

第 2-2 讲：有源蜂鸣器鸣响控制

1. 学习目的

1. 了解有源蜂鸣器和无源蜂鸣器的区别，掌握有源蜂鸣器驱动电路的设计。
2. 掌握编写驱动有源蜂鸣器鸣响与停止的程序。

2. 硬件电路设计

开发板上的蜂鸣器使用的是 3V 有源蜂鸣器，这里，我们有必要了解一下有源蜂鸣器和无源蜂鸣器的区别：

- 有源蜂鸣器：有源蜂鸣器内部带震荡源，所以只要通电就会鸣响。
- 无源蜂鸣器：内部不带震荡源，用直流信号无法令其鸣响。必须用频率信号去驱动他才会鸣响。

所以，对于有源蜂鸣器来说，只要通电就会鸣响。为了实现控制有源蜂鸣器的鸣响，我们的电路中就需要一个“开关”来实现有源蜂鸣器接通或断开电源。在开发板上，是使用 ULN2003 达林顿阵列来实现“开关”的功能的。

ULN2003 是高耐压、大电流达林顿阵列，由七个 NPN 达林顿管组成。所有单元共用发射极，每个单元采用开集电极输出。每一对达林顿都串联一个 2.7K 的基极电阻，直接兼容 TTL 和 5V CMOS 电路，可以直接处理原先需要标准逻辑缓冲器来处理的数据。

ULN2003 工作电压高，工作电流大，灌电流可达 500mA，并且能够在关态时承受 50V 的电压，输出还可以在高负载电流下并行运行，很好的提供了需要多接口驱动电路的解决方案。被广泛应用于显示驱动、继电器驱动、照明灯驱动、电磁阀驱动、伺服电机驱动、步进电机驱动等电路中。

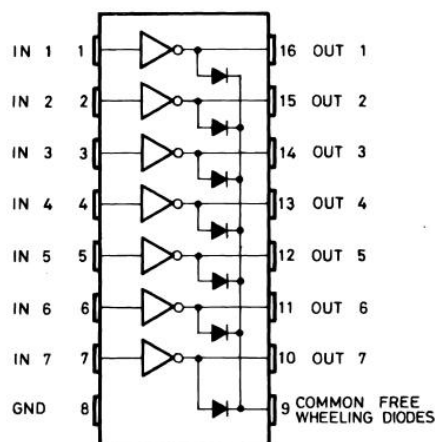


图 1：ULN2003 内部电路框图

本开发板控制有源蜂鸣器即是使用 ULN2003 中的一路实现，不过 ULN2003 输入端信号是 74HC573 锁存器 U11 提供。具体硬件电路如下：

74HC138译码器+74HC02输出真值

A0(P25)	A1(P26)	A2(P27)	Y4C	Y5C	Y6C	Y7C
L	L	H	H	L	L	L
H	L	H	L	H	L	L
L	H	H	L	L	H	L
H	H	H	L	L	L	H

Y5C为高电平时，U11锁存器打开，P0口可控制ULN2003的输出。

开发板实物图

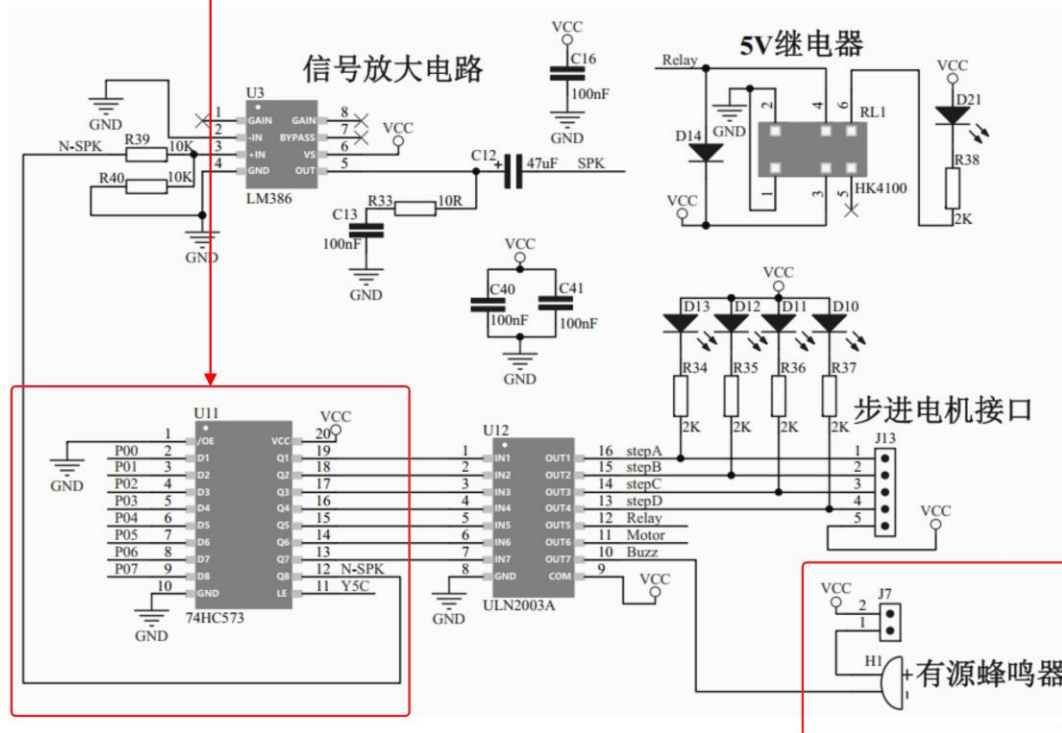
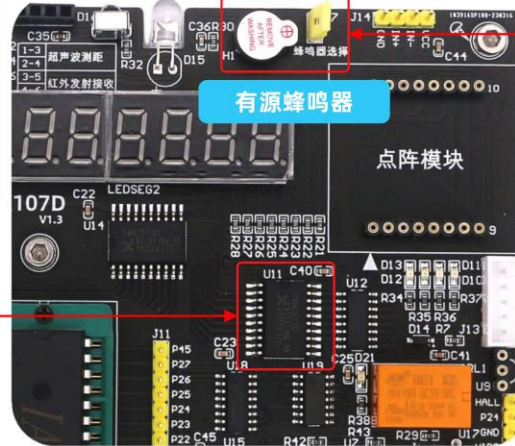


图 2：有源蜂鸣器驱动电路

不难看出，当 U11 锁存器的第 11 引脚为高电平时，可通过单片机 P0.6 控制有源蜂鸣器工作。ULN2003 驱动电路引脚分配如下：

表 1：ULN2003 驱动电路引脚分配

名称	引脚	说明
步进电机	P0.0	非独立 GPIO

	P0.1	非独立 GPIO
	P0.2	非独立 GPIO
	P0.3	非独立 GPIO
继电器	P0.4	非独立 GPIO
电机	P0.5	非独立 GPIO
有源蜂鸣器	P0.6	非独立 GPIO

3. 软件设计

3.1. GPIO 配置

本实验中需将单片机的 P0 引脚配置为准双向口，并配置译码器用到的 P2.5、P2.6、P2.7 为准双向口，原理和 LED 实验一样，参见“第 2-1 讲：流水灯实验”中 GPIO 部分的描述即可。

3.2. 蜂鸣器鸣响实验

✧ 注：本节的实验是在“实验 2-1-1：流水灯”的基础上修改，本节对应的实验源码是：“实验 2-2-1：有源蜂鸣器鸣响控制”。

3.2.1. 实验内容

1. 配置用到的 GPIO 为准双向口。
2. 主循环中驱动蜂鸣器间隔性鸣响：每 200ms 改变一次 P0.6 的输出电平，即蜂鸣器以 200ms 的间隔鸣响。

3.2.2. 代码编写

1. 新建一个名称为“beep.c”的文件及其头文件“beep.h”并保存到工程的“Source”文件夹，并将“beep.c”加入到 Keil 工程中的“SOURCE”组。
2. 本例中编写了三个操作蜂鸣器的函数：ULN74hc573_on、beep_on 函数和 beep_off 函数，分别用于打开控制 ULN2003 的锁存器、鸣响蜂鸣器和关闭蜂鸣器鸣响，他们的代码清单如下。

代码清单：打开控制 ULN2003 的锁存器函数

```

1. /*****
2. 功能描述：打开控制 ULN2003 的锁存器（U11:74HC573）
3. 参 数：无
4. 返 回 值：无
5. *****/
6. void ULN74hc573_on(void)
7. {
8.     P25=1;          //控制 P2.5 端口输出高电平
9.     P26=0;          //控制 P2.6 端口输出低电平
10.    P27=1;          //控制 P2.7 端口输出高电平
11.}

```

代码清单：beep_on 函数

```
1. /*****
2. 功能描述：鸣响蜂鸣器
3. 参    数：无
4. 返 回 值：无
5. *****/
6. void beep_on(void)
7. {
8.     ULN74hc573_on();    //打开控制 ULN2003 的锁存器，就可以用 P0.6 口控制蜂鸣器了
9.     BEEP_P06 = 1;       //蜂鸣器鸣响
10. }
```

代码清单：beep_off 函数

```
1. /*****
2. 功能描述：蜂鸣器停止鸣响
3. 参    数：无
4. 返 回 值：无
5. *****/
6. void beep_off(void)
7. {
8.     ULN74hc573_on();    //打开控制 ULN2003 的锁存器，就可以用 P0.6 口控制蜂鸣器了
9.     BEEP_P06 = 0;       //蜂鸣器停止鸣响
10. }
```

3. 主程序“main.c”中先配置用到的 GPIO，之后在主循环中以 200ms 间隔轮流鸣响和关闭蜂鸣器鸣响，代码清单如下：

代码清单：主函数

```
1. /*****
2. 功能描述：主函数
3. 入口参数：无
4. 返回值：int 类型
5. *****/
6. int main(void)
7. {
8.     P2M1 &= 0x1F;    P2M0 &= 0x1F;    //设置 P2.5、P2.6、P2.7 为准双向口
9.     P0M1 &= 0x00;    P0M0 &= 0x00;    //设置 P0.0 ~ P0.7 为准双向口
10.
11.     SEG_off();       //控制 8 位数码管/点阵不显示
12.     leds_off();      //熄灭 D1~D8 指示灯
13.     ULN2003_off();   //控制步进电机、蜂鸣器、继电器等不工作
14.     delay_ms(10);    //延时
15. }
```

```

16. while(1)
17. {
18.     beep_on();    //鸣响蜂鸣器
19.     delay_ms(200); //延时 200ms
20.     beep_off();   //蜂鸣器停止鸣响
21.     delay_ms(200); //延时 200ms
22. }
23.}

```

3.2.3. 硬件连接

本实验程序的编写都是基于 IO 模式，所以 J24 端子需要使用短路帽将该端子第 1 引脚和第 2 引脚短接，即选择为 IO 模式。有源蜂鸣器的供电引脚没有直接和 VCC 相连，需要使用短路帽将 J7 端子短接才可实现有源蜂鸣器供电引脚与 VCC 相连。

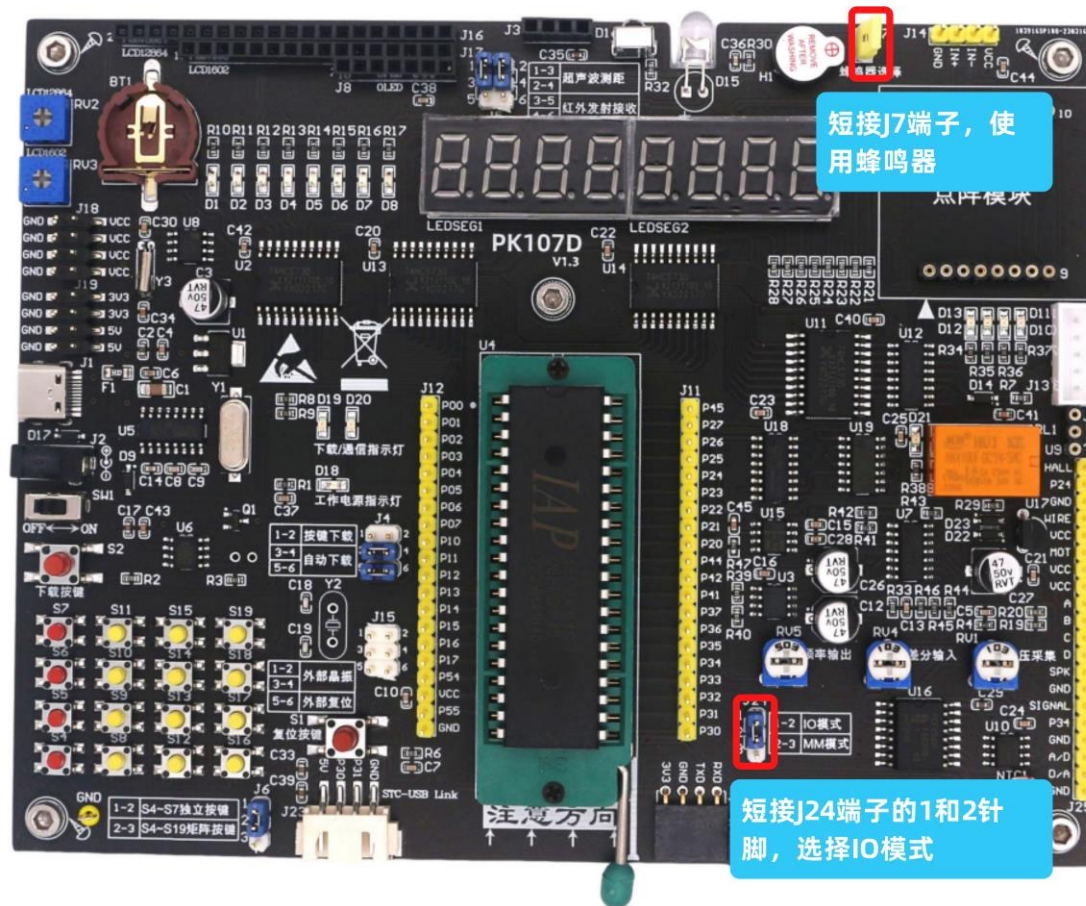


图 3：硬件连接图

3.2.4. 实验步骤

1. 解压“···\第 3 部分：配套例程源码”目录下的压缩文件“实验 2-2-1：有源蜂鸣器鸣响控制”，将解压后得到的文件夹拷贝到合适的目录，如“D:\STC15”（这样做的目的是为了防止中文路径或者工程存放的路径过深导致打开工程出现问题）。
2. 双击“···\beep\project”目录下的工程文件“beep.uvproj”。

3. 点击编译按钮编译工程，编译成功后生成的 HEX 文件“beep.hex”位于工程的“...\beep\project\Objects”目录下。
4. 打开 STC-ISP 软件下载程序，下载使用内部 IRC 时钟，IRC 频率选择：12MHz。
5. 程序运行后，可以观察到蜂鸣器鸣响 200ms，之后关闭鸣响 200ms，如此往复。