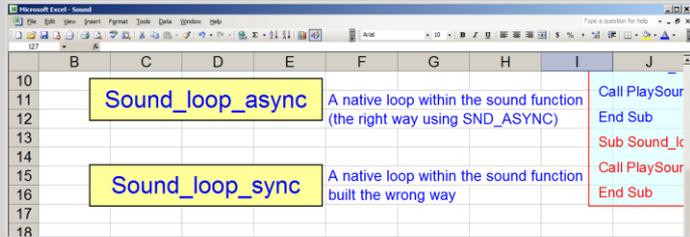


Playing Sounds in Excel Using VBA #2

- This tutorial will continue the presentation started in the first section by taking the knowledge of playing sounds in Excel one step further.
- First an example will be shown in which the sound is played in a loop which means a native internal loop within the "PlaySound" function (using the flag option: `SND_LOOP = &H8`).
- A second function (the "Beep" function) will be introduced in the form of a simple tone generator but with adjustable frequency.
- The Beep function will also be use in a fourth macro to create a sound which goes through a series of frequency steps from 100Hz to 6000Hz

The looping feature of the PlaySound function:

- Insert a second worksheet and rename it "Sound_Macros_Part#2"
- Insert a new module (Module2)
- In the new module write the following code
- Add a couple of buttons to the worksheet and assign the two macros to them



```
Private Declare Function PlaySound Lib "winmm.dll" (ByVal lpszName As String, _  
ByVal hModule As Long, ByVal dwFlags As Long) As Long
```

```
Const SND_SYNC = &H0
```

```
Const SND_ASYNC = &H1
```

```
Const SND_LOOP = &H8
```

```
Sub Sound_loop_async()
```

```
Call PlaySound(ThisWorkbook.Path & "\two.wav", 0&, SND_ASYNC Or SND_LOOP)
```

```
End Sub
```

```
Sub Sound_loop_sync()
```

```
Call PlaySound(ThisWorkbook.Path & "\three.wav", 0&, SND_SYNC Or SND_LOOP)
```

```
End Sub
```

- Testing these two macros you see that the first one works right but the second one, while producing the sound once, does not work in a loop. **You can stop the first macro by clicking on another sound macro button.**
- The software instructions are clear here (see the first tutorial) and we can expect the second loop to fail the way it is written except I was not sure in which way it would fail.
- As a rule of thumb when using these functions, the experimentation is a very important phase in the process of figuring out how they work. Many details of operation are not written anywhere and you need to do an array of experiments by yourself.
- If you go on various forums you will get good answers from MVP's which will save you a lot of time but if the problem is not common what you usually get is silence.

The "Beep" macro:

- Insert a new module (Module3) and in the new module write the code to the right:
- This new sound function generates a tone with adjustable frequency and duration. The function is called as the word "Beep" followed by two numbers. The first number is the tone frequency in Hertz and the second number is the duration in milliseconds.
- Add a couple of buttons to the worksheet and assign the two macros to them
- Create a spin button which can change the value of cell B21 from 100 to 10000 in steps of 100. Cell B21 contains the frequency info. for the macro Beep_1
- The way the first macro is set up it generates a tone with programmable frequency (pitch), and the sounds lasts 1000 milliseconds (1s).
- The second macro generates a series of 30 frequency steps from 200Hz to 6KHz in 200Hz increments
- The problem with this macro is that the frequency steps are separated by short but annoying sound lapses and the duration of the steps is not that constant as it is supposed to be.
- I need to do more work with these macros in order to improve their usefulness.

```
Private Declare Function Beep Lib "kernel32" _  
(ByVal dwFreq As Long, ByVal dwDuration As Long) As Long
```

```
Sub Beep_1()  
    DoEvents  
    Beep [B21], 1000  
End Sub  
Sub Beep_steps()  
    For i = 1 To 30  
        DoEvents  
        [B27] = 200 * i  
        Beep 200 * i, 400  
        DoEvents  
    Next i  
End Sub
```

