重庆大学城市科技学院
CITY COLLEGE OF SCIENCE AND TECHNOLOGY,CHONGQING UNIVERSITY

毕　业　设　计

题目　基于 AT89C52单片机的音乐盒设计

学生所在学院　　电气信息学院

专　　业　　电子信息工程

学　　号　　*********

学 生 姓 名　　XXXXXX

指 导 教 师　　XXXXXXXXXX

助理指导教师　　XXXXXXXXXX

起 止 日 期　　2013.3.10-2013.5.15

目录

摘　要

本设计是一个基于 AT89C52系列单片机的音乐盒，依据单片机技术原理，通过硬件电路制作以及软件编译，设计制作出一个多功能音乐盒。该音乐盒主要由按键电路、复位电路、内部振荡器、数码管显示、中断控制电路以及蜂鸣器电路组成。使用三个按键控制音乐盒，两个用来切换歌曲，且数码管显示当前曲目。另一个用来控制开始 暂停。播放歌曲时，蜂鸣器发出某个音调，数码管显示与之对应的曲目。本设计利用 Keil 编程软件对音乐盒源程序进行编程并调试，配合 Proteus 仿真软件对硬件进行仿真调试，节约了设计时间。

关键字：音乐盒 AT89C52 单片机 Keil Proteus

1 引言

二十世纪九十年代以来计算机、信息、电子、控制、通信等技术得到迅速发展促使了社会生产力的提高,也使人们的生产方式和生活方式产生了日新月异的变化。随着人们生活水平的提高及对音乐的喜爱,对音乐播放器的品质、功能、品种等提出了越来越多的要求，表现在对控制系统性能、可靠性等要求越来越高。而品质的提高功能的更新、可靠性的增强、品种的变化无不与产品的核心控制部分水平的提高密不可分。家用音乐播放器产品及其它有关消费电器产品都是一些开环或闭环控制系统，都由核心控制部分、执行部分与人机界面三部分组成。而最为重要的控制部分一般是由单片机来执行完成的。这就必将导致和促进单片机在音乐领域应用的发展。现在这些由单片机实现的音乐播放器的功能越来越强、费用越来越低。例如：就目前市场上的MP3的功能越来越强大体积却越来越小，价格也逐渐便宜，被大多数人所能接受。但这些音乐播放器也或多或少的存在着一些问题。解决这些问题，非智能化的单片机莫属。本设计由硬件电路设计和软件程序设计两大部分组成。整个硬件电路是由中心控制、播放、按键、和数码显示模块组成。中心控制模块采用 AT89S52单片机，播放模块是由蜂鸣器组成，显示模块是由七段显示数码管 LED灯组成，按键模块由 3 个按键组成，其中 1 个作为开始/暂停切换，另外 2 个作为曲目切换按键。软件程序运用汇编语言编程实现。

1 概述

本设计是以AT89C52芯片的电路为基础，加上外部按键及放音设备，以此来实现音乐盒的硬件电路，通过软件程序来控制单片机内部的定时器使其演奏出优美动听的音乐。用户可以按照自己的喜好选择音乐并将其转化成机器码存入单片机的存储器中。该系统具有很好的通用性，很高的实际使用价值。

1.1 课题意义

音乐盒的起源，可追溯至中世纪欧洲文艺复兴时期。当时为使教会的钟塔报时，而将大小的钟表装上机械装置，被称为“可发出声音的组钟”。音乐盒有着 300 多年的发展历史，是人类文明发展的历史见证。传统的音乐盒多是机械音乐盒，其工作原理是通过齿轮带动一个带有铁钉的铁桶转动，铁桶上的铁钉撞击铁片制成的琴键，从而发出声音。但是，机械式的音乐盒体积比较大，比较笨重，且曲目单一。容易受水、灰尘等外在因素，使内部金属发音条变形，从而造成发音跑调。另外，机械音乐盒放音时为了让音色稳定，必须放平不能动摇，而且价格昂贵，不能实现大批量生产。本文设计的音乐盒，是基于单片机设计制作的电子式音乐盒。与传统的机械式音乐盒相比更小巧，能演奏多个曲目且携带方便。电子式音乐盒动力来源是电池，制作工艺简单，可进行批量生产，且价格便宜。基于单片机制作的电子式音乐盒，控制功能强大，可根据需要选歌，使用方便。根据存储容量的大小，可以尽可能多的存储歌曲。另外，可以设计彩灯外观效果，使音乐盒的功能更加丰富。

1.2 设计任务和要求

利用 I/O 口产生一定频率的方波，驱动蜂鸣器，发出不同的音调，从而演乐曲(内存五首乐曲)。采用七段数码管显示当前播放的歌曲序号。可通过按键选择乐曲，暂停 播放，上一曲，下一曲。

1.3 应用软件介绍

本设计利用 KEIL编程软件对音乐盒源程序进行编程并调试，配合 PROTEUS仿真软件对硬件进行仿真调试，两种软件的简介如下：

1.3.1 Proteus 软件简介

Proteus 软件是英国 Labcenter electronics 公司出版的 EDA工具软件（该软件中国总代理为广州风标电子技术有限公司）。它不仅具有其它 EDA工具软件的仿真功能，还能仿真单片机及外围器件。它是目前最好的仿真单片机及外围器件的仿真工具。虽然目前国内推广刚起步，但已受到单片机爱好者、从事单片机教学的教师、致力于单片机开发应用的科技工作者的青睐。Proteus 是世界上著名的 EDA工具(仿真软件)，从原理图布图、代码调试到单片机与外围电路协同仿真，一键切换到 PCB设计，真正实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB设计软件和虚拟模型仿真软件三合一的设计平台，其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DsPIC33 、AVR、ARM、8086 和 MSP430等，2010 年即将增加 Cortex 和 DSP系列处理器，并持续增加其他系列处理器模型。在编译方面，它也支持 IAR、Keil 和 MPLAB等多种编译器。

1.3.2 KEIL 简介

单片机开发中除必要的硬件外，同样离不开软件，我们写的汇编语言源程序要变为 CPU可以执行的机器码有两种方法，一种是手工汇编，另一种是机器汇编。机器汇编是通过汇编软件将源程序变为机器码，用于 MCS-51单片机的汇编软件有早期的 A51，随着单片机开发技术的不断发展，从普遍使用汇编语言到逐渐使用高级语言开发，单片机的开发软件也在不断发展，Keil 软件是目前最流行开发 MCS-51系列单片机的软件，这从近年来各仿真机厂商纷纷宣布全面支持 Keil 即可看出。Keil 提供了包括 C编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（uVision ）将这些部份组合在一起。运行 Keil 软件需要 Pentium 或以上的 CPU，16MB或更多 RAM、20M以上空闲的硬盘空间、WIN98、NT、WIN2000、WINXP等操作系统。掌握这一软件的使用对于使用 51 系列单片机的爱

好者来说是十分必要的，如果你使用 C语言编程，那么 Keil 几乎就是你的不二之选（目前在国内你只能买到该软件、而你买的仿真机也很可能只支持该软件），即使不使用 C语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。Keil C51 生成的目标代码效率非常之高，多数语句生成的汇编代码很紧凑，容易理解。在开发大型软件时更能体现高级语言的优势。

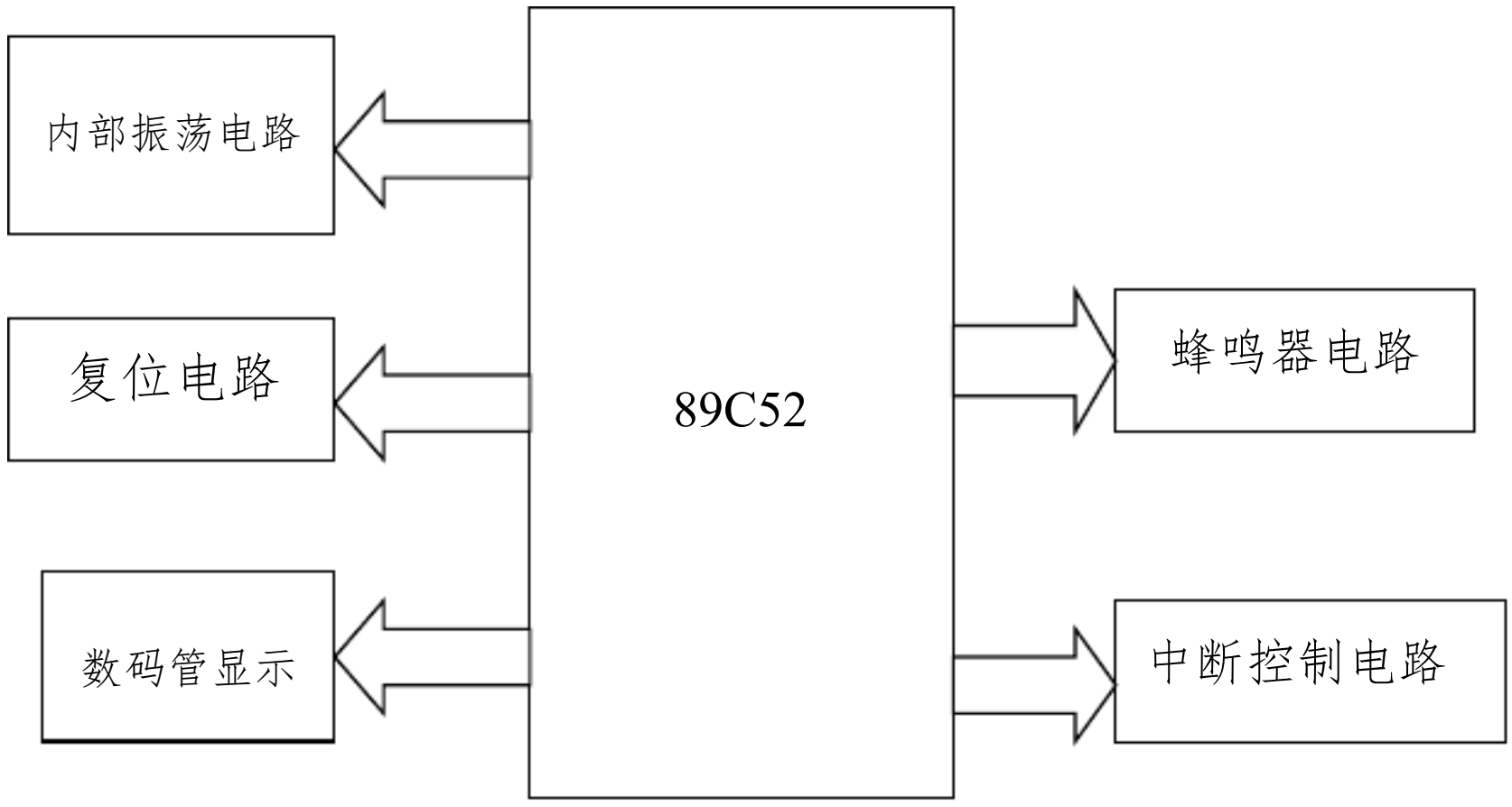
2 设计方案

本文提出了一种基于 AT89C52单片机的音乐盒设计方案，实现对蜂鸣器发音控制。AT89C52是一个低电压，高性能 CMOS 8位单片机，片内含 8k bytes 的可反复擦写的 Flash 只读程序存储器和 256 bytes 的随机存取数据存储器（RAM），器件采用 ATMEL公司的高密度、非易失性存储技术生产，兼容标准 MCS-51指令系统，片内置通用 8 位中央处理器和 Flash 存储单元，AT89C52单片机在电子行业中有着广泛的应用。

本方案以 AT89C52单片机与按键组成核心主控制模块。在主控模块上设有 3 个按键；根据用户需要可以编写出曲目代码并有数码显示对应曲目序号，利用其内部定时器 T0 实现定时中断。本方案中应用定时计数器 T0、T1，计数器 0 工作于方式 1，计数器 1 以计数方式，工作于方式 2。

本次系统中应用中断指令。当键盘有键按下时，判断键值，启动计数器 T0，产生一定频率的脉冲，驱动蜂鸣器，放出乐曲。同时启动定时器 T1,显示歌曲号。

单片机音乐盒的系统结构框图



3 时钟电路

单片机内有时钟电路，与振荡器共同产生单片机工作所需要的时钟信号。它使单片机在唯一的时钟信号控制下，严格地按一定的节拍进行工作（按一定的时序进行工作）。

振荡器可由单片机内振荡元件外接振荡元件实现，构成内部时钟方式，即在 XTAL1 和 XTAL2 引脚跨接振荡器元件（如晶振），则可构成一个稳定的自激振荡器。如果振荡器元件是晶振，则在晶振两个引脚上接上两个电容。电容主要是起频率微调和稳定的作用。电容容量一般为 20-40pf 基本都可以。本系统选择 12MHZ 的晶体振荡器，30pf 电容。如图 3 所示。

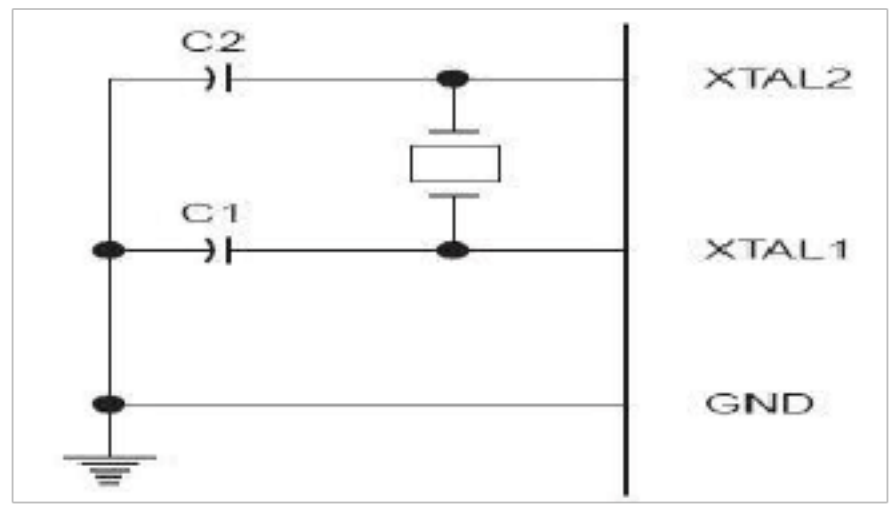


图 3

4 复位电路

复位是令单片机初始化的操作，其主要功能是初始化单片机的工作状态。例如，把程序计数器 PC的值初始化为 0000H，这样单片机在复位后就从程序存储器 ROM的 0000H单元开始执行程序。另外，当程序运行出错或因操作错误而使系统处于死锁状态时，按复位键来重新初始化单片机。RST引脚是复位信号的输入端。实现复位操作，必须使 RST引脚至少保持两个机器周期的高电平（一般用手按下去再放开，都能达到两个机器周期），再从高电平变为低电平，完成复位。复位后，单片机从 ROM中的 0000H单元开始执行程序。复位操作有上电自动复位、按键复位等方式。

4.1 上电复位

是通过外部复位电容充电来实现复位。由于电容通交流隔直流，在上电瞬间可等效交流电，在这一瞬间 RST引脚的电位与 VCC的电位一样。由于电容两端电压差不断增大，则 RST端的电压逐渐变小，直至电容充电完毕。电容的充电时间必须大于两个机器周期。如图 4-1。

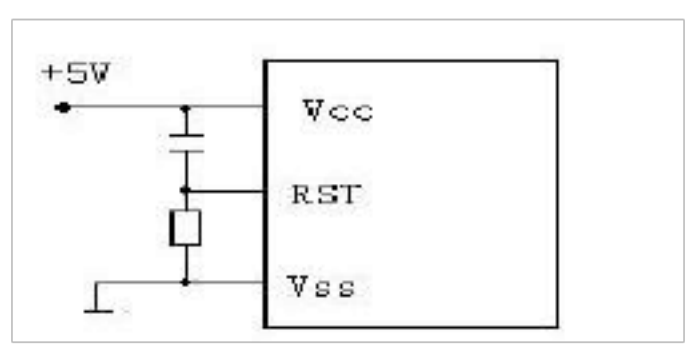


图 4-1

4.2 按键复位

是通过按键接通的瞬间，使 RST为高电平，实现复位功能。有些单片机内部设置 RST接收低电平时，实现复位功能 RST是高电平还是低电平时复位，要根据单片机的类型而定。如图 4-2。

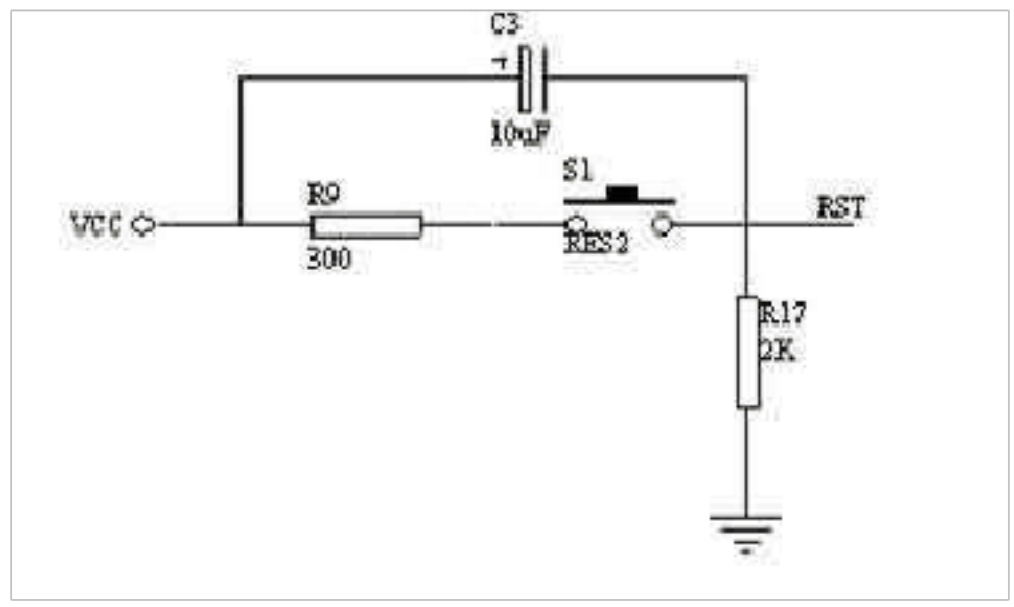


图 4-2

本系统选择上电复位。其中电容为 10uf，复位电阻为 10k。

硬件设计

音乐盒控制系统包括 2 大部分，即音乐盒控制模板（AT89C52主控模块）和音乐盒发音模块（数码管显示模块）。前者是主控模块，具有按键功能，并利用 AT89C52 的 P3 口输出控制信号；后者是受控模块，利用 AT89C52的 P1、P2 口输入控制信号，上面焊有蜂鸣器和数码显示器。

主控模块主要设计器件有 AT89C2 3 个按键。通过软件设计，使单片机 p3 口作为蜂鸣器和数码管的信号输出口。其中 k1 接 p3.2，作为上一曲的按键；k2 接 p3.3，作为下一曲的按键；k3 接 p3.5，作为开始/暂停的按键。如图 5-1。

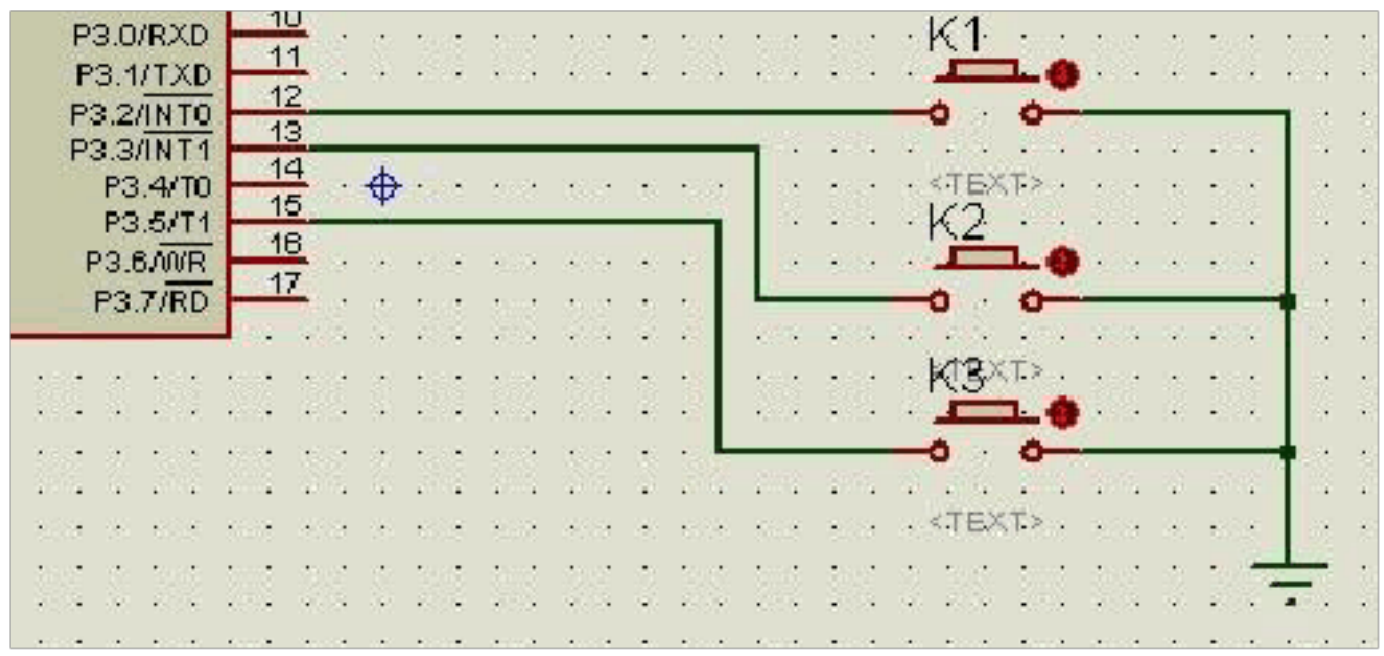


图 5-1

通过 K1,K2,K3 三个按键，分别实现上一曲，下一曲和开始/暂停的切换。

根据实际需要，使其与 AT89C52的 p2.0 口相接，另一引脚接地，实现演奏曲目的功能。

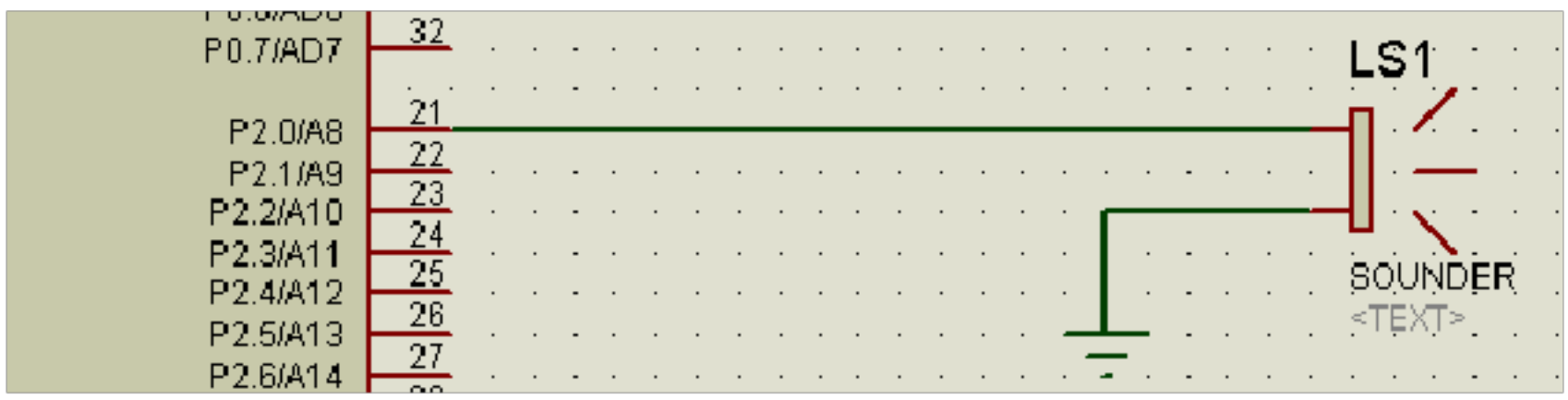


图 5-2-1

数码显示电路设计主要器件是数码管。使其与 AT89C52的 p1 口相接，另一引脚接电源，实现显示当前曲目的功能。

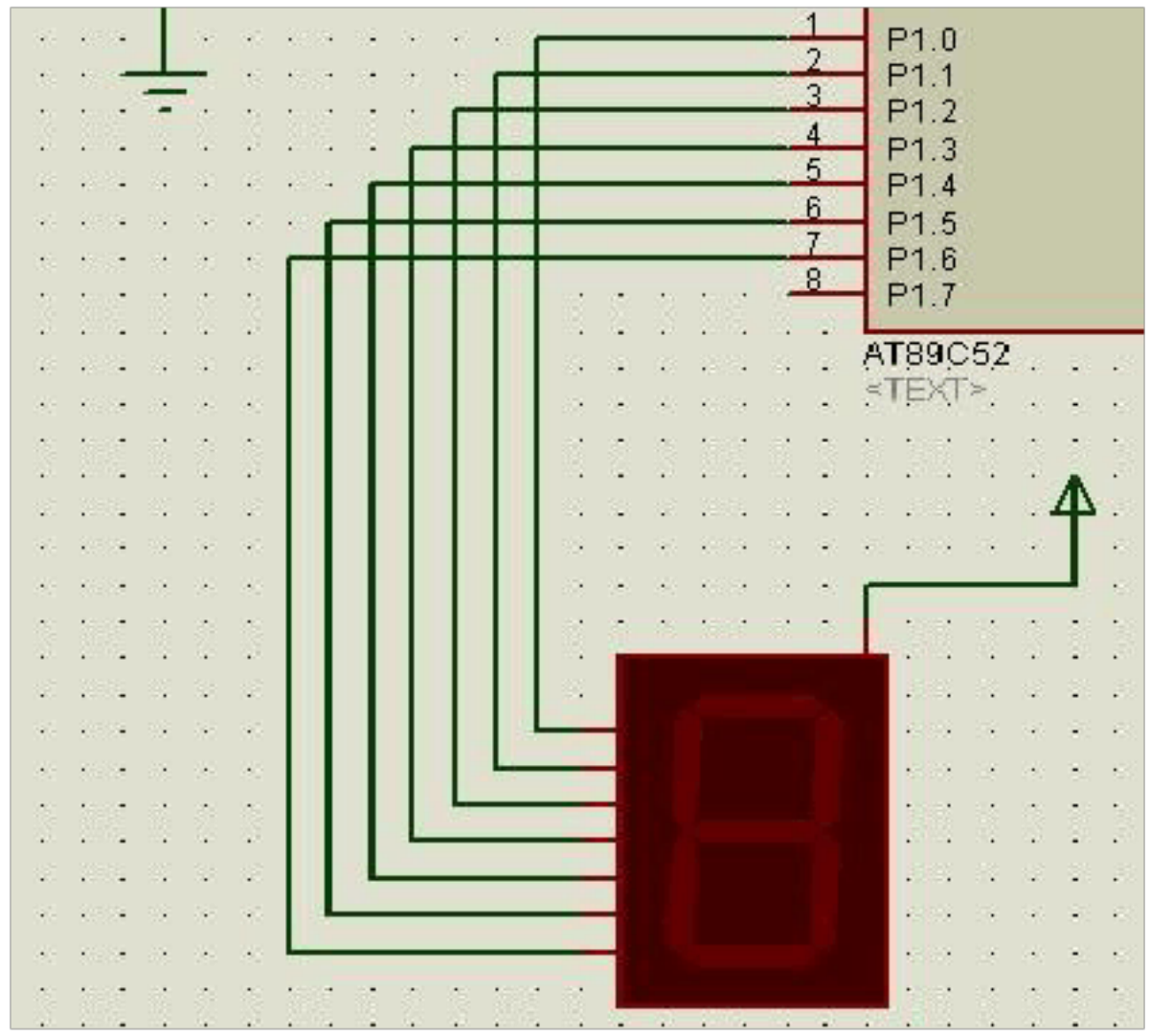
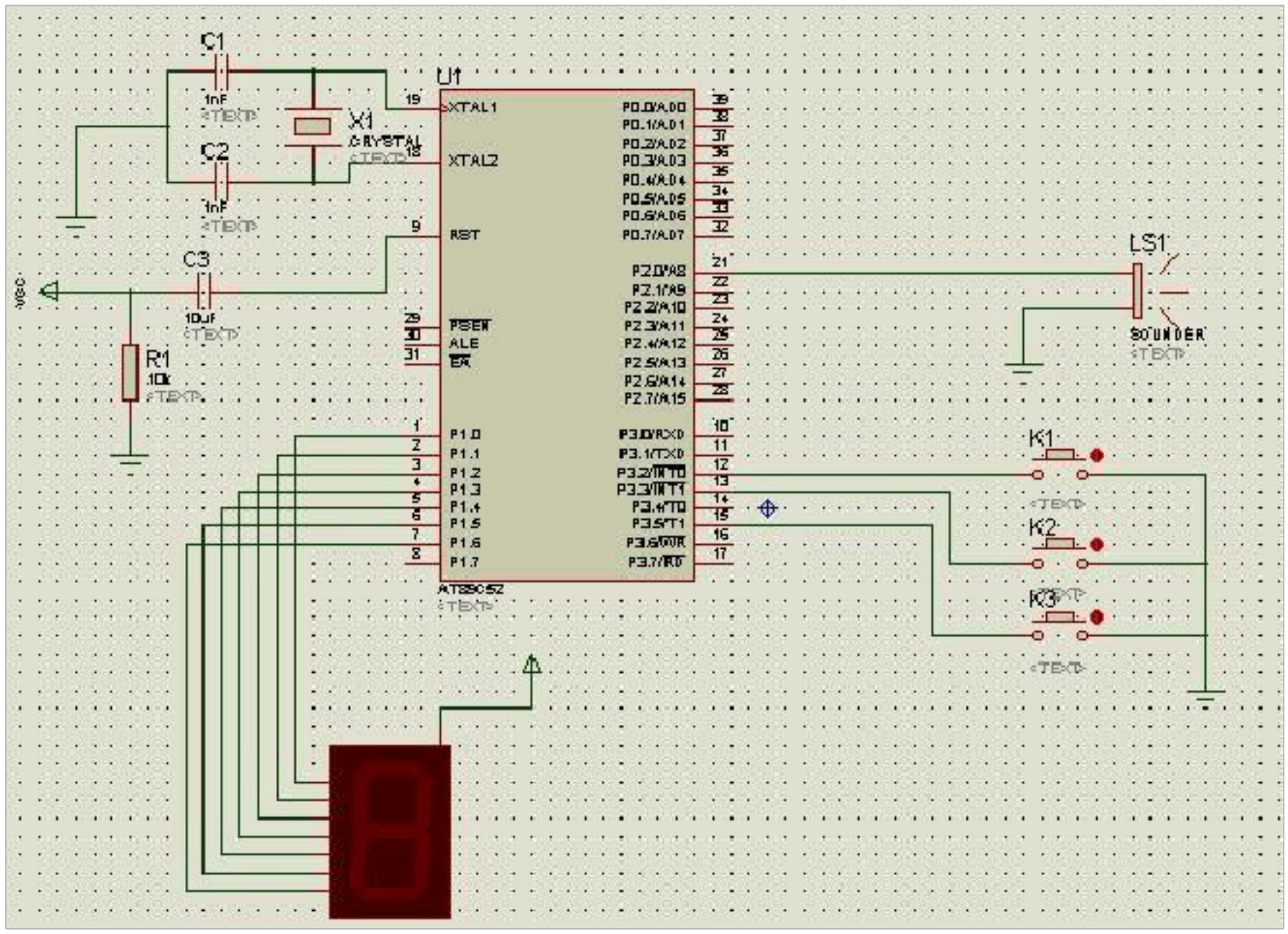


图 5-2-2



软件设计

在本程序中主要有节拍，音调、节拍以及编码的确定方法、开始/暂停、上一曲、下一曲四个软件模块组成。

6.1 音调、节拍以及编码的确定方法

一般说来，单片机演奏音乐基本都是单音频率，它不包含相应幅度的谐波频率，也就是说不能像电子琴那样能奏出多种音色的声音。因此单片机奏乐只需弄清楚两个概念即可，也就是“音调”和节拍表示一个音符唱多长的时间。

6.1.1音调的确定

不同音高的乐音是用C、D、E、F、G、A、B来表示，这7个字母就是音乐的音名，它们一般依次唱成DO、RE、MI、FA、SO、LA、SI,即唱成简谱的1、2、3、4、5、6、7，相当于汉字“多来米发梭拉西”的读音，这是唱曲时乐音的发音，所以叫“音调”，即Tone。把C、D、E、F、G、A、B这一组音的距离分成12个等份，每一个等份叫一个“半音”。两个音之间的距离有两个“半音”，就叫“全音”。在钢琴等键盘乐器上，

－D、D－E、F－G、G－A、A－B两音之间隔着一个黑键，他们之间的距离就是全音；E－F、B－C两音之间没有黑键相隔，它们之间的距离就是半音。通常唱成1、2、3、4、5、6、7的音叫自然音，那些在它们的左上角加上#号或者b号的叫变化音。#叫升记号，表示把音在原来的基础上升高半音，b叫降记音，表示在原来的基础上降低半音。例如高音DO的频率（1046Hz）刚好是中音DO的频率（523Hz）的一倍，中音DO的频率（523Hz）刚好是低音DO频率（266 Hz）的一倍；同样的，高音RE的频率（1175Hz）刚好是中音RE的频率（587Hz）的一倍，中音RE的频率（587Hz）刚好是低音RE频率（294 Hz）的一倍。

1）要产生音频脉冲，只要算出某一音频的周期（1/频率），然后将此周期除以2，即为半周期的时间。利用定时器计时这半个周期时间，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时间再对I/O 反相，就可在I/O 脚上得到此频率的脉冲。

2）利用AT89C51的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。

此外结束符和休止符可以分别用代码 00H和FFH来表示，若查表结果为00H，则表示曲子终了；若查表结果为FFH，则产生相应的停顿效果。

3）例如频率为 523Hz，其周期 T=1/523=1912us，因此只要令计数器计时956us/1us=956在每次计算956次时将I/O反相，就可得到中音DO（523Hz）。

计数脉冲值与频率的关系公式如下：

N=Fi/Fr

N：计算值；　　Fi：内部计时一次为1us，故其频率为1MHz

4）其计数值的求法如下：

T=65536-N=65536-Fi/Fr

例如：设K=65536，F=1000000=Fi=1MHz 球低音DO（261Hz）。中音DO（523Hz）。高音的DO（1046Hz）的计算值

T=65536-N=65536-Fi/Fr=65536-1000000/Fr=65536-500000/Fr

低音DO的 T=65536-500000/262=63627

低音DO的 T=65536-500000/523=64580

低音DO的 T=65536-500000/1047=65059

5）C调各音符频率与计数值T的对照表如表4.1所示。

表4.1 C 调各音符频率与计数值T的对照表

低音	频率	T 参数		中音	频率	T 参数		高音	频率	T 参数	
Do	262	1908	229	Do	523	956	115	Do	1046	57	57
Do#	277	1805	217	Do#	554	903	108	Do#	1109	54	54
Re	294	1701	204	Re	587	852	102	Re	1175	51	51
Re#	311	1608	193	Re#	622	804	97	Re#	1245	48	48
Mi	330	1515	182	Mi	659	759	91	Mi	1318	45	45
Fa	349	1433	172	Fa	698	716	86	Fa	1397	43	43
Fa#	370	1351	162	Fa#	740	676	81	Fa#	1480	41	41
So	392	1276	153	So	784	638	77	So	1568	38	38
So#	415	1205	145	So#	831	602	72	So#	1661	36	36

	440	1136	136	La	880	568	68	La	1760	34	34
La	464	1078	129	La	932	536	64	La	1865	32	32
#				#				#			
Si	494	1012	121	Si	988	506	61	Si	1976	30	30

节拍的确定

若要构成音乐,光有音调是不够的,还需要节拍,让音乐具有旋律(固定的律动),而且可以调节各个音的快满度。“节拍”,即 Beat，简单说就是打拍子，就像我们听音乐不自主的随之拍手或跺脚。若 1 拍实 0.5s ，则 1/4 拍为 0.125s 。至于 1 拍多少 s，并没有严格规定，就像人的心跳一样，大部分人的心跳是每分钟 72 下，有些人快一点，有些人慢一点，只要听的悦耳就好。音持续时间的长短即时值，一般用拍数表示。休止符表示暂停发音。

一首音乐是由许多不同的音符组成的，而每个音符对应着不同频率，这样就可以利用不同的频率的组合，加以与拍数对应的延时，构成音乐。了解音乐的一些基础知识，我们可知产生不同频率的音频脉冲即能产生音乐。对于单片机来说，产生不同频率的脉冲是非常方便的，利用单片机的定时/计数器来产生这样的方波频率信号。因此，需要弄清楚音乐中的音符和对应的频率，以及单片机定时计数的关系。

表 4.2 节拍与节拍码对照

节拍码	节拍数	节拍码	节拍数
1	1/4 拍	1	1/8 拍
2	2/4 拍	2	1/4 拍
3	3/4 拍	3	3/8 拍
4	1 拍	4	2/1 拍
5	1 又 1/4 拍	5	5/8 拍
6	1 又 1/2 拍	6	3/4 拍
8	2 拍	8	1 拍
A	2 又 1/2 拍	A	1 又 1/4 拍
C	3 拍	C	1 又 1/2 拍
F	3 又 3/4 拍		

每个音符使用 1 个字节,字节的高 4 位代表音符的高低,低 4 位代表音符的节拍,图 5.2 为节拍码的对照。如果 1 拍为 0.4 秒，1/4 拍实 0.1 秒，只要设定延迟时间就可求得节拍的时间。假设 1/4 拍为 1DELAY,则 1 拍应为 4DELAY,以此类推。所以只要求得 1/4 拍的 DELAY 时间，其余的节拍就是它的倍数，如图 5.3 为 1/4 和 1/8 节拍的时间设定。

表 4.3 1/4 和 1/8 节拍的时间设定

曲调值	DELAY	曲调值	DELAY
调 4/4	125 毫秒	调 4/4	62 毫秒
调 3/4	187 毫秒	调 3/4	94 毫秒
调 2/4	250 毫秒	调 2/4	125 毫秒

do re mi fa so la si　　分别编码为 1~7，重音 do 编为 8,重音 re 编为 9，停顿编为 0。播放长度以十六分音符为单位（在本程序中为 165ms)，一拍即四分音符等于 4 个十六分音符，编为 4,其它的播放时间以此类推。音调作为编码的高 4 位，而播放时间作为低 4 位，如此音调和节拍就构成了一个编码。以 0xff 作为曲谱的结束标志。

举例 1：音调 do,发音长度为两拍，即二分音符，将其编码为 0x18。

举例 2：音调 re, 发音长度为半拍，即八分音符，将其编码为 0x22

歌曲播放的设计。先将歌曲的简谱进行编码，储存在一个数据类型为 unsigned char 的数组中。程序从数组中取出一个数，然后分离出高 4 位得到音调，接着找出相应的值赋给定时器 0，使之定时操作蜂鸣器，得出相应的音调；接着分离出该数的低 4 位，得到延时时间，接着调用软件延时。

表 4.4 简谱对应的简谱码、T 值、节拍数

简谱	发音	简谱码	T 值	节拍码	节拍数
5	低音 SO	1	64260	1	1/4 拍
6	低音 LA	2	64400	2	2/4 拍
7	低音 TI	3	64524	3	3/4 拍
1	中音 DO	4	64580	4	1 拍
2	中音 RE	5	64684	5	1 又 1/4 拍
3	中音 MI	6	64777	6	1 又 1/2 拍
4	中音 FA	7	64820	8	2 拍
5	中音 SO	8	64898	A	2 又 1/2 拍
6	中音 LA	9	64968	C	3 拍
7	中音 TI	A	65030	F	3 又 3/4 拍
1	高音 DO	B	65058		
2	高音 RE	C	65110		
3	高音 MI	D	65157		
4	高音 FA	E	65178		
5	高音 SO	F	65217		

计数器 0 工作于方式 1，计数器 1 以计数方式，工作于方式 2，接开始和暂停

```
CLR PT0                    ;                    计数器 0 为低优先级
SETB PT1                                        ;计数器 1 为高优先级
```

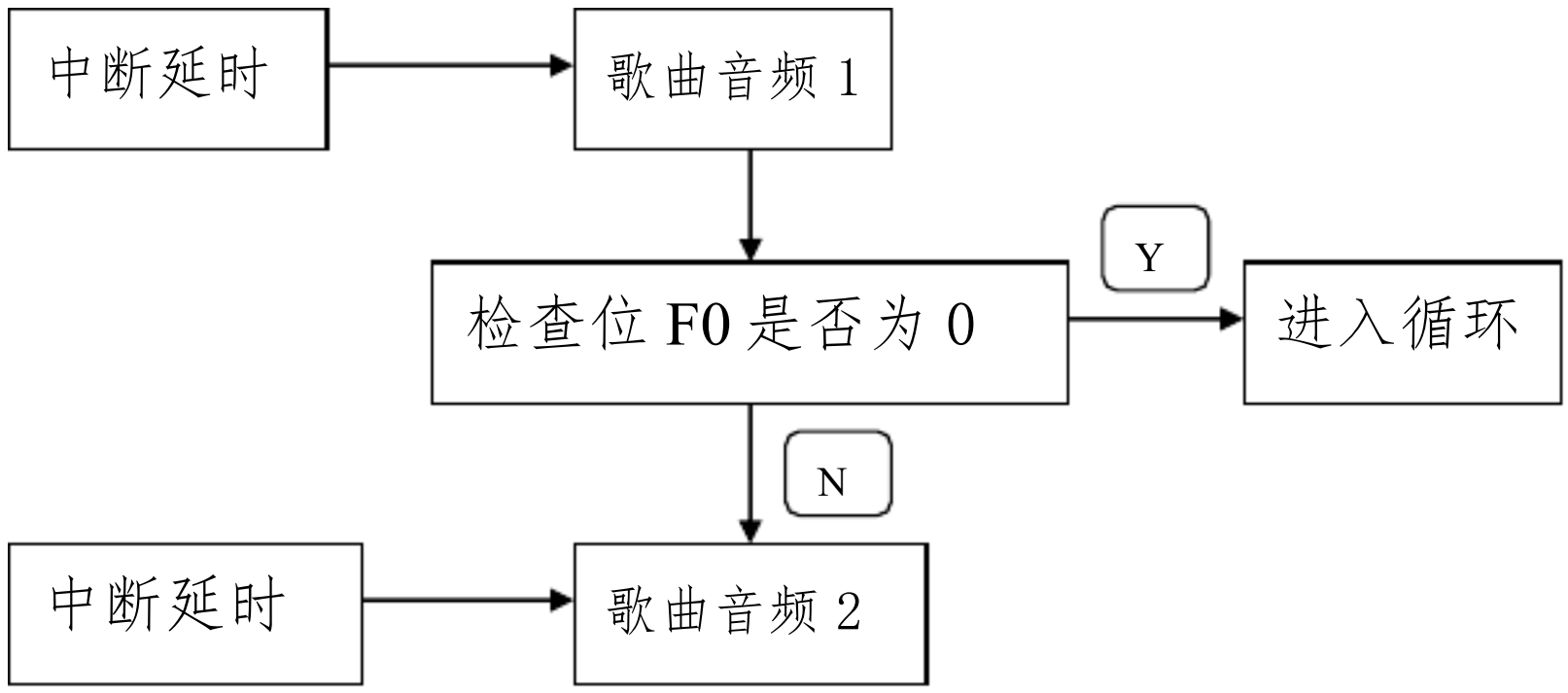
计数器 1 以计数方式，工作于方式 2，相当于外部中断，将 TH1, TL1, 都设置为#0FF, 利用重载形成循环

```
MOV TL1,#0FFH
MOV TH1,#0FFH
```

取一个位 F0，初始值为 0，CLR F0;

一旦 T1 进入中断服务就将取反，CPL F0，同时也在每一个音符开始唱时，设置检验 F0 是否为一。

```
    LCALL DELAY                ;              每个音符唱多久
    JB F0,FOR                                 ; 是否暂停
```



歌曲播放 暂停

6.3 上一曲程序设计

通过外部中断 0，$\overline{\text{INT0}}$，下降沿触发方式，每次通过下降沿触发一次触发中断，将当前地址压栽，保护现场　PUSH DPH

PUSH DPL

```
判断是否指向第一曲          MOV A,22H          ;          曲目数送 A
                            CJNE A,#1,QQ       ;          是否是第一首
  回到上一曲首地址将 R7 减 4    MOV A,R7                  ;R7     减 4
                          SUBB A,#4
```

通过查表，改变数码管显视相应歌曲的编号

```
      MOV DPTR,#OUT_TAB
        MOVC A,@A+DPTR
        MOV OUT_NUM,A  ;  数码管显视相应歌曲的编号
```

OUT_TAB:DB0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H,88H,83H,0C6H,0A1H,86H,8EH,0FFH ； 共阳

对即将播放的歌曲的几分拍进行调整,利用已经改变的首地址，调用节拍。最后中断返回

上一曲总程序：

```
LAST_SONG: PUSH ACC          ;      上一曲中断程序处理开始，保护现场
        PUSH DPH
        PUSH DPL
        CLR EA               ;              关中断
        MOV A,22H            ;              曲目数送 A
          CJNE A,#1,QQ                    ;是否是第一首
        MOV 22H,#N           ;              是第一首歌曲
        MOV B,#4             ;              将 R7指向最后一首歌曲
          MOV A,#N-1
          MUL AB
          MOV R7,A
          AJMP BACK2         ;          处理结束
QQ:       DEC 22H
        MOV A,R7                        ;R7     减 4
          SUBB A,#4
          MOV R7,A
BACK2:     MOV R4,#00H
          MOV A,22H
            MOV DPTR,#OUT_TAB
            MOVC A,@A+DPTR
            MOV OUT_NUM,A     ;     数码管显视相应歌曲的编号
;*****************************************
   MOV  B,R0;对下首要演唱的歌曲的几分拍进行调整,同时对R0中的内容进行保护
   MOV R4,#00H
   MOV R0,#30H
```

```
MOV A,R7
ADD A,R0
MOV R0,A
MOV DPH,@R0
INC R0
MOV DPL,@R0
INC R0
MOV A,R4
INC R4
MOVC A,@A+DPTR
    MOV  26H,A                    ; 结果存入 26H单元中
    DEC R4
    MOV R0,B                      ;R0 中的内容恢复
;*********************************************
      POP DPL                     ; 恢复现场
     POP DPH
     POP ACC
       SETB EA
       RETI          ;            中断返回
```

6.4下一曲程序设计

通过外部中断 1，$\overline{INT1}$，下降沿触发方式，每次通过下降沿触发一次触发中断，将当前地址压栈。

```
保护现场      PUSH DPH    ;
              PUSH DPL;
判断是否是最后一曲
      CJNE A,#N,Q                 ; 是最后一首吗？
       MOV R7,#00H                ; 是最后一首，则 R7指向第一首，演唱第一首
```

通过查表，改变数码管显视相应歌曲的编号