Experiment 2: General Purpose Input Output

(03-05).10.2022
Res. Asst. Batuhan Cengiz
cengiz16@itu.edu.tr

Veni. Vidi. Vici.

-Julius Caesar

1 Introduction

This lab aims to help students to gain more experience in the MSP430 Education Board, MSP430G2553 microcontroller and its assembly language. Students are recommended both to read the supplementary material **”Supplementary Chapter 6 General Purpose IO”** on Ninova. Also, if preferred, they could bring their own computers to the laboratory on which Texas Instruments Code Composer Studio IDE is installed.

2 Part 1

The general purpose input and output (GPIO) using the ports of MSP430G2553 (i.e. Port 1 and 2) can be performed by configuring and reading/setting the corresponding registers of the selected port. The following two instructions read P1.2 and conditionally branch depending on the state of the button.

```
    ;read the switch at P1.2 and set flags
    bit.b    #00000100b,&P1IN
    jnz      ON
```

The following two instructions clear and set LED 5 respectively.

```
    ;read the switch at P1.2 and set flags
    bic.b    #00010000b,&P1OUT ;clear P1.4
    bis.b    #00010000b,&P1OUT ;set P1.4
```

Remember; bic and bis instructions are bit clear and bit set instructions respectively. bit instruction is bit test instruction. Note that bic and bis use masks and they are different than mov instruction. In MSP430 bit test compares two operands and sets status bits.
Note that, this document is written to describe the objectives of the experiment.

You are required to know(learn) how to use the MSP430. Examples in the first part are not sufficient enough to successfully complete the experiment. They can only give you some insights. You should read the necessary supplementary material before coming to the laboratory and in the laboratory.

Overall, in the first part of the experiment, you are required to write a simple code that:

- Waits for the user to press P2.4.
- Turns LED P1.4 on.
- Waits on an infinite loop.

```
    ; Infinite loop example
    loop    jmp loop
```

3 Part 2

Write an assembly program that switches back and forth between the LED P2.2 & LED P2.3 using BUTTON P1.5. If the LED P2.2 is on when the BUTTON P1.5 is pressed, it turns off LED P2.2 & turns on LED P2.3. If the LED P2.3 is on when the BUTTON P1.5 is pressed, it turns off LED P2.3 & turns on LED P2.2. Also, if you hold the BUTTON, the state of the LEDs should not keep changing (Hint: What is switch bouncing?). Lastly, you should set the initial state of LEDs as P2.2 at on & P2.3 at off state.

4 Part 3

Write an assembly program that counts how many times the push button P2.1 is pressed. In this manner, you should first ”define a variable in memory rather than using an accumulator”. to store how many times the button is pressed. You are asked to use a 4-bit variable which resets at #16d. Your program should display the value of the variable on Port 1. Note that the value should not continue to increase if you keep the push button pressed. (You should check how to declare a variable can be declared and initialized in the data section. Please check MSP430 Introduction document.)