

Lab#1p 单片机汇编语言基本编程 以及 GPIO 操作

2.1 实验内容

使用 8051 单片机和蜂鸣器，使用 C51 汇编语言编程，实现播放音乐功能。理解使用汇编语言实现循环、分支、函数调用等功能。理解 GPIO 的基本操作。

2.2 单片机产生乐曲的原理

(<https://wenku.baidu.com/view/2eaa076da45177232f60a280.html?re=view>)

利用单片机产生乐曲音符，再把乐曲音符翻译成计算机音乐语言，由单片机进行信息处理,再通过蜂鸣器或喇叭放出音乐。

2.2.1 音频脉冲的产生：

音乐的产生主要是通过单片机的 I/O 口输出高低不同的脉冲信号来控制蜂鸣器发音，要想产生音频脉冲信号，需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。利用单片机定时器计时这个半个周期时间，每当计时到后就将输出脉冲的 I/O 口反相，然后重复计时此半周期时间再对 I/O 口反相，这样就能在此 I/O 口上得到此频率的脉冲。

通常，利用单片机的内部定时器 0，工作在方式 1 下，改变计数初始值 TH0 和 TL0 来产生不同的率。

例如，若单片机采用 12MHz 晶振，要产生频率为 587Hz 的音频脉冲时，其音频信号的脉冲周期 $T=1/587=1703.5775 \mu s$ ，半周期的时间为 $852 \mu s$ ，因此只要令计数器计数 $=852 \mu s / 1 \mu s = 852$ ，在每计数 852 时将 I/O 口反相，就可得到 C 调中音 Re。

计数脉冲值与频率的关系如下：

音符	频率 Hz	周期 μs	音符	频率 Hz	周期 μs	音符	频率 Hz	周期 μs
低 1Do	262	3816	中 1Do	523	1912	高 1Do	1047	955
低 2Re	294	3401	中 2Re	587	1703	高 2Re	1175	851
低 3Mi	330	3030	中 3Mi	659	1517	高 3Mi	1319	758
低 4Fa	349	2865	中 4Fa	698	1432	高 4Fa	1397	751
低 5So	392	2551	中 5So	784	1275	高 5So	1568	637
低 6La	440	2272	中 6La	880	1136	高 6La	1760	568
低 7Si	494	2024	中 7Si	988	1012	高 7Si	1967	508

2.2.2 音频节拍的产生:

每个音符使用一个字节，字节的高四位代表音符的高低，低四位代表音符的节拍，表 5-2 为节拍与节拍码的对照表。如果 1 拍为 0.4s，1/4 拍是 0.1s 只要设定延迟时间就可求得节拍的时间。假设 1/4 拍为 1 个延迟单位时间，那么 1 拍对应的就是 4 个延迟单位时间，以此类推。所以只要求得 1/4 拍的延迟单位时间，其余的节拍就是它的倍数，表 5-3 为 1/4 拍和 1/8 拍的时间设定。

表 5-2 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	1/2 拍
5	$1\frac{1}{4}$ 拍	5	5/8 拍
6	$1\frac{1}{2}$ 拍	6	3/4 拍
8	2 拍	8	1 拍
A	$2\frac{1}{2}$ 拍	A	$1\frac{1}{4}$ 拍
C	3 拍	C	$1\frac{1}{2}$ 拍
F	$3\frac{3}{4}$ 拍		

表 5-3 1/4 和 1/8 拍的时间设定

1/4 拍的时间设定		1/8 拍的时间设定	
曲调值	DELAY (延迟时间)	曲调值	DELAY
调 4/4	125ms	调 4/4	62ms
调 3/4	187ms	调 3/4	94ms
调 2/4	250ms	调 2/4	125ms

1/4 拍的延迟时间=187ms。

2.2.3 简单音乐示例

以下为使用函数调用和循环方式产生闹钟声音的一段示例代码。编写音乐代码的过程如下：

1. 构建音符表，决定当前演奏的音调，如下列例子中的 notes；
2. 构建延迟时间表，决定当前音的持续时间，根据节拍决定，如下列例子中的 duration；
3. 使用函数、循环、分支等方式，编写单个频率程序，通过控制 I/O 口，如下列例子中的 P1.3，实现频率输出。

2.3 作业

1. 编写汇编程序，播放一段音乐片段（需要有不同节拍）。阐述实现原理，分析代码延时，使用程序框图等描述编程思路。(50%)
2. 使用 Protues 仿真或者在试验箱硬件播放该音乐片段。(10%)
3. 通过按键连接 I/O 口，控制 8051 选择播放不同的预设音乐。阐述编程思路。(20%)
4. 自由发挥部分。(20%)

```

1      org 0000H
2      sjmp start
3  notes:      db 24      ; the delay time of high do
4  duration:   db 255    ; the duration lasting for 25 ms
5
6      org 0040H
7  start:      acall beep
8              acall delay_short
9              acall beep
10             acall delay_short
11             acall beep
12             acall delay_long
13             acall delay_long
14             acall delay_long
15             sjmp start
16
17  beep:       mov a,#0
18             mov dptr,#duration
19             movc a,@a+dptr
20             mov R2,a
21  loop3:      mov a,#0
22             mov dptr,#notes
23             movc a,@a+dptr
24             mov R1,a
25  loop2:      mov R0,#10
26  loop1:      djnz R0,$
27             djnz R1,loop2
28             cpl P3.1
29             djnz R2,loop3
30             ret
31
32  delay_short: mov R1,50
33  delay2:      mov R0,#50
34  delay1:      djnz R0,$
35             djnz R1,delay2
36             ret
37
38  delay_long:  mov R1,255
39  delay4:      mov R0,#255
40  delay3:      djnz R0,$
41             djnz R1,delay4
42             ret
43
44             sjmp $
45             end

```

示例程序 1：闹钟

```

1      org 0000H
2      sjmp start
3notes: db 96,85,76,72,64,57,51,48 ; the delay time of do, re, mi, fa, sol, la, ti
4duration: db 131,147,165,175,196,220,255 ; the duration of each note lasting for 25 ms
5
6      org 0040H
7start: mov R3,#7
8loop4: jb P1.0,forward
9      dec R3 ; play ti, la, sol, fa, mi, re, do
10     sjmp mod7
11forward: inc R3 ; play do, re, mi, fa, sol, la, ti
12mod7:  mov a,R3
13      mov b,#8
14      div ab
15      mov a,b
16      mov R3,b
17      mov dptr,#duration
18      movc a,@a+dptr
19      mov R2,a
20loop3: mov a,b
21      mov dptr,#notes
22      movc a,@a+dptr
23      mov R1,a
24loop2: mov R0,#10
25loop1: djnz R0,$
26      djnz R1,loop2
27      cpl P3.1
28      djnz R2,loop3
29      mov R1,255
30delay2: mov R0,#255
31delay1: djnz R0,$
32      djnz R1,delay2
33      sjmp loop4
34      sjmp $
35      end

```

示例程序 2：音阶播放