

Lab 6

Timers



Spring 2025

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

A handwritten signature in black ink that reads "Mohsin Sajjad".

Student Signature: _____

Submitted to:

Engr. Faheem Jan

Month Day, Year (23 03, 2025)

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

Timers

- Timer use to generate a delay
- Event counter to count events happening outside the microcontroller

TASKS:

Task 01:

Create a delay of 2.5 sec .. the LED should ON after 2.5 sec and OFF for 2.5 seconds.

CODE:

```
*main.c  cmpf_i.h
1#include <msp430.h>
2int main(void)
3{
4    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
5    // Configure GPIO
6    P1DIR |= BIT0; // Set P1.0 as output
7    P1OUT |= BIT0; // Turn on LED
8    PM5CTL0 &= ~LOCKLPM5; // Disable high-impedance mode
9    // Configure Timer_A
10   TA0CTL0 |= CCIE; // Enable Timer A0 interrupt
11   TA0CCR0 = 50000; // Set compare value
12   TA0CTL |= TASSEL__SMCLK | MC__UP; // Use SMCLK, Up mode
13   __bis_SR_register(LPM0_bits | GIE); // Enter LPM0, enable global interrupts
14   __no_operation(); // For debugging
15}
16volatile unsigned int counter = 0;
17// Timer A0 interrupt service routine
18#pragma vector = TIMER0_A0_VECTOR
19__interrupt void Timer_A(void)
20{
21    counter++;
22    if (counter >= 50) // 50 * 50ms = 2.5 seconds
23    {
24        P1OUT ^= BIT0; // Toggle LED
25        counter = 0; // Reset counter
26    }
27    TA0CCR0 += 50000; // Add offset for next interrupt
28}
```

OUTPUT:



conclusion:

This code uses Timer_A on the MSP430 to toggle an LED (P1.0) every 2.5 seconds using an interrupt-based approach. The timer generates an interrupt every 50ms, and after 50 counts, it toggles the LED state. This ensures efficient power usage by running in low-power mode (LPM0) while maintaining precise timing.

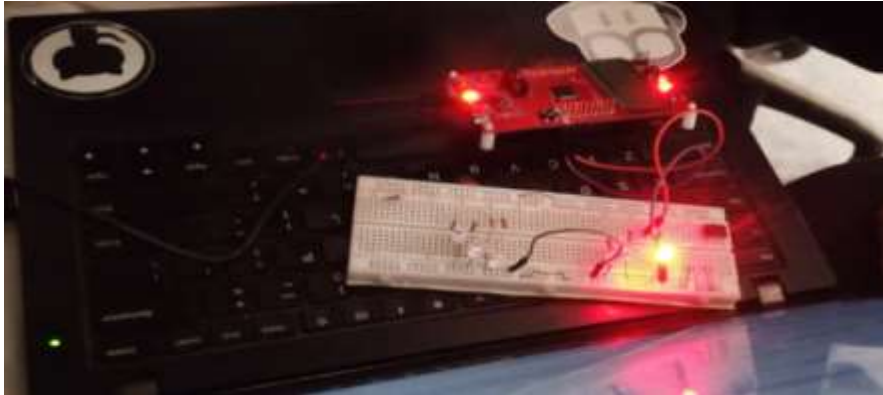
Task 2:

Create a delay of 1 sec .. the LED should ON after 1 sec and OFF for 1 sec ...

CODE:

```
#main.c  cmpf_ih
1 #include <msp430.h>
2 int main(void)
3 {
4     WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
5     // Configure GPIO
6     P1DIR |= BIT0; // Set P1.0 as output
7     P1OUT |= BIT0; // Turn on LED
8
9     PM5CTL0 &= ~LOCKLPM5; // Disable high-impedance mode
10
11    // Configure Timer_A
12    TA0CTL0 |= CCIE; // Enable Timer A0 interrupt
13    TA0CCR0 = 50000; // Set compare value
14    TA0CTL |= TASSEL_SMCLK | MC_UP; // Use SMCLK, Up mode
15
16    __bis_SR_register(LPM0_bits | GIE); // Enter LPM0, enable global interrupts
17    __no_operation(); // For debugging
18
19    volatile unsigned int counter = 0;
20    // Timer A0 interrupt service routine
21    #pragma vector = TIMER0_A0_VECTOR
22    __interrupt void Timer_A(void)
23    {
24        counter++;
25        if (counter >= 20) // 20 * 50ms = 1 second
26        {
27            P1OUT ^= BIT0; // Toggle LED
28            counter = 0; // Reset counter
29        }
30        TA0CCR0 += 50000; // Add offset for next interrupt
31    }
```

OUTPUT:



CONCLUSION:

This MSP430 program uses Timer_A to toggle an LED (P1.0) every 1 second using an interrupt-based approach. The timer triggers an interrupt every 50ms, and after 20 counts, it toggles the LED. The low-power mode (LPM0) ensures efficient power consumption while maintaining accurate timing.

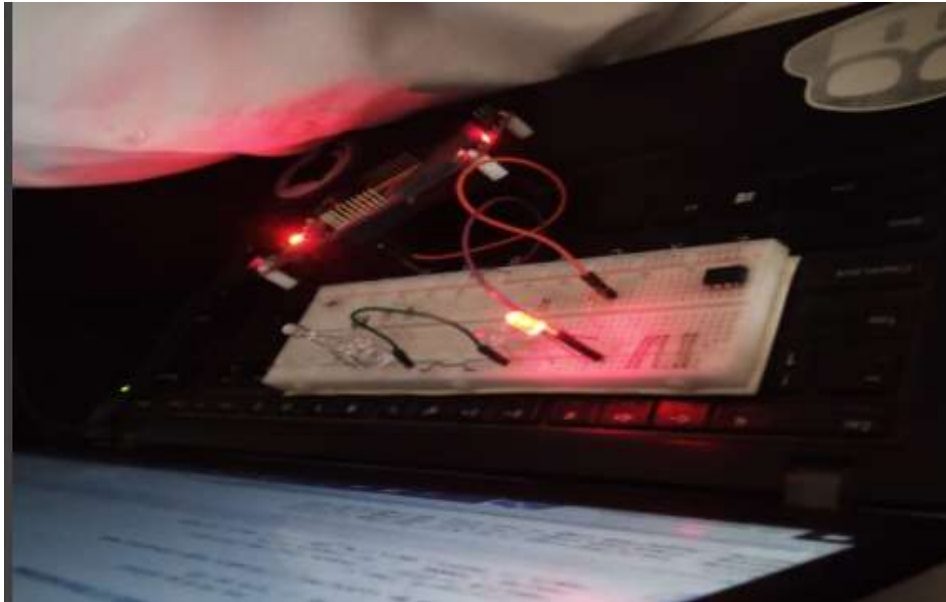
TASK 03:

Create a delay of 500 msec .. the LED should ON after 500 msec and OFF for 500 msec ...

CODE:

```
*main.c | cmpf.h
1#include <mcp430.h>
2int main(void)
3{
4    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
5    // Configure GPIO
6    P1DIR |= BIT0; // Set P1.0 as output
7    P1OUT |= BIT0; // Turn on LED
8    PM5CTL0 &= ~LOCKLPM5; // Disable high-impedance mode
9    // Configure Timer_A
10   TA0CTL0 |= CCIE; // Enable Timer A0 interrupt
11   TA0CCR0 = 50000; // Set compare value
12   TA0CTL |= TASSEL_SMCLK | MC_UP; // Use SMCLK, Up mode
13   __bis_SR_register(LPM0_bits | GIE); // Enter LPM0, enable global interrupts
14   __no_operation(); // For debugging
15}
16volatile unsigned int counter = 0;
17// Timer A0 interrupt service routine
18#pragma vector = TIMER0_A0_VECTOR
19__interrupt void Timer_A(void)
20{
21    counter++;
22    if (counter >= 10) // 10 * 50ms = 500 milliseconds
23    {
24        P1OUT ^= BIT0; // Toggle LED
25        counter = 0; // Reset counter
26    }
27    TA0CCR0 += 50000; // Add offset for next interrupt
28}
```

OUTPUT:



CONCLUSION:

This MSP430 program uses Timer_A to toggle an LED (P1.0) every 500ms with an interrupt-based approach. The timer fires an interrupt every 50ms, and after 10 counts, the LED toggles. The low-power mode (LPM0) optimizes power consumption while maintaining accurate timing.