Timer Exercises on MSP430FR2433

Embedded Systems Lab Report

September 22, 2025

Chapter 1

Introduction

This report discusses the configuration and use of **Timer_A** on the MSP430FR2433 microcontroller using the example code provided. The aim is to understand timer counting, interrupts, and how the Capture/Compare Register (CCR0) affects the blinking frequency of an LED connected to `P1.0`.

Chapter 2

Reference Code

Below is the reference code discussed in class:

```c
/* DESCRIPTION: COUNT USING TIMERS
 * Timer0_A is configured with an SM clk in UP MODE.
 * Timer overflows when TAR counts to CCR0.
 * TA0CCR0 register is loaded with 50000 counts.
 * ISR toggles LED on P1.0 at each interrupt.
 * SMCLK = 1 MHz, ACLK = 32.678 kHz.
 */

#include <msp430.h>

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;   // stop watchdog timer
    PM5CTL0 &= ~LOCKLPM5;       // Disable GPIO high-impedance mode

    // Configure GPIO
    P1DIR |= BIT0;
    P1OUT |= BIT0;              // p1.0 is high initially

    // Configure Timer: SMCLK, UP mode, clear TAR
    TA0CTL = TASSEL__SMCLK | MC__UP | TACLR;

    // Enable CCR0 interrupt
    TA0CCTL0 |= CCIE;

    // Load CCR0 with required value
    TA0CCR0 = 50000;

    __bis_SR_register(LPM0_bits | GIE);   // Enter LPM0 with interrupts
    __no_operation();                     // For debug
}

// ISR
#pragma vector = TIMER0_A0_VECTOR
```

```
__interrupt void Timer_A (void)
{
    P1OUT ^= BIT0;
}
```

Chapter 3

Analysis and Exercises

1. Current Blinking Frequency

- Timer clock: $SMCLK = 1\,\text{MHz} \implies 1\,\mu s$ per count.
- CCR0 value: 50000 counts.
- Time per interrupt:
$$T_{int} = 50000 \times 1\,\mu s = 50\,ms$$
- Each interrupt toggles the LED (ON/OFF), so full cycle $= 2 \times 50\,ms = 100\,ms$.
- Blinking frequency:
$$f = \frac{1}{0.1\,s} = 10\,Hz$$

Answer: The LED is blinking at **10 Hz**.

2. Blinking at 25 Hz

We want LED to blink at 25 Hz $\implies$ period $= 40\,ms$, half-period $= 20\,ms$.

$$T_{int} = \frac{1}{25 \times 2} = 20\,ms$$

Since timer increments every $1\,\mu s$, required counts:

$$N = \frac{20\,ms}{1\,\mu s} = 20000$$

Answer: Set `TA0CCR0 = 20000` to achieve $25\,Hz$.

3. Blinking at 0.5 Hz (Heartbeat)

We want LED period $T = 2\,s$ (1 s ON + 1 s OFF). Thus half-period $= 1\,s = 1,000,000\,\mu s$.

$$N = \frac{1,000,000\,\mu s}{1\,\mu s} = 1,000,000$$

Answer: We need $1,000,000$ counts per half-period.

4. Feasibility at 1 MHz Clock

The Timer_A module in MSP430FR2433 is a 16-bit timer, meaning:

$$\text{Max count} = 65535$$

Since $1,000,000 > 65535$, it cannot be achieved directly with the current $1\,MHz$ clock.

Solution:

- Use a clock divider (`ID` or `TAIDEX`) to slow down timer increments.
- For example, with a divider of 16:

  $$T_{tick} = 16\,\mu s \quad \Rightarrow \quad N = \frac{1,000,000}{16} = 62500$$

  which fits within 16-bit.
- Alternatively, use `ACLK = 32.768 kHz` with divider, which is common for low-frequency applications.

Answer: Not possible directly. Use prescalers or ACLK to achieve 0.5 Hz blinking.

Chapter 4

Modified Codes

For 25 Hz Blinking

```
TA0CCR0 = 20000; // 25 Hz blink
```

For 0.5 Hz Blinking (using divider)

```
TA0CTL = TASSEL__SMCLK | MC__UP | ID__16 | TACLR; // SMCLK/16
TA0CCR0 = 62500; // Achieves ~0.5 Hz blinking
```

Chapter 5

Conclusion

The exercises demonstrate how timer configuration and the CCR0 register determine the interrupt period, which directly controls the LED blink frequency.

- Current frequency: 10 Hz.
- At 25 Hz, CCR0 must be set to 20000.
- At 0.5 Hz, CCR0 would require 1,000,000 counts, which exceeds 16-bit capacity.
- Prescalers or alternate clock sources must be used for such long delays.