EE 337: Microprocessors Laboratory (Spring 2022)
Indian Institute of Technology, Bombay

Assignment 6: 20 points — Date: March 01, 2022

Before attempting this lab, do the following:

- Go through the video lectures uploaded in the **Files** → **Class Materials** → **Timer Lectures by Prof. DKS** folder on MS Teams (optional).
  - These video lectures are from the Spring 2021 EE 309 Microprocessors course taught by Prof. Dinesh K Sharma
  - There are three video lectures named **Recording-02-02.mp4**, **Recording-02-04.mp4**, and **Recording-02-08.mp4**.
  - The last video needs to be watched only till the 6 minute mark.
  - The slides used in the videos are uploaded in the same folder. See `8051-Timers.pdf`.
- Also, study the notes on interrupts and timers shared in the course website. `https://ee337.github.io/dks.html`

1. [10 points] You will use the built-in hardware timers in 8051 to generate delays. This is in contrast to the software delay subroutines you had used in the previous labs.

   - Write a subroutine that will use a 16-bit number as the count value to program the timer `T0` 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 $-N$ (i.e., 2's complement of N).

     So the subroutine should subtract the 16-bit number from `0000H` and load the result as the initial count in `T0`. While debugging the program with single stepping, you could consider a count value of `1`, and load the timer register with `-1`, so that it overflows at the first increment itself. In actual use, a different count has to be stored.
   - Write a program that will use the above subroutine to blink **ALL** the onboard LEDs such that these are ON for T seconds and OFF for T seconds continuously. Adjust the timer count and the number of times the delay subroutine is called to make the ON and OFF period as close to T seconds as possible. The value of T will be one of the values in $\{1, 2, 3, 4, 5\}$. The value needs to be read from memory location `30h`.

2. [10 points] You will configure the timers to generate a periodic 4-level pulse amplitude modulated (PAM) signal that takes integer values in the range `0-15`.

   - Level 2 and Level 1 amplitudes should be stored as the most significant nibble and the least significant nibble, respectively, of the data in location `70H`.
   - Level 4 and Level 3 amplitudes should be stored as the most significant nibble and the least significant nibble, respectively, of the data in location `71H`.
   - The period of the PAM signal should be 4 seconds with each level occupying 1 second.
   - For example, if `70H` contains `23H` and `71H` contains `45H`, then the PAM signal should look like the following.

Amplitude	Duration
3	$t = 0$ to $t = 1$
2	$t = 1$ to $t = 2$
5	$t = 2$ to $t = 3$
4	$t = 3$ to $t = 4$
3	$t = 4$ to $t = 5$
2	$t = 5$ to $t = 6$
5	$t = 6$ to $t = 7$
4	$t = 7$ to $t = 8$

- The PAM levels have to be written out on the port pins `P1.7-P1.4`. These are also to be written on the LCD in the following format.
  - First row of the LCD should display text correponding to the current level as "Level 1", "Level 2", "Level 3" or "Level 4".
  - Second row should display the corresponding value as "Value: xxxx", where 'xxxx' is the binary representation of the level amplitude.

TA Checkpoints

1. For question 1, ask the student to show the LEDs blinking and check that they use the timer.
2. For question 1, ask the student to change the value in memory location `30H` and run the program again. Check that the PAM levels have changed accordingly.
3. For question 2, check that the 8-bit values in the memory locations `70H` and `71H` match the levels and binary values displayed on the LCD.