

项目 A 电子琴学习机

1 任务要求	2
1.1 1. 基本功能	2
1.1.1 自由模式	2
1.1.2 自动演奏模式	2
1.1.3 学习模式	2
1.1.4 模块化设计	3
1.2 优化功能	3
1.2.1 自由模式:	3
1.2.2 自动演奏模式:	3
1.2.3 学习模式:	4
1.3 Bonus	4
2 一些帮助	5
2.1 EGO1 用户手册 (必要的引脚描述)	5
2.2 硬件	5
2.3 代码	6
2.4 你的代码模块可能看起来像这样	10
3 附件说明	10

您的团队预计要实现一个小型电子琴，以帮助演奏音乐，该电路支持模式选择和演奏功能，可选 VGA 作为输出设备。

注意：

1. 由于需要演示，每个音符的持续时间可以相对于实际应用而延长。
2. 以下电子琴学习方法仅供参考，如果您有更好的实现方法（合理且用户友好），可以相应调整实现。
3. 演奏节奏由您的团队设计，音乐可以粗略识别即可（我们对音乐是否优雅的判定并不苛刻）。
4. 建议在灵活性、流畅性、可视性和趣味性方面优化您的学习机。

1 任务要求

当电子琴学习机启动并正常工作时，有三种可用模式：自由模式、自动演奏模式、学习模式。

1.1 1. 基本功能

1.1.1 自由模式

随意按键，电子琴播放相应的音符，例如，随意按下几个按钮，学习机播放 do、re、mi 等音符。

1.1.2 自动演奏模式

学习机自动演奏歌曲，例如，进入自动演奏模式，自动演奏”小星星”。

1.1.3 学习模式

根据音符顺序和持续时间点亮，以引导用户正确演奏，例如，根据”小星星”的顺序，应该按灯下方对应的键，用户按下后灯会熄灭，然后打开另

一盏灯。

1.1.4 模块化设计

合理划分模块来处理用户输入、音乐存储、控制蜂鸣器、LED、七段数码管等输出。

1.2 优化功能

1.2.1 自由模式：

在自由按键的基础上，电子琴播放相应的音符。

- 增加八度调整音符：

更高的八度：按下特定键，可以弹出比原始八度更高的音符，同样适用于低音八度。

1.2.2 自动演奏模式：

学习机在增加音乐库的基础上自动演奏歌曲。

- 在七数码管上显示曲目编号：

进入自动模式后，按下按钮实现音乐库的前后翻页，选择三首或更多歌曲，七段数码管显示音乐库的内容（至少可以播放两首歌曲）。

- 在自动演奏时，使用灯光指示用户演奏位置和持续时间：

按下按钮确认曲目后，学习机开始自动演奏，当音符出现时从音符上方点亮灯光，音符演奏后熄灭灯光，直到歌曲结束。

1.2.3 学习模式：

- 在引导的基础上提高整个演奏的评级：对于每个音符，确定演奏间隔，根据用户正确演奏的次数，在七段数码管或 LED 灯上显示用户的评级
- 用户可以根据他们的使用习惯调整键的位置：当学习设备进入调整模式时，设备按顺序演奏音符，如 do re mi。选择与'do' 相对应的音符后，按下确认按钮，机器将演奏下一个音符，等待用户再次选择并确认。完成所有确认后，用户练习期间按下的键对应的音符将与调整模式一致。
- 对于用户的演奏记录，可以选择是否将其保存在音乐库中：进入录音模式，记录用户按下的键的顺序和位置，当用户按下结束键时停止录音，然后自动回放。用户按下确认键后，将其保存在设备中，以后可以通过自动播放模式重新播放它（提供删除和覆盖操作选项）。

1.3 Bonus

用户操作更加方便，显示更加直观，性能更加流畅。

包括但不限于以下内容：

- VGA 显示更多关于音乐的信息以辅助指导演奏和增加趣味性：例如，使用一些图形提示音符播放的时机，显示歌曲名称而不是编号。
- 关于评分的实时更新更加精细和合理：在学习模式中，根据用户按键的时间与标准时间的差异提供实时变化的评分。评分方法将作为评估的基础。

- 音乐的节奏更多变化：在自动模式和学习模式中，每个音符的持续时间可以不一致。

2 一些帮助

2.1 EGO1 用户手册（必要的引脚描述）

EGO1 用户手册

Little Star

小星星-中级版

1=C $\frac{4}{4}$ ♩=120 img895

(1)

1	1	5	5		6	6	5	-		4	4	3	3		2	2	1	-	
1	1	1	1		4	4	1	1		5	5	1	1		5	5	1	1	

(5)

5	5	4	4		3	3	2	-		5	5	4	4		3	3	2	-	
1	1	5	5		1	1	5	5		1	1	5	5		1	1	5	5	

(9)

1	1	5	5		6	6	5	-		4	4	3	3		2	2	1	-	
1	1	1	1		4	4	1	1		5	5	1	1		5	5	1	1	

图 1: 小星星

2.2 硬件

课程提供的蜂鸣器是一种被动蜂鸣器。被动蜂鸣器可以是压电蜂鸣器或磁性蜂鸣器。它们的工作原理相对简单。它们的音调由它们固有的振动频率

决定。为了使蜂鸣器正确发声，需要根据开发板的时钟和 C 大调'do' 的频率来调整输出频率。

也许您会有些担忧自己没有音律方面的天赋，但不用担心！我们已经为您准备好了根据 EGO1 时钟频率调好的代码，就在之后的代码区块，只要输入对应的数字，比如输入 1，它将输出 do 的发声频率。

在使用时，要注意蜂鸣器线的连接。通过测试发现，长时间通电但没有适当振动会导致导线烧断。如果确认代码正确但蜂鸣器没有发声，可能是导线已经烧断，尽管看不出来，或接口连接错误。

正确的接口应将一端连接到正极，将另一端连接到输出特定频率的引脚。有关具体引脚，请参考用户手册。

如果您真的不幸烧断了您的蜂鸣器导线，也不要惊慌，它还可以继续工作。解决方法：撕开导线上的橡胶包装，使导线的金属部分接触到原来连接的位置。（来自先驱探索者）

如果您实在是疑惑并且确定不是您的代码，引脚或者一些其它问题导致的发声问题，您可以尝试以下方法确定它是否还能工作。

验证蜂鸣器是否正常工作的方法：左侧 IO 列从上数第一个和第二个 IO 口（第二个是正极 3V），连通时会产生电流，这个时候如果反复插拔，就会强制产生间断电流。蜂鸣器会发出“吡”的声音。（来自先驱实验者）

蜂鸣器的声音可能没有您想象的大，需要注意它一闪而过的微弱发声。

2.3 代码

Listing 1: Buzzer

```
1 module Buzzer(  
2     input wire clk,           // Clock signal
```

```
3     input wire[3:0] note, // Note (Input 1 outputs a
        signal for 'do,' 2 for 're,' 3 for 'mi,' 4, and
        so on)
4     output wire speaker // Buzzer output signal
5 );
6
7 wire[31:0] notes[7:0];
8 reg [31:0] counter;
9 reg pwm;
10 // Frequencies of do, re, mi, fa, so, la, si
11 // Obtain the ratio of how long the buzzer should be
    active in one second
12 assign notes[1]=381680;
13 assign notes[2]=340136;
14 assign notes[3]=303030;
15 assign notes[4]=285714;
16 assign notes[5]=255102;
17 assign notes[6]=227273;
18 assign notes[7]=202429;
19 initial
20 begin
21     pwm=0;
22 end
23 always @(posedge clk) begin
```

```
24     if (counter < notes[note] || note == 1'b0) begin
25         counter <= counter + 1'b1;
26     end else begin
27         pwm = ~pwm;
28         counter <= 0;
29     end
30 end
31
32 assign speaker = pwm; // Output a PWM signal to the
    buzzer
```

您也许还好奇这样的频率是如何计算出来的，do 的频率是 261.6Hz，时钟的频率是 100MHz， $100\text{M}/261.6=381680$ 。那么，接下来任务中的调整八度就请您自行探索啦。

以及，使用二维数组时，请注意：`wire[31:0] notes[7:0]` 中，`[31:0]` 是位宽，`[7:0]` 表示数组大小为 8（从 0-7），两者意义不同。（来自先驱探索者）

Listing 2: for 的使用 (just a hint, 也许会帮助您减少一些工作量)

```
1     integer i;
2     for ( i = 0; i < 8; i = i + 1)
3     begin
4         if (key[i] == inputs )
5         begin
6             note <= i;
7             light <= key[i];
8         end
```


9 | `end`

2.4 你的代码模块可能看起来像这样

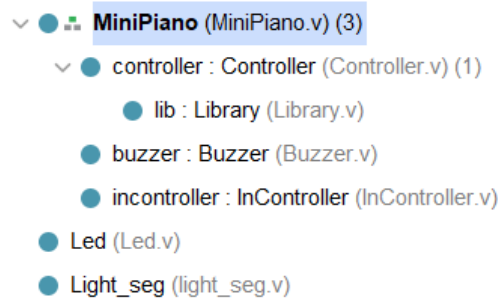


图 2: a module example

3 附件说明

我们准备了一个测试用的 Bit 流文件并且录制了使用时的视频，最右边的三个娇小白色的开关（R3 T3 T5）可以控制它的状态:011 自由模式，010 自动播放模式，101 调整按键模式。

希望可以对您有所帮助，祝项目顺利