



西安建筑科技大学

课程设计（论文）

课程名称：____单片机应用课程设计____

题 目：____电子贺卡____

院（系）：____信息与控制工程学院____

专业班级：____计算机 1802____

姓 名：____李杨____

学 号：____1808010222____

指导教师：____李宣妮____

2020 年 08 月 20 日

西安建筑科技大学单片机课程设计任务书

专业班级：计算机 1802

学生姓名：李杨

指导教师（签名）：_____

1. 课程设计题目

电子贺卡

2. 本次课程设计应达到的目的

大作业的基本教学目的是培养学生综合运用所学的基础理论、专业知识和基本技能，提高分析与解决问题的能力以及科技写作或设计能力。要求学生在做大作业的过程中要有意识地培养自己的实际能力。通过本次大作业的实践，了解单片机工作的原理及应用技术，掌握根据硬件电路设计软件的方法，了解设计过程中的各个基本环节，也为今后的实际应用奠定基础。

3. 本次大作业（论文）任务的主要内容和要求（包括原始数据、技术参数、设计要求等）

操作对象：Protues单片机仿真。

操作要求：完成电路绘制，软件编制及实验报告。

设计内容：A设计基于单片机的伴奏乐曲 B LCD12864显示图片 C通过矩阵键盘按键显示不同图片，并伴有不同乐曲 D创新部分（自主设计）

要求：1) 完成设计方案。

2) 设计电路图，用 proteus 绘制图纸。

3) 编制程序。

4) 仿真调试成功

5) 撰写工作报告。

6) 提交报告并演示作品。

4. 应收集的资料及主要参考文献

[1] 丁有军 段中兴 何波 花新峰 周方晓 丁莉，《单片机原理及应用教程（c语言）》（第一版），人民邮电出版社，2018.9。

[2] 潘永雄，《新编单片机原理与应用》，西安电子科技大学出版社，2011.12。

[3] 彭伟，《单片机C语言程序设计实训100例——基于PIC+Proteus仿真》，电子工业出

5. 审核批准意见

教研室主任（签字）_____



摘要

本次实验的主要内容是基于 AT89C51 单片机实现电子贺卡，其中包括音乐播放功能，图片显示功能，以及矩阵键盘控制音乐和图片的输出功能，此外拓展创新了字幕滚动的功能，丰富本次项目的趣味性，此次实验涉及的几个方面所用到的知识，内容独立，通过整合学习这几个方面的知识以及多次的实践应用，逐步完成整个项目。

项目基于 KEIL 5 和 Protues 仿真设计完成，首先初步完成硬件设备的整体搭和设备的连接，之后构建程序的基本框架，分别完成 LCD12864 的 129*64BMP 格式图片的显示功能，利用定时器和延时函数构建单片机通过蜂鸣器播放音乐的功能，进一步通过矩阵键盘 P1 口的输入控制切换图片和背景音乐的功能，最后增加了一个字幕滚动作为项目的结尾。

关键词：单片机、LCD12864、蜂鸣器、矩阵键盘



Content summary

The main content of this experiment is to implement electronic greeting cards based on AT89C51 microcontroller, which includes music playback function, picture display function, and matrix keyboard control music and picture output function. In addition, the function of subtitle scrolling has been expanded and innovated, which enriches the interest of this project. The content of the knowledge used in several aspects of this experiment is independent, and the whole project is gradually completed through integrated learning of these aspects and multiple practical applications.

The project was completed based on KEIL 5 and Protues simulation design. First, the overall hardware equipment and the connection of the equipment were initially completed, and then the basic framework of the program was constructed to complete the display function of the LCD12864 129*64BMP format picture, using the timer and delay function. Build the function of single-chip microcomputer to play music through the buzzer, and further control the function of switching pictures and background music through the input of the matrix keyboard P1 port, and finally add a subtitle scroll as the end of the project.

Keywords: MCU, LCD12864, buzzer, matrix keyboard



目录

第一章：绪论.....	1
1.1 概述.....	1
1.1.1 课题研究的意义.....	1
1.1.2 个人所做的工作.....	1
1.1.3 系统的主要功能.....	1
1.1.4 课程设计思维导图.....	2
第二章：系统设计.....	3
2.1 系统组成.....	3
第三章：硬件设计.....	4
3.1 硬件电路设计及描述.....	4
3.1.1 单片机的选择.....	4
3.1.2 复位电路.....	5
3.1.3 时钟电路.....	6
3.2 键盘基本电路设计.....	7
3.2.1 输入键盘的选择.....	7
3.2.2 输入键盘的电路设计.....	7
3.3 显示屏基本电路设计.....	8
3.3.1 显示屏的选择.....	8
3.3.2 显示屏的电路设计.....	8
3.4 音源的选择.....	9
3.5 字模提取工具的选择.....	10
第四章：软件设计.....	11
4.1 编程语言选择.....	11
4.2 128*64BMP 图片的取模和显示.....	11
4.2.1 原理解释.....	11
4.2.2 程序流程图.....	13
4.2.3 程序设计.....	13
4.3 音乐的播放.....	17
4.3.1 原理解释.....	17
4.3.2 程序流程图.....	19
4.3.3 程序设计.....	19



4.4 矩阵键盘的控制	22
4.4.1 原理解释	22
4.4.2 程序流程图	23
4.4.3 程序设计	24
第五章：结论与总结	27
5.1 结论	27
5.2 总结	27
附录	28
参考文献	29



第一章：绪论

1. 1 概述

1.1.1 课题研究的意义

电子贺卡在生活中已经并不常见了，但是在一些生日以及节日中人们仍然能够随处看到其衍生出的电子设备，如朋友在手机上发来的动态图画邮件祝福，以及在广场大屏幕上播放的广告和音乐，这些电子设备的显示图片和播放音乐的功能都与此项目有着相应的关联。

此次项目设计能够学习到显示器和音乐播放器等多媒体的设计基础原理，同时能够培养自己的学习能力。为学习其他硬件知识培养基础。

1.1.2 个人所做的工作

在此项目中个人独立自主的完成电子贺卡的硬件设计、软件设计、调试运行、书写实验报告等内容。

在项目设计中查询资料，学习 LCD12864 的成像原理、实际应用、字模生成等知识，编程实现蜂鸣器的音乐播放、音源文件存储格式、搜索音源文件等工作，进一步使用 4*4 的矩阵键盘控制音乐的播放和图像的显示。

1.1.3 系统的主要功能

- 1、通过 LCD12864 以字模生成的形式显示 128*64 的 BMP 格式图片；
- 2、通过蜂鸣器播放给定形式存储的音乐文件；
- 3、通过 4*4 的矩阵键盘控制同步图像显示与音乐播放；
- 4、个人增添功能滚动字幕；



1.1.4 课程设计思维导图

本次课程设计项目的主要工作流程如图 1.1 所示：

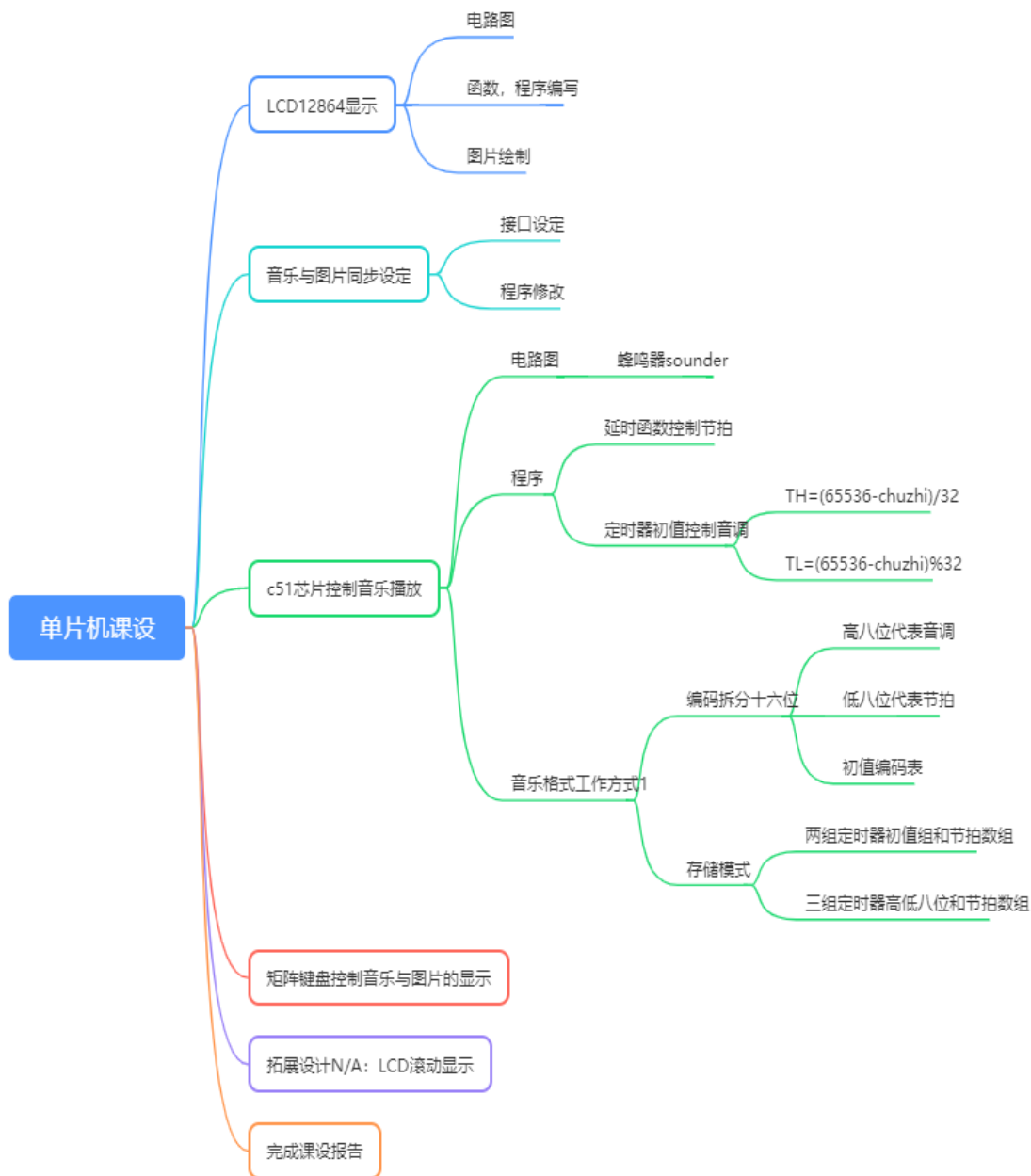


图 1.1 课程设计思维导图



第二章：系统设计

对此次下个项目的需求进行分析后，硬件电路所需要的元器件进行挑选，最终的使用的元器件有 LCD12864、AT89C51、上拉电阻、时钟电路、矩阵键盘、以及蜂鸣器、电源等。

通过软件设计的过程中发现的问题对硬件的选取做出部分调整，选择最优的元器件，以及最好的电路连接方式，通过软件设计，使硬件达到最佳的使用效果，完成系统整体的设计。

2.1 系统组成

系统工作原理如图 2.1 所示

硬件：

1. 根据需实现的键盘输入判断，数据处理，显示等功能进行整体设计结合。
2. 选用设备包括矩阵键盘，AT89C51 单片机，LCD12864（不带字库）显示屏，蜂鸣器。
3. 设计所需要的电路图，包括键盘、单片机、显示屏等。

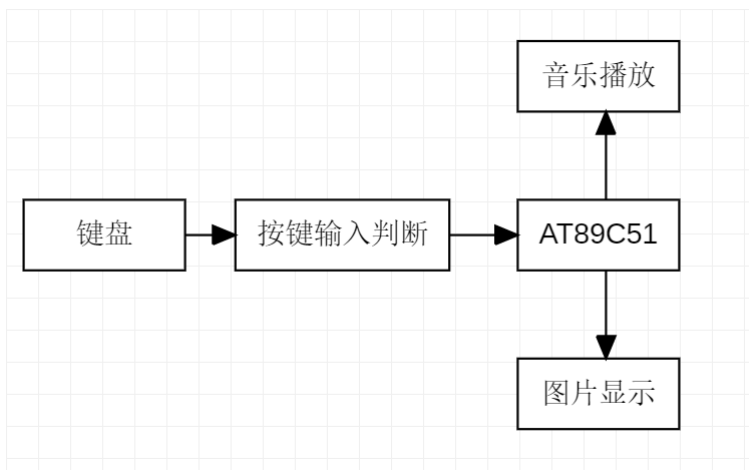


图 2.1系统工作原理图

电子贺卡的工作流程是：通过矩阵键盘的按键，使不同的图片和相应的音乐同时显示和播放，当键盘有一键按下显示器和蜂鸣器同时工作，一组音乐和图片效果完成后可以通过再次按键，展现下一组的效果。



第三章：硬件设计

该单片机结构的功能设计分为两部分：硬件设计与软件设计。其中硬件设计用来搭建一个键盘的结构，软件设计用来使该结构发挥应有的效果，本章仅讨论硬件设计。

3.1 硬件电路设计及描述

3.1.1 单片机的选择

目前，单片机技术得到了飞速发展，在全球范围内诸多集成电路生产制造商都研发出各自的单片机。单片机的种类很多，其中 51 系列单片机在众多型号中脱颖而出，具有先进的技术、性价比高等有点，目前已经成为自动化工程、工业测控等应用领域中的主力军，已经在国内广泛应用。

AT89C51 单片机硬件结构有如下一些主要特点：

- (1) 8位中央处理器（CPU） 核心部件
- (2) 内部数据存储器（内部RAM） 共256个RAM单元
- (3) 内部程序存储器（内部ROM） 4KB的闪存程序存储器(Flash ROM)
- (4) 定时器/计数器 T0（TH0、TL0）
- (5) 并行I/O口 共有4个8位的并行I/O端口
- (6) 串行口 有1个全双工的可编程串行I/O端口
- (7) 中断控制系统 共有5个中断源
- (8) 时钟电路（OSC） 由反向放大器反馈连接组成
- (9) 内部总线（三总线） 系统总线

(10) I/O端口引脚（32条）P0.0~P0.7（39~32脚）：P0口的8位准双向I/O端口线。 第一功能：普通的I/O口。用于传送用户输入/输出的信息。 第二功能：在对单片机总线扩展时，用于分时传送用户输入/输出的数据和低8位地址（A0~A7） 负载能力：每位能驱动8个LSTTL负载。P1.0~P1.7（1~8脚）：P1口的8位准双向I/O端口线。 功能：用于传送用户输入/输出的信息（即普通的I/O口）。 负载能力：每位能驱动4个LSTTL负载。



(11) 引脚如图 3.1 所示:

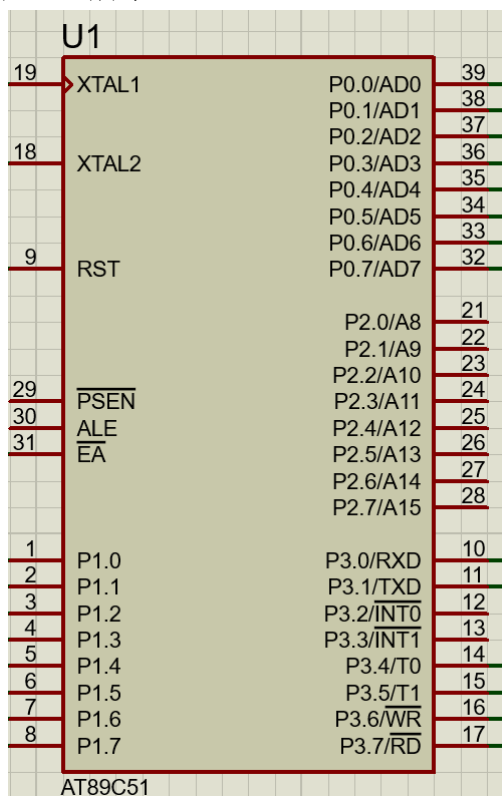


图 3.1 AT89C51 引脚图

3.1.2 复位电路

在单片机中具有复位功能当 RST 引脚得到大于两毫秒的高电平后单片机将自动复位, CPU 将恢复初始状态, 专用寄存器全部清零, 管脚置于 FFH。PC 寄存器恢复到 0000H, 系统重新开始运行。

复位的意义是为了防止单片机出现如假死或其他系统异常, 导致系统无法正常运行或无法正常启动的一项保护措施, 当需要进行复位操作时, 按下复位按钮即可实现复位, 复位电路可分为两部分——片内电路和片外电路。复位操作的方式一般有两种: 按键复位于上电复位, 本设计采用按键复位的方式, 按下按钮电容出现短路, 复位引脚激活至高电平状态, 复位执行即可实现。



复位电路如图 3.2 所示。

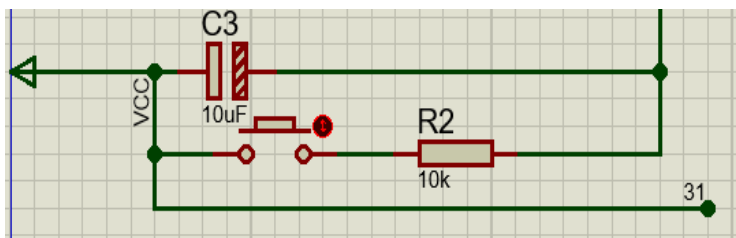


图 3.2 复位电路

3.1.3 时钟电路

单片机的时钟信号可以通过内部或外部震荡的方式得到，内部振荡可以得到稳定的时钟信号，利用引脚 XTAL1 与 XTAL2 外接晶体振荡器的方式实现，通过单片机内置的高增益反相放大器产生脉冲信号。

电容器 C1、C2 具有稳定振荡频率的功能，一般大小选用 20~30 pF 之间的电容，本次使用电容为 22pF，晶振频率为 20MHz。

内部振荡的外部电路如图 3.3 所示：

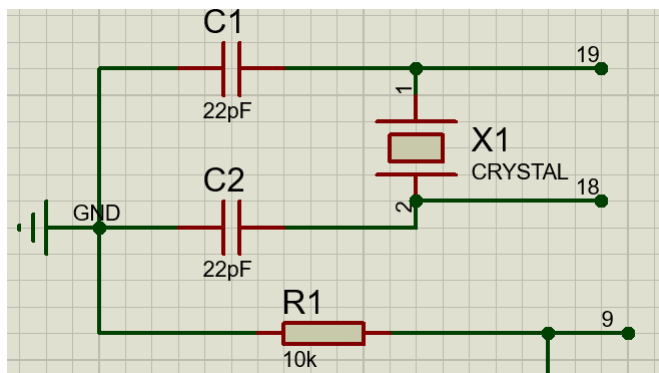


图 3.3 时钟电路



3.2 键盘基本电路设计

3.2.1 输入键盘的选择

由于本设计基于 Proteus 仿真模拟软件进行, 软件的标准库中并没有合适的键盘可供使用, 因此选择自行搭建 4×4 矩阵式键盘使用, 实现高度的定制化和可控化。

矩阵式键盘连接方式: 矩阵式键盘由行线和列线组成, 按键放置在行和列的交叉点上, 按键的两端分别连接到行线和列线上。这样, 4 条行线、4 条列线就可以构成有 16 个按键的键盘(称为 4×4 矩阵式键盘), 通过 P1 口实现键盘信息的输入。

3.2.2 输入键盘的电路设计

系统功能的基础实现需要使用的个按键, 其实并不多, 可以按照需求随意调整, 键盘进行标签化, 方便用户根据手册选择相应的媒体内容。

键盘的排列及连接方式如图 3.4 所示, 键盘连接单片机 P1 口使用, 第四位作为行号来使用, 高四位作为列号来使用, 使用线翻转法确定按钮位置。

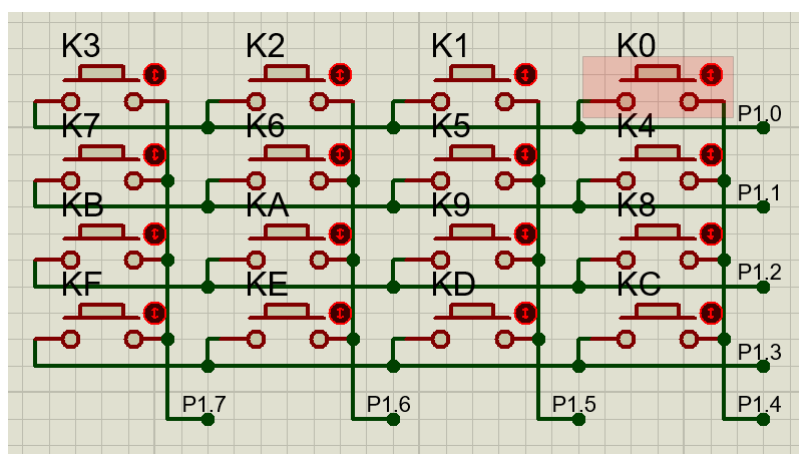


图 3.4 键盘系统



3.3 显示屏基本电路设计

3.3.1 显示屏的选择

在 Proteus 仿真模拟软件并没有带字库的 LCD12864 显示屏,因此需要自己通过特定的软件进行对图片字模的提取。

LCD12864基本参数:

- (1) 低电源电压 (VDD: +3.0—+5.5V)
- (2) 显示分辨率: 128×64 点
- (3) 内置128 个16×8 点阵字符
- (4) 2MHZ 时钟频率
- (5) 显示方式: STN、半透、正显
- (6) 驱动方式: 1/32DUTY, 1/5BIAS
- (7) 视角方向: 6 点
- (8) 背光方式: 侧部高亮白色LED, 功耗仅为普通LED 的1/5—1/10
- (9) 通讯方式: 串行、并口可选
- (10) 内置DC-DC 转换电路, 无需外加负压
- (11) 无需片选信号, 简化软件设计

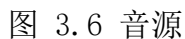
3.3.2 显示屏的电路设计

显示屏的电路设计如图 3.5 所示。

其中, 由于连接的是 P0 口, P0 口内无上拉电阻, 无法输出高电平, 因此外接排阻来提供高电平的输出, 排阻具有装配方便、安装密度高等优点。



此次使用的音源是sounder 如图3.6所示,可以避免音源的有源和无源造成的无法播放音乐的问题,实际使用中还是使用有源的蜂鸣器较好一些。





3.5 字模提取工具的选择

此次使用的字模提取工具是PctoLCD2002 软件如图3.7所示, 该软件提供汉字以及图片字模的提取模式, 可以完成对BMP格式图片文件的字模提取。要设置好相应的C51提取模式, 以及对提取格式的设定。

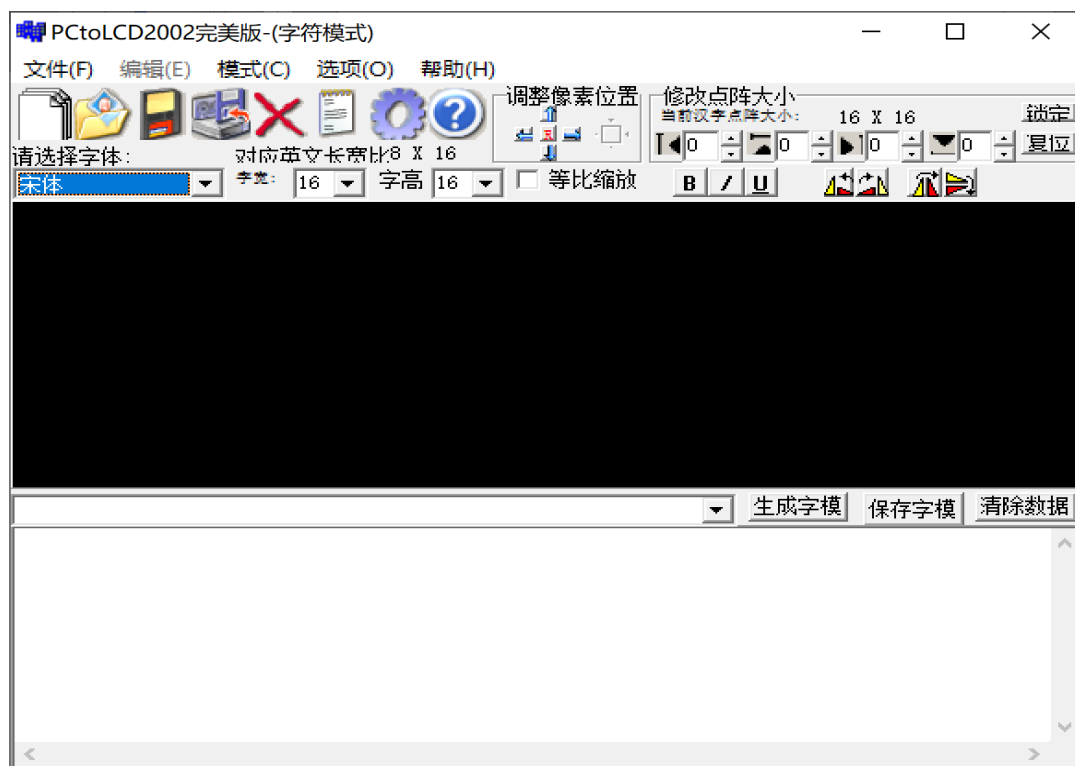


图 3.7 字模提取工具



第四章：软件设计

程序是单片机系统实现的基础之一，良好的软件系统配合硬件可以高效的解决问题。

4.1 编程语言选择

本次开发选用的是 C 语言，C 语言属于高级语言，即具有高级语言的具有可读性好，编写容易的特点，又具有汇编语言的效率高的特点，在降低开发成本的基础上还不损失效率，Keil C51 配合 Proteus 仿真模拟软件即可实现利用 C 语言实现单片机的开发。

4.2 128*64BMP 图片的取模和显示

4.2.1 原理解释

想要理解显示原理首先，要了解LCD12864 的显示格式，LCD12864与带字库的液晶不同，此块液晶含有两个液晶驱动器，每块驱动器都控制 64*64 个点，分为左右两个屏幕显示，总共为 128*64 个点（即有 128×64 个点）。这就是为什么 AMPIRE128*64 有 CS1和 CS2两个片选端的原因。此液晶有 8 页，一页有 8 行点阵点，左右各 64 列，共 128 列，如图4.1 所示。

经过取模的128*64的BMP文件时按照8*128的数组存储的，大小为1KB, 想要完整的显示图片必须要两个64*64的屏幕同时工作，以128为总列数交替交替显示自己所在部分的图片信息，若是出现半屏黑屏，则是因为程序出错，未开启屏幕或者数组溢出显示器的大小。

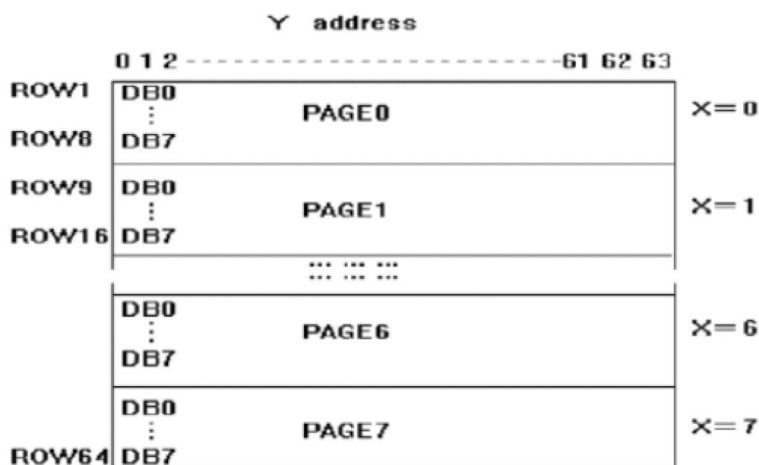


图 4.1 LCD12864 的列和页

LCD12864的引脚功能如图表4. 1所示：

引脚功能：

引脚符号	状态	引脚名称	功能
$\overline{CS1}$, $\overline{CS2}$	输入	芯片片选端，都是低电平有效	$CS1=0$ 开左屏幕， $CS1=1$ 关左屏幕 $CS2=0$ 开右屏幕， $CS2=1$ 关右屏幕
RS	输入	数据 / 命令选择信号	$RS=1$ 为数据操作， $RS=0$ 为写指令或读状态
RW	输入	读写选择信号	$R/W=1$ 为读选通， $R/W=0$ 为写选通
E	输入	读写使能信号	在 E 下降沿，数据被锁存（写）入液晶，在 E 高电平期间，数据被读出
DB0—DB7	三态	数据总线	数据或指令的传送通道
\overline{RST}	输入	复位信号，低电平时复位	复位时，关闭液晶显示，使显示起始行为 0，可以跟单片机的复位引脚 RST 相连，也可以直接接 VCC，使之不起作用
V0			液晶显示器驱动电压
-Vout	-10V		LCD驱动负电压

表4. 1 LCD12864的引脚功能



4.2.2 程序流程图

LCD12864 显示图片的流程如图 4.2 所示:

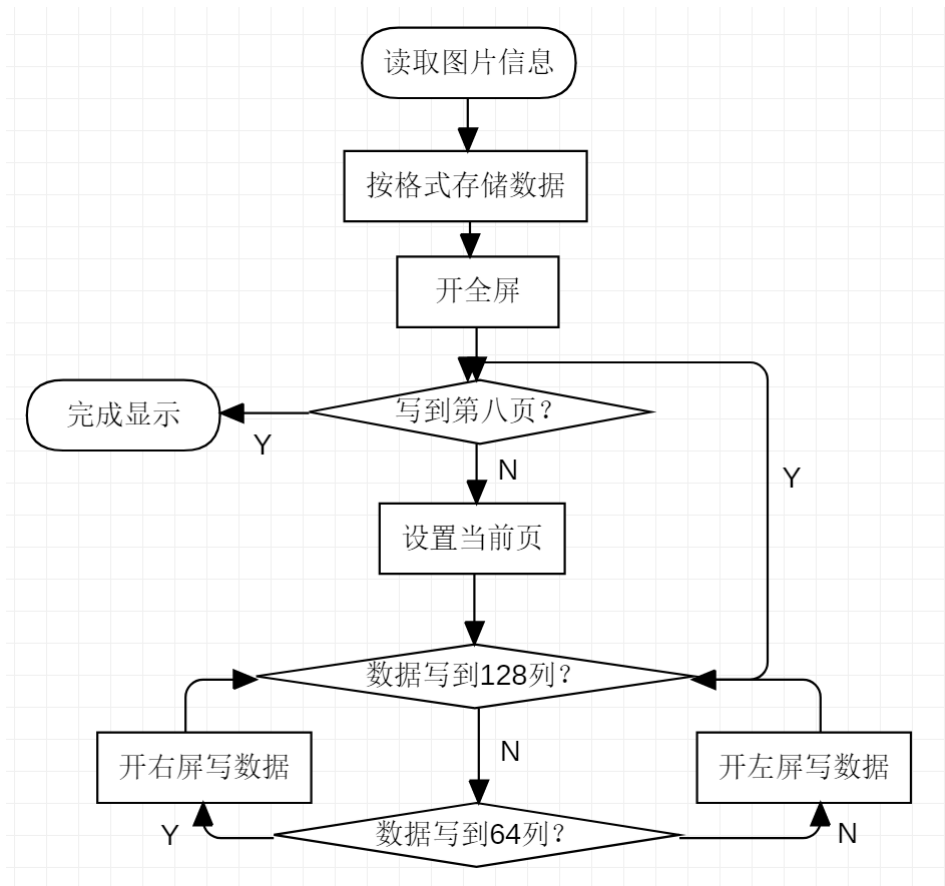


图 4.2 LCD12864 显示流程

4.2.3 程序设计

采用模块化设计方便后期更新以及维护。

```
void busy()// 忙检测函数
{
```

```
    P0 = 0X00; // 这句不能少 RS=0;
```



```
RW = 1; E = 1;
while (P0 & 0X80); //P0&0X80 结果为真则一直执行空操作, 也就是说 P0&0X80 为 0
则结束循环
E = 0;
}
void wcmd(uchar cmd)
{
    busy(); // 每次写入指令或数据之前都进行忙检测
    RS = 0;
    RW = 0;
    P0 = cmd;
    E = 1; // 在 E 下降沿写入数据和指令
    E = 0;
}
void wdata(uchar dat) // 写命令和写数据只在 RS=0或 RS=1上不同, 其余都相同
{
    busy(); RS = 1;
    RW = 0;
    P0 = dat;
    E = 1; // 在 E 下降沿写入数据和指令
    E = 0;
}

void set_page(uchar page) // 设置页, 12864LCD共有 8 页, 每页有 8 行点阵点。
{
    page = 0xb8 | page; // 首页地址为 0XB8
    wcmd(page); //page 取值范围为 0~7, 表示第 1 到 8 页
}
void set_line(uchar line) // 设置显示的起始行, 共有 0——63 行, 一般从 0 行开始显示
{
    line = 0xc0 | line; // 起始行地址 0XC0
    wcmd(line); //line 取值范围为 0~63, 表示第 1 到 64 行
}
void set_column(uchar column) // 设置显示的列
{
    //column=column&0x3f; // 列的最大值为 0X3F, 即 64
    column = 0x40 | column; // 列的首地址为 0X40,
    wcmd(column); //column 的取值范围为 0~63, 共
    第 14 页, 共 29 页
}
```



```
}  
  
void set_onoff(uchar onoff)// 设置显示开关, onoff 取值为 0 或 1  
{  
    onoff |= 0x3e;//0X3E 是关显示, 0X3F是开显示  
    wcmd(onoff);// 所以若 onoff 为 0,则表示关显示, onoff 为 1,则表示开显示  
}  
  
void select_screen(uchar screen)// 选屏, screen 取值范围为 0,1,2  
{  
    switch (screen)  
    {  
        case 0:CS1 = 0; CS2 = 0; break; // 全屏,左右两个屏幕显示同样的内容  
        case 1:CS1 = 0; CS2 = 1; break; // 左半屏  
        case 2:CS1 = 1; CS2 = 0; break; // 右半屏  
        default:break;  
    }  
}  
  
void clear_screen(uchar screen)// 清屏  
{  
    uchar i, j;  
    select_screen(screen);// 先选屏  
    for (i = 0; i < 8; i++)// 控制页数 0—— 7, 共 8 页  
    {  
        set_page(i);// 设置页  
        set_column(0); // 设置列, 每页都从第 1 列开始, 共 64 列  
        for (j = 0; j < 64; j++)// 控制列数 0—— 63, 共 64 列  
            wdata(0x00);// 写入 0, 列地址指针会自动加 1  
    }  
}  
  
void init() //LCD 初始化  
{  
    busy(); // 忙检测  
    select_screen(0);// 选屏, 全屏  
    set_onoff(0); // 关显示  
    select_screen(0); // 选屏, 全屏  
    set_onoff(1); // 开显示  
    select_screen(0); // 选屏, 全屏  
    clear_screen(0); // 清屏
```



```
set_line(0); // 起始行: 0
}

void show(uchar screen, uchar page, uchar column, uchar p[]) // 显示图片一个图有
8*128=1024 字节数据
{
    uchar i, j; //screen 表示选择屏幕,    page 表示页, column 表示列, *p 表示
汉字的数组
    select_screen(screen); // 选屏
    for (i = 0; i < 8; i++) {
        // 设置页, 写第一页
        set_column(column); // 设置列
        for (j = 0; j < 128; j++) // 控制 每一页的 128 列的数据输出
            if (j < 64) {
                select_screen(1);
                set_page(page + i);
                wdata(p[j + i * 128]); // 写入该页的数据
            }
            else {
                select_screen(2);
                set_page(page + i);
                wdata(p[j + i * 128]);
            }
        }
    }
}
```

//滚动字幕的相关函数设计如下:

```
void delay2(unsigned char t)
{
    unsigned char t1;
    unsigned long t2;
    for(t1 = 0; t1 < t; t1++)
        for(t2 = 0; t2 < 100; t2++);
    TR0 = 0;
}

void hz_displ6(unsigned char page, unsigned char col, unsigned char code* hzk)
{
    unsigned char j = 0, i = 0;
```



```
for (j = 0; j < 2; j++)  
{  
    wcmd(0xb8 + page + j);  
    delay2(0);  
    wcmd(0x40 + col);  
    delay2(0);  
    for (i = 0; i < 16; i++)  
    {  
        wdata(hzk[16 * j + i]);  
        delay2(1);  
    }  
}}
```

图片的存储略:

在提取字模时要先设置好点阵格式、取模方式、取模走向以及格式。

4.3 音乐的播放

4.3.1 原理解释

发声原理:

一首乐曲是由多个音符构成的。每个音符都对应着一个确定的频率，乐曲中不同的音符，实质就是不同频率的声音；另外每个音符会根据乐曲的要求设定一个确定的节拍。可以控制单片机产生不同频率不同节拍的脉冲信号，由蜂鸣器发出就产生了美妙和谐的乐曲。

单片机产生不同频率脉冲信号的原理:

1、要产生音频脉冲，只要算出某一音频的脉冲（1/频率），然后将此周期除以2，即为半周期的时间，利用定时器计时这个半周期的时间，每当计时到后就将输出脉冲的I/O 反相，然后重复计时此半周期的时间再对I/O 反相，就可以在I/O脚上得到此频率的脉冲。

2、利用单片机内部定时器使其工作在计数器模式MODE1 下，改变计数



值TH0及TL0 以产生不同频率的方法如下:

例如:

频率为523Hz, 其周期 $T=1/523 \text{ S}=1912\mu\text{S}$, 因此只要令计数器计时 $956\mu\text{S}/1\mu\text{S}=956$, 在每计数956 次时就将I/O 反接, 就可得到中音D0(532Hz)。

计数脉冲值与频率的关系公式如下:

$$N=Fi/2/Fr \quad (4-1)$$

其中N 表示计数值; Fi 表示内部计时一次为1 μS , 故其频率为1MHz; Fr 表示要产生的频率。

3、其计数值的求法如下:

$$T=65536-N=65536-Fi/2/Fr \quad (4-2)$$

4、建立音乐步骤:

(1) 先把乐谱的音符找出, 然后由上表建立 T 值表的顺序。

(2) 把 T 值表构成一个编码表, 构成发音符的计数值放在计数初值编码表里。

(3) 简谱码(音符)为高位, 节拍为(节拍数)为低 4 位, 音符节拍码放在程序的简谱编码表里。

建立编码表具体如下:

1) 定时值为十六进制 4 位数, 拆分为两组, 如 5 对应的定时值为FD80H, 拆分 FDH 和 80H 两组. 前者装入定时器的高位 TH0, 后组装入定时器的低位 TL0。

2) 在程序中使用定时器 T0 方式 1 来产生来产生歌谱中各音符对应频率的音频脉冲, 由 P3.7 输出, 经三极管将信号放大后驱动蜂鸣器发出不同音节的声音。

3) 程序中的节拍控制是通过调用延时子程序 DELAY 的 次数 来实现, 1 拍为DELAY。

4) 节拍的控制码在表 TABLE 中位于音符码的后面。

5) 当一个音符的发音时间到时, 再查下一个音符的定时常数和延时常数。依此进行下去, 就可演奏出悦耳动听的乐曲。



4.3.2 程序流程图

蜂鸣器播放音乐的程序流程如图 4.3 所示：

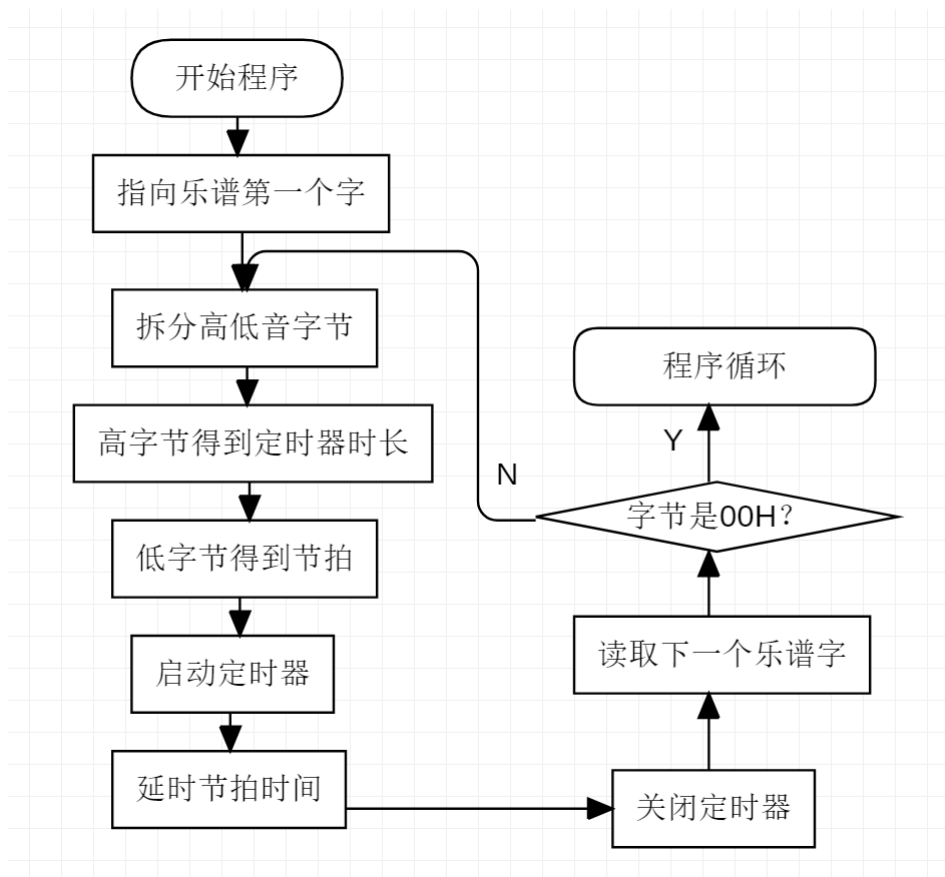


图 4.3 音乐播流程

4.3.3 程序设计

```
unsigned char timeH, timeL, time;  
//-----  
void song();  
/*存储计算好的定时器初值*/  
code unsigned char FREQH[] = {
```



```
0xF2, 0xF3, 0xF5, 0xF5, 0xF6, 0xF7, 0xF8, //1234567
0xF9, 0xF9, 0xFA, 0xFA, 0xFB, 0xFB, 0xFC, 0xFC, //1, 2, 3, 4, 5, 6, 7, i
0xFC, 0xFD, 0xFD, 0xFD, 0xFD, 0xFE, // 234567
0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFF }; // 1234567

code unsigned char FREQL[] = {
    0x42, 0xC1, 0x17, 0xB6, 0xD0, 0xD1, 0xB6, //1234567
    0x21, 0xE1, 0x8C, 0xD8, 0x68, 0xE9, 0x5B, 0x8F, //1, 2, 3, 4, 5, 6, 7, i
    0xEE, 0x44, 0x6B, 0xB4, 0xF4, 0x2D, // 234567
    0x47, 0x77, 0xA2, 0xB6, 0xDA, 0xFA, 0x16 };
//-----
'void timer0() interrupt 1
{
    TRO = 0;
    speaker = !speaker;
    TH0 = timeH;
    TLO = timeL;
    TRO = 1;
}

void delay(unsigned char t)
{
    unsigned char t1;
    unsigned long t2;
    for (t1 = 0; t1 < t; t1++)
        for (t2 = 0; t2 < 1900; t2++);
    TRO = 0;
}

void song()
{
    TH0 = timeH;
    TLO = timeL;
    TRO = 1;
    delay(time);
}

//延时函数控制音乐的节拍
void delay(unsigned char t)
{
    unsigned char t1;
```



```
unsigned long t2;
for (t1 = 0; t1 < t; t1++)
    for (t2 = 0; t2 < 1900; t2++);
TRO = 0;
}
//-----
void song()
{
    TH0 = timeH;
    TL0 = timeL;
    TRO = 1;
    delay(time);
}
//
void play(unsigned char* p) {
    unsigned char k;
    unsigned long int i;
    ///while(1) {
    i = 0;
    time = 1;
    while (time) {
        k = p[i] + 7 * p[i + 1] - 1;
        timeH = FREQH[k];
        timeL = FREQL[k];
        time = p[i + 2];
        i += 3;
        song();
    }
}
```



4.4 矩阵键盘的控制

4.4.1 原理解释

矩阵键盘又称为行列式键盘,它是用4条I/O线作为行线,4条I/O线作为列线组成的键盘。在行线和列线的每一个交叉点上,设置一个按键。这样键盘中按键的个数是 4×4 个。这种行列式键盘结构能够有效地提高单片机系统中I/O口的利用率。

1、先读取键盘的状态,得到按键的特征编码。

先从P1口的高四位输出低电平,低四位输出高电平,从P1口的低四位读取键盘状态。再从P1口的低四位输出低电平,高四位输出高电平,从P1口的高四位读取键盘状态。将两次读取结果组合起来就可以得到当前按键的特征编码。使用上述方法我们得到16个键的特征编码。

举例说明如何得到按键的特征编码:

假设“1”键被按下,找其按键的特征编码。从P1口的高四位输出低电平,即P1.4—P1.7为输出口。低四位输出高电平,即P1.0—P1.3为输入口。读P1口的低四位状态为“1101”,其值为“0DH”。

再从P1口的高四位输出高电平,即P1.4—P1.7为输入口。低四位输出低电平,即P1.0—P1.3为输出口,读P1口的高四位状态为“1110”,其值为“E0H”。将两次读出的P0口状态值进行逻辑或运算就得到其按键的特征编码为“EDH”。用同样的方法可以得到其它15个按键的特征编码。

2、根据按键的特征编码,查表得到按键的顺序编码。

将用上述方法得到的16个按键的特征编码按图2按键排列的顺序排成一张特征编码与顺序编码的对应关系表,然后用当前读得的特征编码来查表,当表中有该特征编码时,它所在的位置就是对应的顺序编码。

3、矩阵键盘键值查找程序的具体编程



4.4.2 程序流程图

键盘控制的程序流程如图 4.4 所示：

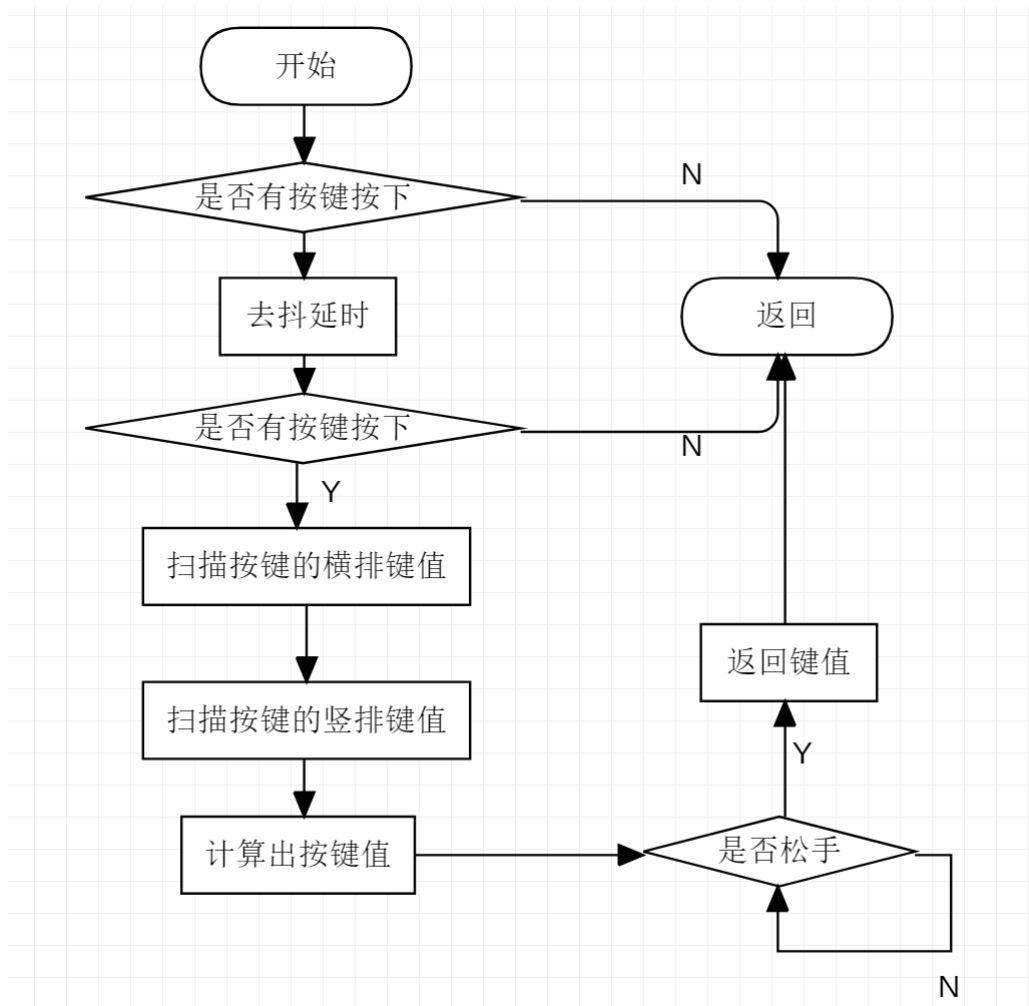


图 4.4 矩阵键盘工作流程



4.4.3 程序设计

```
/*键盘*/
//子函数声明
void SCANkey(); //扫描键盘函数
bit hitkey(); //判断是否有键盘按下，有则返回1，没有返回0
uchar scan_key(); //扫描键盘，返回键盘值
void delay1(uint N); //延迟函数
void key_manage(uchar keycode); //键值处理函数
//函数名: SCANkey
//函数功能: 扫描键盘程序，有键按下完成按键处理，无按键按下时直接返回
//形式参数: 无
//返回值: 无
void SCANkey()
{
    uchar key_code;
    if (hitkey()) //判断是否有按键按下
    {
        delay1(2); //延时20ms，消除按键抖动干扰
        if (hitkey()) //再次判断是否是有效按键
        {
            key_code = scan_key();
            while (hitkey()); //等待按键释放
            { EA = 1;
              key_manage(key_code); //键盘扫描、键值处理跳转、按键处理
            }
        }
    }
}
//函数名: hitkey
//函数功能: 判断是否有键按下
//形式参数: 无
//返回值: 1表示有按键按下，0表示无按键按下
bit hitkey()
{
    uchar scancode, keycode;
    scancode = 0x0f; //P1.4~P1.7输出全0
```



```
P1 = scandcode;    //扫描按键，换回键值
keycode = P1;      //读P1.0~P1.3的状态
if ((keycode & 0x0f) == 0x0f) //屏蔽高4位
    return(0);      //全1则无按键闭合，换回0
else
    return(1);      //全0则有按键闭合，换回1
}

//函数名: delay1
//函数功能: 延时效果，实现 (N*10ms) 的延时。系统采用12MHZ的时钟时，延时满足要求，其他情况需要改动。
//形式参数: N
//返回值: 无
void delay1(uint N)
{
    int i, j, k;
    for (i = 0; i < N; i++)
        for (j = 0; j < 10; j++)
            for (k = 0; k < 120; k++);
}

//函数名: sacn_key
//函数功能: 扫描按键，换回键值。高4位代表行，低4位代表列
//形式参数: 无
//返回值: keycode，相应的按键按下扫描出一个按键扫描码，送到key_manage函数提取扫描码
uchar scan_key()
{
    uchar scandcode, keycode, keycodeh, keycodel;
    scandcode = 0xef;    //扫描键盘，采用逐行扫描法，P1.4=0, 扫描第一行
    while (scandcode != 0xff)
    {
        P1 = scandcode;    //输入扫描码，扫描P1.4对应的行
        keycode = P1;      //读出数据，看是否在此行的某列键盘被按下
        if ((keycode & 0x0f) != 0x0f) break; //扫描到按下十位键，怎退出
        scandcode = (keycode << 1) | 0x0f; //否则，左移，更新扫描码继续扫描
    }
    keycode = ~keycode;    //键盘扫描码变反，返回
    keycodeh = keycode << 4;
    keycodel = keycode >> 4;
    keycode = keycodeh | keycodel;
}
```



```
return(keycode);  
}  
  
//函数名: key_manage  
//函数功能: 键值处理跳转, 根据keycode而对应display参数  
//形式参数: keycode扫描键盘返回的值。高4位代表行, 低4位代表列  
//返回值:  ) 函数  
void key_manage(uchar keycode)  
{  
    switch (keycode)  
    {  
        case 0x11:show(0, 0, 0, &picture0); m = music; play(m); break;  
        case 0x12:show(0, 0, 0, &picture1); m = music1; play(m); break;  
        case 0x14:show(0, 0, 0, &picture2); m = music2; play(m); break;  
        case 0x18:show(0, 0, 0, &picture3); m = music3; play(m); break;  
        case 0x21:show(0, 0, 0, &picture4); m = music4; play(m); break;  
        case 0x22:show(0, 0, 0, &picture5); m = music5; play(m); break;  
        case 0x24:show(0, 0, 0, &picture6); m = music6; play(m); break;  
        case 0x28:show(0, 0, 0, &picture7); m = music7; play(m); break;  
        case 0x41:show(0, 0, 0, &picture8); m = music8; play(m); break;  
        case 0x42:show(0, 0, 0, &picture9); m = music9; play(m); break;  
        case 0x44:show(0, 0, 0, &picture10); m = music10; play(m); break;  
        case 0x48:gundong(); break;  
        /*case 0x81:show(0,0, 0, &picture2);m=music12;play(m);break;  
        case 0x82:m=music13;break;  
        case 0x84:m=music14;break;  
        case 0x88:m=music15;break;*/  
    }  
}
```




第五章：结论与总结

5.1 结论

在此次项目中深入的学习了单片机显示和播放音乐的原理，通过自主学习和实践较好的完成了项目需求，同时增加了项目设计的经验，在项目中学会利用LCD12864，AT89C51 单片机等硬件，实现图片格式的转换，取模处理等未接触过的技术。

就本项目而言，利用 LCD12864 进行显示、蜂鸣器播放音乐，矩阵键盘进行控制整体的运行和选择，通过模块化功能的设计，既能够完成各个需求所必要的功能，又能够构成了一个相对完整的系统。通过自主画图取模等过，对项目的实用性、趣味性有了更为直观的体验，51单片机的超强性能以及微小的体积，都意味着在未来生活高度智能的情况下，其应用范围将得到更大的扩展，其次各种微处理器对于实现智能家居，互联生活会起到至关重要的作用。

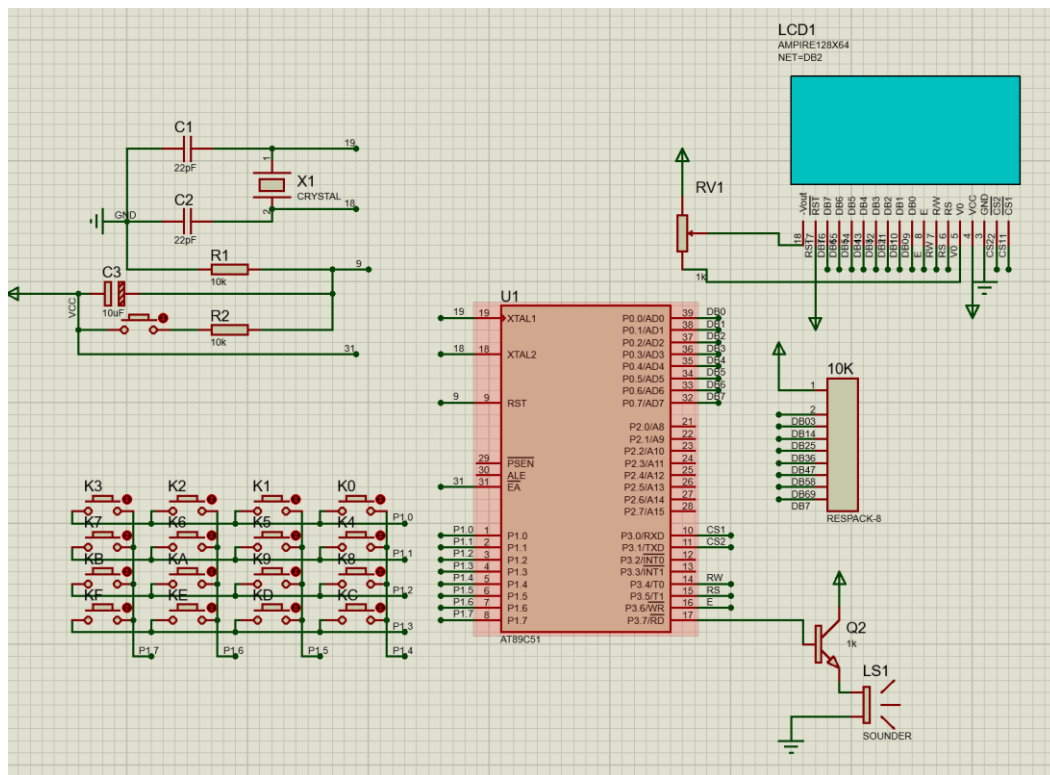
5.2 总结

经过本次课程设计，我充分体会到了单片机及硬件开发的乐趣，不仅巩固了专业知识，而且通过自学实现了对未知技能的掌握，在这个过程中，不断的从自己始料未及的地方发现问题，通过不断的尝试以及修改，一点点的完成一个大的项目，对自己的编程能力和硬件基础知识有所巩固，过程虽然艰难却也很充实，同时还提高了自己的论文撰写能力。

能够将自己所学的知识应用到实际生活中是一件令人自豪的事情，既可以丰富生活，又可以增加自己的实践经验。这也提醒自己在将来的学习和生活中不要放弃实践的机会，要勇于实践自己所学的知识，通过实践改变生活，成就更加美好的未来。

附录

本此课程设计的电子贺卡的全局电路图如图所示:





参考文献

- [1]周亚辉. 运用单片机与计算机通信进行温度显示监控[J]. 电子技术与软件工程, 2019, (12):232.
- [2]黄丽真. 单片机系列 4:矩阵式键盘控制设计[N]. 电子报, 2019-02-10(008).
- [3] 盛华 . 对单片机在电子领域当中的应用 [J]. 电 子 技 术 与 软 件 工 程,2018,(24):242.
- [4]高铭泽. C51 单片机的开发与应用[J]. 硅谷, 2011, (23):61-62.