ANKARA YILDIRIM BEYAZIT UNIVERSITY

EE226 – INTRODUCTION TO MICROPROCESSORS LABORATORY

LABORATORY MANUAL –3–

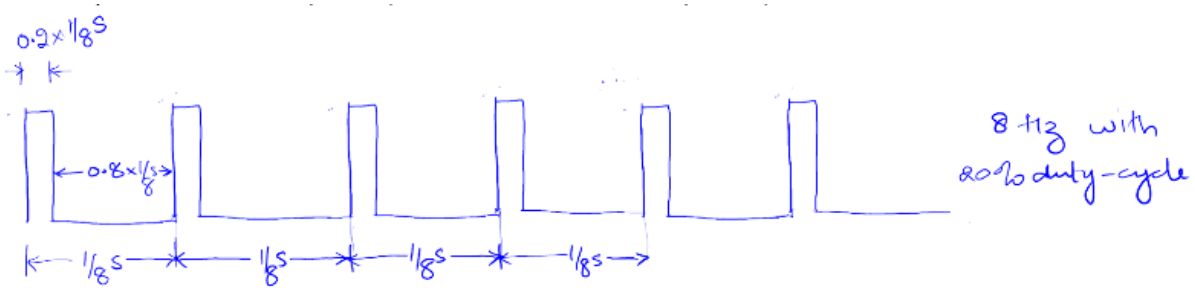
Breathing LED and Interfacing

Background Work:

- Compose a new project.
- Select your device, TM4C123GH6PM.
- Add a new C file to the project, named "Lab3Main".
- Repeat the same remaining steps given in Background Work of Lab 2 manual.
- Note: You may utilize your Lab 2 codes for Lab 3 work. However, you will add a new project. Do not use the old project.
- Bring proper cables, LEDs, resistors for protection of LEDs, a breadboard and on-off switches (switches are not compulsory, but better for your experience) to Lab 3 for your demonstrations.

Laboratory Work:

1. Write a C code such that an RGB LED blinks at 8 Hz with duty cycle of 20%. Duty cycle of 20% means that your led will be on 20% of time and will be off 80%. It will complete the cycle within 1/8 seconds:



2. Write a C code so that the RGB LED on the board breaths. You can use any color for breathing LED. You can watch the video to understand what is "breathing" for LEDs:

https://www.youtube.com/watch?v=ZT6siXyIjvQ

- A breathing LED increases in brightness gradually and once it reaches its full brightness it decreases it brightness gradually till it reaches zero brightness. At which point it again repeats the increase.

- A frequency of 8Hz is low enough to be visible to the naked eye. We need the toggle the LED at a higher frequency (say 80Hz) to be able to see the desired effect of duty-cycle impacting brightness.

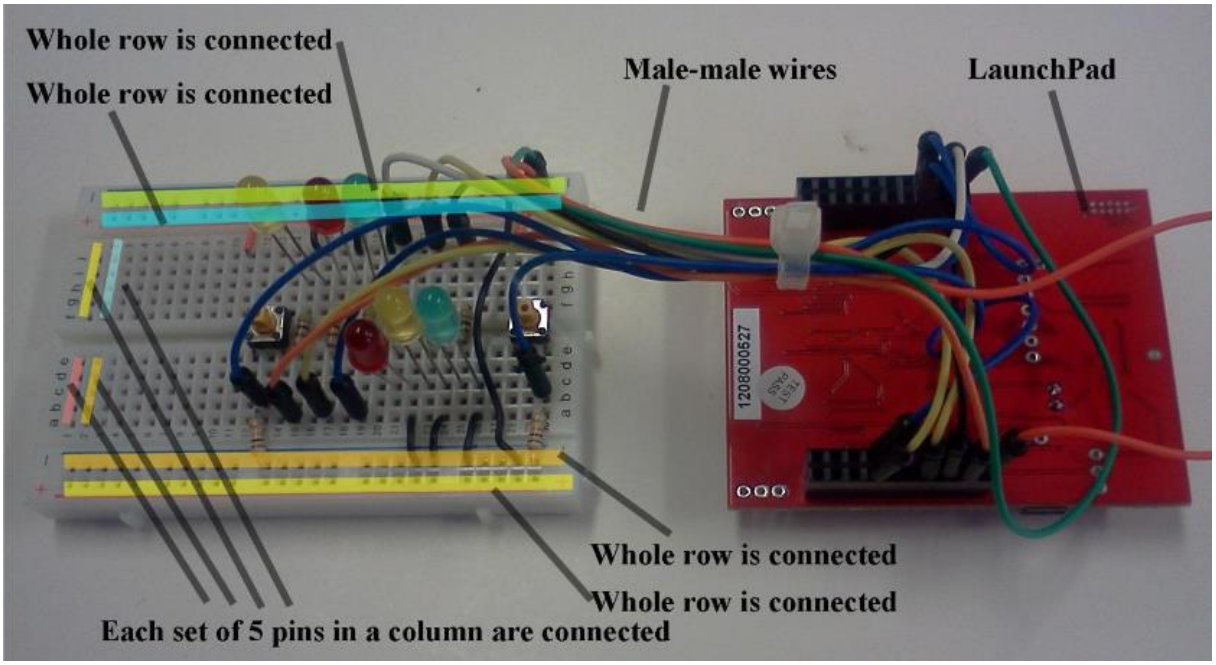
- Varying brightness is achieved by varying duty-cycle. You may need more than 5 levels of duty-cycle for better breathing feel.
- Consider changing the duty-cycle at a programmable rate. That is, if your current duty-cycle is x%, stay at this duty-cycle for N iterations before changing to (x ± d)%.

- Remember, you can play with both the frequency and the duty-cycle.

Hint: 5 different duty cycles: first, LED is 20% on and 80% off in time. In second 40% on and 60% off; and so on. A whole breathing cycle starts from 100% off, reaching to 100% on and returning to 100% off again. Use the delay function you built in Lab 1. However, here you may use it in millisecond values. Try to implement a good cycle, like the video above. The hint is only for 5 different steps; more steps will provide you better breathing.

3. Make improvements on the code so that the buttons SW1 controls the two steps above. When SW1 is on, 8Hz LED blinking will be on. When SW1 is off, LED breathing will be on.

4. Now, do the same job above with an outer LED connected to one of PORT-E pins. You will use an interface circuit here. An exemplar circuit is given in the picture below:



As different from the picture above, you will use only one LED on your breadboards. In this part, you may use SW1 or an outer switch just like in the picture.

Hint: *Do not forget to define the GPIO ports and unlock the switches.*
Hint: *Color codes for RGB LEDs: RED 0x02, YELLOW 0x0A, GREEN 0x08.*
Note: Be careful about anode and cathode sides of LEDs.