

## 蜂鸣器驱动模块

在单片机应用的设计上，很多方案都会用到蜂鸣器，大部分都是使用蜂鸣器来做提示或报警，比如按键按下、开始工作、工作结束或是故障等等。这里对中颖电子的单片机在蜂鸣器驱动上的应用作一下描述。

### 1. 驱动方式

由于自激蜂鸣器是直流电压驱动的，不需要利用交流信号进行驱动，只需对驱动口输出驱动电平并通过三极管放大驱动电流就能使蜂鸣器发出声音，很简单，这里就不对自激蜂鸣器进行说明了。这里只对必须用 1/2duty 的方波信号进行驱动的他激蜂鸣器进行说明。

单片机驱动他激蜂鸣器的方式有两种：一种是 PWM 输出口直接驱动，另一种是利用 I/O 定时翻转电平产生驱动波形对蜂鸣器进行驱动。

PWM 输出口直接驱动是利用 PWM 输出口本身可以输出一定的方波来直接驱动蜂鸣器。在单片机的软件设置中有几个系统寄存器是用来设置 PWM 口的输出的，可以设置占空比、周期等等，通过设置这些寄存器产生符合蜂鸣器要求的频率的波形之后，只要打开 PWM 输出，PWM 输出口就能输出该频率的方波，这个时候利用这个波形就可以驱动蜂鸣器了。比如频率为 2000Hz 的蜂鸣器的驱动，可以知道周期为  $500\ \mu\text{s}$ ，这样只需要把 PWM 的周期设置为  $500\ \mu\text{s}$ ，占空比电平设置为  $250\ \mu\text{s}$ ，就能产生一个频率为 2000Hz 的方波，通过这个方波再利用三极管就可以去驱动这个蜂鸣器了。

而利用 I/O 定时翻转电平来产生驱动波形的方式会比较麻烦一点，必须利用定时器来做定时，通过定时翻转电平产生符合蜂鸣器要求的频率的波形，这个波形就可以用来驱动蜂鸣器了。比如为 2500Hz 的蜂鸣器的驱动，可以知道周期为  $400\ \mu\text{s}$ ，这样只需要驱动蜂鸣器的 I/O 口每  $200\ \mu\text{s}$  翻转一次电平就可以产生一个频率为 2500Hz，占空比为 1/2duty 的方波，再通过三极管放大就可以驱动这个蜂鸣器了。

### 2. 蜂鸣器驱动电路

由于蜂鸣器的工作电流一般比较大，以致于单片机的 I/O 口是无法直接驱动的，所以要利用放大电路来驱动，一般使用三极管来放大电流就可以了。

蜂鸣器的驱动电路有很多种，这里举两个常用的例子，也是建议使用的驱动电路：

#### 1. 1. 无源压电式蜂鸣器、无源电磁式蜂鸣器（他激）

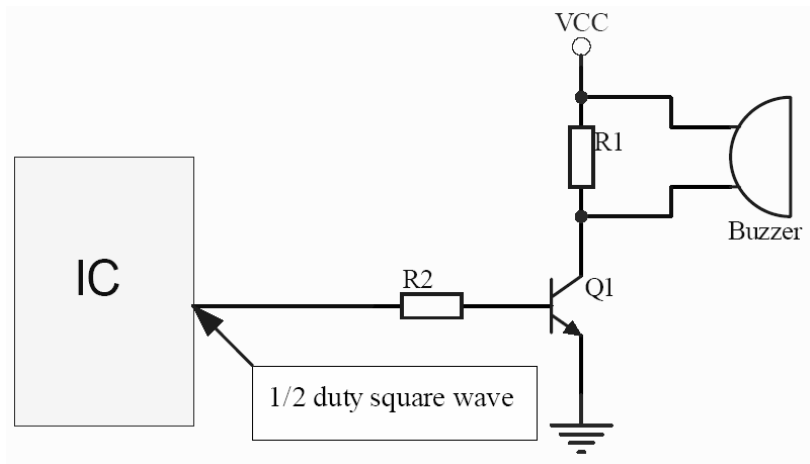


图 1-1 无源压电式蜂鸣器、无源电磁式蜂鸣器驱动电路

### 1.2. 有源压电式蜂鸣器、有源电磁式蜂鸣器（自激）

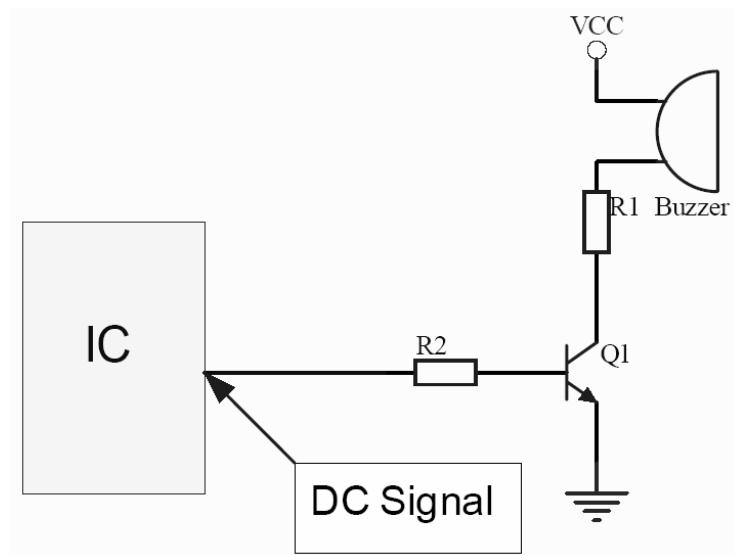


图 1-2 有源压电式蜂鸣器、有源电磁式蜂鸣器驱动电路

## 3. 蜂鸣器驱动设计

由于这里要介绍两种驱动方式的方法，所以在设计模块系统中将两种驱动方式做到一块，即程序里边不仅介绍了 PWM 输出口驱动蜂鸣器的方法，还要介绍 I/O 口驱动蜂鸣器的方法。所以，我们将设计如下的一个系统来说明单片机对蜂鸣器的驱动：系统有两个他激蜂鸣器，频率都为 2000Hz，一个由 I/O 口进行控制，另一个由 PWM 输出口进行控制；系统还有两个按键，一个按键为 PORT 按键，I/O 口控制的蜂鸣器不鸣叫时按一次按键 I/O 口控制的蜂鸣器鸣叫，再按一次停止鸣叫，另一个按键为 PWM 按键，PWM 口控制的蜂鸣器不鸣叫时按一次按键 PWM 输出口控制的蜂鸣器鸣叫，再按一次停止鸣叫。

## ■ 电路原理图

如图 1-3 所示，使用 SH69P43 为控制芯片，使用 4MHz 晶振作为主振荡器。PORTC. 3/T0 作为 I/O 口通过三极管 Q2 来驱动蜂鸣器 LS1，而 PORTC. 2/PWM0 则作为 PWM 输出口通过三极管 Q1 来驱动蜂鸣器 LS2。另外在 PORTA. 3 和 PORTA. 2 分别接了两个按键，一个是 PWM 按键，是用来控制 PWM 输出口驱动蜂鸣器使用的；另一个是 PORT 按键，是用来控制 I/O 口驱动蜂鸣器使用的。连接按键的 I/O 口开内部上拉电阻。

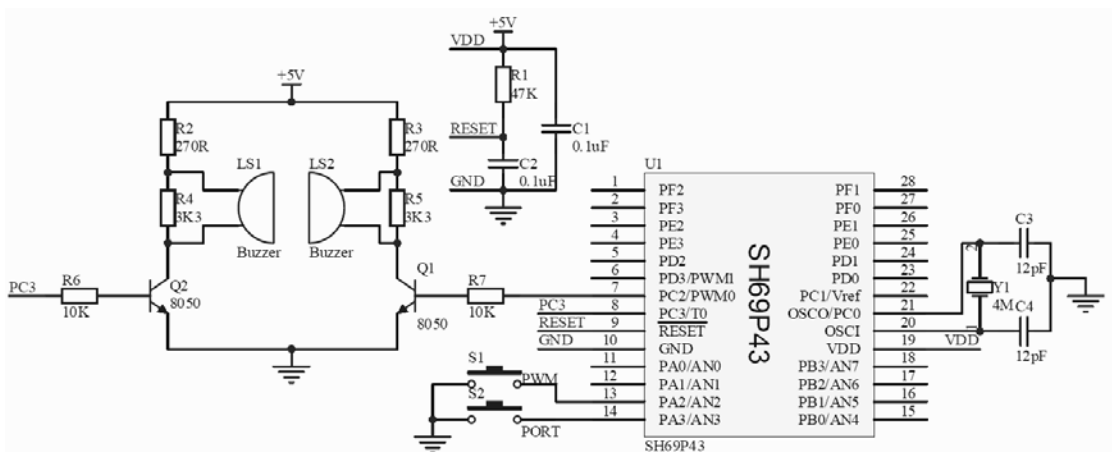


图 1-3 SH69P43 驱动蜂鸣器原理图

## ■ 软件设计方法

先分析一下蜂鸣器。所使用的蜂鸣器的工作频率是 2000Hz，也就是说蜂鸣器的驱动信号波形周期是  $500\mu\text{s}$ ，由于是  $1/2\text{duty}$  的信号，所以一个周期内的高电平和低电平的时间宽度都为  $250\mu\text{s}$ 。

软件设计上，我们将根据两种驱动方式来进行说明。

### a) PWM 输出口直接驱动蜂鸣器方式

由于 PWM 只控制固定频率的蜂鸣器，所以可以在程序的系统初始化时就对 PWM 的输出波形进行设置。

首先根据 SH69P43 的 PWM 输出的周期宽度是 10 位数据来选择 PWM 时钟。系统使用 4MHz 的晶振作为主振荡器，一个  $t_{\text{osc}}$  的时间就是  $0.25\mu\text{s}$ ，若是将 PWM 的时钟设置为  $t_{\text{osc}}$  的话，则蜂鸣器要求的波形周期  $500\mu\text{s}$  的计数值为  $500\mu\text{s}/0.25\mu\text{s} = (2000)_{10} = (7D0)_{16}$ ，7D0H 为 11 位的数据，而 SH69P43 的 PWM 输出周期宽度只是 10 位数据，所以选择 PWM 的时钟为  $t_{\text{osc}}$  是不能实现蜂鸣器所要的驱动波形的。

这里我们将 PWM 的时钟设置为  $4t_{\text{osc}}$ ，这样一个 PWM 的时钟周期就是  $1\mu\text{s}$  了，由此可以算出  $500\mu\text{s}$  对应的计数值为  $500\mu\text{s}/1\mu\text{s} = (500)_{10} = (1F4)_{16}$ ，即分

别在周期寄存器的高 2 位、中 4 位和低 4 位三个寄存器中填入 1、F 和 4，就完成了对输出周期的设置。再来设置占空比寄存器，在 PWM 输出中占空比的实现是通过设定一个周期内电平的宽度来实现的。当输出模式选择为普通模式时，占空比寄存器是用来设置高电平的宽度。 $250\mu s$  的宽度计数值为  $250\mu s/1\mu s=(250)_{10}=(0FA)_{16}$ 。只需要在占空比寄存器的高 2 位、中 4 位和低 4 位中分别填入 0、F 和 A 就可以完成对占空比的设置了，设置占空比为 1/2duty。

以后只需要打开 PWM 输出，PWM 输出口自然就能输出频率为 2000Hz、占空比为 1/2duty 的方波。

#### b) I/O 口定时翻转电平驱动蜂鸣器方式

使用 I/O 口定时翻转电平驱动蜂鸣器方式的设置比较简单，只需要对波形分析一下。由于驱动的信号刚好为周期  $500\mu s$ ，占空比为 1/2duty 的方波，只需要每  $250\mu s$  进行一次电平翻转，就可以得到驱动蜂鸣器的方波信号。

在程序上，可以使用 TIMER0 来定时，将 TIMER0 的预分频设置为 /1，选择 TIMER0 的始终为系统时钟(主振荡器时钟/4)，在 TIMER0 的载入/计数寄存器的高 4 位和低 4 位分别写入 00H 和 06H，就能将 TIMER0 的中断设置为  $250\mu s$ 。

当需要 I/O 口驱动的蜂鸣器鸣叫时，只需要在进入 TIMER0 中断的时候对该 I/O 口的电平进行翻转一次，直到蜂鸣器不需要鸣叫的时候，将 I/O 口的电平设置为低电平即可。不鸣叫时将 I/O 口的输出电平设置为低电平是为了防止漏电。

### ■ 程序

以下是带有两种驱动方式的蜂鸣器驱动模块程序，其中黄色背景的内容是关于 I/O 口定时翻转电平驱动方式的，绿色背景的内容则是关于 PWM 输出口直接驱动方式的：

#### 例[1-1]

LIST P=69P43

ROMSIZE=3072

\*\*\*\*\*

; 系统寄存器

\*\*\*\*\*

IE EQU 00H ;中断使能标志

IRQ EQU 01H ;中断请求标志

TMO EQU 02H ;Timer0 模式寄存器

TLO EQU 04H ;Timer0 装入/记数寄存器低四位

THO EQU 05H ;Timer0 装入/记数寄存器高四位

PORTA EQU 08H ;Port A 数据寄存器

```

PORTC    EQU    0AH        ;Port C 数据寄存器
TBR      EQU    0EH        ;查表寄存器
INX      EQU    0FH        ;间接寻址伪索引寄存器
DPL      EQU    10H       ;INX 数据指针低四位
DPM      EQU    11H       ;INX 数据指针中三位
DPH      EQU    12H       ;INX 数据指针高三位
PACR     EQU    18H       ;PortA 输入/输出控制寄存器
PCCR     EQU    1AH       ;PortC 输入/输出控制寄存器
PWM0     EQU    20H       ;位 0:选择 PWM 输出, 位 2-1:设置 PWM0 时钟, 位 3:设置 PWM0 占空比输出模
式
PP0L     EQU    22H       ;PWM0 周期低四位
PP0M     EQU    23H       ;PWM0 周期中四位
PP0H     EQU    24H       ;PWM0 周期高二位
PD0L     EQU    25H       ;PWM0 占空比低四位
PD0M     EQU    26H       ;PWM0 占空比中四位
PD0H     EQU    27H       ;PWM0 占空比高二位
;*****
; 用户定义寄存器(Bank0)
;*****
AC_BAK   EQU    30H       ;AC 值备份寄存器
PA_BAK   EQU    31H       ;PortA 缓冲寄存器
PC_BAK   EQU    32H       ;PortC 缓冲寄存器
TMP_TO   EQU    33H       ;临时寄存器用于 TIMERO
;-----
F_TIMER  EQU    34H
;
;位 0=1, 5ms 到
FLAG1    EQU    35H
;
;位 0=1, 按键未松开
;位 3=1, 需使用 I/O 口驱动蜂鸣器
;-----
KCODE_NEW EQU    36H       ;旧的按键码
KCODE_OLD EQU    37H       ;新的按键码
KEYS_CT   EQU    38H       ;按键扫描次数
;-----
T5MS_CT1 EQU    39H       ;5ms 计数器低位
T5MS_CT2 EQU    3AH       ;5ms 计数器高位
;*****
; 程序

```

```

;*****
ORG      0000H
JMP      RESET
RTNI
JMP      TIMERO_ISP
RTNI
RTNI

;*****
; 子程序: TIMERO 中断服务程序 (250us 中断)
;*****

TIMERO_ISP:
    STA      AC_BAK, 00H      ;备份 AC 值
    ANDIM IRQ, 1011B        ;清 TIMERO 中断请求标志
;*****
; 模块: I/O 口驱动蜂鸣器模块
;*****
PORT_BUZ:
    ADI      FLAG1, 1000B
    BA3      PORT_BUZ_NO      ;不需要 I/O 口驱动蜂鸣器, 跳转
    EORIM    PC_BAK, 1000B
    STA      PORTC, 00H      ;PORTC. 3 翻转电平
    JMP      PORT_BUZ_END
PORT_BUZ_NO:
    ANDIM    PC_BAK, 0111B
    STA      PORTC, 00H      ;PORTC. 3 口驱动的蜂鸣器未鸣叫时, 输出低电平以保证不漏电
PORT_BUZ_END:
;*****

J5MS:                                     ;判断 5ms 到
    SBIM     T5MS_CT1, 01H
    LDI      TMP_T0, 00H
    SBCM     T5MS_CT2, 00H
    OR       T5MS_CT1, 00H
    BNZ      TIMERO_ISP_END ;5ms 未到, 跳转
    LDI      T5MS_CT1, 04H
    LDI      T5MS_CT2, 01H      ;重新加载 5ms 计数值
    ORIM     F_TIMER, 0001B      ;设置"5ms 到"标志

TIMERO_ISP_END:
    LDI      IE, 0100B          ;开 TIMERO 中断

```

```

        LDA        AC_BAK, 00H        ;恢复 AC 值
        RTNI          ;返回
;*****
; 上电程序
;*****

RESET:
        NOP

;-----
; 清用户寄存器
POWER_RESET:
        LDI        DPL, 00H
        LDI        DPM, 02H
        LDI        DPH, 00H

POWER_RESET_1:
        LDI        INX, 00H
        ADIM       DPL, 01H
        LDI        TBR, 00H
        ADCM       DPM, 00H
        BA3        POWER_RESET_2
        JMP        POWER_RESET_3

POWER_RESET_2:
        ADIM       DPH, 01H

POWER_RESET_3:
        SBI        DPH, 01H
        BNZ        POWER_RESET_1
        SBI        DPM, 04H
        BNZ        POWER_RESET_1

;-----
; 初始化系统寄存器
SYSTEM_INITIAL:
        ;TIMER0 初始化
        LDI        TMO, 07H          ;设置 TIMER0 预分频为/1
        LDI        TLO, 06H
        LDI        TH0, 00H          ;设置中断时间为 250us
        LDI        T5MS_CT1, 04H
        LDI        T5MS_CT2, 01H
        ;I/O 口初始化
        LDI        PORTC, 0011B

```

```

LDI      PCCR, 1100B      ;设置 PORTC. 2 和 PORTC. 3 为输出口
LDI      PORTA, 0FH      ;打开 PORTA 上拉电阻
;PWM 初始化
LDI      PWM0, 04H       ;设置 PWM0 时钟为 4tosc, 占空比输出模式为正常模式
LDI      PPOH, 01H
LDI      PPOM, 0FH
LDI      PPOL, 04H       ;设置周期为 500us
LDI      PDOH, 00H
LDI      PDOM, 0FH
LDI      PDOL, 0AH       ;设置占空比电平为 250us, 得频率为 2KHz, 占空比为
1/2duty 的
;信号
;-----
MAIN_PRE:
LDI      IRQ, 00H
LDI      IE, 0100B      ;打开 Timer0 中断
;*****
MAIN:
ADI      F_TIMER, 0001B
BAO      HALTMODE      ;未到 5ms, 跳转
ANDIM    F_TIMER, 1110B ;清“5ms 到”标志
;*****
; 按键扫描
;*****
KEYSCAN:
LDA      PORTA, 00H
STA      TBR, 00H
SBI      TBR, 0FH
BAZ      NO_KEY        ;没有扫描到按键按下, 跳转

LDA      FLAG1, 00H
BAO      KEYSKAN_END   ;之前的按键未松开, 不再扫描按键, 跳转
SBI      TBR, 0111B
BAZ      KS_PORT       ;PORT 键按下, 跳转
SBI      TBR, 1011B
BAZ      KS_PWM        ;PWM 键按下, 跳转
JMP      KEYSKAN_END

KS_PORT:

```



```

        LDI        KCODE_NEW, 0001B ;设置 PORT 键的键码为 01H
        JMP        KEYSKAN_1

KS_PWM:

        LDI        KCODE_NEW, 0010B ;设置 PWM 键的键码为 02H

KEYSCAN_1:

        ADIM       KEYS_CT, 01H      ;扫描次数加一
        SBI        KEYS_CT, 01H
        BAZ        KEYSKAN_2        ;第一次扫描到该按键，无需与旧按键进行比较，跳转
        LDA        KCODE_OLD, 00H
        SUB        KCODE_NEW, 00H   ;新扫描的按键码与上次扫描的按键码比较
        BAZ        KEYSKAN_2        ;同一个按键，跳转
        LDI        KEYS_CT, 01H     ;不是同一按键，将扫描次数置为 1，为新按键

KEYSCAN_2:

        LDA        KCODE_NEW, 00H
        STA        KCODE_OLD, 00H   ;将新的按键码赋值给旧的按键码
        SBI        KEYS_CT, 0AH
        BNC        KEYSKAN_END      ;扫描未到 10 次（50ms），跳转
        ;已扫描到该按键 10 次
        LDI        KEYS_CT, 00H     ;清按键扫描次数
        ORIM       FLAG1, 0001B     ;设置“按键未松开”标志
        LDA        KCODE_OLD, 00H
        BAO        KEY_PORT

KEY_PWM:                                ;PWM 键
        EORIM      PWM0, 0001B      ;PWM 输出开启或关闭，PWM 口驱动的蜂鸣器鸣叫或停止
        JMP        KEYSKAN_END

KEY_PORT:                                ;PORT 键
        EORIM      FLAG1, 1000B     ;I/O 定时翻转电平开启或停止翻转
        ;I/O 口驱动的蜂鸣器鸣叫或停止
        JMP        KEYSKAN_END

NO_KEY:

        ANDIM      FLAG1, 1110B     ;清“按键未松开”标志
        LDI        KEYS_CT, 00H     ;清按键扫描次数

KEYSCAN_END:

;*****

HALTMODE:

        NOP
        HALT        ;进入 HALT 模式
        NOP

```

NOP

JMP       MAIN

END