摘 要

单片机又称单片微控制器，它不是完成某一个逻辑功能的芯片，而是相当于一个微型的计算机。本文主要上网查找了一个单片机应用系统实例——单片机音乐盒，其通过单片机控制蜂鸣器发声的频率和时间从而发出音乐。

关键词：单片机；蜂鸣器

目录

[1引言 1](#_Toc135574397)

[2系统分析设计 2](#_Toc135574398)

[3程序流程图设计 3](#_Toc135574399)

[4源程序设计 4](#_Toc135574400)

[5系统调试与仿真及调试结果分析 8](#_Toc135574401)

1引言

单片机（Single-Chip Microcomputer）是一种集成电路芯片，是采用超大规模集成电路技术把具有数据处理能力的中央处理器CPU、随机存储器RAM、只读存储器ROM、多种I/O口和中断系统、定时器/计数器等功能（可能还包括显示驱动电路、脉宽调制电路、模拟多路转换器、A/D转换器等电路）集成到一块硅片上构成的一个小而完善的微型计算机系统，在工业控制领域广泛应用。从上世纪80年代，由当时的4位、8位单片机，发展到现在的300M的高速单片机。其使用领域已十分广泛，如智能仪表、实时工控、通讯设备、导航系统、家用电器等。

从二十世纪九十年代开始，单片机技术就已经发展起来，随着时代的进步与科技的发展，目前该技术的实践应用日渐成熟，单片机被广泛应用于各个领域。现如今，人们越来越重视单片机在智能电子技术方面的开发和应用，单片机的发展进入到新的时期，无论是自动测量还是智能仪表的实践，都能看到单片机技术的身影。当前工业发展进程中，电子行业属于新兴产业，工业生产中人们将电子信息技术成功运用，让电子信息技术与单片机技术相融合，有效提高了单片机应用效果。作为计算机技术中的一个分支，单片机技术在电子产品领域的应用，丰富了电子产品的功能，也为智能化电子设备的开发和应用提供了新的出路，实现了智能化电子设备的创新与发展。

单片机的体积比较小，内部芯片作为计算机系统，其结构简单，但是功能完善，使用起来十分方便，可以模块化应用。其有着较高的集成度，可靠性比较强，即使处于长时间的工作也不会存在故障问题。而且其在应用时低电压、低能耗，是人们在日常生活中的首要选择，为生产与研发提供便利。单片机对数据的处理能力和运算能力较强，可以在各种环境中应用，且有着较强的控制能力。

2系统分析设计

蜂鸣器是一种将电信号转换为声音信号的器件，常用来产生设备的按键音、报警音等提示信号。按驱动方式可分为有源蜂鸣器和无源蜂鸣器，有源蜂鸣器是内部自带振荡源，将正负极接上直流电压即可持续发声，频率固定，而无源蜂鸣器则是内部不带振荡源，需要控制器提供振荡脉冲才可发声，调整提供振荡脉冲的频率，可发出不同频率的声音。

运用51单片机的一个引脚来控制无源蜂鸣器的震荡脉冲的频率，根据音乐知识，设置不同的音符相应的频率，达到发出音符的声音，之后在控制每个音符发出声音的时间，来控制节奏，这样使蜂鸣器演奏音乐。原理图设计如图 2.1所示。

图表

描述已自动生成

图 2.1 原理设计图

3程序流程图设计

程序流程图如图 3.1所示。

图片包含 日程表

描述已自动生成

图 3.1 程序流程图

4源程序设计

本次作业采用C语言进行程序设计，运用Keil软件进行编写，程序代码如下：

#include <REGX52.H>

sbit speaker = P2^5;

unsigned char timer0h, timer0l, time;

//--------------------------------------

//单片机晶振采用11.0592MHz

// 频率-半周期数据表 高八位 本软件共保存了四个八度的28个频率数据

code unsigned char FREQH[] = {

0xF2, 0xF3, 0xF5, 0xF5, 0xF6, 0xF7, 0xF8, //低音1234567

0xF9, 0xF9, 0xFA, 0xFA, 0xFB, 0xFB, 0xFC, 0xFC,//1,2,3,4,5,6,7,i

0xFC, 0xFD, 0xFD, 0xFD, 0xFD, 0xFE, //高音 234567

0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFE, 0xFF}; //超高音 1234567

// 频率-半周期数据表 低八位

code unsigned char FREQL[] = {

0x42, 0xC1, 0x17, 0xB6, 0xD0, 0xD1, 0xB6, //低音1234567

0x21, 0xE1, 0x8C, 0xD8, 0x68, 0xE9, 0x5B, 0x8F, //1,2,3,4,5,6,7,i

0xEE, 0x44, 0x6B, 0xB4, 0xF4, 0x2D, //高音 234567

0x47, 0x77, 0xA2, 0xB6, 0xDA, 0xFA, 0x16}; //超高音 1234567

//--------------------------------------

code unsigned char sszymmh[] =

{

4,2,4, 2,2,2, 4,2,2, 3,2,4, 1,2,2, 6,1,2,

4,2,1, 3,2,1, 2,2,1, 1,2,1, 2,2,2, 6,2,2, 3,2,4, 6,1,4,

2,2,2, 1,2,1, 7,1,1, 6,1,1, 5,1,1, 6,1,1, 7,1,1, 6,1,2, 3,2,4, 6,1,2,

2,2,1, 2,2,1, 1,2,1, 7,1,1, 6,1,1, 5,1,1, 6,1,1, 7,1,1, 6,1,2, 3,2,4,

};//歌曲

//世上只有妈妈好数据表 要想演奏不同的乐曲, 只需要修改这个数据表

//code unsigned char sszymmh[] = {

//6, 2, 3, 5, 2, 1, 3, 2, 2, 5, 2, 2, 1, 3, 2, 6, 2, 1, 5, 2, 1,

//一个音符有三个数字。前为第几个音、中为第几个八度、后为时长（以半拍为单位）。

//6, 2, 3代表：6, 中音, 3个半拍;

//5, 2, 1代表：5, 中音, 1个半拍;

//3, 2, 2代表：3, 中音, 2个半拍;

//5, 2, 2代表：5, 中音, 2个半拍;

//1, 3, 2代表：1, 高音, 2个半拍;

//

//6, 2, 4, 3, 2, 2, 5, 2, 1, 6, 2, 1, 5, 2, 2, 3, 2, 2, 1, 2, 1,

//6, 1, 1, 5, 2, 1, 3, 2, 1, 2, 2, 4, 2, 2, 3, 3, 2, 1, 5, 2, 2,

//5, 2, 1, 6, 2, 1, 3, 2, 2, 2, 2, 2, 1, 2, 4, 5, 2, 3, 3, 2, 1,

//2, 2, 1, 1, 2, 1, 6, 1, 1, 1, 2, 1, 5, 1, 6, 0, 0, 0,,,,,,,,,,,,,,,,,,,,,6,2,3, 6,2,1, 1,3,2, 7,2,1, 6,2,1, 5,2,2, 3,2,1, 5,2,1, 6,2,4, 3,2,3,};

//--------------------------------------

void t0int() interrupt 1 //T0中断程序，控制发音的音调

{

TR0 = 0; //先关闭T0

speaker = !speaker; //输出方波, 发音

TH0 = timer0h; //下次的中断时间, 这个时间, 控制音调高低

TL0 = timer0l;

TR0 = 1; //启动T0

}

//--------------------------------------

void delay(unsigned char t) //延时程序，控制发音的时间长度

{

unsigned char t1;

unsigned long t2;

for(t1 = 0; t1 < t; t1++) //双重循环, 共延时t个半拍

for(t2 = 0; t2 < 8000; t2++); //延时期间, 可进入T0中断去发音

TR0 = 0; //关闭T0, 停止发音

}

//--------------------------------------

void song() //演奏一个音符

{

TH0 = timer0h; //控制音调

TL0 = timer0l;

TR0 = 1; //启动T0, 由T0输出方波去发音

delay(time); //控制时间长度

}

//--------------------------------------

void main(void)

{

while(1)

{

unsigned char k, i;

TMOD = 1; //置T0定时工作方式1

ET0 = 1; //开T0中断

EA = 1; //开CPU中断

while(1) {

i = 0;

time = 1;

while(time) {

k = sszymmh[i] + 7 \* sszymmh[i + 1] - 1;

//第i个是音符, 第i+1个是第几个八度

timer0h = FREQH[k]; //从数据表中读出频率数值

timer0l = FREQL[k]; //实际上, 是定时的时间长度

time = sszymmh[i + 2]; //读出时间长度数值

i += 3;

song(); //发出一个音符

}

}

}

}

编辑好代码之后，进行编译和构建，生成HEX文件用于仿真。

5系统调试与仿真及调试结果分析

仿真结果如图 5.1所示，仿真结果比较好的符合设计要求，播放出了预设的音乐《草原啊草原》。

图表

描述已自动生成

图 5.1 仿真图