HACETTEPE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ELE 417 EMBEDDED SYSTEM DESIGN

PRELIMINARY WORK #3- TIMER USAGE ON MSP430

Nazlı CANATAN

21590872

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. Timers are a fundamental concept in embedded systems and they have many use cases such as executing a periodic task, implementing a PWM output or capturing the elapsed time between two events to name a few.

The Examples of Usages for Timers:

- Controlling the brightness of LEDs.
- Controlling the angle of servo shafts.
- Receiving sensor data that transmit in PWM (Pulse Width Modulation)

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

In the embedded systems, waiting time is necessary. If an empty loop is used to create a wait time, the memory is kept busy. This situation leads to unnecessary CPU usage.

If an timer is used to create a wait time, the CPU cannot be used unnecessarily because the CPU knows when to run and it stops when the task is finished. Interrupt can be performed when the timer is used.

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

The main purpose of the watchdog timer is to protect the system against failure of the software, such as the program becoming trapped in an unintended, infinite loop. Also we can use it for a normal timer application.

4. What are the registers used to program timers in MSP430FG4618? Explain what they do.

TACTL –Timer A Control Register: This register used to configure how the timer runs.

TACCTL0 –Capture/Compare Control Register: For enabling and disabling TimerA0 interrupt.

TACCR0 –Capture/Compare Register: This register holds the value YOU define to configure the timing.

TACTL(Timer A Control Register), TBCTL(Timer B Control Register.

TACTL= 16bit register;

(bits 15-10) = unused,

(bits 9-8)= TASSELx Timer_A clock source select,

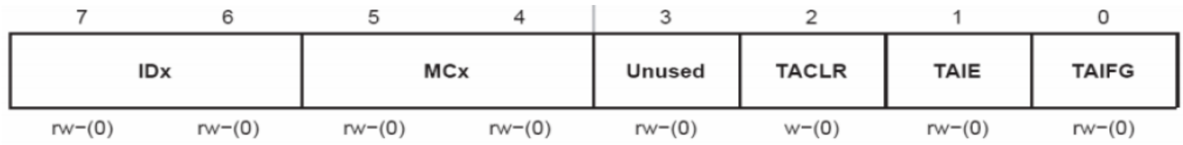
(bit 7-6)= IDx Input Divider(extends period),

(bits 5-4)= MCx Mode Control(up, continuous, up-down),

(bit2)= TACLR resets TAR, ID, MC,

(bit1)= TAIE TA Interrupt Enable,

(bit0)= TAIFG TA Interrupt Flag TBCTL is similar to TACT



7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw−(0)	rw−(0)	rw−(0)	rw−(0)	rw−(0)	w−(0)	rw−(0)	rw−(0)

5. What are the registers used to program timers in your choice of MSP430 kit you obtained for your project?

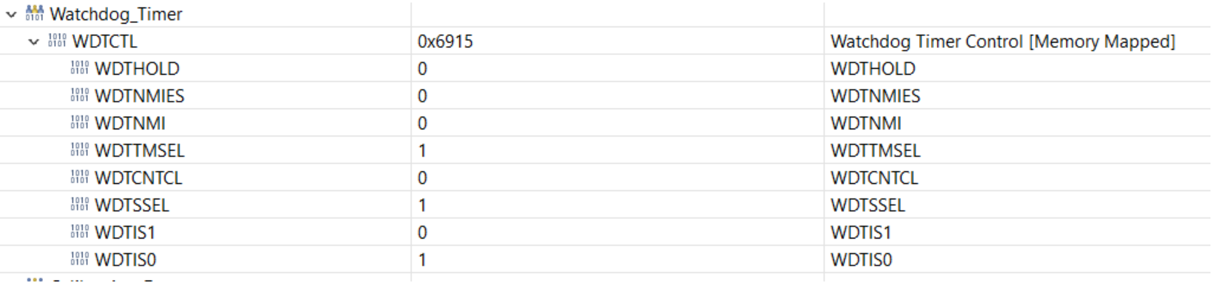
My development kit is MSP430G2553 and the registers are same as MSP430FG4618.

6. How can you reset an MSP430 in software? What are the usages of software reset?

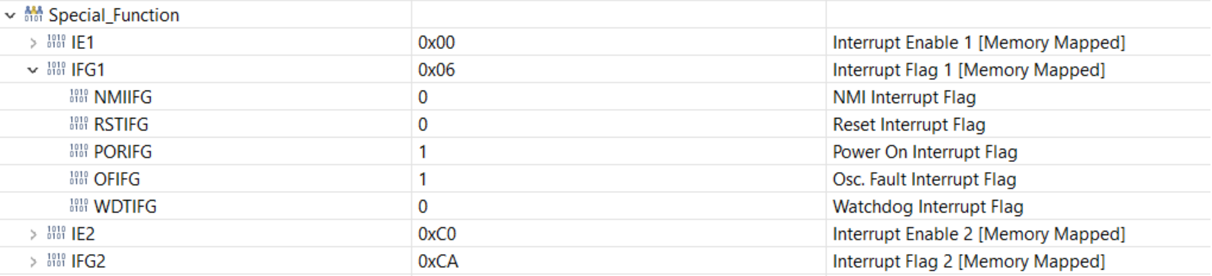
The watchdog counts up and resets the MSP430 when it reaches its limit. The code must therefore keep clearing the counter before the limitis reached to prevent a reset.

7.

```
#include <msp430.h>
void main(void)
{
WDTCTL = WDTPW | WDTCNTCL | WDTTMSEL | WDTSSEL | WDTIS0;// watchdog timer
P1DIR |= BIT0 | BIT6; // configure P1.0 and P1.6 as output
P1OUT &= ~BIT0;
P1OUT &= ~BIT6;
for(;;)
{
if((IFG1 & WDTIFG)==1) //wait the flag=1
{
P1OUT ^= BIT0|BIT6; //toggle leds
WDTCTL = WDTPW | WDTCNTCL | WDTTMSEL | WDTSSEL | WDTIS0;
IFG1 &= ~WDTIFG;
}
}
}
```



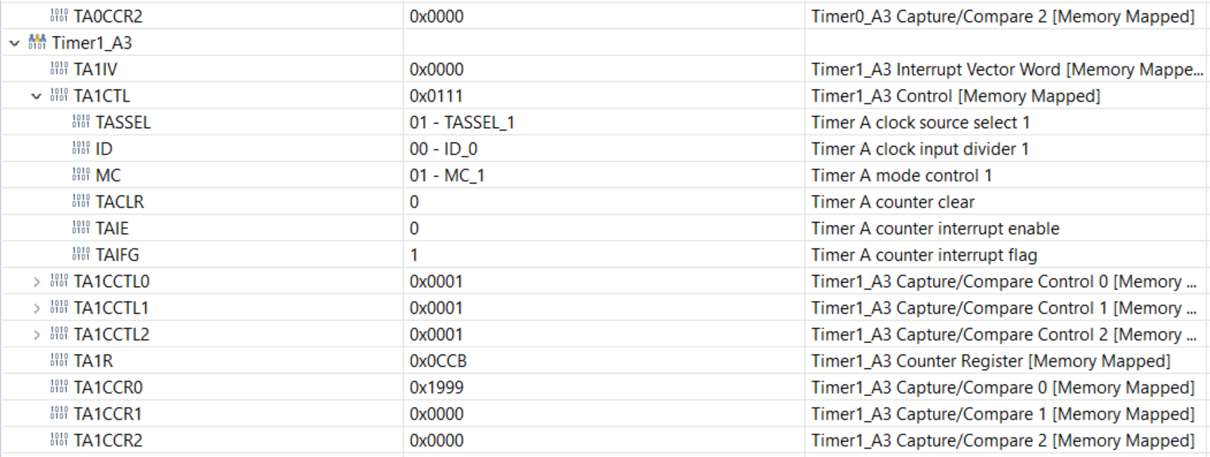
Watchdog_Timer		
WDTCTL	0x6915	Watchdog Timer Control [Memory Mapped]
WDTHOLD	0	WDTHOLD
WDTNMIES	0	WDTNMIES
WDTNMI	0	WDTNMI
WDTTMSEL	1	WDTTMSEL
WDTCNTCL	0	WDTCNTCL
WDTSSEL	1	WDTSSEL
WDTIS1	0	WDTIS1
WDTIS0	1	WDTIS0



Special_Function		
IE1	0x00	Interrupt Enable 1 [Memory Mapped]
IFG1	0x06	Interrupt Flag 1 [Memory Mapped]
NMIIFG	0	NMI Interrupt Flag
RSTIFG	0	Reset Interrupt Flag
PORIFG	1	Power On Interrupt Flag
OFIFG	1	Osc. Fault Interrupt Flag
WDTIFG	0	Watchdog Interrupt Flag
IE2	0xC0	Interrupt Enable 2 [Memory Mapped]
IFG2	0xCA	Interrupt Flag 2 [Memory Mapped]

8.

```
#include <msp430.h>
void main(void)
{
WDTCTL = WDTPW | WDTHOLD; //stop watchdog timer
P1DIR |= BIT0 | BIT6; // set LEDs as an output
P1OUT &= ~(BIT0 | BIT6) ; //firstly,LEDs should be OFF mode
TA1CCR0= 6554 -1; //set timer number
TA1CTL = MC0| TASSEL0 |TACLR; //// Set up and start Timer A
// Halt mode select, clock from TACLK , clear timer
for(;;)
{
while((TA1CTL & TAIFG)==1) //wait the flag=1
{
P1OUT ^= (BIT0 | BIT6); // toggle LEDs
TA1CTL &= ~TAIFG;
}
}
}
```



TA0CCR2	0x0000	Timer0_A3 Capture/Compare 2 [Memory Mapped]
Timer1_A3		
TA1IV	0x0000	Timer1_A3 Interrupt Vector Word [Memory Mappe...
TA1CTL	0x0111	Timer1_A3 Control [Memory Mapped]
TASSEL	01 - TASSEL_1	Timer A clock source select 1
ID	00 - ID_0	Timer A clock input divider 1
MC	01 - MC_1	Timer A mode control 1
TACLR	0	Timer A counter clear
TAIE	0	Timer A counter interrupt enable
TAIFG	1	Timer A counter interrupt flag
TA1CCTL0	0x0001	Timer1_A3 Capture/Compare Control 0 [Memory ...
TA1CCTL1	0x0001	Timer1_A3 Capture/Compare Control 1 [Memory ...
TA1CCTL2	0x0001	Timer1_A3 Capture/Compare Control 2 [Memory ...
TA1R	0x0CCB	Timer1_A3 Counter Register [Memory Mapped]
TA1CCR0	0x1999	Timer1_A3 Capture/Compare 0 [Memory Mapped]
TA1CCR1	0x0000	Timer1_A3 Capture/Compare 1 [Memory Mapped]
TA1CCR2	0x0000	Timer1_A3 Capture/Compare 2 [Memory Mapped]

9.

```
#include<msp430.h> //Nazlı Canatan
void main()
{
WDTCTL = WDTPW | WDTHOLD;//stop watchdog timer
TA1CCR0 = 3277; //Set Timer Number
P1OUT |= 0x41; // set LEDs as an output
P1DIR |= 0x41;
for(;;)
{
TA1CTL = MC0| ID1|TASSEL0 | TACLR ; // Set up and start Timer A
// Halt mode select,divide clock by 2, clock from TACLK , clear timer
while((TA1CTL & TAIFG) == 0){} //long delay
P1OUT ^= 0x41; //firstly,LEDs should be OFF mode
TA1CTL &= ~TAIFG;
TA1CTL = MC0| ID1|TASSEL0 | TACLR;// Set up and start Timer A again
while((TA1CTL & TAIFG) == 0){} // long wait when LEDs off
P1OUT ^= 0x41; //toggle LEDs on
TA1CTL &= ~TAIFG;
TA1CTL =  MC0|TASSEL0 | TACLR;
while((TA1CTL & TAIFG) == 0){} //short wait
P1OUT ^= 0x41;
TA1CTL &= ~TAIFG;
TA1CTL =  MC0| TASSEL0 | TACLR;
while((TA1CTL & TAIFG) == 0){} //short delay
P1OUT ^= (BIT0 | BIT6);
TA1CTL &= ~TAIFG;
}
}
```