

华中科技大学电子与信息工程系

2011 年 TI 杯电子设计大赛

项目总结报告（团队）

课 题： 数字化语音存储与回放系统

组 员： 柳涛 U200812939 电信 0801 班

 黄攀 U200812864 电信 0801 班

 邵小飞 U200812950 电信 0801 班

指导老师： 曾喻江

日 期： 2011.7.9

目录

1、设计背景.....	3
2、设计目标.....	3
3、团队介绍.....	3
4、系统设计方案与应用场景.....	4
4.1 系统设计分析	4
4.2 系统方框图	4
4.3 应用介绍	4
5、硬件设计与实现.....	4
5.1 硬件设计分析	4
5.2 硬件结构框图	5
5.3 各硬件模块设计	6
5.3.1 音频采集模块	6
5.3.2 音频回放模块	7
5.3.3 单片机最小系统模块	8
5.3.4 人机交互模块	9
6、软件设计与实现.....	10
6.1 系统软件分析	10
6.2 软件流程图	11
6.3 状态转换	11
6.4 重要软件模块设计	13
6.4.1 时钟初始化模块	13
6.4.2 系统初始化模块	13
6.4.3 LCD 显示模块	14
6.4.4 按键模块	14
6.4.5 定时器模块	15
6.4.6 Flash 操作模块	15
6.4.7 数据采集模块	16
6.4.8 数据输出模块	17
7、系统测试与结果.....	18
7.1 单音频率测试	18
7.2 语音测试	19
7.3 音乐测试	19
8、结束语.....	20
8.1 问题与解决	20
8.2 待扩展功能	20
8.3 小结与感悟	20
9、参考文献.....	22
10、附录	23
10.1 作品实物图	23
10.2 整机电路图	24
10.3 源码清单	24

1、设计背景

随着消费类电子市场规模不断扩大，消费类电子产品拥有非常广阔的市场前景。近年来，各种消费类电子产品应运而生，从简单的电子表、录音笔、音乐播放器、多媒体播放器，到智能手机终端、PDA、Laptop 等等层出不穷。出于对消费类电子产品的喜爱和深入了解简单电子产品内部原理和构造，我们选择了这一题目。

2、设计目标

我们主要致力于完成一个具有语音采集，量化，数字语音存储、语音回放、人机交互等功能的数字化语音存储回放系统。系统选择了 TI 公司超低功耗的 16 位单片机 MSP430F169 作为主控制器，结合外部时钟、电源、复位和模拟信号采集、放大、滤波等模块完成系统的设计开发。

系统采用外接键盘和 LED 指示灯作为用户输入和状态显示接口，同时采用液晶显示器 LCD12864 作为信息提示界面，具有良好的用户可操作性和友好性。

- **基本功能：**

1. 实现语音信号的采集、存储、回放；
2. 对信号进行基本和必要的变换，如采集、放大、A/D、D/A、滤波、偏置等；
3. 利用按键和 LCD 液晶显示实现有良好的人机交互性；

- **扩展功能：**

1. 实现系统的低功耗控制；
2. 信号采集的失真度控制；
3. 实现扩展大容量存储器存取控制；

3、团队介绍

- **柳涛**

模拟音频放大器、功率放大器设计，硬件标准板焊制，液晶模块驱动实现，PPT 制作

- **黄攀**

模数/数模转换实现，MSP430 定时器研究，MCU 控制程序设计、实现、调试，文档制作

- **邵小飞**

模拟滤波器设计，FLASH 读、写、擦除等操作研究，硬件电路图绘制

4、系统设计方案与应用场景

4.1 系统设计分析

录音回放功能的实现，关键在于准确的对原音频数据进行记录并存储。音频采集部分将实时采集外界声源信号，作为系统的数据输入接口。数据存储部分将输入的数据保存起来，这是实现回放的基础。音频输出部分能够将存储的数据按照一定的速率恢复成原音频信号，实现回放功能。除了数据处理，系统需要按照用户的意愿执行各种动作，人机交互界面就是负责提供给用户操作选择的接口，并随时反馈给用户当前系统的运行状况。有了数据和控制的来源，系统能够准确全面的体现其价值。

4.2 系统方框图

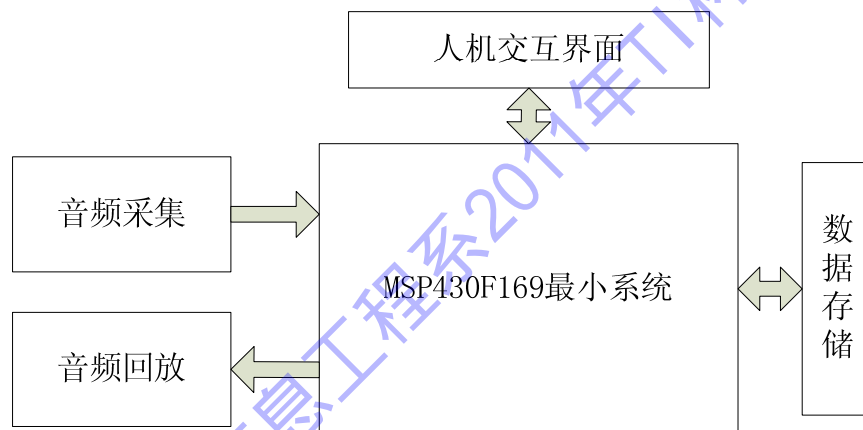


图 1、系统结构框图

4.3 应用介绍

本设备提供四个功能选择按键，分别为录音，播放，停止，内存格式化，每个键按下，如果逻辑合理，会响应相应的动作，LED 和 LCD 作为反馈接口会直观的显示结果用户；如果逻辑不合理，也会提示相应的错误动作。

5、硬件设计与实现

5.1 硬件设计分析

信号录制流程大致为从 MIC 进入，前置放大后经 AD 采样后得到数字化信息。MSP430F169 的 AD 转换器精度为 12 位，满足本课题的精度需求。由于一般语音信号频率峰值在 3.4KHz 左右，因此根据奈奎斯特采样定律，AD 的采样频率应为这一频率的两倍，保留余量，我们选择得到采样频率为 8KHz。采样数据点存储于 Flash 中。音频回放时不

断从 Flash 中读出 16 位数据（高四位无用），由 MSP430F169 自带的 DA 模拟化后输出。该 DA 也是 12 位，因此转换无需作其他处理可直接输出。输出后先滤波，再经过放大和功放之后驱动音响设备回放音频。MCU 主要作为逻辑控制枢纽保证各模块稳步协调工作。

5.2 硬件结构框图

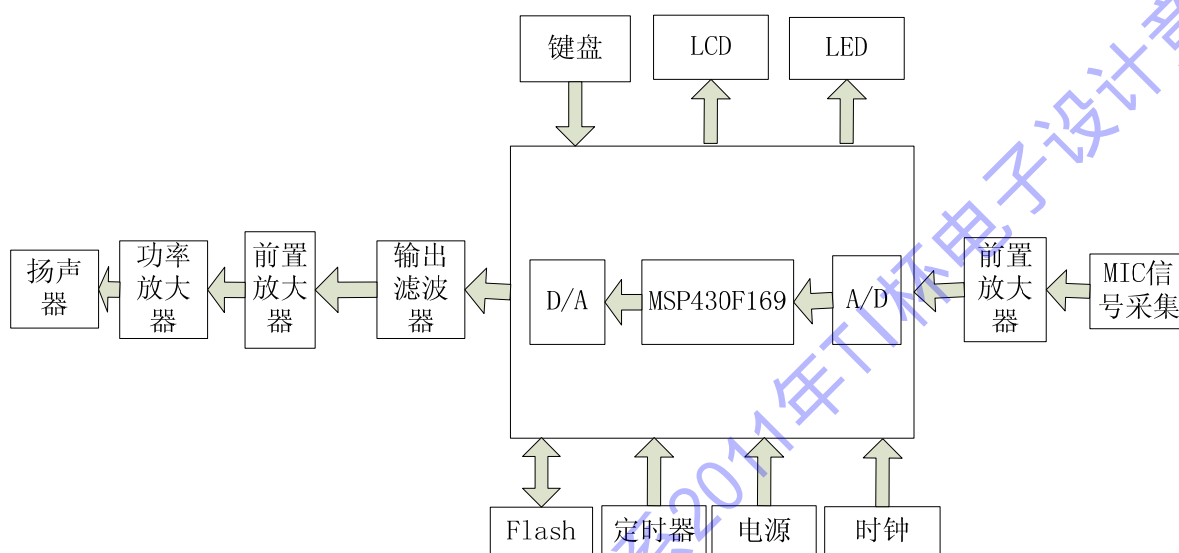


图 2、硬件结构框图

1. 数据采集处理部分：利用拾音器拾取外部语音信号，利用运放实现语音信号的前置放大的，且倍数可调。利用 MSP430F169 内部集成的 12 位 A/D 转换器将放大后的信号进行模拟 - 数字转换。
2. MSP430F169 控制部分：系统设计的核心部分。负责控制整个系统各个模块的工作，实现 A/D 和 D/A 转换以及数字语音信号到 FLASH 的存储控制。
3. 电源部分：系统供电电压及 LCD、放大器等外围模块使用 5V 电压，而 MSP430 微控制器使用 3.3V 电压。由于整个系统各种芯片的工作电压不一致，因此需要电平转换，系统采用电压转换芯片 AMS1117 将 5V 电压转换为 3.3V 供微控制器使用。
4. 按键模块：通过单片机具有中断功能的引脚设计独立按键，通过按键执行相应功能，包括：录音、放音、停止、清空闪存，使用户对系统控制完成相应的操作。
5. 时钟模块：采用了 8MHz 和 32KHz 的无源晶体振荡器为系统提供时钟信号，同时亦可使用 MSP430F169 内部的 DCO 压控振荡器产生时钟，8MHz 是 MSP430F169 所支持的最高时钟频率，而 32KHz 是系统在低功耗模式下工作时的外围器件可用时钟频率。
6. LED 模块：通过 LED 的点亮和熄灭显示电路工作状态：录音或放音、待机等。
7. FLASH 闪存模块：经过采样转换后的语音数据在控制器的控制下存储到外挂 FLASH 存储器中。
8. 回放模块：微控制器将从 FLASH 中取出的语音数据经 D/A 转换后，经过低通有源滤波器、前置放大器，送入功放电路经扬声器回放输出。
9. LCD 显示模块：采用 LCD 点阵字符型液晶显示屏显示时间、操作提示、系统工作

状态、记录时间等内容，提供一个人机交互的友好菜单界面。

5.3 各硬件模块设计

5.3.1 音频采集模块

➤ 受话器模块

受话器我们采用市场上常见的驻极体 MIC，输出信号峰峰值在 20mV 左右，输出阻抗小于 2k 欧。加偏置电阻的目的是为了给 MIC 内部的 FET 放大器提供一个漏极偏置，使其工作在饱和区，完成信号放大的任务。其接口电路如下所示：

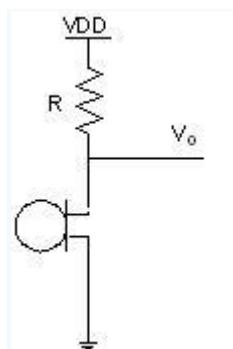


图 3、受话器模块电路

通过查阅相关手册得到 MIC 的最大耗电为 500uA，外围电路供电电压为 5V，如果偏置电阻为 5K 欧，则偏置电压为 2.5V 左右，恰好是供电电压 5V 的一半，这样可以使输出信号的正负半周动态范围相等。因此，将偏置电压设为电源的中点，可以最大化输出的动态范围。经实际测试，发现输出信号峰峰值约为 30~50mV。

➤ 话筒前置放大模块

由于 MIC 输出的电压幅度过小（峰峰值 20mV）左右，所以在输入滤波之间需要先接入前置放大器，目的是无失真的放大语音信号。前置放大器采用 TI 公司的高性能集成语音运放 NE5532，前置放大器采用同向比例放大电路，放大倍数可以根据信号幅度进行调节，这里约为 10 倍，输出电压幅度为 200mV 左右。电路原理图如下所示。

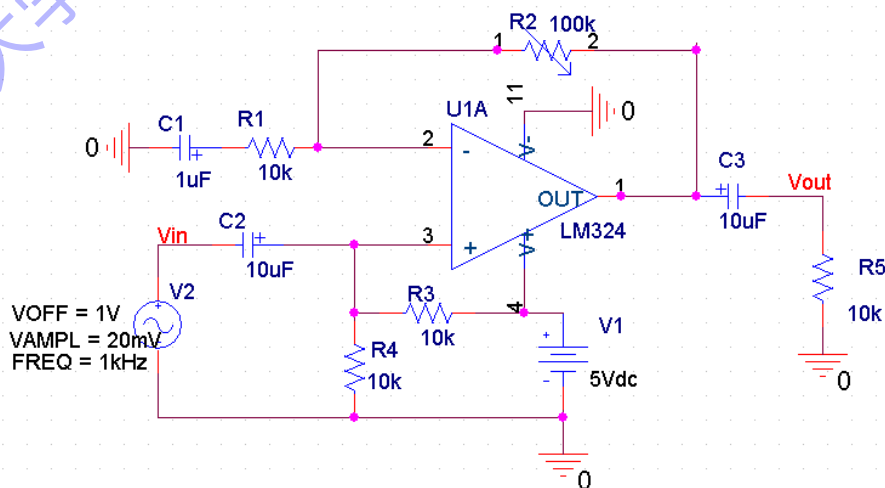


图 4、话音放大电路

电路为反向放大，放大倍数计算公式为： $1+R2/R1$ ，电路设计的放大倍数为 11 倍左右，放大后电压约为 200mV。

5.3.2 音频回放模块

➤ 有源滤波器模块

由于 DA 输出为阶梯状波形，存在很大的量化噪声（高频分量），需要一个低通滤波器进行低通滤波，范围为 $0 \sim 4\text{kHz}$ 。由于有源滤波器效果较之无源滤波器要好，因此我们采用一个二阶低通滤波器来进行滤波，电路如下：

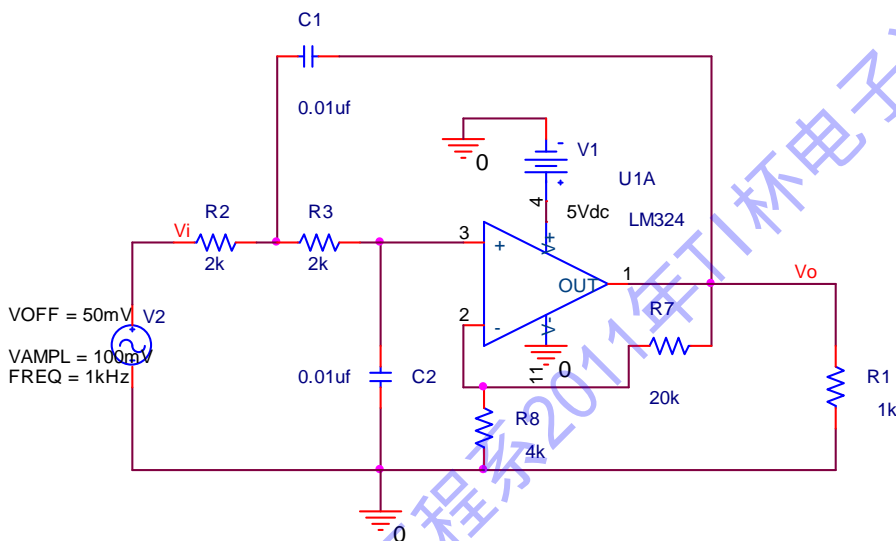


图 5、有源滤波器设计

经过仿真，效果如下：

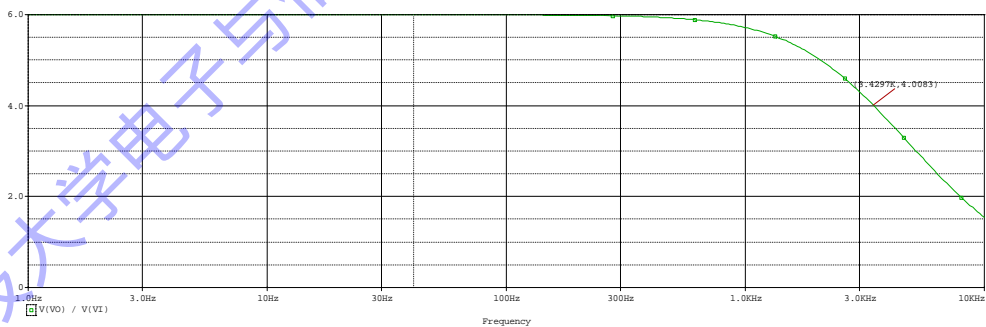


图 6、有源滤波器仿真结果

经硬件调测得出，滤波器带宽达标，并且将输入信号放大约 2 倍。

➤ 前置放大模块

存储的数字语音信号经 D/A 转换后，首先送有源滤波器进行滤波以去除量化噪声，然后送输出级前置放大模块进行初级放大，初级放大电路同样采用集成运放 NE5532，放大倍数约为 50 倍，并通过精密电位器将信号耦合到下一级，实现音量的调节功能，防止信号失真。

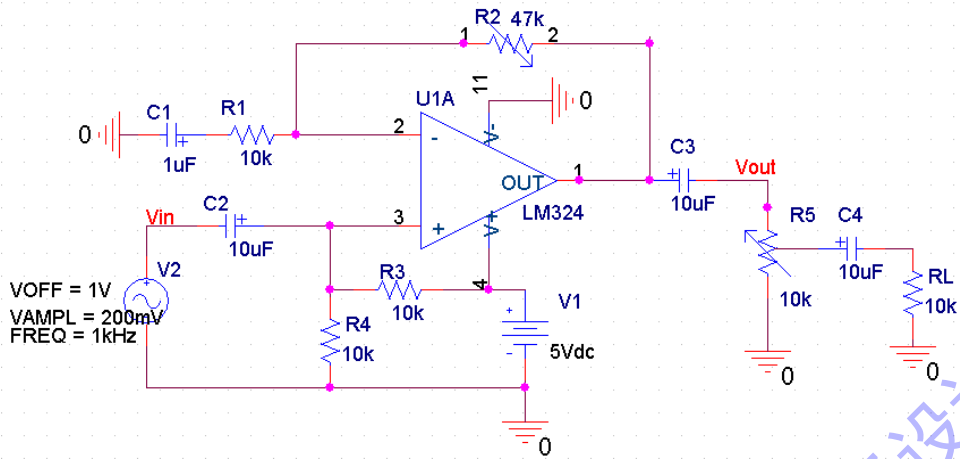


图 7、输出级前置放大电路

➤ 功率放大模块

存储的数字语音信号经 D/A 转换，滤波，前置放大后的信号需要经过功放电路驱动扬声器进行输出。功率放大器采用集成功放 LM386，供电电压为 5V，设计的电路原理图如下所示。

经检验，功放电路的放大倍数约为 20 倍左右。

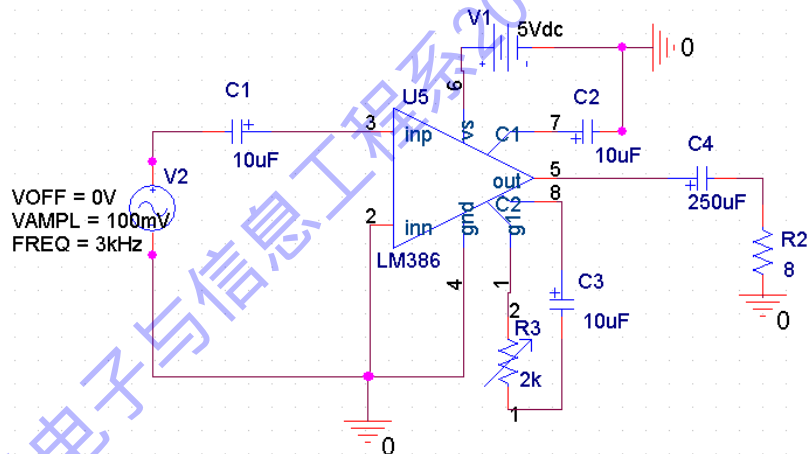


图 8、输出级功率放大电路

5.3.3 单片机最小系统模块

➤ 电源模块

整个系统各种芯片的工作电压不一样，分别为+5 V 和+3.3 V。因此需要有电压从+5 到+3.3V 转换，这里采用 AMS1117 稳压芯片，可将 5V-12V 电压稳压到 3.3V，同时也给单片机提供复位信号。

➤ 时钟模块

时钟是驱动单片机系统各模块工作的根本。MSP 单片机提供了三种时钟：MCLK（系统主时钟），SMCLK（用于高速外围模块）和 ACLK（用于低速外围模块）。其中 ACLK 固定为 32KHz。MCLK 和 SMCLK 可以选择单片机内部的 DCO 或者外部 8MHz

的无源晶体振荡。时钟频率越高，运行速度越快，但一些外围模块可能跟不上 CPU 速度而导致工作不正常，如 LCD 显示模块。速度过低，则可能导致采样速率不能得到实时存储而导致信息丢失。因此，经过协调与测试，我们采用 DCO 的中等频率作为系统主时钟频率，大约为 1.046MHz。SMCLK 与 MCLK 相同。

➤ 定时器模块

对定时器进行编程，可使其产生 8KHz 的频率中断。从而作为 AD 和 DA 的同步时钟驱动，保证回放音频信号与原信号的频率一致性。

➤ AD 采样模块

AD 是模拟信号到数字信号的转换媒介，其精度越高，量化误差越小。单片机自带的 12 位 AD 很好的满足了采样精度需求。采样的频率越高，相对而言每个频率周期波形能够采样的信息点越多，对于恢复原波形越有利。但对于 Flash 的容量要求也就越大。经过权衡，我们设定的采样率为 8KHz，选择的模拟输入通道为 A0，即 P6.0 端口。

➤ DA 转换模块

DA 是数字信号到模拟信号转换的媒介。由于其精度与 AD 一致，只要保证输出数据速率和参考电压与 AD 一致，就能够保证同频同幅的还原信号。但由于输出为阶梯状波形，且音频越高，阶梯状越明显，输出波形会携带大量高频分量，这样的波形是不能直接驱动功放的，必须先滤波，去除高频分量。DA 选择的输出通道为 0 通道，即 P6.6 口。

➤ 数据存储模块

为了保证实时读写速率，我们采用片内 Flash 存储数据。Flash 的非易失性保证了存储的数据能够多次的播放。而其可擦除可编程特性满足了多次录制，随时更新的需求。片内 Flash 的擦写速率主要由主频率决定，因此主频不能过低。

5.3.4 人机交互模块

➤ 按键模块

按键是用户操作本机的唯一方式，由不同的功能按键的按下触发相应端口中断，从而执行不同的功能。本系统设置了四个按键，分别为格式化内存，录制，播放，停止四个功能。由于按键较少，直接将按键接上拉电阻后与 P2.4-P2.7 端口相连。按键没按下时，端口为高电平，一旦按下，变为低电平，其下降沿引发相应端口中断。

➤ LED 模块

通过 LED 的亮和灭显示状态信息。LED 反应的操作中的过程分别为程序是否在正常运行，清空内存中，录制音频中，播放音频中。四个 LED 分别通过串联电阻与 P6.1-P6.4 端口相连，端口高电平则亮，否则不亮。

➤ LCD 模块

LCD 是唯一可以显示非数字文本的设备，是一个很好的人机交互界面。可以将

所有文本交由 LCD 显示。本系统选用的 LCD12864 是一个 4 行 8 列的液晶屏，一方面，可以提示用户相应的按键功能，方便用户选择；另一方面，可以显示当前系统的运行状态，以便实时掌控。LCD 共 13 根引脚，其中两个是电源与地线，另外三根控制线分别与 P3.5-P3.7 相连，8 根数据线与 P4.0-P4.7 端口相连。

6、软件设计与实现

6.1 系统软件分析

系统的软件实现主要是 MSP430 单片机程序控制和数字信号存放，通过实现各个模块的功能来实现系统工作。IAR 的 Embedded Workbench 是一个适应不同 CPU 的目标系统开发集成环境，它提供方便丰富的窗口界面，使开发效率大大提高。其中 IAR 的 C430 编译器提供了 C 语言的标准特性，并且添加了许多有利于 MSP430 系列特性而设计的扩展功能。由于 MSP430F169 有 JTAG 调试接口和电擦写的 FLASH 存储器，因此采用先下载程序到 FLASH 内，再在器件内通过软件控制程序运行，由 JTAG 接口读取片内信息观察程序运行情况，可方便调试和修改程序。

考虑到本系统是以中断驱动的，用户通过功能选择按键进行选择按键后，就会产生不同的中断，系统必须实时响应这些中断，通过中断服务程序作出相应的动作。而每次作出相应，意味着程序状态的跳转。共设置了四种状态，即：空闲状态 (IDLE)，音频录制状态 (Recording)，音频播放状态 (Playingback) 和内存格式化状态 (Erasing)。空闲状态意味着不做任何操作；处于音频录制状态时，进行音频数据采集与存储操作；处于音频播放状态时，进行数据读取与回放状态；如果在内存格式化状态时，则进行系统内部操作，将 Flash 中存放的旧数据全部擦除。

对于 AD 采集数据信息以及 DA 回放数据信息模块，要保证原始音频频率不漂移，我们使用定时器来保证 8KHz 的中断操作。对 AD 和 DA 的操作都放在定时器中断服务程序中去执行，这样不会被一些误操作打断。而在对 AD 读数之后和 DA 放入数据之前，都要对 Flash 进行操作，一写一读，Flash 的读写可以作为单独的 Flash 驱动函数，读 Flash 的函数输入参数为 16 位地址，返回参数为 16 位数据；写 Flash 的函数输入参数为 16 位地址和 16 位数据，无返回参数。

还有一个很重要的模块是用户交互显示模块。这里采用 LED 二极管和 LCD 液晶两种显示方式。点亮 LED 发光二极管只需对相应的端口值赋值“1”，灭灯为“0”。而写 LCD 需要用到专门的驱动函数。附录中提供的驱动函数每次可以从任意一行的任意一列开始，写一个汉字字符串。每次操作开始或结束后，都可以刷新 LCD 进行相应的界面显示提示。

6.2 软件流程图

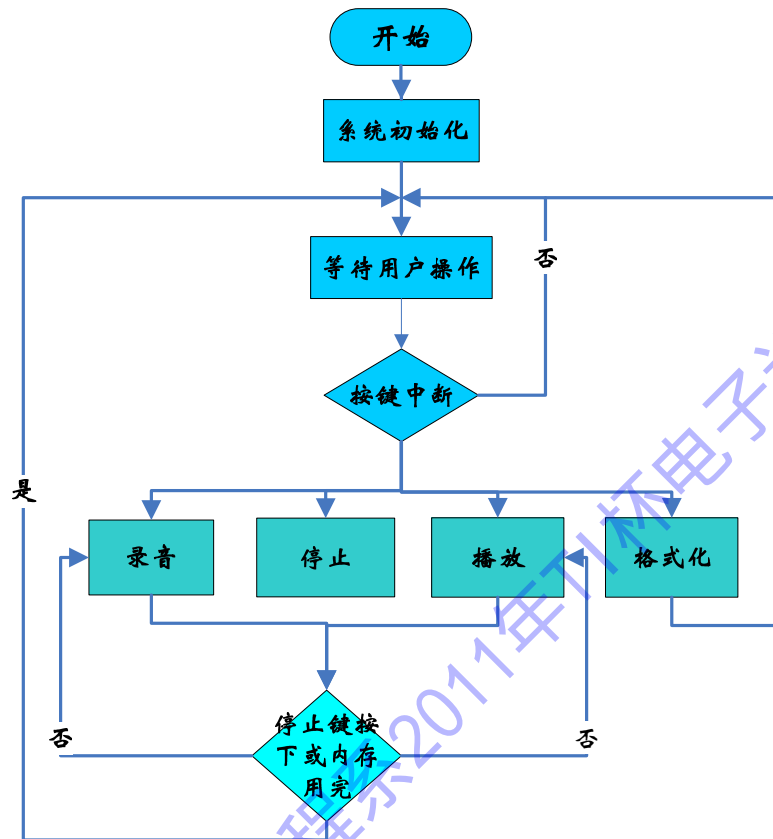


图 9、软件流程图

6.3 状态转换

状态转换是程序的核心。设计合理有效的状态转换逻辑是关键。

系统初始化后默认进入等待状态“IDLE”，等待状态下等待用户的输入，准备状态转移。

当用户在等待状态下按下录音键，应启动录音，工作状态转移到录音“Recording”。录音状态下执行录音操作直到按下停止键表示停止录音，或者用于存放音频的 Flash 空间占满，或者录音过程中遇到错误，这时切回等待状态。

当用户在等待状态下按下播放键，应启动播放，工作状态转移到播放“Playing”。播放状态下执行播放操作直到按下停止键表示停止播放，或者已经播放的长度大于等于 Flash 内数据表示播放完成，这时切换回等待状态。

当用户在等待状态下按下擦除键，应启动内存格式化，工作状态转移到格式化“Erasing”。格式化状态下其他按键失效，只有等到格式化完毕时才能切换回等待状态。最终状态转换图如下：

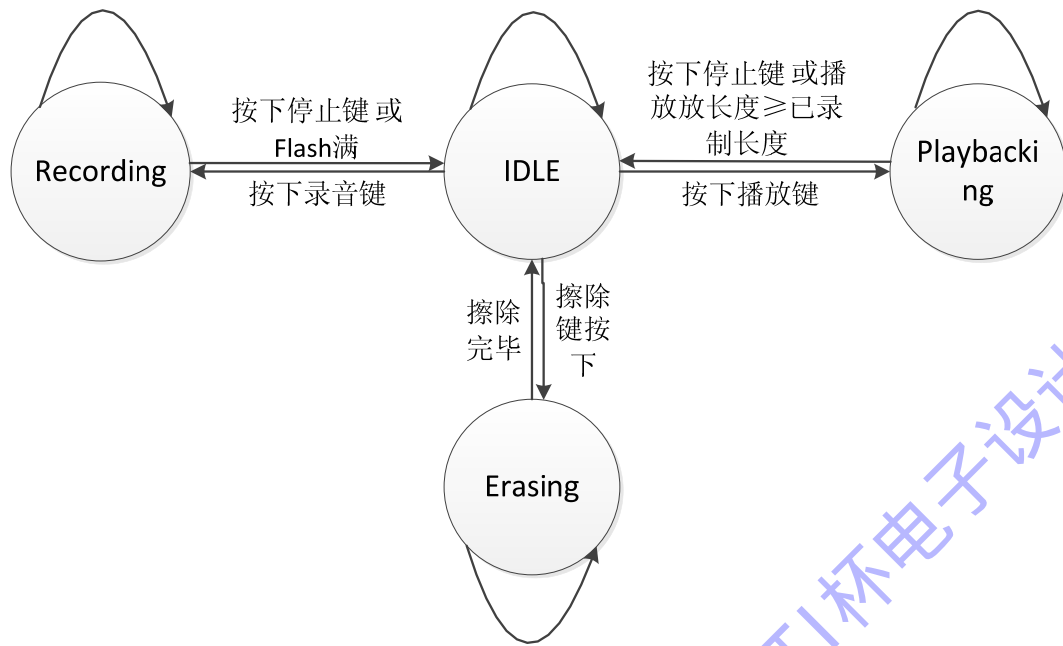


图 10、状态转换图

状态转换核心代码如下：

```

while(1)
{
    //每次都通过检查状态标志进入相应的状态处理
    if(fun_flag==SEL_PLAYBACK)
    {
        P2IE=0x20; //只允许停止按键的中断
        Playback(); //调用播放处理函数
        fun_flag=IDLE; //修改状态标志
        P2IE=0xf0; //恢复所有按键中断
    }
    else if(fun_flag==SEL_RECORD)
    {
        P2IE=0x20; //只允许停止按键的中断
        Record(); //调用录制处理函数
        fun_flag=IDLE; //修改状态标志
        P2IE=0xf0; //恢复所有按键中断
    }
    else if(fun_flag==SEL_ERASE)
    {
        _DINT(); //暂停所有中断
        FlashErase(); //调用内存格式化函数
        fun_flag=IDLE; //修改状态标志
        hanzi_Disp(1,0,"请按键");
        hanzi_Disp(2,0,"0 清除 1 播放");
        hanzi_Disp(3,0,"2 停止 3 录制");
    }
}
  
```

```

    _EINT(); //开中断
}
else
{
    _NOP();
}
}

```

6.4 重要软件模块设计

6.4.1 时钟初始化模块

开机运行程序时进行时钟的初始化操作，选择 DCO 作为时钟源，其振荡频率选择为中等振频，约为 1.046MHz。代码如下：

```

void Init_Clock(void)
{
    DCOCTL=DCO2+DCO1+DCO0; //DCO 为最高频率
    BCSCCTL1=RSEL2;//+RSEL1+RSEL0; //最高的标称频率
    _NOP(); //MCLK 时钟源为 DCO,CLK 不分频, SMCLK 时钟源为 DCO
}

```

6.4.2 系统初始化模块

时钟初始化结束后开始端口以及各外围模块初始化。其中端口 6 的 0 号和 6 号口分别作为 AD 和 DA 与外围模拟电路的通道口，端口 2 的高四位口作为按键中断口，Flash 的时钟源为 MCLK 二分频，LCD 有单独的初始化函数，直接调用即可。代码如下：

```

void Init_Sys(void)
{
    P1OUT = 0x00;
    P1DIR = 0xff;
    P1IES = 0x00;
    P2OUT = 0x00;
    P2DIR = 0x0f;
    P3OUT = 0x00;
    P3DIR = 0xff;
    P4OUT = 0x00;
    P4DIR = 0xff;
    P5OUT = 0x00;
    P5DIR = 0xff;
    P6OUT = 0x00;
    P6DIR = 0xfe; //P6.1/2/3/4 点亮 LED
    P6SEL = 0x01; // ADC12 A0 通道
    P2IES = 0xf0; //P2IES 高四位作为按键中断口，下降沿触发
}

```

```

P2IFG = 0;
P2IE  = 0xf0;
fun_flag = IDLE;
FCTL2 = FWKEY + FSSEL_2 + FN0;    //FLASH 时钟源选择:f=SMCLK/2
Lcd_Init();
}

```

6.4.3 LCD 显示模块

LCD 的驱动函数在 lcd12864.c 文件中有定义，这里直接可调用函数 hanzi_Displ(x, y, s) 来显示汉字，x 为行序号，y 为列序号，s 为汉字字符串。

6.4.4 按键模块

所有的端口 2 上的中断共享一个中断处理函数，在中断处理的时候要对是哪个口触发的中断进行判断，这时候只要看哪个口上的输入电平为低，则中断由它触发。然后修改相应的状态标志即可在主程序中进入相应的处理。端口中断处理代码如下：

```

#pragma vector=PORT2_VECTOR
__interrupt void Port_2(void)
{
    unsigned int ii=0;
    for(;ii<100;ii++);
    switch(P2IN&0xf0)
    {
        case STOP_FLAG : //停止键按下
            fun_flag=IDLE;
            record_flag=OFF;
            playback_flag=OFF;
            break;
        case PLAYBACK_FLAG://录音键按下
            fun_flag=SEL_PLAYBACK;
            break;
        case RECORD_FLAG://播放键按下
            fun_flag=SEL_RECORD;
            break;
        case ERASE_FLAG: //擦除键按下
            fun_flag=SEL_ERASE;
            break;
        default:
            break;
    }
    P2IFG=0; //清空中断标志位
}

```

6.4.5 定时器模块

定时器选择 TimerA，用来产生 8KHz 的中断，供 AD 和 DA 的频率校准。定时器的时钟源选择为主频，分频系数为 125，大约每隔 0.125ms 会产生一次定时计数中断。下面的代码是定时器初始化函数：

```
void Init_TimerA(void)
```

```
{
```

```
    TACCTL0=CCIE; //定时中断使能
```

```
    TACTL=TASSEL1+TACLK+MCO; //时钟源选择 SMCLK:1024KHz 时钟源,增计数模式
```

```
    if(record_flag==ON)
```

```
        TACCRO=TIMERFN; //N 分频，产生 8KHz 定时中断
```

```
    else if(playback_flag==ON)
```

```
        TACCRO=TIMERFN+30;
```

```
    else
```

```
        _NOP();
```

```
}
```

6.4.6 Flash 操作模块

Flash 的操作分为三种：读、写、擦出。其中读的速度最快，每次只读出 16 位数据。写较耗时，每次写 16 位数据。擦除很耗时，每次擦出是清空全部存储空间。

Flash 读函数代码如下：

```
unsigned int FlashRW(unsigned Adr)
```

```
{
```

```
    unsigned int data;
```

```
    data*((unsigned int *)Adr); //数值强制转换为指针
```

```
    return data;
```

```
}
```

Flash 写函数代码如下：

```
void FlashWW(unsigned int Adr,unsigned int data)
```

```
{
```

```
    FCTL1=FWKEY+WRT; //Flash 写使能
```

```
    FCTL3=FWKEY; //解锁操作
```

```
    *((unsigned int *)Adr)=data; //写入数据
```

```
    FCTL1=FWKEY;
```

```
    FCTL3=FWKEY+LOCK;
```

```
}
```

Flash 擦除函数代码如下：

```
void FlashErase(void)
```

```
{
```

```
    Print_LCD(0,"内存初始化，等候"); //LCD 显示当前状态
```

```
    P6OUT|=0x08; //点亮 LED
```

```

unsigned int *pMemory = (unsigned int *)Memstart;
do
{
    FCTL1 = FWKEY + ERASE;
    FCTL3=FWKEY;
    *pMemory = 0; // 段擦除
    pMemory += 0x0100; //指针移动到下一段
} while (pMemory < (unsigned int *)Memend);
P6OUT^=0x08;
Print_LCD(0,"内存初始化完毕");
}

```

6.4.7 数据采集模块

AD 采用单通道单次采样模式，P6.0 端口接输入信号，ADC12MEM0 存放输出数据。由 Timer_A 产生频率为 8KHz 的定时中断，每次中断处理到来时，读入转换完成的值并存储。然后开启下一次的采样转换使能之后，中断处理结束，等到下次定时中断就能读取新的数据。

录制函数代码如下：

```

void Record(void)
{
    address=Memstart; //存储起始点
    record_flag=ON; //录音标志置位
    Init_ADC(); //初始化 AD
    Init_Flash(); //每次录制都要先格式化内存
    Print_LCD(0,"开始录音");
    Init_TimerA(); //初始化时钟
    P6OUT|=0x02; //点亮相应 LED
    while(record_flag==ON); //等定时中断采样存储，若停止按键按下，则退出
    _DINT();
    Close_ADC(); //关闭 AD
    Close_TimerA(); //关闭定时器
    P6OUT^=0x02; //LED 灭
    _EINT();
    Print_LCD(0,"录音结束，请按键");
}

```

定时器中断处理函数中关于录音部分的代码如下：

```

if(fun_flag==SEL_RECORD)//进行录音，采样数据点
{
    if(address<Memend)
    {
        temp=ADC12MEM0; //读取转换数据
        FlashWW(address,temp); //写入 Flash 中
    }
}

```



```

address+=2; //全局变量地址加 2
ADC12CTL0&=~ENC;
ADC12CTL0|=ENC+ADC12SC; //开启一次新的采样
}
else
{
    record_flag=OFF; //Flash 写满，自动退出
}
}
}

```

6.4.8 数据输出模块

DAC12 也采用 8KHz 定时中断，每次中断从 FLASH 中顺序读取数据，放到 D/A 输出寄存器中，使能输出。DA 采用单量程、中等转换速率。DA 的参考电压与 AD 相同。

回放函数代码如下：

```

void Playback(void)
{
    address=Memstart; //读取起始点
    playback_flag=ON; //回放标志位置位
    Print_LCD(0,"播放中"); //显示当前操作
    Init_DAC(); //初始化 DA
    Init_TimerA(); //初始化定时器
    P6OUT|=0x04; //点亮 LED
    while(playback_flag==ON); //等定时中断播放，若停止按键按下，则退出
    _DINT();
    Close_TimerA(); //关闭定时器
    Close_DAC(); //关闭 DA
    P6OUT^=0x04; //灯灭
    _EINT();
    Print_LCD(0,"播放完毕，请按键");
}

```

定时器中断处理函数中关于回放部分的代码如下：

```

else if(fun_flag==SEL_PLAYBACK)//进行播放，读出数据
{
    if(address<Memend)
    {
        temp=FlashRW(address); //从 Flash 中读出数据
        DAC12_ODAT=temp; //从 DA 输出
        address+=2; //地址加 2
    }
    else
    {
        playback_flag=OFF; //所有数据播放完，自动退出
    }
}
}

```

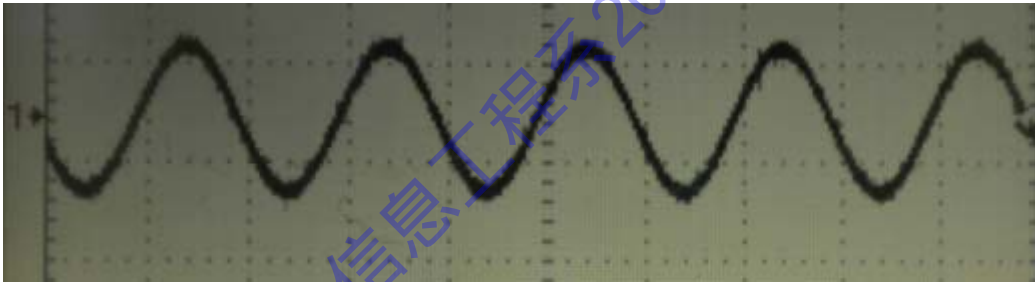
7、系统测试与结果

7.1 单音频率测试

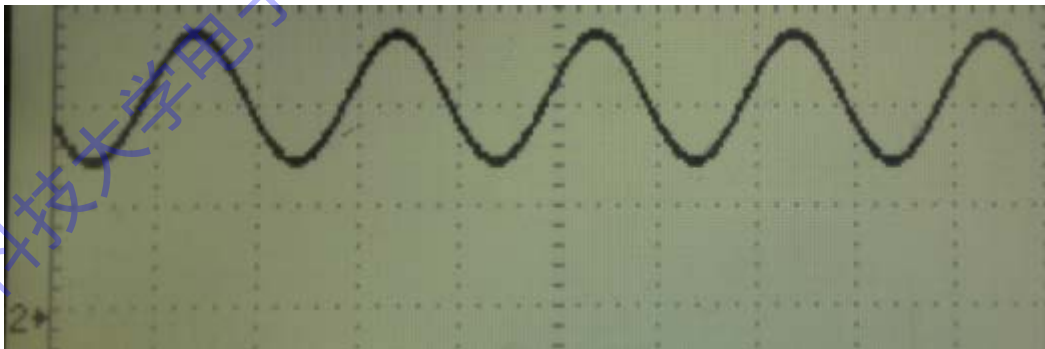
首先选用频率分别为 500Hz, 1KHz, 1500KHz, 2000KHz 的正弦波从音频采集模块的放大器输入端输入, 经过放大, AD, DA, 滤波, 再放大, 功放, 在功放输出端接一 10 欧功率电阻作为负载。

分别用示波器观察各个模块的输出波形。经过测试, 发现, 音频采集端前置放大器放大倍数约为 8 倍, 滤波器滤波效果良好, 放大倍数约为 1 倍, 音频输出端放大器放大倍数约为 1.5 倍, 功放输出端波形峰峰值可达到 3V 左右且不失真。测试结果在预期之中。下列图片为频率为 1KHz 时的各级输出波形: 模块测试采用逐级输出测试的测试方法, 由信号发生器提供峰峰值约为 80mV, 频率为 1KHz 的正弦信号, 作为模拟 MIC 输入的测试信号, 用探针分别接以后各级输出, 观测波形的幅度和频率, 分析失真情况, 并据此进行系统的调试。

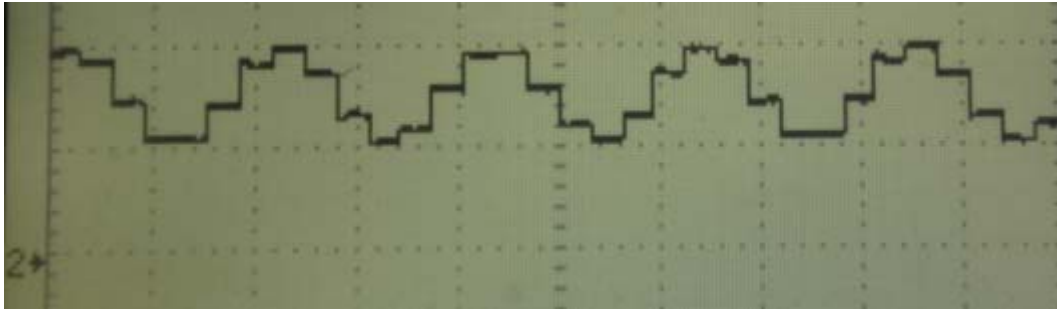
- 1、信号发生器输入信号 (峰峰值: 80mV, 频率: 1KHz)



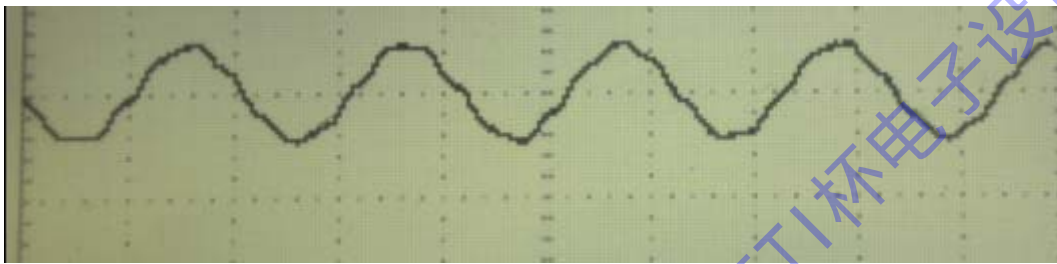
- 2、语音放大器输出信号 (加直流偏置 0.8V, 峰峰值: 680mV, 频率: 1KHz)



- 3、D/A 转换输出信号 (含直流偏置, 峰峰值: 500mV, 频率: 980Hz)



4、有源低通滤波输出信号（峰峰值：960mV，频率：950HZ）



5、功率放大电路输出信号（峰峰值：2.96V，频率：950HZ）



从模块测试结果可以看出，在输入信号频率较小时，系统可以实现几乎无失真的恢复采样信号。但是从频率上可以看出，恢复出的波形信号与原采样信号之间存在一定的频率失真，这是由于 A/D 转换时的速率与 D/A 输出速率不匹配导致的，可以通过更加精确的时延来实现两者的同步。

7.2 语音测试

使用 MIC 采集人的说话语音，并通过外接音箱回放，能够清楚的听清语音。

7.3 音乐测试

用耳机接线从音频输入端输入音乐信号，从外接音箱回放，音乐恢复的效果良好。

8、结束语

8.1 问题与解决

本次硬件课设中，我们碰到了很多平时未曾碰到的问题，最终都通过查阅资料，向老师请教，自己摸索等方式得到解决，下面是一些例子：

问题 1：我们购买了 MSP430 芯片，最小系统板，和板上的元件，开始是自己动手焊 430 的芯片，结果由于经验不足，芯片焊失败作废了。但这块芯片很贵，再买一块会超支很多，没有芯片又无法开展工作，让人很沮丧。

解决办法：听说可以向 TI 公司申请样片，我们抱着一丝希望申请了一片 430 芯片，居然成功了，芯片问题解决。

问题 2：AD 和 DA 最低参考电压为 0，不能够采集负的波形信号，直接采样音频信号会截止失真。

解决办法：我们在 AD 输入之前给输入信号加一个直流偏置，使得信号负电位抬高到正电位，DA 输出的时候加一个隔直流电容，滤除直流电压。

问题 3：在焊接好 LED，键盘到标准板上以后，发现之前正确的中断逻辑、二极管放光逻辑都变得混乱起来。开始我们以为是元件焊接有问题，但仔细用万用表检查后发现不是焊接问题。

解决办法：后来发现单片机与板上的电源不是共地端的，造成了电位逻辑的混乱。我们把两个“GND”连起来后，问题得到很好解决。

问题 4：在单独的一个工程中测试 LCD12864 可以很好的工作，一旦加入到系统中，显示就不正常了。

解决办法：时钟问题。LCD 承受的时钟频率有限，而我最初采用的系统时钟是 32MHz 晶振，这个对 LCD 来说太高了。后来经调试改为 1MHz，可以工作。

问题 5：本来定时器的时钟源是 ACLK 32KHz，四分频后即可产生 8KHz 中断。后来测试的时候确怎么进入不了定时中断了。

解决办法：网上看到有人说，JTAG 与 ACLK 可能会有冲突，我们把程序加载到系统板上后拔掉 JTAG，果然能正常工作了。后来就把定时器时钟源换成了 SMCLK，就可以在 JTAG 下正常中断了。

8.2 待扩展功能

由于我们使用的是片内 60KB Flash，以 8KHz 的采样率只能存放几秒钟的数据，若果能够扩展到外围 Flash，便能大大增加数据容量，效果会更好。

8.3 小结与感悟

柳涛

历时两个星期的硬件课程设计终于落下帷幕。在这两个星期的时间内，我们尝过了太多的酸甜苦辣，有过太多痛苦的挣扎，失败的伤感以及成功的喜悦。

在项目开始后的前三天，由于我们之前准备的 430 单片机最小系统板主控芯片出现故障（烧坏），而出于预算经费的考虑，我们不得不临时在 TI 的官方网站上申请样

片，并且购买最小系统空板及外围器件，自行完成了 MSP430F169 最小系统板的焊接和调测。

终于，我们在第一周周三拿到了试验箱并在周四收到了 TI 的样片，完成了最小系统的焊接测试，并且成功的使用购买的 USB 型 JTAG 仿真器将程序代码下载到 430 单片机并且实现跟踪调试。这让我们对以后的工作有了信心。

截至第一周周五，我们完成了 LCD、A/D、D/A、FLASH、按键等模块代码的编写工作，并在周六完成了硬件电路板的焊接和模拟电路部分的调测工作。第一周周六和周日两天，我们开始了艰难的软硬件连调工作。由于我们是第一次接触 430 单片机，遇到了各种各样的问题，诸如：调试器与系统板的接口兼容性问题，JTAG 调试器与系统辅助时钟的冲突问题（在 JTAG 在线调试时，系统的 ACLK 时钟无法正常工作），FLASH 存储器读写速率和采样频率的匹配问题，最小系统板和外围电路的电平不匹配问题等等。

经过两天的艰难调试，我们终于可以在第一周周日晚实现了系统的基本功能测试，当从 D/A 的输出端看到了复原后的波形时，我们无不畅快的舒了一口气，两天的调试终于开始见到了转机。

从第二周开始，我们开始系统的健壮性完善工作和细节调试。在这时，我们发现了 LCD 液晶显示器读写时钟问题（LCD 的读写速率不能跟上 CPU 的工作速率），起初出于读写 FLASH 的考虑，我们将时钟频率设置在最高频率 8M，由此导致了 LCD 显示器速率无法匹配上系统时钟，导致了显示器不工作，为此我在寝室调试到凌晨 3 点，终于发现了系统的时钟频率与 LCD 读写的矛盾。

紧接着，各种各样的问题慢慢浮现，电路虚焊、外围电路与最小系统电路公共接地端问题、外部 FLASH 扩展问题、功率放大信号失真问题等等接踵而至，面对一个又一个棘手的问题，我们仿佛已经有了抗体一般，终于成功的予以一一击破！

我认为这次课程设计给我最大的收获就是如何在各种各样的问题中寻找解决方案，从项目一开始，我们便一直问题不断，然而，在解决一个个问题的过程中，我们一直采取积极的心态和主动的学习态度，我们积极查询各方面的资料、向老师和同学求助、与老师交流我们遇到的问题、在队内积极展开讨论。

相信经过这次课程设计的锻炼，每一个成员都能够获得充分的锻炼，感谢曾喻江老师的督促和意见，感谢汪小燕老师每天监督、义务加班为了我们能够在晚上继续留在实验室进行课程设计，感谢队友的给力，让我们能够如期完成设计任务！

黄攀

从一个以前都没碰过单片机的门外汉，到几乎把单片机功能模块，包括时钟、定时器、AD、DA、IO 端口、中断、Flash、DMA 都摸索一遍，只是短短十几天的事。虽然不能号称熟悉，但不会再畏惧各种管脚，各种读写时序。这样一次硬件课设给我带来的，是能力上的提升，更是自信心上的提升。即使这期间常常碰到各种明明在原理上无懈可击在板子上就是跑不通的问题，亦或是本来已经跑通了却在后面的测试中十分不给力的模块，我们都硬着头皮，忍着火气，拔掉一颗颗钉子，挺了过来。很多问题都容易让人有崩溃放弃的想法，但问题越困难，解决困难之后的那份自豪与喜悦就越发有分量。动手实践真的是检验真知的一大法宝。以前我们学奈奎斯特采样率，学 AD/DA，只知道记公式采样频率要是最大频率的两倍以上才行，只知道 AD/DA 是模数转换方式，却不知道为什么采样频率要这么大，模数信号相互转换之间的关系。真正做了测试，看来示波器，调试了程序之后，才明白恢复一个正弦信号要至少两个点，所以采样频率至少要是信号频率两倍，才真切感受到采样量化后再输出的阶梯状波形

究竟会带来怎样的失真。同时，团队的力量第一次真真切切的给我留下了深刻印象，分工合作，集思广益这些词语已经深深刻在脑海中。

邵小飞

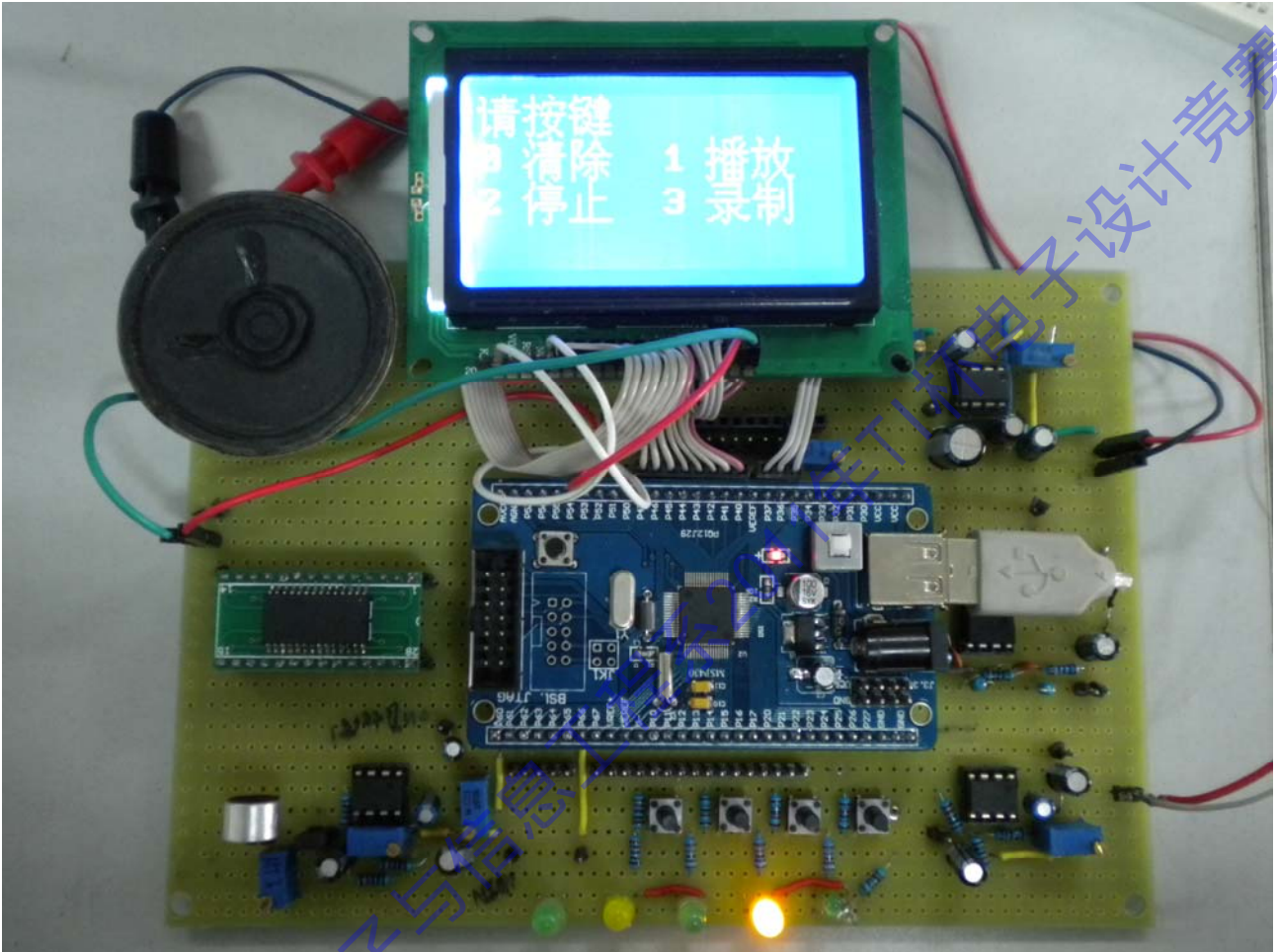
通过参加 TI 杯电子设计大赛，让我学到了很多的东西，让我不在拘泥于书本，将书本上的知识转化为实际操作应用时用以解决问题的能力，锻炼了自己的手动能力和技巧，知识活学活用，反过来对书本上的东西有了更加深刻的理解。通过 TI 杯电子设计大赛，让我对做出一个作品的整体过程有了一个系统的了解，课题设想，初步方案提出，具体方案设计，硬件选择，各功能模块设计，系统联调，扩展延伸，锻炼了自己的思维，对以后工作起了一定的影响。当然，参加这个竞赛，重要的不是结果，而是过程，期间我们自己挑选器材，自己制作硬件，团队商讨方案，团队合作联调，克服一个又一个困难，在这个过程中，我学到了很多，体会到了很多，相信在以后的学习工作中，都会有很大的好处。

9、参考文献

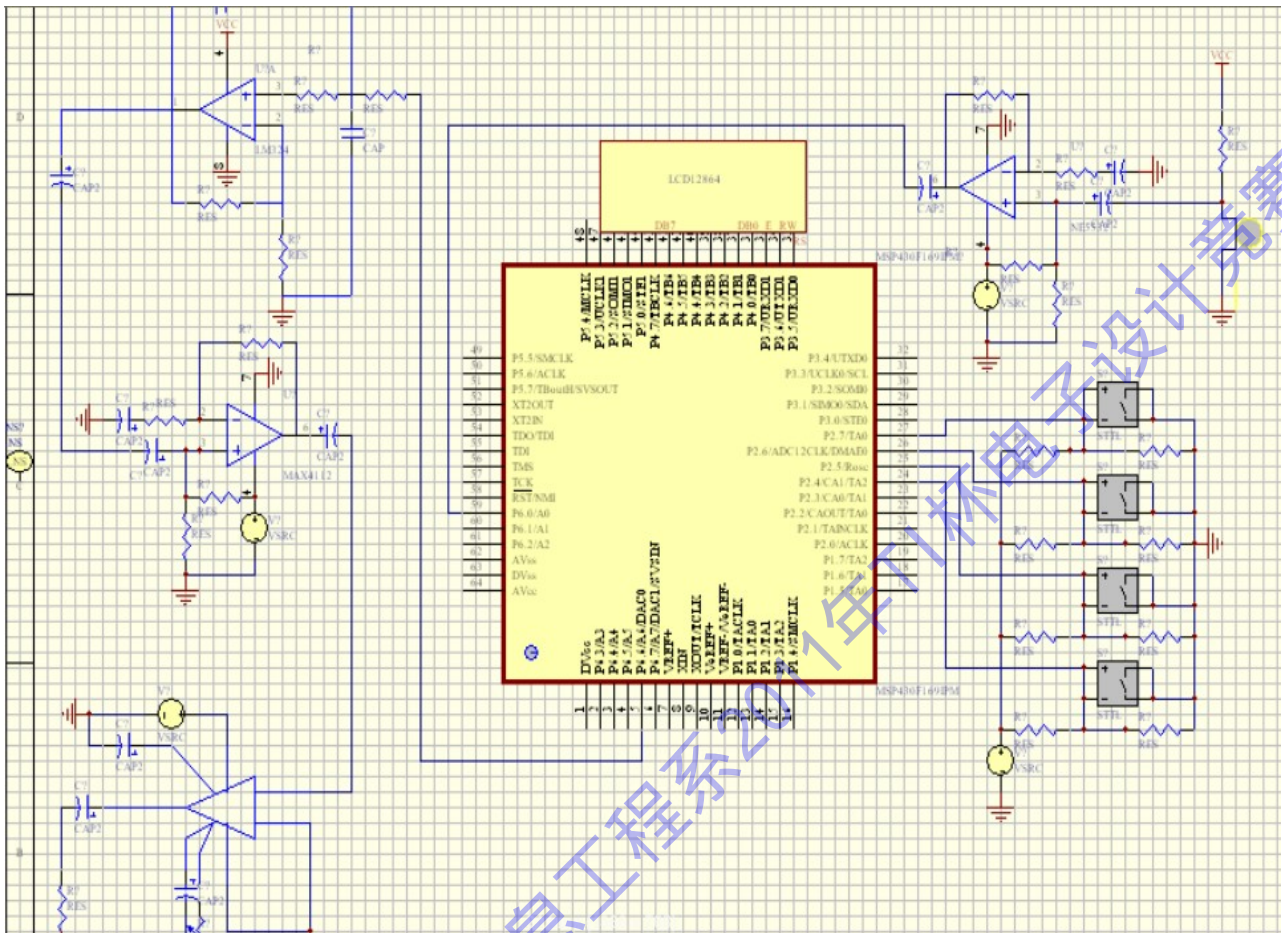
1. 《MSP430 系列 16 位超低功耗单片机原理与实践》 沈建华、杨艳琴 编著，北京航空航天大学出版社
2. 《MSP430 单片机基础与实践》 谢兴红、林凡强、吴雄英 编著，北京航空航天大学出版社
3. 《MSP430 系列 Flash 型超低功耗 16 位单片机》 胡大可 编著，北京航空航天大学出版社
4. 《MSP430 系列单片机 C 语言程序设计与开发》 胡大可 编著，北京航空航天大学出版社
5. TI. MSP430F169 datasheet
6. TI. NE5532 datasheet
7. NS. LM386 datasheet

10、附录

10.1 作品实物图



10.2 整机电路图



10.3 源码清单

- Main.c — 主函数文件
- Lcd12864.c — 液晶模块驱动程序函数文件
- Lcd12864.h — 液晶模块驱动程序头文件
- Msp430f169.h — msp430 单片机头文件