

实验简介：

本次实验，我们借助可编程并行接口芯片 8255，通过 PC 机编程在实验仪平台上实现了一个简易的电子琴。

实验原理：

- 1) 对蜂鸣器输入不同频率的方波，会发出各个音阶的声音；
- 2) 通过编程设定或按键，由 8255A 芯片控制发出不同频率的方波，即产生不同音阶的声音。

实验过程：

通过 8255 的 PAD, 使 F5 区的 1~7 号键由低到高发出 1-7 的音阶。

实验原理图：

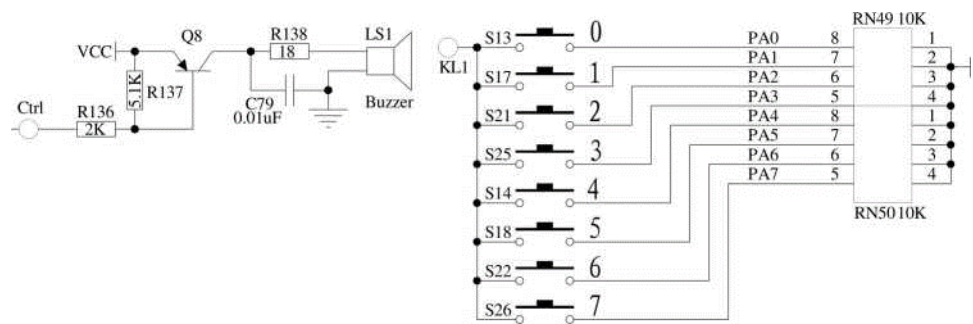


图 1 简易电子琴实验原理图

实验原理图如图 1 所示。

实验步骤：

主机连线说明如表 1 所示。

表 1 简易电子琴连线说明图

D3 区：CS、A0、A1	A3 区：CS1、A0、A1
D3 区：PC7	F8 区：Ctrl
D3 区：JP23 (PA 口)	F5 区：JP37 (A)
C1 区：GND	F5 区：KL1

连线的目的是编程或按键控制 F5 区的按键使 G1 区的蜂鸣器发出不同音阶的声音。

本实验的演示分为两个部分，首先是对 8255 芯片编程使之发布不同频率的方波，控制蜂鸣器发出不同的音阶，演奏一首完整的简易乐曲；其次是通过按控制板上的键盘，手动演奏乐曲。

实验说明：

本次实验所用代码放在附录中以供查阅，接下来对代码做一些必要的说明。

程序首先定义了最终可执行文件的格式，由于本次实验的代码行数较少、访存空间较小，因此采用了 tiny 格式。接着设置控制口地址、输入口地址与输出口地址，需要注意的是，不同类型的实验仪的口地址不一定相同，在编程时应予以注意。程序所用的栈空间为 100 (dw)，数据段定义了由各函数组成的《两只老虎》的乐谱。程序的功能部分主要由以下七个部分组成：

- 1) 调用初始化 8255 芯片以及播放默认乐曲的函数。显然，为了 8255 芯片可以正常工作，我们需要根据实验的实际情况，编程决定 8255 芯片的方式选择控制字与工作方式。本部分主要由两个函数调用组成：

```
call init8255           ;8255 初始化
```

```
call demo               ;播放一段音乐
```

- 2) 按键查询有关的一系列操作。我们的设计是先由 CPU 发送一段默认的乐谱给 8255 芯片，使之控制蜂鸣器演奏乐曲，再允许操作者做其他演奏。本部分主要由 start1~start7 等 7 个分支组成，其中 start1 完成有无按键按下的查询并检测所按键是否为 1，其余 6 个语句块完成的功

能基本相同：查询对应的按键是否按下。

- 3) 自动播放乐曲的主函数。为了实现自动播放的功能，我们实现了一个函数 demo，主要定义了歌曲的节拍数并顺序发出各个音阶。值得一提的是，我们使用寄存器 CX 来保存节拍数，为了避免在函数调用时其值被覆盖，因此需要提前压入堆栈中保护。
- 4) 响应按键的一系列函数。这个部分主要由 music1~music7 等 7 个函数组成，它们的功能基本相似，区别仅在于为了能够区别不同的音阶而设置的延时，即蜂鸣器在一定时间内震荡的频率。为了保证音阶可以方便地被识别，我们应当取各震荡周期基本一致。
- 5) 响应自动播放的一系列函数。本部分代码结构与上一部分在形式上具有很大的相似性，主要由 m1~m7 等 7 个不同函数组成，所不同的是此部分代码的功能是响应自动播放。不得不说的是，各个函数为了使其对应的音阶更明显，通过寄存器 CX 设置 LOOP 循环的次数，即延时时间。另外，上部分与本部分的延时如何调整才能使音阶达到更好的效果需要一定的音乐知识，并加以调试修改。
- 6) 初始化 8255 芯片与读写控制有关的函数。这个部分的代码比较简单，由 w_l、w_h 与 init8255 等三个函数构成，其中 w_l 定义向 PC 口的第 7 位写 0 时则蜂鸣器响，w_h 的功能相反，而 init8255 定义了 8255 芯片的方式选择控制字与具体采用的芯片口，PC 口第 7 位用于输出，PA 口用于输入。
- 7) 为了满足拍子的要求而构建的一些列延时函数。最后这部分的代码具体实现了各个延时函数，本部分的关键在于函数复用，即使用已定义的函数完成新的功能。

至此，本次实验所用代码的总体架构介绍完毕。

心得体会：

在本次实验中……（后面正式写时再补充）