



Hacettepe University
Computer Engineering Department
BBM434: Embedded Systems Laboratory
2nd Lab Assignment Report
(2nd Section)

Şerafettin Berk SEVGİ
21328414

Cemal ÜNAL
21328538

03.03.2017

OBJECTIVE

The main objective of this assignment was to learn general purpose input-output concepts in the launchpad, for example, obtaining different colors on leds or using switches to change the lightning times and lightning times of the leds.

THEORETICAL BACKGROUND FOR THE LAB

In order to implement the project, it's need to get know input&output interfaces of development kit. General Purposed Input&Output interface (GPIO) on Tiva C series development kit has been programmed on Keil platform for getting output signal of PF1, PF2 and PF3 ports (blue, red and green LEDs).

The implementation to ensure the LED in launch pad which lights with blue, red and green color order will light in reverse order by pressing PF4 input (SW1 switch) for a short time has been designed by defining a counter which will counts from time of pressing until releasing. It's important to get know about **debouncing** concept regarding SW1 switch for optimization of pressing time. Debouncing is a oscillation from the moment a key is pressed in the development kit.

Each component in the development kit corresponds to a numerical value. The SW1 switch programmed in the Keil environment is handled according to this numeric value of the variable `GPIO_PORTF_DATA_R`. Accordingly, if the `GPIO_PORTF_DATA_R` variable is set to 0x00, the SW1 switch is pressed. If the value is 0x10, it is not pressed.

PROCEDURE AND RESULTS

In this lab assignment only Launchpad was used. There was no other external material. The first purpose was to lighting the LEDs in turn with blue, red, green and again blue, red, green and so on. This was the first problem. And the second problem was, when the button was pressed at any time, it was necessary to reverse the lighting sequence of the LEDs. (e.g. green, red, blue)

In the code file there is an array of hexadecimal color codes to keep green, red and blue. And there is also a delay function to create time delay between the LED lightnings. This function takes a parameter to create different time delays between each different colored lights. For blue light it takes 1 as parameter, for red light it takes 2 as parameter, for green light it takes 3 as parameter. To make this difference easy function parameter is generated according to the index of the color array. This method solves the first problem.

To solve the second problem, a variable named flag was used in the code file. It is checking whether the button is pressed or not. Initially the default value of this variable is **false**. The color order is blue-red-green as long as this variable is **false**. When the button is pressed once, the flag variable becomes **true**. And the color order reversed to green-red-blue. The color order would be in reverse order as long as flag variable is **true**. This method solves the second problem.

ANSWERS

- 1) Detecting the pressing of a button via polling would not be the best practice. Because preception can be change according to how long you hold it. For example, it may not percieve pressing a button if the button is not holded enough.
- 2) Perform delaying via looping may not be the best solution. Because, the variable inside the loop must be changed until get the required time of delay. To avoid this problem, pre-defined functions in Keil can be used to perform delaying.
- 3) The time interval from starting the loop until final of an iteration was measured as 3.000151 second in simulation. In development kit, that time interval was measured as 2.58 second via timing in chronometer.

Video can be viewed on this link:

<https://youtu.be/wRHjQComldk>