

# 마이크로컨트롤러 기초(#514112 )

## #.10 Timer A3-2

한림대학교  
전자공학과 이선우

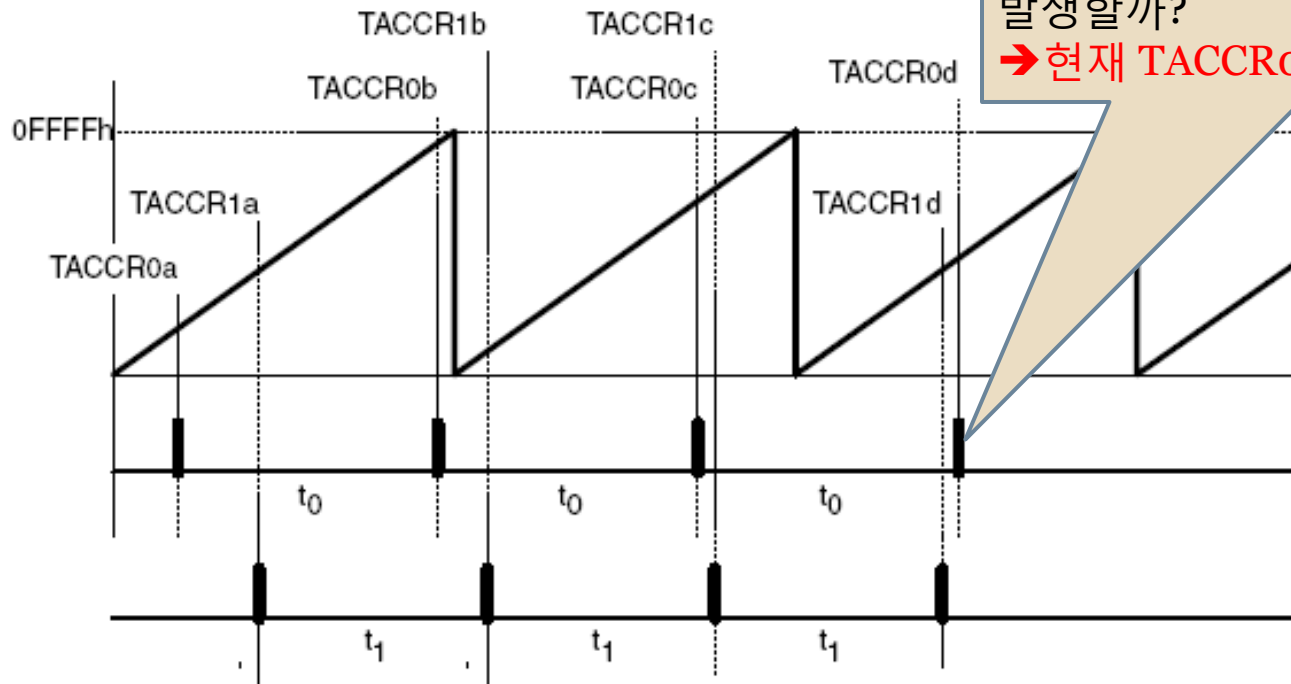
# Contents

---

- ▶ **Timer A3**
  - ▶ Continuous mode application
  - ▶ Up-down mode
  - ▶ Output mode
  - ▶ PWM pulse output

# Continuous Mode의 이용 방법

- ▶ 하나의 TAR(free-running counter)과 여러 개의 CCR<sub>x</sub>에 저장된 값과의 비교를 이용하여 복수개의 인터벌 발생이 가능



CCR<sub>0</sub> ISR에서 어떤 작업을 해야 동일한 시간간격 후에 다시 인터럽트가 발생할까?  
→ 현재  $TACCR_0 += TAR + INTERVAL$

# 복수 인터벌 만들기 (연속모드 이용법)

```
#include <msp430xG46x.h>
#define CCR1INTVAL 8000           //8msec
#define CCR2INTVAL 3000         //3msec

#pragma vector=TIMER_A1_VECTOR
__interrupt void ccr_handler(void)
{
switch (TAIV)
{
case 10: break; //TAIFG not used
case 2: //use TACCR1
    P1OUT ^= 0x01; //toggle P1.0
    TACCR1 += CCR1INTVAL;
    break;

case 4: //use TACCR2
    P1OUT ^= 0x02; //toggle P1.1
    TACCR2 += CCR2INTVAL;
    break;
}
}
```

현재의 CCR2 값에 원하는 인터벌의 크기를 더하여 다음 CCR2 값을 갱신함.  
Ex. 현재 TAR=3000이 상태에서 TACCR2=6000이 되어 3000usec 후에 CCR2 IRQ 발생!

요구사항: **CCR1** 이용 **8msec** 인터벌마다 P1.0 출력 toggle, **CCR2** 이용 **3msec**마다 P1.1 toggle 함

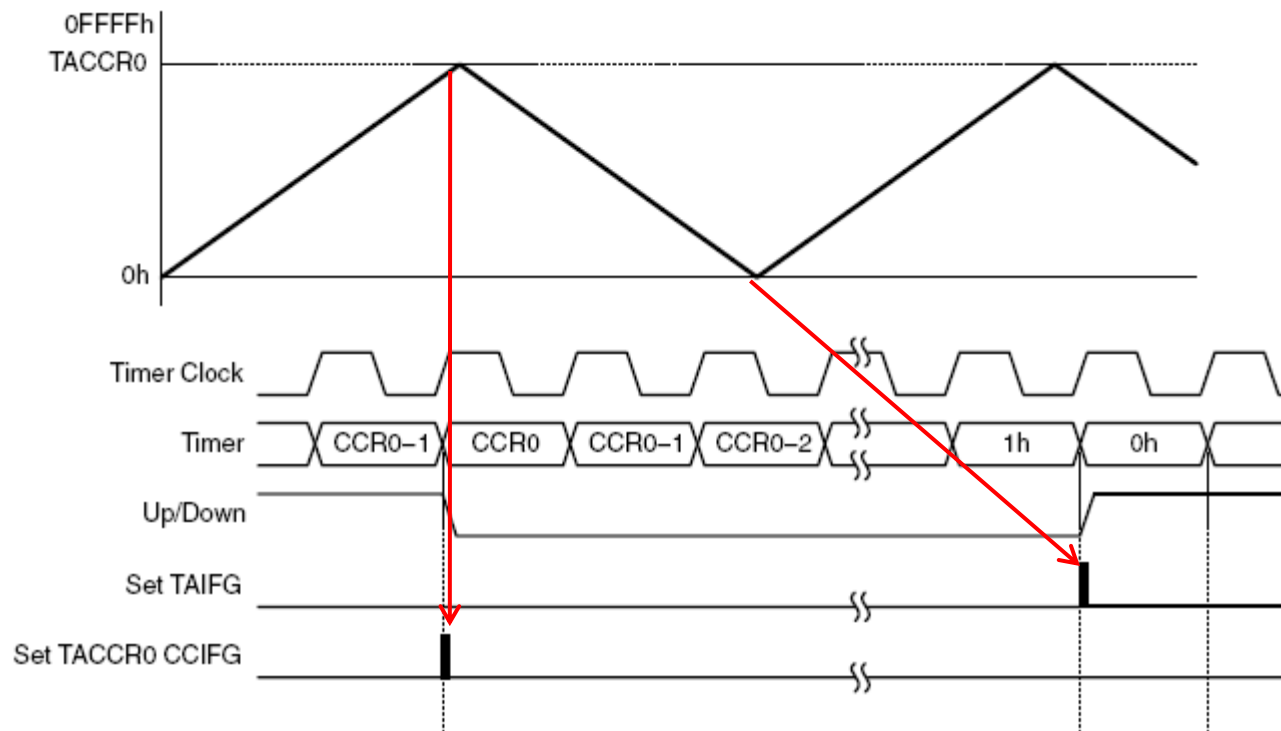
```
void main(void)
{
    WDTCTL = WDTPW + WDTM0;

    P1DIR |= 0x03;

    //setup Timer_A
    TACCTL1 = 0x0010; // CCR1 Interrupt Enable
    TACCR1 = 8000; //8ms
    TACCTL2 = 0x0010; // CCR2 Interrupt Enable
    TACCR2 = 3000; //8ms
    TACTL = 0x0220 ; //SSEL=SMCLK,
    MCx=continuous
    __enable_interrupt();
    __low_power_mode_0();
}
```

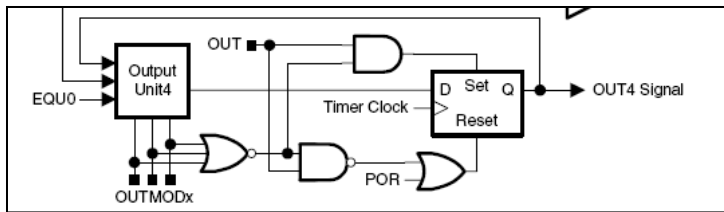
# Up/down mode

- ▶ Square pulse를 만들고자 할 때 이용
- ▶ TAR이 증가 후 감소함



# TA3의 입출력 핀 & Output Mode

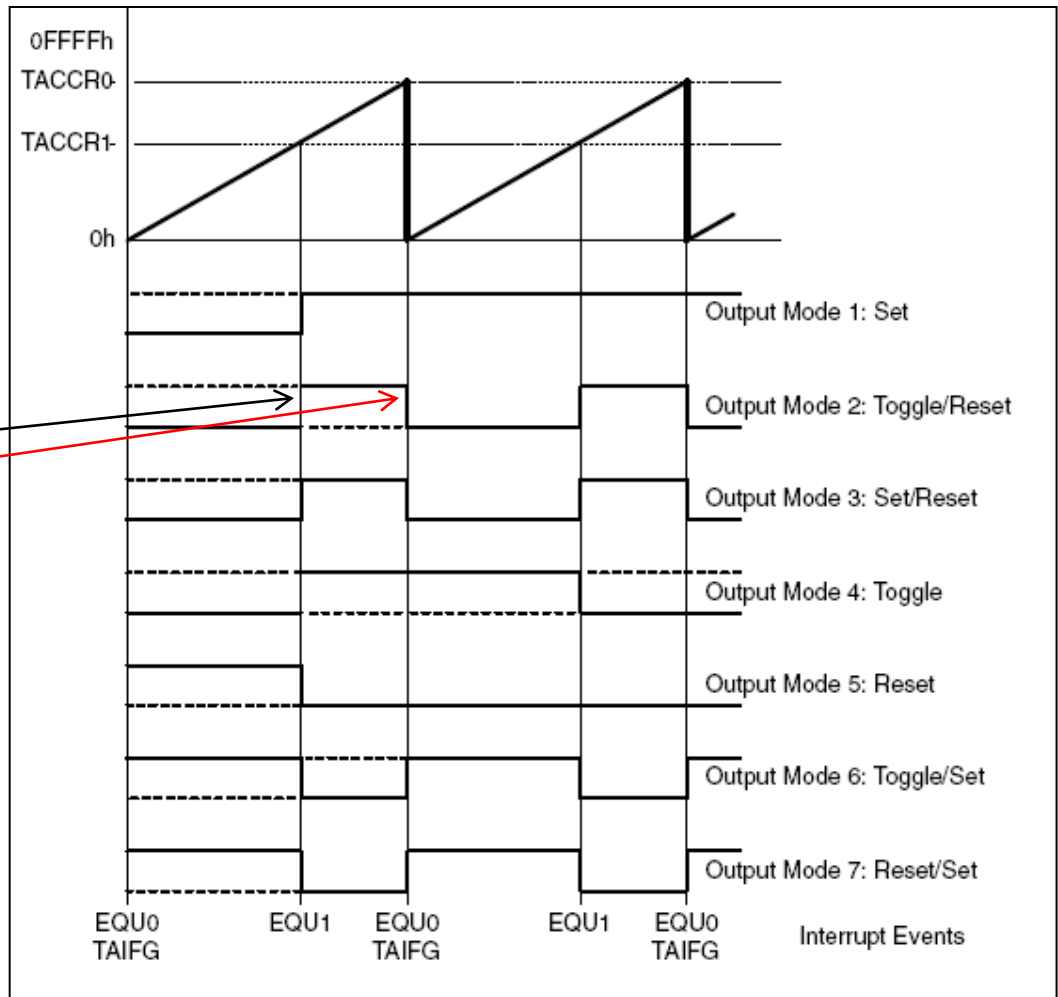
- ▶ MSP430FG461x의 timer\_A3의 입출력 핀 사양
  - ▶ 입력: P1.5(TACLK), P1.0(CCI0A), P1.2(CCI1A), P2.0(CCI2A)
  - ▶ 출력: P1.0(TA0), P1.2(TA1), P2.0(TA2)
- ▶ 각 CC block은 Output Unit이라는 장치를 가지며, 이를 통해 연결된 핀의 값을 주어진 설정에 따라 set/clear하여 원하는 신호를 만들어 낼 수 있다.



Timer_A3 Signal Connections					
Input Pin Number	Device Input Signal	Module Input Name	Module Block	Module Output Signal	Output Pin Number
PZ/QZW					PZ/QZW
82/B9 - P1.5	TACLK	TACLK	Timer	NA	
	ACLK	ACLK			
	SMCLK	SMCLK			
82/B9 - P1.5	TACLK	INCLK	CCR0	TA0	
87/A7 - P1.0	TA0	CCI0A			87/A7 - P1.0
86/E7 - P1.1	TA0	CCI0B			
	DV <sub>SS</sub>	GND			
	DV <sub>CC</sub>	V <sub>CC</sub>			
85/D7 - P1.2	TA1	CCI1A	CCR1	TA1	85/D7 - P1.2
	CAOUT (internal)	CCI1B			ADC12 (internal)
	DV <sub>SS</sub>	GND			
	DV <sub>CC</sub>	V <sub>CC</sub>			
79/A10 - P2.0	TA2	CCI2A	CCR2	TA2	79/A10 - P2.0
	ACLK (internal)	CCI2B			
	DV <sub>SS</sub>	GND			
	DV <sub>CC</sub>	V <sub>CC</sub>			

# Output Unit 이용법: UP Mode

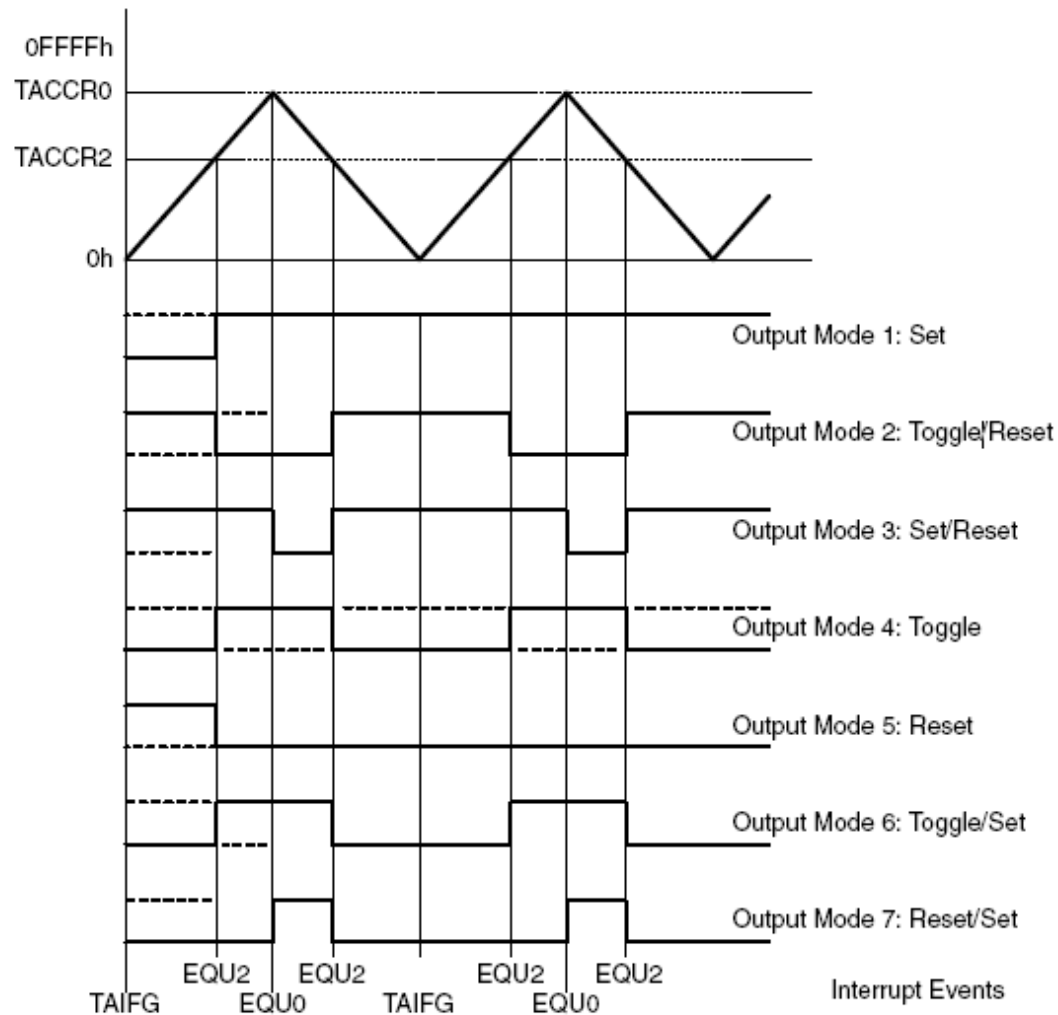
15	14	13	12	11	10	9	8
CMx		CCISx		SCS	SCCI	Unused	CAP
rw-(0)		rw-(0)		rw-(0)	r	r0	rw-(0)
7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw-(0)			rw-(0)	r	rw-(0)	rw-(0)	rw-(0)



## ▶ 7개 모드

- ▶ Set
- ▶ Toggle/Reset
- ▶ Set/Reset
- ▶ Toggle
- ▶ Reset
- ▶ Toggle/Set
- ▶ Reset/Set

# Output example: up/down mode





# Example codes: Output unit이용 클럭신호발생

```
/*
 * Just use OUTPUT mode, even not use IRQ.
 * Use UP mode, CCR0
 *
 * Result in P1.0/TA0 = 32768/200 Hz signal.
 */
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD

    P1DIR |= 0x01;
    P1SEL |= 0x01; //set P1.0 to TA0

    //setup Timer_A
    TACCTL0 = 0x0080; // OUTMODE=4 (Toggle)
    TACCR0 = 100-1; //32768Hz/100
    TACTL = 0x0110 ; //SSEL=ACLK, MCx=Up
    __low_power_mode_3();
}
```

```
/*
 * Just use OUTPUT mode, even not use IRQ.
 * Use UP/DOWN mode, CCR0
 *
 * Result in P1.0/TA0 = SMCLK/1000 Hz signal.
 */
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD

    P1DIR |= 0x01;
    P1SEL |= 0x01; //set P1.0 to TA0

    //setup Timer_A
    TACCTL0 = 0x0080; // OUTMODE=4 (Toggle)
    TACCR0 = 250;
    TACTL = 0x0230 ; //SSEL=SMCLK,
    MCx=Up/Down mode
}
```

# Pulse Width Modulation이란?

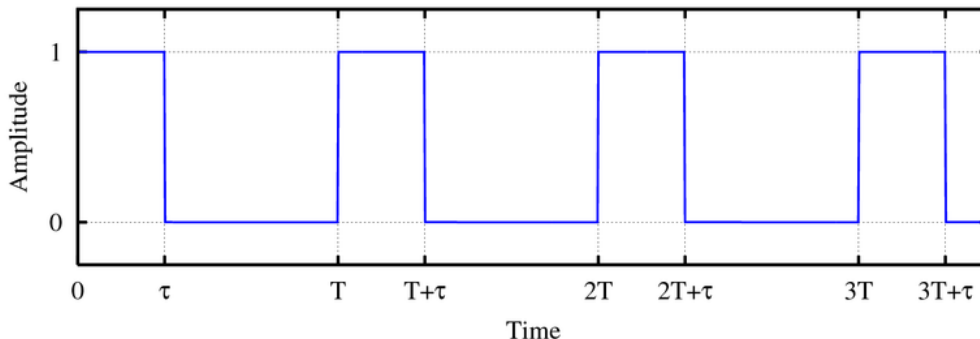
## ▶ PWM: Pulse Width Modulation

▶ ON/OFF만을 이용하여 중간 크기의 에너지를 전달할 수 있는 효과적인 방법

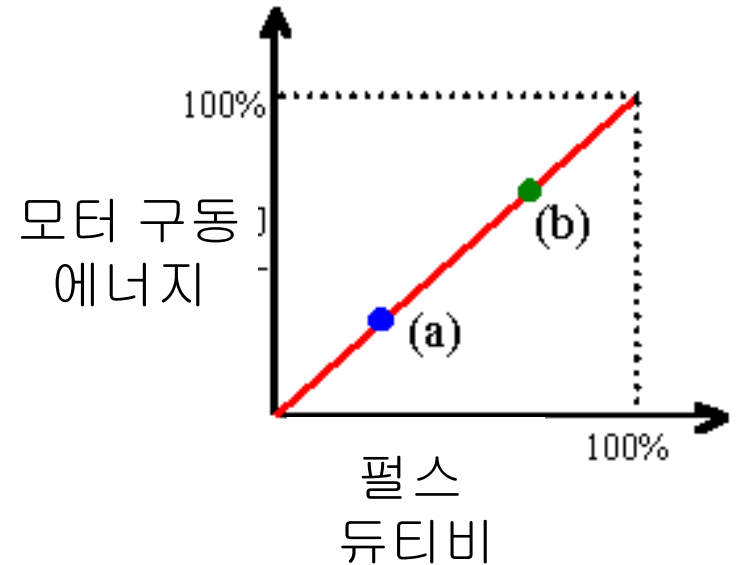
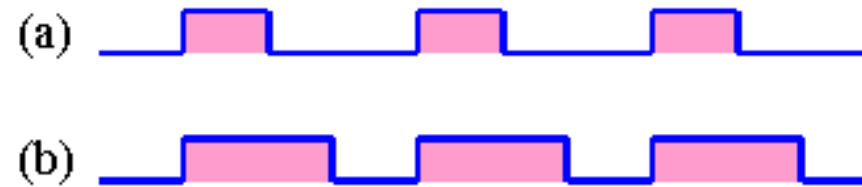
▶ Duty cycle:

▶ ON 시간/주기의 비율로 0~100%로 나타냄.

$$D = \frac{\tau}{T}$$



PWM에 의한 모터 속도제어 원리



# Example code: PWM 신호 발생

```
void main(void)
{
    WDTCTL = WDTPW + WDTHOLD

    P1DIR |= 0x04; /P1.2 Output
    P1SEL |= 0x04; //set P1.2 to TA1
    P2DIR |= 0x01; /P2.0 Output
    P2SEL |= 0x01; //set P2.0 to TA2

    //setup Timer_A
    TACCR0 = 512-1; //PWM Period
    TACCTL1 = 0x00E0; // OUTMODE=7 (reset/set)
    TACCR1 = 384; //duty cycle=75%
    TACCTL2 = 0x00E0; // OUTMODE=7 (reset/set)
    TACCR2 = 128; //duty cycle =25%
    TACTL = 0x0110 ; //SSEL=ACLK, MCx=Up mode
    __low_power_mode_3();
}
```

OUTMODE=reset/set 이용  
-EQU1에서 RESET: 주기 시작  
-CCRO overflow에서 SET: ON  
Timer End  
→ Duty cycle=384/512 = 75%

만약 set/reset OUTMODE를  
이용한다면??