


目录 Contents

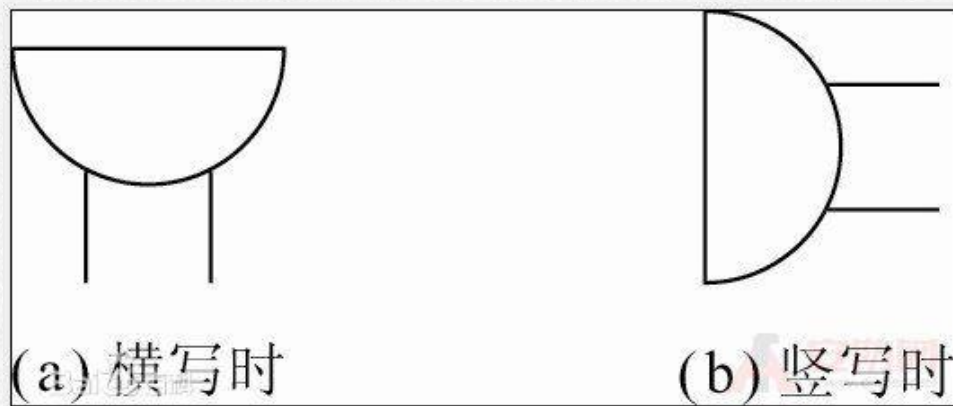
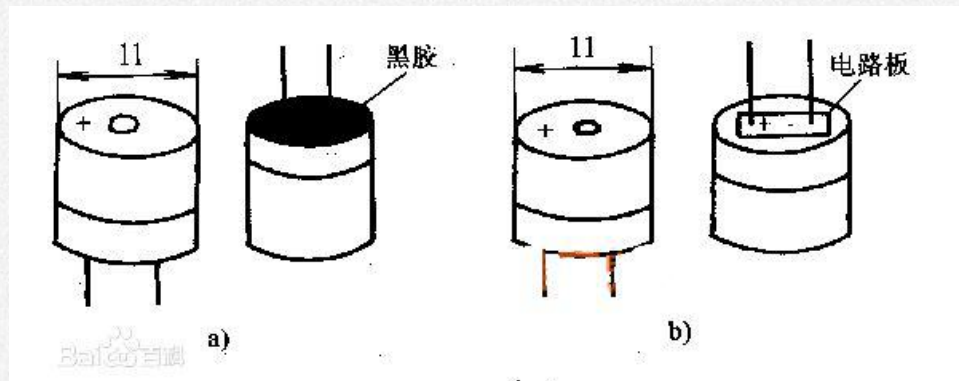
第一部分  蜂鸣器

第二部分  调节PWM频率

第三部分  一首乐曲

蜂鸣器介绍

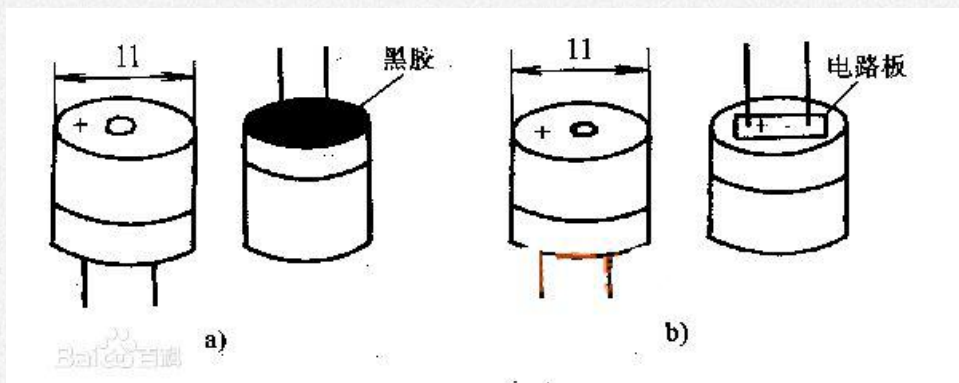
蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器可以分为有源蜂鸣器和无源蜂鸣器两种类型。蜂鸣器在电路中用字母“H”或“HA”（旧标准用“FM”、“ZZG”、“LB”、“JD”等）表示。



蜂鸣器介绍

有源蜂鸣器与无源蜂鸣器的区别：

首先大家要了解有源和无源这里的“源”不是指电源，而是指震荡源。也就是说，有源蜂鸣器内部带震荡源，所以只要一通电就会叫。而无源内部不带震荡源，所以如果用直流信号无法令其鸣叫。必须用 $2K\sim 5K$ 的方波去驱动它。有源蜂鸣器往往比无源的贵，就是因为里面多个震荡电路。这就是通过驱动原理来分别的方法。（图中a为有源蜂鸣器，b为无源蜂鸣器）



■ 蜂鸣器介绍

无源蜂鸣器的优点是：

- 1、便宜
- 2、声音频率可控，可以做出“多来米发索拉西”的效果
- 3、在一些特例中，可以和LED复用一個控制口

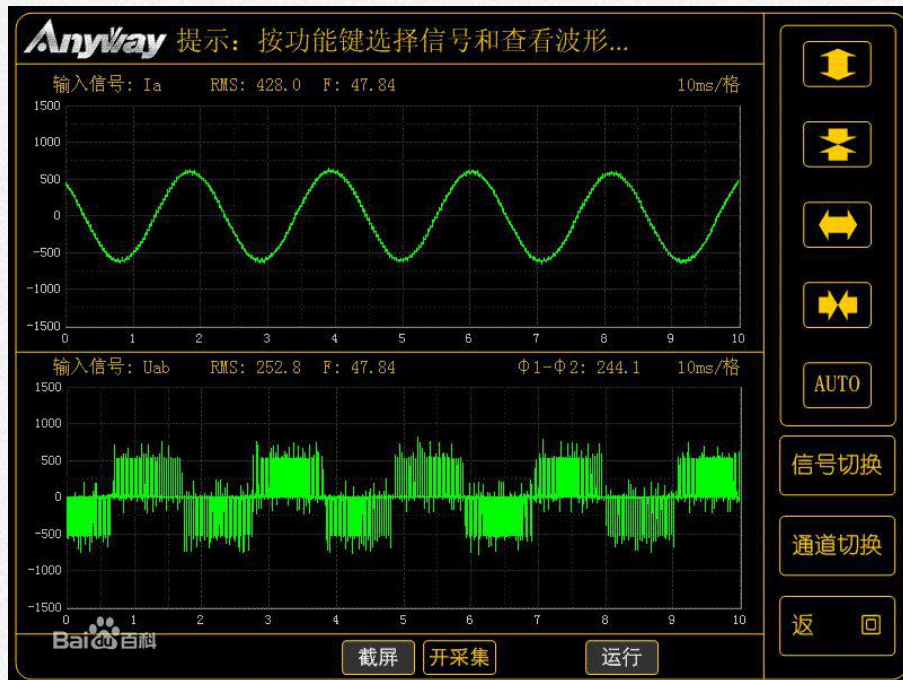
有源蜂鸣器的优点是：程序控制方便。

因为无源蜂鸣器便宜，所以大部分人手里拿到的都是无源蜂鸣器，本次实验使用的是也是无源蜂鸣器。

PWM介绍

脉冲宽度调制(PWM), 是英文“Pulse Width Modulation”的缩写, 简称脉宽调制, 是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术, 广泛应用于从测量、通信到功率控制与变换的许多领域中。

基本原理: 控制方式就是对逆变电路开关器件的通断进行控制, 使输出端得到一系列幅值相等的脉冲, 用这些脉冲来代替正弦波或所需要的波形。



■ PWM介绍

PWM是Timer的一种工作模式，它需要使用到Timer和Pin两个库。
先看一个基本的PWM例子，驱动LED4（蓝色LED灯）。

```
from pyb import Pin, Timer
tm3=Timer(3, freq=200)
led3.pulse_width_percent(10)
led3=tm3.channel(1,Timer.PWM,pin=Pin.cpu.B4,pulse_width_percent=50)
```


PWM介绍

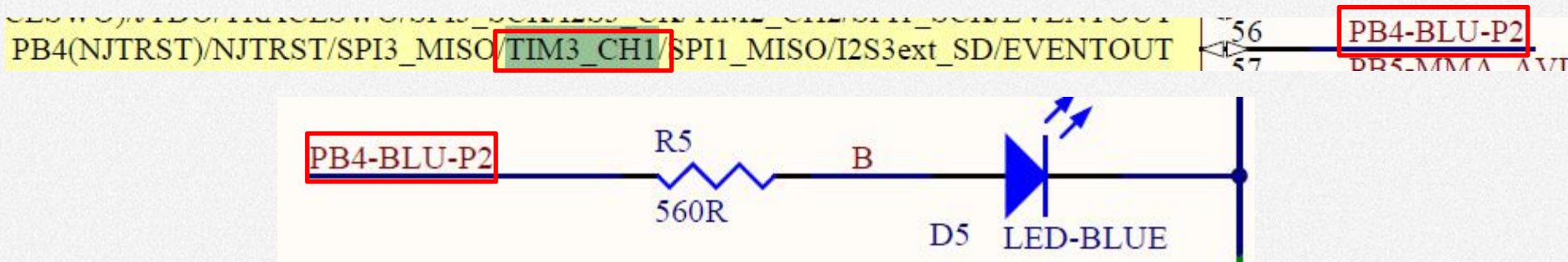
从上例可以看出，首先使用Timer设定定时器，然后指定Timer的通道，并设定PWM模式、关联的Pin等参数，最后设置输出脉冲宽度或者脉冲宽度百分比（占空比）。

其中，关于Pin口的Timer设定与通道选择需要参考电路图。

例：

```
tm3=Timer(3, freq=200)
```

```
led3=tm3.channel(1,Timer.PWM,pin=Pin.cpu.B4,pulse_width_percent=50)
```



蜂鸣器演奏乐曲











无源蜂鸣器内部结构相当于电磁场扬声器，可以通过给它输出一定频率的信号来发声。一段音乐就是不同频率的声音按一定的时间节拍转换发出。所以音乐包含音调和节拍信息

音符	频率 Hz	音符	频率 Hz	音符	频率 Hz
低1 Do	262	中1 Do	523	高1 Do	1047
低2 Re	294	中2 Re	587	高2 Re	1175
低3 Mi	330	中3 Mi	659	高3 Mi	1319
低4 Fa	349	中4 Fa	698	高4 Fa	1397
低5 So	392	中5 So	784	高5 So	1568
低6 La	440	中6 La	880	高6 La	1760
低7 Si	523	中7 Si	988	高7 Si	1976

蜂鸣器演奏乐曲

延迟时长和节拍的对应如下：

附录三： 五线谱与简谱的音符时值对照表

名 称	五线谱记谱	时 值 (以四分音符为一拍)	简谱记法
全 音 符		四 拍	$\overset{5}{\text{—}} \overset{\text{—}}{\text{—}} \overset{\text{—}}{\text{—}} \overset{\text{—}}{\text{—}}$ (或 $\underset{\cdot}{5} \quad \underset{\cdot}{5} \quad \underset{\cdot}{5} \quad \underset{\cdot}{5}$)
二 分 音 符		二 拍	$\overset{5}{\text{—}} \text{—}$ (或 $\underset{\cdot}{5} \quad \underset{\cdot}{5}$)
四 分 音 符		一 拍	$\underset{\cdot}{5}$
八 分 音 符		$\frac{1}{2}$ 拍 (半拍)	$\underline{\underset{\cdot}{5}}$
十六分音符		$\frac{1}{4}$ 拍 (四分之一拍)	$\underline{\underline{\underset{\cdot}{5}}}$
全 休 止 符		四 拍	0 0 0 0
二分休止符		二 拍	0 0
四分休止符		一 拍	0
八分休止符		$\frac{1}{2}$ 拍 (半拍)	$\underline{0}$
十六分休止符		$\frac{1}{4}$ 拍 (四分之一拍)	$\underline{\underline{0}}$

(注：简谱在音符的后面每加一条短横线表示增加一拍的长度。)

蜂鸣器演奏乐曲

知道了音符对应的频率及节拍时延，就可以设定PWM的频率来使蜂鸣器奏出响应的音符，之后根据节拍的不同设置延迟函数即可完成一首乐曲的演奏，本次实验将使用梁祝的名曲《化蝶》

The image displays a musical score for the piece 'The Butterfly Lovers' (化蝶) using numbered notation (1-7). The score is organized into three horizontal staves, each with a series of notes and lyrics underneath. The notes are connected by horizontal lines, indicating melodic phrases. The lyrics are written in Chinese characters.

Staff 1:

3 5. 6 1. 2 6 1 5 | 5. 1 6 5 3 5 2 - | 2 2 3 7 6 5. 6 1 2 |

碧草 青 青 花 盛 开 彩 蝶 双 双
同窗 共 读 整 三 载 促 膝 并
楼台 一 别 恨 如 海 泪 染 双 翅 身

Staff 2:

3 1 6 5 6 1 5 - | 3. 5 7 2 6 1 5 5 | 3 5 3 5 6 7 2 6. 5 6 |

人 排 徊 千 古 传 颂 深 深 爱
两 翩 无 猜 十 八 相 送 情 切 切
翩 花 丛 历 尽 磨 难 真 情 在

Staff 3:

1. 2 5 3 2 3 2 1 6 5 | 3 1 6 1 6 5 3 5 6 1 | 5 - 0 0 ||

山 伯 永 恋 祝 英 台
谁 知 一 别 在 楼 分 开
天 长 地 久 不 分

蜂鸣器演奏乐曲

知道了音调的频率与时值后，将其写入程序中。代码如下：

定义低音音名

L1=262 # c
L2=294 # d
L3=330 # e
L4=349 # f
L5=392 # g
L6=440 # a1
L7=494 # b1

定义中音音名

M1=523 # c1
M2=587 # d1
M3=659 # e1
M4=698 # f1
M5=784 # g1
M6=880 # a2
M7=988 # b2

定义高音音名

H1=1047 # c2
H2=1175 # d2
H3=1319 # e2
H4=1397 # f2
H5=1568 # g2
H6=1760 # a3
H7=1976 # b3

定义时值单位，决定演奏速度（数值单位：ms）

T=3600

蜂鸣器演奏乐曲

根据乐谱与时值，我们将《化蝶》其中一部分翻译为代码，使用一个二维数组来表示音乐，代码如下：

```
MyScore = [[L3, T/4], [L5, T/8+T/16], [L6, T/16], [M1, T/8+T/16], [M2,  
T/16], [L6, T/16], [M1, T/16], [L5, T/8], [M5, T/8+T/16], [H1, T/16],  
[M6, T/16], [M5, T/16], [M3, T/16], [M5, T/16], [M2, T/2], [ 0, 0] ]  
# 省略后续乐曲数据，请感兴趣的读者补充完整  
# 结束
```

蜂鸣器演奏乐曲

完成乐谱后，使用循环语句改变PWM的频率设定，同时为了方便调试，在每次循环的开头输出频率值。代码如下：

```
i=0
x1 = Pin('X3',Pin.OUT_PP)
while True:
    print(MyScore[i][0])
    tm3=Timer(2)
    tm3.freq(MyScore[i][0])
    led3=tm3.channel(3, Timer.PWM, pin=x1,pulse_width_percent=50)
    pyb.delay(int(MyScore[i][1]))
    i=(i+1)%15
```


蜂鸣器演奏乐曲

在终端的运行结果如图所示，频率值随着音乐的播放被打印在终端上：

```
admin1@admin1-virtual-machine:/media/admin1/PYBFLASH$ sudo python3 pyboard.py music.py
[sudo] admin1 的密码：
330
392
440
523
587
440
523
392
784
1047
880
784
659
784
587
330
```

注：在opern.py文件中有《最炫民族风》的数字乐谱



THANK YOU

Concise business business report PPT

牛爱科技