

마이크로컨트롤러 기초(#514112)

#.9 Timer A3-1

한림대학교
전자공학과 이선우

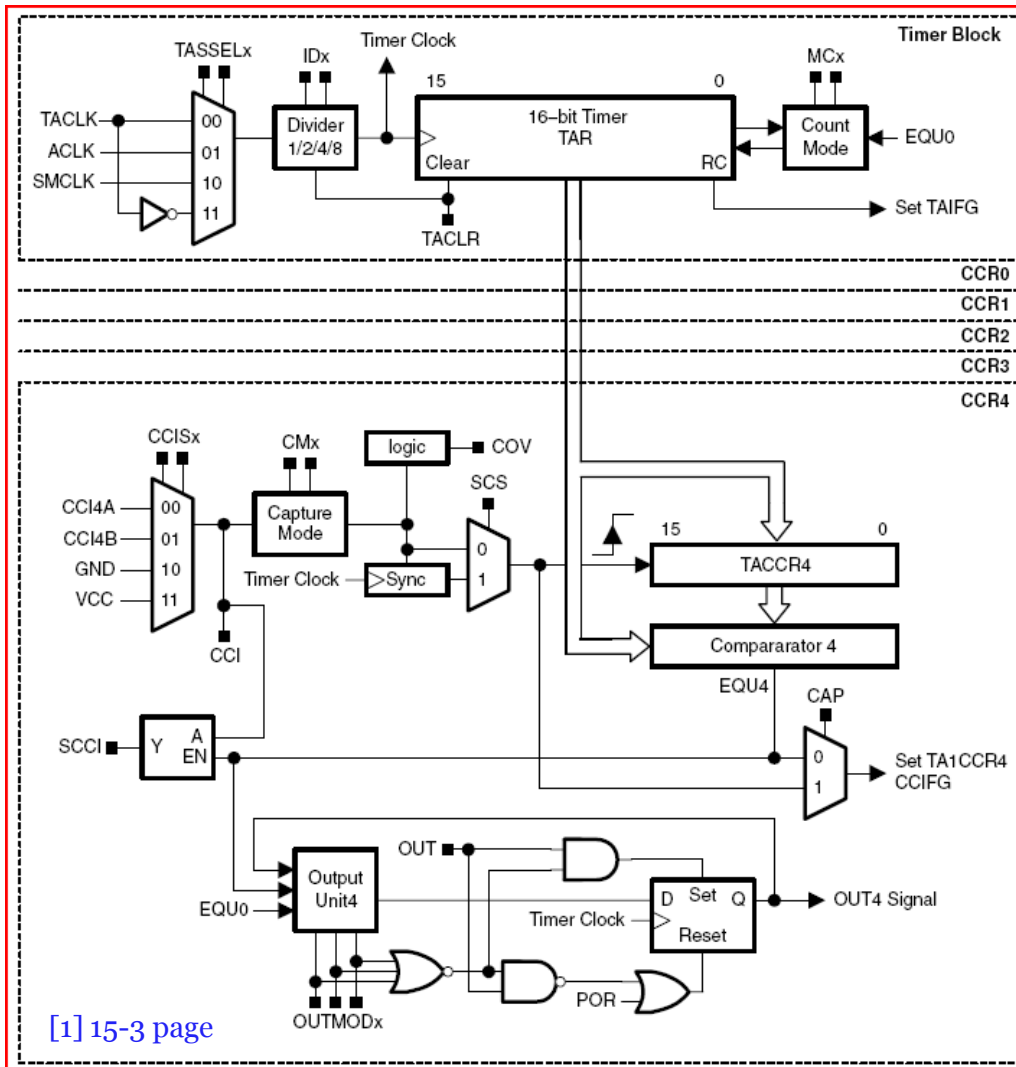
Contents

- ▶ **Timer A3**
 - ▶ Overview
 - ▶ Related control registers
 - ▶ Operations
 - ▶ Up mode
 - ▶ Continuous mode

Features of Timer_A

- ▶ 4개의 동작 모드를 가지는 비동기 16-bit timer/counter
- ▶ 클럭 소스 선택/설정 가능
- ▶ 칩에 따라 3개 혹은 5개의 capture/compare regs. 가짐
 - ▶ 의미: 변하는 카운터 값과 비교하여 동작 하는 제어 장치가 3/5개 있음을 뜻함.
 - ▶ MSP430FG4618: Timer_A3 (3개의 CCRs이 있음)
- ▶ PWM 기능을 가지는 설정 가능한 출력 기능
 - ▶ Pin을 통해 H/L 출력 가능
- ▶ 비동기 입/출력 레칭(latching)
- ▶ Timer_A 관련 인터럽트 발생 소스에 대한 빠른 해석을 위한 Interrupt vector reg.(TAIV)를 가짐.

Block diagram



▪ MSP430xG461x의 경우
 CCR0~CCR2까지 3개의
 제어장치만 존재
 → TAR 값을 이용하여 별도
 3개의 TACCR0/1/2를 이용

Related control registers

Table 15-3. Timer_A3 Registers

Register	Short Form	Register Type	Address	Initial State
Timer_A control Timer0_A3 Control	TACTL/ TA0CTL	Read/write	0160h	Reset with POR
Timer_A counter Timer0_A3 counter	TAR/ TA0R	Read/write	0170h	Reset with POR
Timer_A capture/compare control 0 Timer0_A3 capture/compare control 0	TACCTL0/ TA0CCTL	Read/write	0162h	Reset with POR
Timer_A capture/compare 0 Timer0_A3 capture/compare 0	TACCR0/ TA0CCR0	Read/write	0172h	Reset with POR
Timer_A capture/compare control 1 Timer0_A3 capture/compare control 1	TACCTL1/ TA0CCTL1	Read/write		Reset with POR
Timer_A capture/compare 1 Timer0_A3 capture/compare 1	TACCR1/ TA0CCR1	Read/write	0174h	Reset with POR
Timer_A capture/compare control 2 Timer0_A3 capture/compare control 2	TACCTL2/ TA0CCTL2	Read/write	0166h	Reset with POR
Timer_A capture/compare 2 Timer0_A3 capture/compare 2	TACCR2/ TA0CCR2	Read/write	0176h	Reset with POR
Timer_A interrupt vector Timer0_A3 interrupt vector	TAIV/ TA0IV	Read only	012Eh	Reset with POR

Timer block을 제어하는 레지스터. 즉 **TAR 동작 제어**

Rising edge clock signal 입력에 따라 값 증가/감소

Compare/Capture 제어 블록 #0 제어 레지스터

Compare/Capture 관련 데이터 레지스터 (16bit). 동작설정에 따라 TAR과 비교/TAR값 캡처된 결과 저장

- TACTL,TAoCTL의 차이점: 일부 장치(430x415/417,xW42x)는 동일한 TimerA를 2개 가짐. 430x461x 장치는 1개의 TA를 가지므로 TA만 사용함.
- 2개 Timer_A를 가지는 장치는 TA0/1을 구별하기 위해 TAoCTRL/TA1CTL로 다른 이름의 레지스터를 가짐. (표15-4 참조)

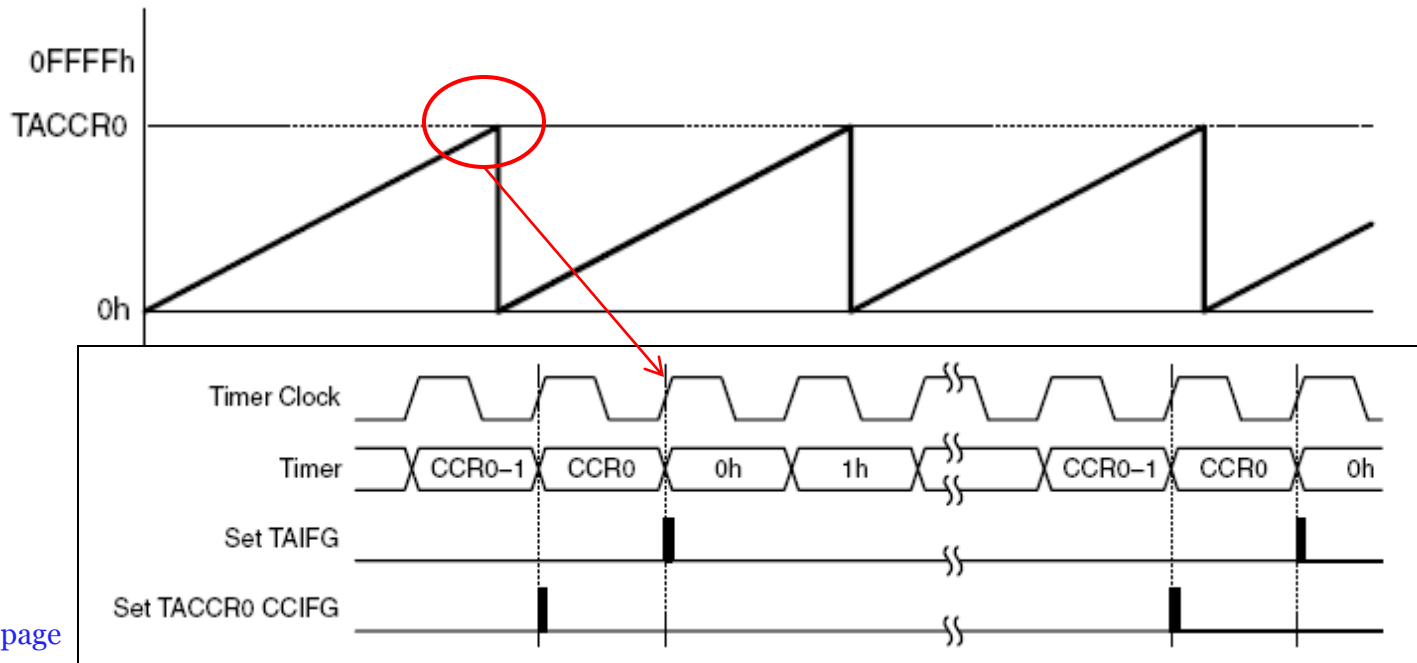
동작 방식

▶ Timer block

- ▶ 일반적인 16비트 up/down 카운터로 동작
- ▶ 클럭 소스: 3개 중 하나 (ACLK, SMCLK, TACLK)
 - ▶ TACLK: 외부에서 공급하는 클럭 신호 (x461x 장치:P1.5)
- ▶ Divider 제공: /2,/4,/8 중 하나 선택 가능 (Idx bits)
- ▶ Reset 기능: TACLR bit=1로 TAR, Idx, MCx clear. Automatic reset.
- ▶ 4개 Mode
 - ▶ MCx bit(in TACTL reg.)로 설정
 - 00: Stop → Timer_A 정지, 저전력 위해 사용하지 않을 때 사용
 - ▶ MCx>0: Running
 - 01:Up, 10:continuous, 11:Up/down

UP MODE

- ▶ 작동방식: TAR 값이 증가하다 CCR0 reg.에 써있는 값 (>0)이 되면 인터럽트를 발생시킴
 - ▶ 정확하게는 다른 종류의 2개 인터럽트 (CCIFG



[1] 15-6 page

Up mode 이용하여 특정 인터벌 만들기

```

#include <msp430xG46x.h>

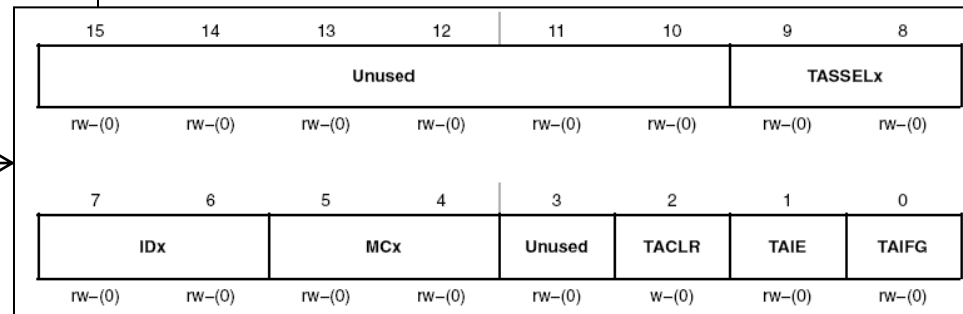
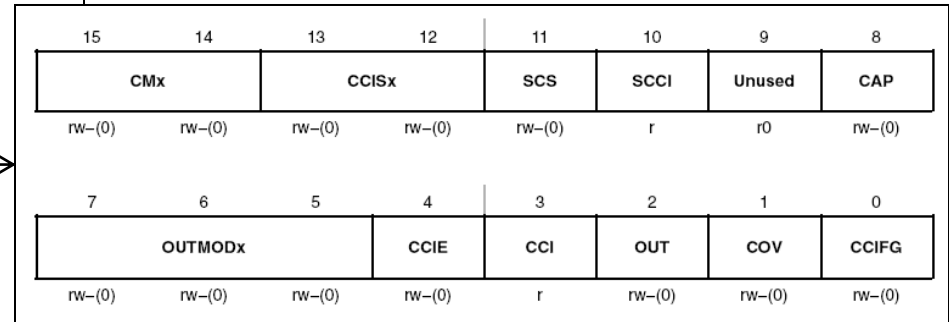
#pragma vector=TIMER_A0_VECTOR
__interrupt void timer_handler(void)
{
    P1OUT ^= 0x01; //toggle P1.0
}

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD

    P1DIR |= 0x01;

    //setup Timer_A
    TACCTL0 = 0x0010; // 0000 0000 0001 0000
    TACCR0 = 13180; //13180 usec
    TACTL = 0x0210 ; //0000 0010 0001 0000
    __enable_interrupt();
    __low_power_mode_0();
}
    
```

요구사항: 13.18msec 의 인터벌마다
P1.0 출력 toggle시켜 클럭 신호 발생



Timer_A3 관련 Interrupts

▶ 2개의 인터럽트 소스

- ▶ TACCR0 CCIFG0 (IAR C에선 TIMERA0_VECTOR)
- ▶ CCIFG1, CCIFG2, TAIFG는 모두 하나의 인터럽트로 처리됨. (P1/2와 동일) (TIMERA1_VECTOR)

ADC12	ADC12IFG (see Notes 1 and 2)	Maskable	0FFEEh	23
Timer_A3	TACCR0 CCIFG0 (see Note 2)	Maskable	0FFECh	22
Timer_A3	TACCR1 CCIFG1 and TACCR2 CCIFG2, TAIFG (see Notes 1 and 2)	Maskable	0FFEAh	21
I/O Port P1 (Eight Flags)	P1IFG.0 to P1IFG.7 (see Notes 1 and 2)	Maskable	0FFE8h	20

- ▶ CCIFG1/2 및 TAIFG는 순서대로 우선순위를 가지며 ISR에서 어떤 소스인가를 보다 빠르게 파악할 수 있도록 TAIV reg.를 제공함.

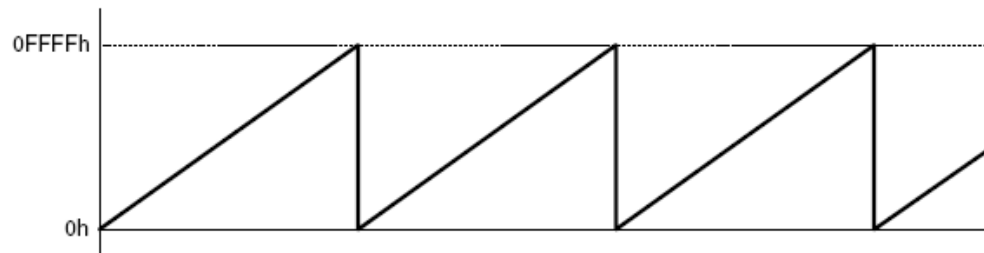
TAIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	-	
02h	Capture/compare 1	TACCR1 CCIFG	Highest
04h	Capture/compare 2	TACCR2 CCIFG	
06h	Capture/compare 3†	TACCR3 CCIFG	
08h	Capture/compare 4†	TACCR4 CCIFG	
0Ah	Timer overflow	TAIFG	
0Ch	Reserved	-	
0Eh	Reserved	-	Lowest

† Timer1_A5 only

Continuous mode

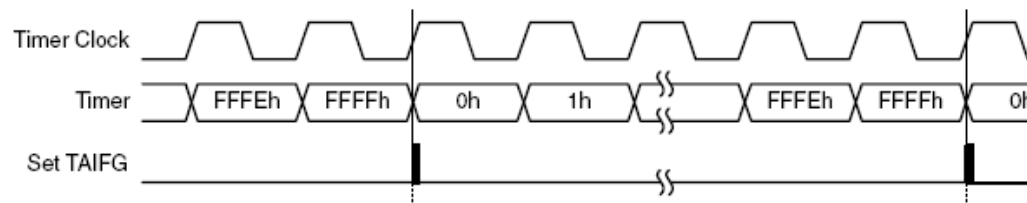
- ▶ 연속 모드: TAR이 TACCR_x의 값과 상관없이 계속 증가하다가 최대 0xFFFF 갔다가 다시 0으로 넘어서 동작함

Figure 15-4. Continuous Mode



The TAIFG interrupt flag is set when the timer *counts* from 0FFFFh to zero. Figure 15-5 shows the flag set cycle.

Figure 15-5. Continuous Mode Flag Setting



[1] 15-7 page

Continuous mode 이용 방법

```
#include <msp430xG46x.h>

#pragma vector=TIMER_A1_VECTOR
__interrupt void ta_cont_isr(void)
{
    switch (TAIV)
    {
        case 2: break; //TACCR1 not used
        case 4: break; //TACCR2 not used
        case 10: //use TAIFG
            P1OUT ^= 0x01; //toggle P1.0
    }
}

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD

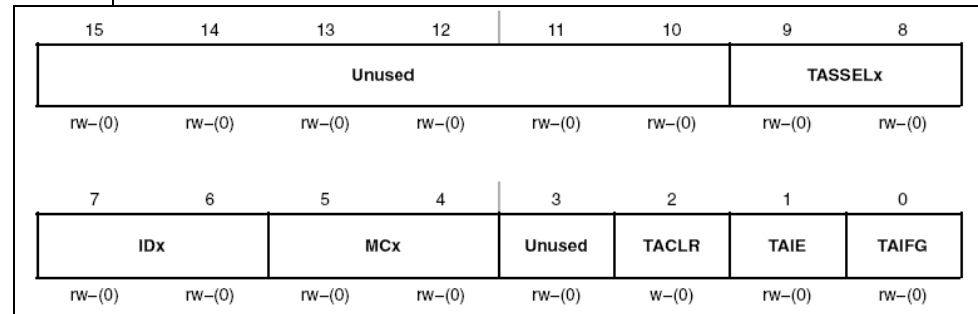
    P1DIR |= 0x01;

    //setup Timer_A
    TACTL = 0x0222 ; //0000 0010 0010 0010
    __enable_interrupt();
    __low_power_mode_0();
}
```

요구사항: 16Hz 의 인터벌마다 P1.0 출력 toggle시켜 클럭 신호 발생 → 8Hz 클럭 발생

TAIV: Timer_A Interrupt Vector register

CCIFGo 이외의 다른 소스는 모두 하나의 인터럽트로 간주하므로 이를 효과적으로 처리하기 위해 만들어진 레지스터



- TASSEL=10: clock source = SMCLK (=1048576Hz)
- MCx=10: continuous mode
- TAIE = 1: enable TAIFG interrupt request