

HACETTEPE UNIVERSITY
ELE417 EMBEDDED SYSTEM DESIGN

PRELIMINARY WORK 2

AHMET ANBAR
21628023

09/12/2020

1) What is a timer in the concept of embedded system design? Give at least three examples of usages for timers.

A timer is a specialized type of clock which is used to measure time intervals. In embedded system design, there is on-chip RC oscillator to use timer. In applications, we could also use external modules as timer, but on-chip models are more favorable.

- Counting running time
- Watch application
- HC-SR04 sends and receives a signal to measure distance between something and itself. In embedded system we need to count the elapsed time between sending and receiving signal.

2) What are the differences between using a timer and using an empty for loop to measure or pass the required amount of time?

When using loop to measure time, we assume approximate elapsed time by the clock speed of our embedded system. By the way, the elapsed time could be change because of running different instruction by our chip. But the timer is independent of our instructions. It is better to use. It is not affected from other code snippets.

3) Is there any difference between Watchdog timer and Timer A (or B)? If so what are the differences?

Watchdog timer is used to prevent the abnormal activity of our program like infinite loop or other failure status. Its main purpose is different. Timer A or B have timer purpose. They have different modes.

4) What are the registers used to program timers in your choice of MSP430 kit you obtained for your project?(Emphasize the model of your kit.)















TAXCTL, TAXCCTLn, TAXR, TAXCCRN, TAXEX0, TAXIV. My kit is MSP430FR4133.

5) How can you reset an MSP430 in software? What are the usages of software reset?(Hint: Check the course slides.)

There are various ways to reset in software.

- Enabling watchdog and calling infinite loop.
- Using PowerManagementModule for simulating reset by using brownout reset and power-on reset.

6)

▼  Watchdog_Timer			
▼  WDTCTL			0x6904
 WDTCTL			0
 WDTSSSEL			00 - WDTSSSEL_0
 WDTTMSSEL			0
 WDTCNTCL			0
 WDTIS			100 - WDTIS_4
▼  SFRIFG1	0x00C2		Interrupt Flag 1 [Memory Mapped]
 JMBOUTIFG	1		JTAG Mail Box output Interrupt Flag
 JMBINIFG	1		JTAG Mail Box input Interrupt Flag
 NMIIIFG	0		NMI Interrupt Flag
 VMAIFG	0		Vacant Memory Interrupt Flag
 OFIFG	1		Osc Fault Flag
 WDTIFG	0		WDT Interrupt Flag

```

#include <mbsp430.h>

int main(void)
{
    WDTCTL = WDT_ADLY_1000;          // WDT 32ms, SMCLK, interval timer

    P1OUT &= ~BIT0;    // Clear P1.0 output latch for a defined power-on state
    P1DIR |= BIT0;      // Set P1.0 to output direction
    P4OUT &= ~BIT0;    // Clear P4.0 output latch for a defined power-on state
    P4DIR |= BIT0;      // Set P4.0 to output direction

    PM5CTL0 &= ~LOCKLPM5; // Disable the GPIO power-on default high-impedance mode
                          // to activate previously configured port settings

    SFRIF1 |= WDTIE;          // Enable WDT interrupt

    __bis_SR_register(LPM0_bits | GIE); // Enter LPM0, enable interrupts
    __no_operation();          // For debugger
}

#pragma vector=WDT_VECTOR
__interrupt void WDT_ISR(void)
{
    P1OUT ^= BIT0;          // Toggle P1.0 (LED)
    P4OUT ^= BIT0;          // Toggle P4.0 (LED)
}

```

7)

```
#include <msp430.h>

/**
 * main.c
 */

unsigned int counter = 0;
int long_or_not = 0;

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

    P1DIR |= BIT0;

    PM5CTL0 &= ~LOCKLPM5;
    TA0CTL = TASSEL__SMCLK + MC__UP + ID__8; // smclock sourcemuz ve mod secimi up ccr0 a kadar
    TA0CCR0 = 40000; // maximum 65535
    TA0CCTL0 |= CCIE;
    __bis_SR_register(GIE + LPM0_bits);

    return 0;
}

#pragma vector = TIMER0_A0_VECTOR
__interrupt void TimerA_ISR()
{
    counter+=1;
    if(counter==3){
        P1OUT ^= BIT0;
        counter = 0;
    }
}
```

8)

```
#include <msp430.h>

/**
 * main.c
 */

unsigned int counter = 0;
int long_or_not = 0;
int turn_on_off_counter = 0;
// short delay 1 second, long delay 2 second
int short_delay_counter = 3;
int long_delay_counter = 6;
int compared_counter = 3;

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer

    P1DIR |= BIT0;

    PM5CTL0 &= ~LOCKLPM5;
    TA0CTL = TASSEL__SMCLK + MC__UP + ID__8;
    TA0CCR0 = 40000; // maximum 65535
    TA0CCTL0 |= CCIE;
    __bis_SR_register(GIE + LPM0_bits);

    return 0;
}

#pragma vector = TIMER0_A0_VECTOR
__interrupt void TimerA_ISR()
{
    counter += 1;

    if (counter == compared_counter)
    {
        P1OUT ^= BIT0;
        turn_on_off_counter++;

        if (turn_on_off_counter == 2)
        {
            compared_counter = long_delay_counter;
        }
        if (turn_on_off_counter == 4)
        {
            compared_counter = short_delay_counter;
            turn_on_off_counter = 0;
        }
        counter = 0;
    }
}
```