADC_Ctrl 单元

b15 ^[2]	b6	B2	b0	控制功能描述
RDY (读)	DAC_I (写)	AGCE (写)	ADE (写)	
0	-	-	-	10 位模/数转换未完成
1	-	-	-	10 位模/数转换完成，输出 10 位数字量
	0			DAC 电流= 3mA @V _{DD} =3V ^[1]

	1			DAC 电流 = 2mA @V _{DD} =3V
-	-	0	-	取消自动增益控制功能 ^[3]
-	-	1	-	设置自动增益控制功能
-	-	-	0	禁止模/数转换工作
-	-	-	1	允许模/数转换工作

注：

- [1] 此为 DAC_I 的缺省选择。
- [2] b15 只用于 MIC_IN 通道输入。
- [3] 当模拟信号通过麦克风的 MIC_IN 通道输入时，可选择 AGCE 为 ‘1’，即运算放大器的增益可在其线性区域内自动调整。AGCE 缺省选择为 ‘0’，即取消自动增益控制功能。
- [4] 写入时需注意 b5 = 1, b4=1, b3 =1 和 b1=0。

P_ADC_MUX_Ctrl 单元

b15	b14	b13-b3	b2	b1	b0	控制功能描述
Ready_MUX (读) ^[1]	FAIL (读) ^[2]	-	Channel_sel (读/写)			
0		-	-	-	-	10 位模/数转换未完成
1	0	-	-	-	-	10 位模/数转换完成
-		-	0	0	0	模拟电压信号通过 MIC_IN 输入
-		-	0	0	1	模拟电压信号通过 LINE_IN1 输入
-		-	0	1	0	模拟电压信号通过 LINE_IN2 输入
-		-	0	1	1	模拟电压信号通过 LINE_IN3 输入
-		-	1	0	0	模拟电压信号通过 LINE_IN4 输入
-		-	1	0	1	模拟电压信号通过 LINE_IN5 输入
-		-	1	1	0	模拟电压信号通过 LINE_IN6 输入
-		-	1	1	1	模拟电压信号通过 LINE_IN7 输入

ADC 的多路 LINE_IN 输入将与 IOA[0~6]共用，即：

多路 LINE_IN 输入与 IOA[0~6]共用

IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
LIN_IN 7	LIN_IN 6	LIN_IN 5	LIN_IN 4	LIN_IN 3	LIN_IN 2	LIN_IN 1

P_ADC_MUX_Data 单元用于读出 LINE_IN[7:1]10 位 ADC 转换的数字数据，即：

10 位 ADC 转换的数据存放格式

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6
D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

我们现在来看 AD 的初始化程序

```
//AD 设定
```

```
r1 = 0x1;
```

```
[P_ADC_Ctrl] = r1;           //允许数模转换
```

```
r1 = 0x7;
```

```
[P_ADC_MUX_Ctrl] = r1;      //选择 Line_IN7(IOA6)输入
```

若要从其它通道输入模拟电压，只需要把相应的通道值赋给 P_ADC_MUX_Ctrl。

在程序中需要 AD 结果的时候，只需要直接去读 P_ADC_MUX_Data 单元，r1 = [P_ADC_MUX_Data]。需要注意的是，AD 的结果放在 P_ADC_MUX_Data 单元的高 10 位。

3.4 数字滤波

3.4.1 概述

我们希望录音得到的只是音频信号，减少其它信号的干扰。因此，我们要进行滤波，滤去音频信号（300~3400Hz）频段以外的信号。

滤波可以用模拟的方法实现，也可以用数字的方法实现。模拟滤波一般采用 RC 电路组成无源滤波，或加上运放构成有源滤波，不管用哪种方法，都需要增加硬件成本，而且，随着阶数的增加，模拟滤波器的调试将越来越困难。数字滤波的好处在于没有硬件成本，当然，不包括因 MCU 的速度不够而必须更换 MCU 引起的成本增加。另外，高阶（比如 100 阶）滤波器对模拟滤波器来说是很难想象的，而用数字滤波器就可以实现。

数字滤波器是对数字信号实现滤波的线性时不变系统。数字滤波的实质是一种运算过程，实现对信号的运算处理。

我们从信号和系统课程中知道，时域离散系统的频域特性如下：

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

其中 $Y(e^{j\omega})$ 、 $X(e^{j\omega})$ 分别是数字滤波器的输出序列和输入序列的频域特性，

$H(e^{j\omega})$ 是数字滤波器的单位取样响应的频谱。按照输入信号频谱的特点和处理信号的目的，适当选择 $H(e^{j\omega})$ ，使得滤波后的 $X(e^{j\omega})H(e^{j\omega})$ 满足设计的要求，这就是数字滤波器的滤波原理。

$H(e^{j\omega})$ 的选择很关键，它决定了滤波器的性能。单片机一般适合做定点运算，而 $H(e^{j\omega})$ 一般都是小数，所以要对 $H(e^{j\omega})$ 进行定点处理，这是一个很麻烦的过程。

滤波运算要进行乘累加的运算，一般 MCU 速度不够的话是无法进行的。SPCE061A 可以工作在 49MHz，更重要的是，它提供了一个 16×16 的硬件乘法器，它的乘累加指令正好可以用来实现滤波运算。

3. 4. 2 $\mu'nSP$ 的乘-累加 (MAC) 功能

$\mu'nSP$ 乘法累加求和指令 (Muls) 之格式如下：

MR = [Rd] * [Rs] , ss, n;

MR = [Rd] * [Rs] , us, n;

式中：

MR 为用于累加的寄存器对，由 R3、R4 充当；

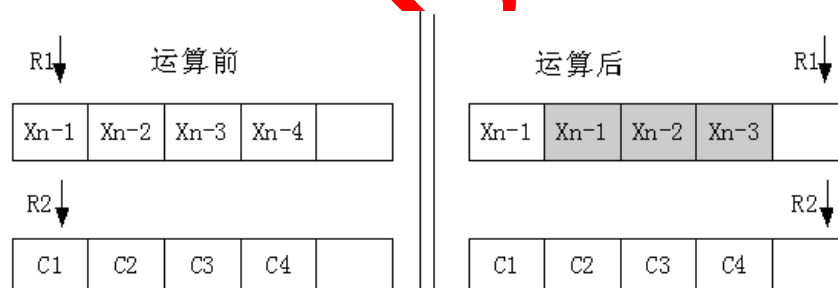
Rd 为目标寄存器，在此用于采样数据指针；

Rs 为源寄存器，在此用于系数寄存器指针；

n 为参加滤波运算的采样样本数目；

ss 为有符号数相乘的设定，缺省设定即为此；us 为无符号数相乘的设定。

假设参加滤波的采样样本数为 4，且由采样数据指针 R1 和系数寄存器指针 R2 分别指向的存储器内容在执行 Mul 指令前后的情况如下图所示。



当执行了指令 MR = [R1] * [R2], 4 以后，会产生如下一些动作：

1) 累加器 MR 清零后进行 MAC 计算：

$$MR = C_1 * X_{n-1} + C_2 * X_{n-2} + C_3 * X_{n-3} + C_4 * X_{n-4};$$

且指针向右移动了 n (n=4) 个字的位置。

2) 值得注意的是，指针 R1 指向的存储器的内容 (采样数据样本) 向前移动了一个字的位置。其意义在于当采样新的数据样本 (X_n) 时，可将 X_n 依序存放在 X_{n-1} 之后，而最旧的数据样本 (X_{n-4}) 会被次旧的数据样本 (X_{n-3}) 取代。

Muls 指令只占一个字 (即 16 位) 的存储空间且指令执行的时间为 (10n+6) 个时

钟周期，具体到本例是 $10 \times 4 + 6 = 46$ 个时钟周期。

3. 4. 3 用 SPCE061A 实现带通 FIR 滤波器

我们再来看看这个公式

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

具体到我们语音存储和回放系统，其中 $Y(e^{j\omega})$ 、 $X(e^{j\omega})$ 分别是数字滤波器的输出和采集到的音频信号， $H(e^{j\omega})$ 是数字滤波器系数。我们最大的工作就是适当选择 $H(e^{j\omega})$ ，使得滤波后的 $X(e^{j\omega})H(e^{j\omega})$ 满足设计的要求。

FIR (有限冲击响应) 滤波器维护一个固定数目 n 的最新样本表。每次 FIR 迭代前，新样本加入表中。FIR 迭代后，最旧的样本被替代。此表表示样本向量，它与系数向量相乘得到当前输出。

设我们要实现一个 15 阶 FIR 低通滤波器。采样频率为 $f_{\text{sample}} = 8k$ ，截止频率为 $fp1 = 300\text{Hz}$, $fp2 = 3400\text{Hz}$ 。

我们可以利用 MATLAB 的 `fir1` 函数来进行设计该滤波器。

`fir1` 的具体语法形式为 `b=fir1(n,W)`。它的输入参数是滤波器的阶数 n 和截至频率 W_n ， W_n 是一个 0 到 1 的数，1 对应采样频率的一半。带通滤波器有两个截至频率，所以 W_n 是含有两个数的向量， $W_n=[w1 \ w2]$ ，带通滤波器的通带范围为 $w1 < W < w2$ 。函数的返回值 b 是一个 $n+1$ 的向量，它包含了设计得到的滤波器的 $n+1$ 个系数。在 MATLAB 的命令窗口下键入 `help fir1` 后回车，将会看到 `fir1` 的说明。

在 MATLAB 的命令窗口下键入 `b=fir1(15,[400 3400]/4000)`，回车得到 `b =`

Columns 1 through 7

0.0007 -0.0110 -0.0018 -0.0424 -0.0434 -0.0317 -0.2345

Columns 8 through 14

0.5130 0.5130 -0.2345 -0.0317 -0.0434 -0.0424 -0.0018

Columns 15 through 16

-0.0110 0.0007

这时有三个问题摆在我们的面前：

1、系数的整数化

由于 SPCE061A 更适于做定点运算，故系数必须以整数的形式存储在 ROM 中。计

算出的系数是小于 1 的小数，所以需要对其进行适当的放大并进行取整处理。

2、 滤波运算的符号问题

因为滤波运算有两种方式： $mr=[r1]*[r2],us,n$ （有符号数与无符号数相乘）和 $mr=[r1]*[r2],ss,N$ （有符号数与有符号数相乘）。那么如果系数均为正数，运算较为简单，使用参数 us 即可得到正确结果。如果系数出现了负数，那么就需要考虑把 AD 采样的数据处理为有符号数的整数（如果 AD 结果最高位可能出现 1，可以通过右移一位，确保最高位为 0；如果信号幅值小，AD 结果最高位一定为 0，则无需进行转换），再进行 ss 滤波运算，才能得到正确结果。此时还涉及到 DA 输出的数据是无符号数，在实际应用中应根据情况进行灵活处理。

3、 滤波运算结果

由于滤波运算结果是 32 位的有符号数，如果结果是大于 0 的非溢出的数据，我们则根据其出现的最高有效位进行移位处理，选择最高有效的 16 位作为真正的输出。但是如果结果出现了负数，那么我们还要进行加一个偏移量，将结果全部转换为非溢出正数，再提取高 16 位有效数据。如果结果溢出，那么就要调整系数的放大倍数，使得滤波运算的结果落在合适的范围。

在设计过程中，我是这样考虑的，将输入的最大值与最大的系数相乘，然后乘以 16，取高 16 位，加上偏移量 $0x7fff$ （最大的有符号数），保证得到的结果不小于 0，也不大于 $0xffff$ ，这样就可以保证运算结果和运算过程没有溢出。照这样一算，最大系数为 139677，最小系数为 -139669。而 R1 只能表示 -32768~32767。考虑到滤波系数 b 的最大值（绝对值）为 0.5，所以将滤波系数放大 65535 倍也是安全的。

这样可以计算 16 个系数为：

$h[0]=h[15]=48$

$h[1]=h[14]=-724$

$h[2]=h[13]=-119$

$h[3]=h[12]=-2779$

$h[4]=h[11]=-2841$

$h[5]=h[10]=-2080$

$h[6]=h[9]=-15369$

$h[7]=h[8]=32768$

于是我们可以定义如下系数表：

.DATA

h0: .dw 48;

h1: .dw -724;

h2: .dw -119;

h3: .dw -2779;

h4: .dw -2841;



```
h5:    .dw -2080;
h6:    .dw -15369;
h7:    .dw 32768;
h8:    .dw 32768;
h9:    .dw -15369;
h10:   .dw -2080;
h11:   .dw -2841;
h12:   .dw -2779;
h13:   .dw -119;
h14:   .dw -724;
h15:   .dw 48;
```

下面建立一个变量序列:

.ISRAM

```
x0:    .dw 1;
x1:    .dw 1;
x2:    .dw 1;
x3:    .dw 1;
x4:    .dw 1;
x5:    .dw 1;
x6:    .dw 1;
x7:    .dw 1;
x8:    .dw 1;
x9:    .dw 1;
x10:   .dw 1;
x11:   .dw 1;
x12:   .dw 1;
x13:   .dw 1;
x14:   .dw 1;
x15:   .dw 1;
```

滤波程序如下

_Fir: .PROC

r1 = x0; //数据指针复位

r2 = h0; //系数指针复位

r3 = [P_ADC_MUX_Data]; //取 AD 采样数据

r3 = r3 & 0xffc0 //取有效位

r3 = r3 lsr 1; //转换成有符号数的形式

[r1] = r3; //更新

```
mr = [r1]*[r2],ss,16      //FIR 滤波运算
r4 += 0x7fff;             //电压偏移
r1=r4;                    //r1 为滤波结果
retf
.ENDP
```

四、系统测试及整机指标

方案二：

声音清晰，基本无失真。

录音时间大于 50s。

方案三：

声音较清晰，有轻度失真与干扰。

录音时间大于 50s。

五、总结

这次尝试了三个方案中最差的两个方案，没有采用最优方案。如果采用最优方案，效果会更好。

在实施方案二的过程中，发现如果录音都用相同频率的中断来做的话，放音速度会比实际录音时快。原因在于录音时往 SPR4096 的 flash 里写数据时需要额外的等待，使得录音的频率小于设定的频率，而放音时从 SPR4096 的 flash 里读数据的时间几乎可以忽略不计，这就造成了放音速度会比实际录音时快。针对这种情况，我用 8KHz 录音，用 4KHz 放音，效果不错。

实施方案三的过程中，一开始回放出来的是噪声，听不到人声，参照方案二的解决办法，我用 8KHz 录音，用 4KHz 放音，这时能从回放中听到人声，但噪音还是很大。用示波器观察各点波形，分析后认定是电源干扰（我用 WY-11 型微机电源提供 +12V、-12V 和 +5V 的电压）。对电源滤波后，效果好了很多。调整可变电阻 R4 和 R10，使人正常的话音得到最好的录放，此时 R4 和 R10 都是 100k。本方案对麦克风信号的放大电路很简单，没有做自动增益控制，导致了微弱的话音录不进，太大的声音有失真，只能保证正常的话音（这里的正常话音特指中等音量的声音）能得到比较好的录放。这里要提一提在 SPCE061A 内部实现的带通滤波，也就是用软件实现数字滤波。这里采用的是一个 15 阶的 Fir 滤波器，在 Matlab 下求得滤波系数，将系数放大了 65535 倍后再四舍五入得到的值作为程序中的滤波系数。滤波后得到的值是有符号数，而 SPCE061A 的 DAC 只能对无符号数进行转换，也就是说，SPCE061A 的 DAC 控制器把所有的数都认为是无符号数，这就需把滤波结果加一个偏移量。采用了 Fir 滤波，既减小了大声音的失真，也提高了小声音的录音效果。本方案提供了一个用

SPCE061A 普通 AD 口录音的例子，大家可以据此实现 SPCE061A 的多路录音。可以设想，在麦克风放大部分加上 AGC 电路，效果会更好。

六、参考资料

《第四届全国大学生电子设计竞赛获奖作品选编（1999）》

《MATLAB 6.x 信号处理》

《NE5532 datasheet》

Supplus