

# Training Project Lab (8051 Assembly Development)

Names: Anjan Kumar Das (D42DQA)

Instructor Name: Kovacs Viktor

## TASK 1.1:

```
$NOMOD51
```

```
$include (SI_EFM8BB3_Defs.inc)
$include (CPanel.inc)
```

```
CSEG  AT 0    //Select Code segment, from address 0
LJMP  Main    //Processor start executing instructions from address 0 after reset
```

```
MYPROG      SEGMENT CODE
             RSEG MYPROG
```

```
Main:      MOV  WDTCN, #0DEh    //Disable Watchdog
             MOV  WDTCN, #0ADh

             CALL CPanel_Init    //Initialize ports for the extension board

             CLR  N_LED_EN
             CLR  SWITCH_EN
```

### LOOP:

```
MOV C, BTN1
MOV N_LD1, C
```

```
JMP LOOP
```

```
END
```

### Description:

To solve this problem to turn on the LED when we press the button 1 and turning on the LED 1. We firstly cleared the LED and the switch . After that we have introduced a LOOP and since we can not move the BTN1 to led directly, we already moved the BTN1 to carry and then move the carry to LED. and we keep running the loop over and over again for every time the button has been pressed.

## TASK 1.2:

```
$NOMOD51

#include (SI_EFM8BB3_Defs.inc)
#include (CPanel.inc)

CSEG  AT 0           //Select Code segment, from address 0
LJMP  Main           //Processor start executing instructions from address 0 after
reset

MYPROG      SEGMENT CODE
RSEG MYPROG

Main: MOV  WDTCN, #0DEh    //Disable Watchdog
      MOV  WDTCN, #0ADh

      CALL CPanel_Init    //Initialize ports for the extension board

      CLR  N_LED_EN
      CLR  SWITCH_EN
      CLR  C

      //JMP $

      //SETB PSW_F0

LOOP: MOV  PSW_F0,C
      MOV  C,BTN1
      JC  LOOP

      JNB  PSW_F0,LOOP
      CPL  N_LD1
      JMP  LOOP

END
```

### Description:

To solve this problem to turn on the LED when we press the button 1 and turning on the LED 1 and keep it on and will be turned off after the we press button again. To solve this problem we have to detect the falling edge from the button and then we will save the previous state in another temp memory and we will compare the present state with the temporary memory and we will turn on the LED when the previous state of the button is 1 and the present state is 0. For this at first we will save the value of the carry to the temp memory and then we save the BTN value to C. So the previous state is PSW and present state is C. We will run the loop again if the C is 1 and if not we will continue further and jump loop again if the PSW is not set directly. If PSW is 1 and the C is 0 we will turn on LED. And when the same thing happens the LED will be turned off.

### TASK 1.3:

\$NOMOD51

\$include (SI\_EFM8BB3\_Defs.inc)  
\$include (CPanel.inc)

CSEG AT 0 //Select Code segment, from address 0  
LJMP Main //Processor start executing instructions from address 0 after reset

MYPROG SEGMENT CODE  
RSEG MYPROG

**Main:** MOV WDTCN, #0DEh //Disable Watchdog  
MOV WDTCN, #0ADh  
  
CALL CPanel\_Init //Initialize ports for the extension board  
  
CLR N\_LED\_EN  
CLR SWITCH\_EN  
CLR C

//SETB PSW\_F0

#### LOOP:

MOV PSW\_F0,C  
MOV C,BTN1  
  
JNB BTN2, RESET  
JC LOOP  
  
JNB PSW\_F0,LOOP  
  
MOV A, SWITCHPORT  
RR A  
RRC A  
ANL A, #0Eh  
ADDC A, #0  
MOV LEDPORT, A  
  
//CPL N\_LD1  
JMP LOOP

**RESET:** MOV LEDPORT, #0FFh  
JMP LOOP

END

**Description:**

In this problem we will try to turn on the light of the LED according to the status of the switches when the Button 1 is pressed. After that, when the button 2 is pressed, all the led will be turned off. To implement this problem, we will shift the all 4 switch ports together. To do that we will use the ROTATE operation to the SWITCHPORT to put all the switch together. After the switch ports are together we will execute the turn on led according to the switch status. And we used the BUTTON 2 for turning all LEDs off. To do that we set a jump in the loop when button 2 is pressed. And made a new label named RESET for this, where will transfer the 0xFF value to the LED PORTS , because they are low active . So, the LEDs will be turned off.