# Speaker, Keypad, and Timer Control

## PURPOSE

In this four-week lab, you will develop a system that allows you to play simple songs and a system that can read from a keypad. They are based on the 6812 I/O ports and timer on an EVB.

**PRELAB (25%):** Complete the prelab assignments as teams. Remove any syntax error before you turn in prelab codes. Code segments that are used many times should be implemented as functions. Arrays should be used so that it will be easy to modify the code to handle a different keypad.

(1) (Week 1: 5%) In order to produce the C tone in Experiment (1), specify the bus clock frequency, calculate the constants needed for output compare relevant registers, and modify tone.c. Note that the bus clock of the EVB runs at 24 MHz with the function MCU_init( ) in tone.c and runs at 8 MHz when the function is commented out.

(2) (Week 2: 10%) Draw a complete circuit diagram and then write the code for Experiment (2).

(3) (Week 4: 10%) Identify the output compare parameters that can produce all the tones in the table at the bottom of this page. Write a code for Experiment (3).

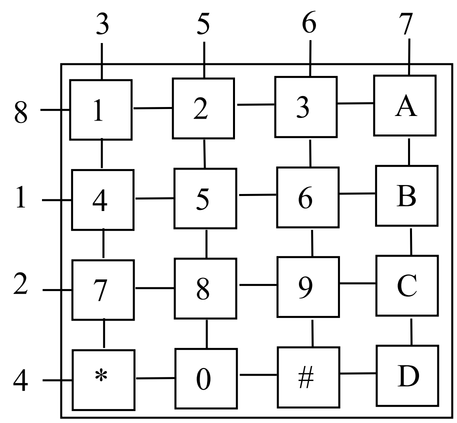
## EXPERIMENT (75%)

(1) (20%) Connect a speaker through a low-pass RC circuit (e.g., R=3K Ohm, C=100pF) to the 6812 EVB PT1 pin. Modify the tone.c program so that you can hear a music C tone (261.6 Hz) continuously from the speaker. (Pick a resistor value to avoid too loud a sound.) √√ If you have extra time, try to produce different music tones as they are needed in Experiment (3).
**Background:** The note frequencies follow a logarithm scale, as shown in the following table. From C to high C, an octave, the frequency is doubled. Each octave is divided into 12 half tones. Using the note C as a starting point, the number of half tones to a note is called the pitch. So the frequency of a note = 261.2 x $2^{(pitch/12)}$ Hz.

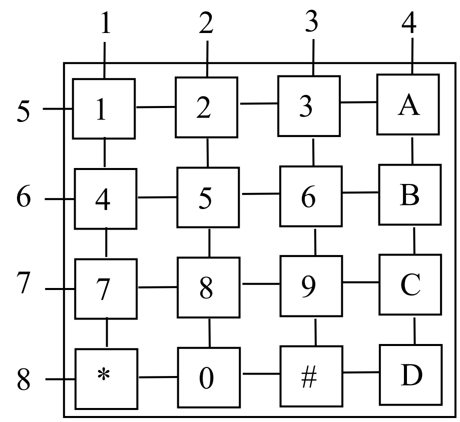
| Note | C | D | E | F | G | A | B | High C |
|---|---|---|---|---|---|---|---|---|
| Pitch | 0 | 2 | 4 | 5 | 7 | 9 | 11 | 12 |
| Freq. (Hz) | 261.6 | 293.6 | 329.6 | 349.2 | 392 | 440 | 493.8 | 523.2 |

(2) (30%) Connect a keypad (refer to the figures below) to the EVB and write a C program so that when any of the 9 non-zero numeric keys is pressed down the corresponding character ('1' to '9') is stored in a variable whose value can be verified using the debug window. You need to use PTAD (pins 2, 3, 4) and PTT (pins 2, 3, 4) as shown in the figures. If necessary, use software to handle debouncing. (You may use an oscilloscope, with single triggering, to check out if there is bouncing.) Assume no multiple keys get pressed at the same time. Note

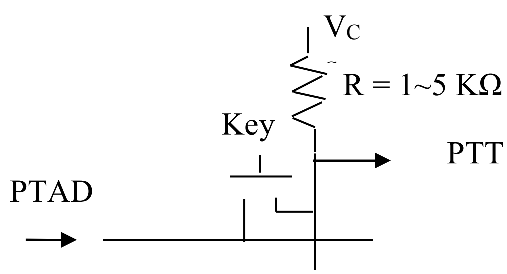
that, after a keypad row is selected, the code should wait for one or two μsec before reading from the keypad.√√




This keypad has <u>white letters on black keys</u>. Viewed from above, Pin 1 is the left-most pin on the keypad circuit board.




This keypad has <u>black letters on white keys</u>. Viewed from above, Pin 1 is the left-most pin on the keypad circuit board.




(3) (25%) Combine the previous two programs and modify them so that a user can use the keypad to play simple songs. Key ‘1’ should be mapped to music note C, ‘2’ to D, ‘3’ to E, and so on. Key ‘9’ is reserved for usage in Lab 4. A music note should last as long as the duration the corresponding key is pressed down. √√

Keep all of your source codes. They are needed later on for system integration.