ELE 417 EMBEDDED SYSTEM DESIGN LABORATORY

PRELIMINARY WORK 3

NAME SURNAME : FATIH KAYA
NUMBER : 21329281

Q.1)

Timer is a hardware which is used when accurate timing is needed. It is one of the most part an embedded system, because without it is not easy to know the exact elapsed time. It can be used

- to make accurate delays.

- as a trigger for another hardware like ADC module or PWM module

- to generate interrupt in a desired time interval.

- to make polling in embedded systems

Q.2)

Using timers gives more accurate results than using loops in timing applications. When loops are used, elapsed time changes due to number of cycles to perform instructions on that loop. A rough calculation can be done but it still will not give reliable results. And this calculation varies according to the architecture of the microcontroller. Besides that, using timer is also leads to less power consumption. We can use low power mode by disabling the clock connection of CPU when using timers but it is not possible when loops are used. So it is better to use timer in timing applications.

Q.3)

The purpose of watchdog timer is to reset the microcontroller when there is an abnormal activity in the program unless it is disabled. It counts to only a couple of certain time intervals. But timer A and timer B are general purpose timers. They can be used in different modes, and the intervals can be changed.

Q.4)

Timer Control Register (TACTL): This register is used to configure the timer.

Timer Counter Register(TAR): This register holds the current count value.

Timer Capture Compare Register(TACCR0): This register holds the upper limit of the counter when it is used in up count mode.

Q.5) I use MSP430G2553.

Timer Control Register(TACTL): This register is used to configure the timer.

Timer Counter Register(TAR): This register holds the current count value.

Timer Capture Compare Register(TACCR0): This register holds the upper limit of the counter when it is used in up count mode.

Q.6)

Power management control register can be used to reset MSP430 in software. There are two bits to trig reset module in that register. These registers are PMMSWBOR and PMMSWPOR which are brown-out reset and power on reset respectively. To set that bits following codes can be used.

- PMMCTL0 = PMMPW | PMMSWBOR; // This triggers a Software BOR
- PMMCTL0 = PMMPW | PMMSWPOR; // This triggers a Software POR

Q.7)

```
#include <msp430.h>
int main(void)
{
      WDTCTL = WDT_ADLY_1000; // stop watchdog timer
      P1DIR |= BIT6;  // P1.6 is configured as output
      P1OUT = ~BIT6; // Led on P1.6 is turned off
      while(1){
          if(IFG1&BIT0 == WDTIFG){
              P1OUT ^= BIT6;
              IFG1 &= ~WDTIFG;
          }
      }
}
```

Q.8)

```
#include <msp430.h>
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer
    P1DIR |= BIT6;  // P1.6 is configured as output
    P1OUT = ~BIT6; // Led on P1.6 is turned off
    TACCR0 = 32797;
    TACTL = TASSEL_1 | MC_1 | TACLR;
    while(1){
        if(TACTL&TAIFG == TAIFG){
            P1OUT ^= BIT6;
            TACTL &= ~TAIFG;
        }
    }
}
```

Q.9)

```
#include <msp430.h>
#define LONG_DELAY 32797
#define SHORT_DELAY LONG_DELAY/4

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD; // stop watchdog timer
    P1DIR |= BIT6;  // P1.6 is configured as output
    P1OUT |= BIT6; // Led on P1.6 is turned on
```

```
    TACCR0 = LONG_DELAY;
    TACTL = TASSEL_1 | MC_1 | TACLR;

    volatile unsigned int isShort = 0;
    volatile unsigned int isLong = 1;
    volatile unsigned int count = 0;
    while(1){
        // blink when interrupt flag is set
        if(TACTL&TAIFG == TAIFG){
            P1OUT ^= BIT6;
            TACTL &= ~TAIFG;
            count++;
        }
        // after one blink set the delay type
        if(count == 2){
            count=0;
            isShort ^= 1;
            isLong ^= 1;
        }
        // Due to delay type set TACCR0 register
        if(isShort == 1){
            TACCR0 = SHORT_DELAY;
        }
        else if(isLong == 1){
            TACCR0 = LONG_DELAY;
        }
    }
}
```