Lab-5: Timers for Musical Note Generation

In this set of experiments, we learn how to use built-in hardware timers (refer Fig.1) in 8051 to generate delays without tying up the processor. We then make a musical note generator using these timers.

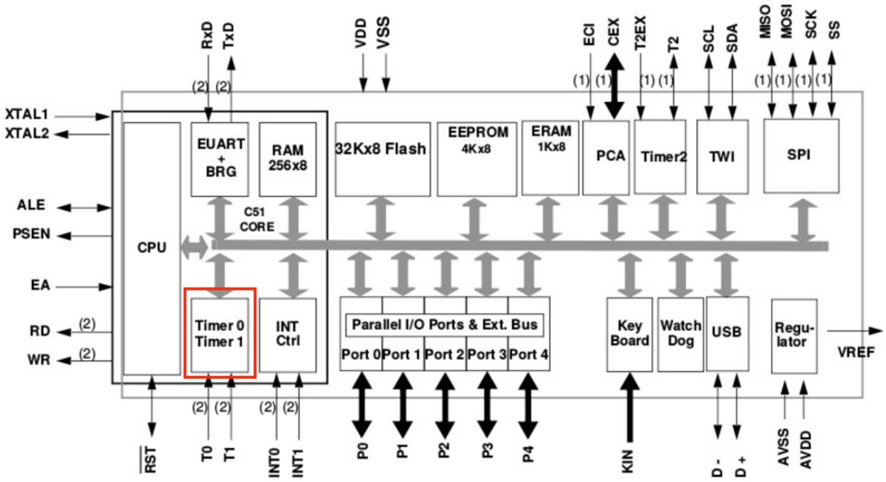

Figure 1: Internal block diagram of AT89C5131 (Ref AT89C5131 datasheet) with Timer section highlighted

Before attempting these exercises, go through the notes on timers and interrupts uploaded in moodle.

Note: The document, "Notes on Timers" describes timers of 8051 with crystal frequency 12 MHz. However, the Pt-51 board has a 24 MHz crystal. So do the calculation accordingly.

Homework

1. Write a subroutine that will use a 16-bit value, stored at 81H/82H in the indirectly addressable memory, as the count value to program the timer T0 in order to generate a proportional delay.

   Recall that the 8051 timers count *up*. These generate an interrupt (if enabled to do so) when the count wraps around from FFFFH to 0000H. If we want a timer to time-out after n cycles, it should be loaded with -n (2's complement of n).

So the subroutine should subtract the stored 16 bit number from 0000H and load the result as the initial count in T0.

(While debugging the program with single stepping, you could initialize locations 81H/82H with 0001, which results in loading the timer with -1 = 0FFFFH, so that it overflows at the first increment itself. In actual use, a different count will be stored, of course.)

2. Write a program that will use the above subroutine to blink onboard LEDs (as in Lab-1) such that these are ON for one second and OFF for one second endlessly. Adjust the timer count and the number of times the delay subroutine is called to make the ON and OFF period as close to 1 second as possible. (Measure the time over a large number of cycles to make this adjustment).

3. Now we shall use the timers to generate musical notes. Write a program to generate a square wave of 240Hz on Port pin P0.3 using Timer T0. Use 240Hz as the fundamental frequency for the various notes in the main octave.

4. Table 1 lists the various notes used in an octave, in Indian classical music. Sa, Re, Ga, Ma, Pa, Dha, Ni, are Shuddha Swars. Out of these seven swars Re, Ga, Dha, and Ni become "Komal" when their frequency is lower than their Shuddha form.

   For generating these notes, it is required to generate various frequencies using the ratios given in Table 1. Complete the table with appropriate frequencies and timer values, to be used during the lab session.

5. Octave refers to one set of 7 major notes Sa, Re, Ga, Ma, Pa, Dha, Ni. The next octave begins from Upper Sa. In lower octave, all the notes will have a frequency half the corresponding frequency in the next octave. i.e., here you have generated octave of 240Hz to 480Hz with all the intermediate frequencies as written in the table. For lower octave, halve the corresponding frequencies and they will vary from 120 Hz to 240Hz.

Lab Work

1. Generate the following notes: "Sa, Re, Ga, Ma, Pa, Dha, Ni, Sa (Upper Octave), (pause), Sa(Upper Octave), Ni, Dha, Pa, Ma, Ga, re, Sa" on Port pin P0.3, each note should play for 1 second and the pause be also for 1 second. You may use Timer 1 to for controlling the duration.

2. Simulate your code in Keil and verify the signal frequency using the logic analyzer.

3. In order to play these notes, we need to interface the port pin to a speaker. The impedance of a speaker is quite low (less than 10 Ω). A port output cannot provide sufficient current to drive this impedance. So, we connect the port pin which provides

Table 1: Base notes and frequencies in Indian classical music.

S. No.	Note	Ratio (x)	Frequency (240*x)	Timer Value (TH0 TL0)
1	Sa	1		
2	Komal Re	16/15		xsx
3	Re	9/8		
4	Komal Ga	6/5		
5	Ga	5/4		
6	Ma	4/3		
7	Tivra Ma	64/45		
8	Pa	3/2		
9	Komal Dha	8/5		
10	Dha	5/3		
11	Komal Ni	16/9		
12	Ni	15/8		
13	Sa (Upper Octave)	2/1		

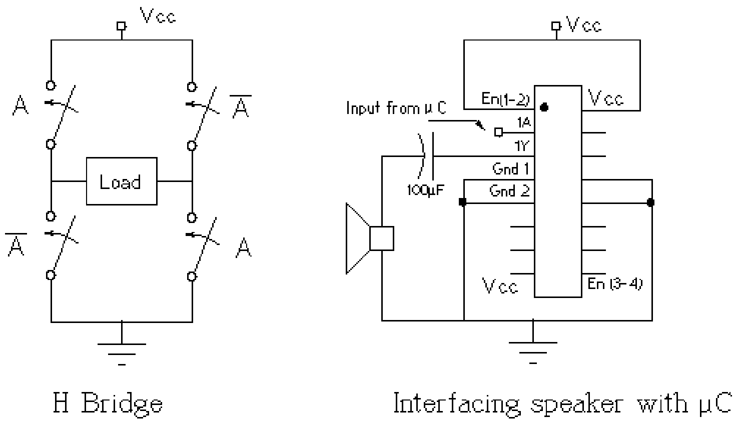

Figure 2: Loudspeaker driver circuit.

the square wave to what is called an H-bridge. This is just a collection of buffers with enable inputs which can connect the output to supply or ground depending on the current input. The equivalent circuit of the H-bridge is shown Figure 2 on the left below. It is called a H-bridge because of the shape of the circuit. The data sheet of the H-bridge has been uploaded on moodle. However, for the speaker driver, we need not use the entire H bridge. We only need to use the simpler circuit shown in Fig. 2 on the right.

4. Connect the circuit shown in Fig. 2 on breadboard and play tones.

5. Let's play a short music sequence now.

Table 2: Notes and durations for a "Sare Jahan se achchha..."

	S. No.	Note		Duration (Seconds)
21.12	1	Komal Ga	f26f	0.66
21.12	2	Re	F187	0.66
10.56	3	Sa	EFB8	0.33
21.12	4	Re	F187	0.66
10.56	5	Ni(in lower Octave)	eea2	0.33
32	6	Sa	EFB8	1
10.56	7	Pause		0.33
10.56	8	Pa (in lower Octave)	ea4b	0.33
10.56	9	Dha (in lower Octave)	ec77	0.33
21.12	10	Sa	EFB8	0.66
10.56	11	Re	F187 F187	0.33
21.12	12	Ga	F2FA	0.66
10.56	13	Ma	F3C9	0.33
32	14	Ga	F2FA	1

	BDA
1	EFB8
2	F0BA
3	F187
4	F26F
5	F2FA
6	F3C9
7	F48D
8	F525
9	F5D3
10	F63A
11	F6D6
12	F751
13	F7DC

- We have generated seven major notes in Indian scale from Sa, Re, Ga, Ma, Pa, Dha, Ni, and Sa (Next octave) in Part 1. To generate a song, along with the notes to be played, the duration of each note should also be accurate.
- Lets try playing "Sare Jahan Se accha, hindostan hamara"!
  We need to generate one more intermediate note. 'Komal Ga' whose frequency ratio is 6/5.
- Table 2 provides the sequence of notes and their duration for the notes to be played for this song.