마이크로컨트롤러 기초(#514112)

#.11 Timer A3-3

한림대학교
전자공학과 이선우

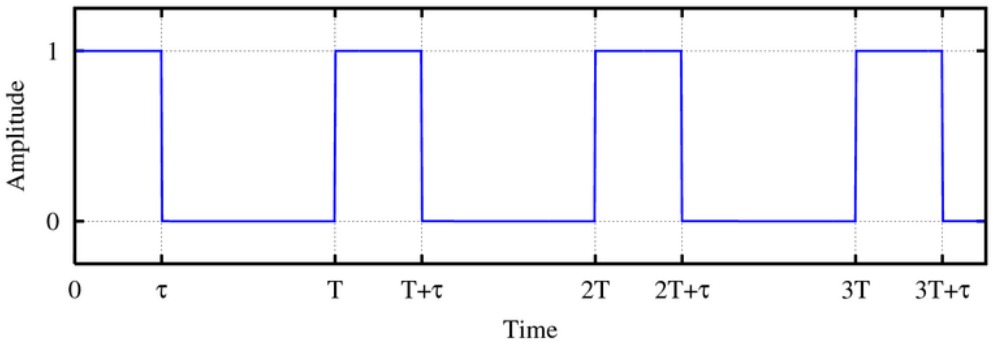
Contents

- Timer A3
  - PWM operation
  - Capture operation

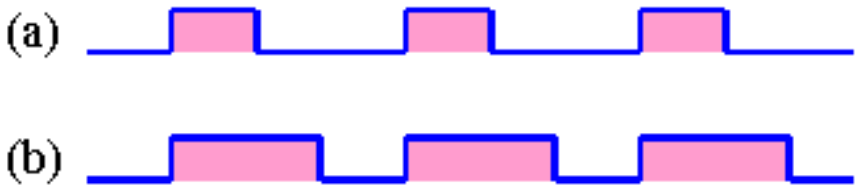
Pulse Width Modulation이란?

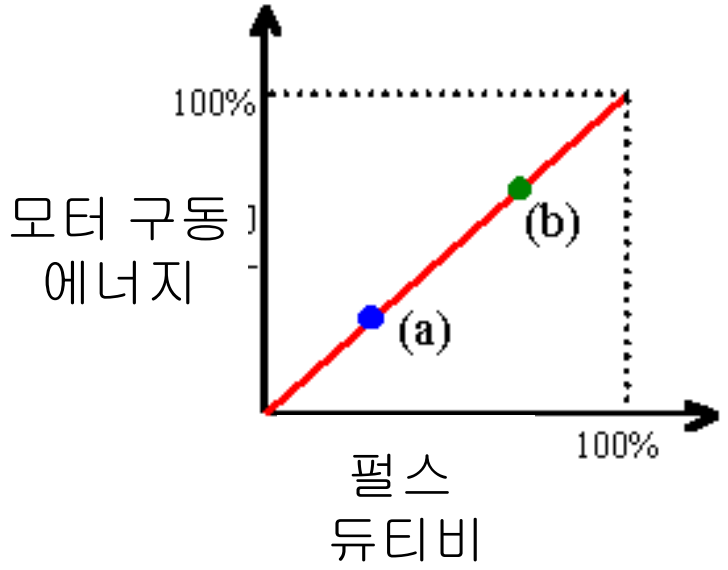
- PWM: Pulse Width Modulation
  - ON/OFF만을 이용하여 중간 크기의 에너지를 전달할 수 있는 효과적인 방법
  - Duty cycle:
    - ON 시간/주기의 비율로 0~100%로 나타냄.

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



PWM에 의한 모터 속도제어 원리





한림대학교 HALLYM UNIVERSITY
senu@hallym.ac.kr

Example code: PWM 신호 발생

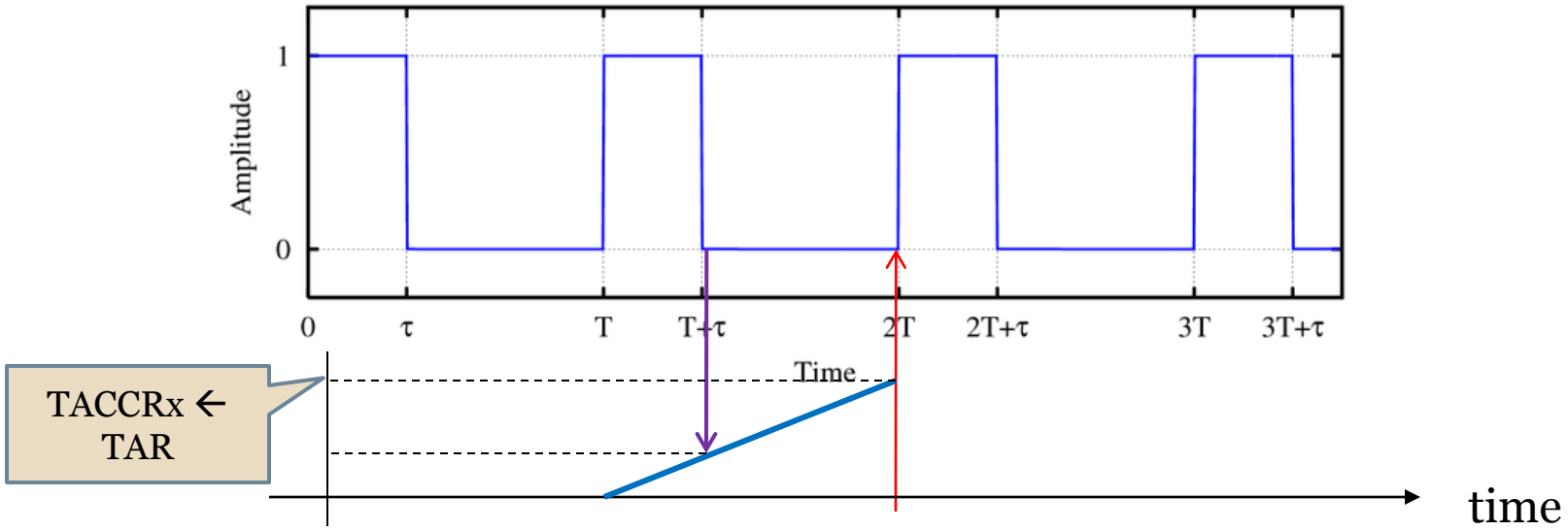
```
void main(void)
{
  WDTCTL = WDTPW + WDTHOLD

  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
-CCR0 overflow에서 SET: ON
Timer End
➔Duty cycle=384/512 = 75%

만약 set/reset OUTMODE를 이용한다면??

Capture mode

- Capture mode는 **시간 이벤트를 기록**하기 위해 이용됨.
  즉, MCU 내/외부 신호의 변화 시간 간격을 측정할 수 있다.
  - How? **내부 타이머의 값을 특정 시간에 일시 저장하여..**
  - 외부 클럭 신호의 주기 측정 → rising edge 사이 측정
  - 외부 신호의 ON/OFF time 측정 → rising/falling edge 사이 측정



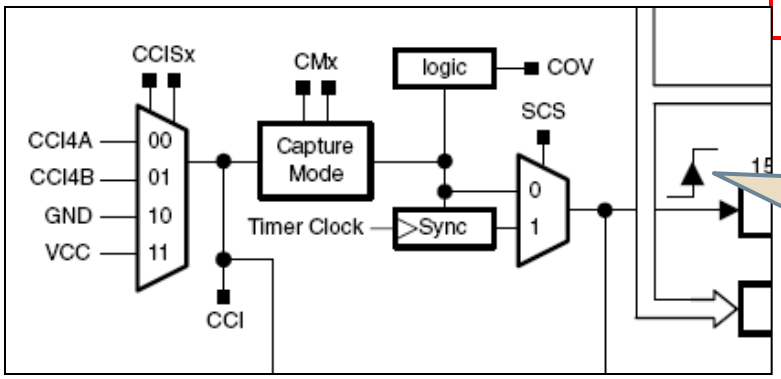
한림대학교 HALLYM UNIVERSITY
senu@hallym.ac.kr

Timer_A3 입력 핀

- MSP430FG461x의 timer_A3의 입출력 핀 사양
  - 입력: P1.5(TACLK), **P1.0(CCI0A), P1.1(CCI0B), P1.2(CCI1A), P2.0(CCI2A)**
- CAP=1
  - CCISx: 측정 대상 선택
  - CMx: input signal의 capture edge 선택→ rising, falling, both

Timer_A3 Signal Connections

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



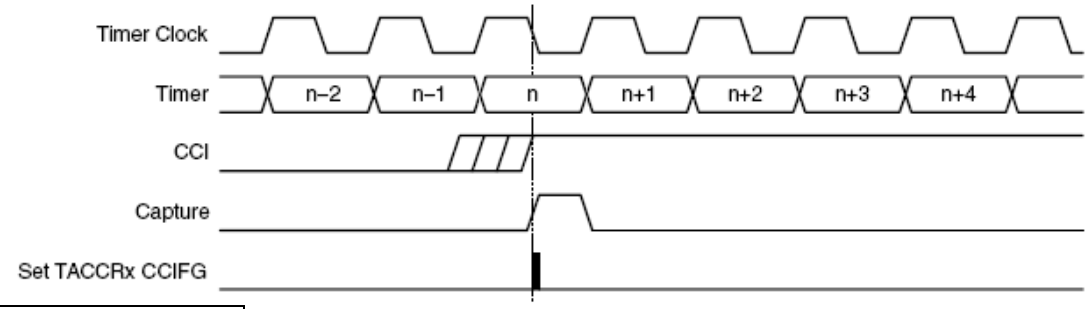
지정된 신호 입력되면(Capture occurs! 캡쳐 ON)
- TAR 값이 TACCRx reg.로 copy
- CCIFGx =1 (IRQ. 발생)

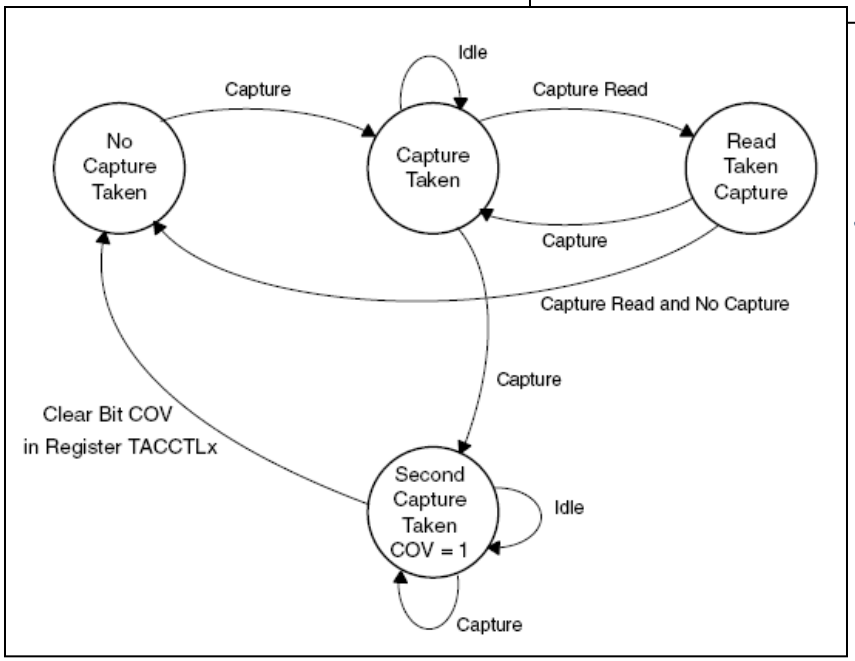
한림대학교 HALLYM UNIVERSITY
senu@hallym.ac.kr

Capture mode timing diagram

SCS: TACCTLx[11]
Synchronize capture source 입력신호 변화를 타이머 클럭과 동기화 시킴.

Figure 15–10. Capture Signal (SCS=1)





Capture Overflow 문제

- 캡쳐 발생 후 일시 저장된 값을 읽지 않은 상태에서 다시 캡쳐ON 되어 기존 값을 잃어버리는 경우➔ overflow라고 함.
- 이 경우 HW가 자동적으로 찾아 COV=1로 설정함.
- COV clearing은 반드시 SW에서 실행되야 함.

한림대학교 HALLYM UNIVERSITY
senu@hallym.ac.kr

Example code: 외부 신호 주기 측정

```
Int prev_cnt, Period;
int main( void )
{
  WDTCTL = WDTPW + WDTHOLD;

  //setup TA0 output
  P2SEL |= 0x01;  //P2.0 --> CCI2A
  P2DIR &= ~0x01;
  P2DIR |= 0x08; //P2.3 ouput

  //setupt TACTL0 for clock generation
  TACCTL0 =  CCIE;
  TACCR0 = 3000;  //3ms
  //setup TTACTL2 for capturing
  TACCTL2 =0x4910;
  TACCR2 = 0;
  TACTL = 0x0220 ;  //continuous mode, SMCLK

  prev_cnt = 0;
    __enable_interrupt();
  __low_power_mode_0();
  return 0;
}

#pragma vector=TIMERA0_VECTOR
__interrupt void ta0_isr(void)
{
  P2OUT ^= BIT3;
  TACCR0 += INTVL1;
}

#pragma vector=TIMERA1_VECTOR
__interrupt void ta_isr(void)
{
  switch(TAIV) {
  case 2: break;
  case 10: break;
  case 4:
    Period = TACCR2 -prev_cnt;
    prev_cnt = TACCR2;

  }
}
```

Capture mode 이용

- IRQ 발생 시점: P2.3 clock rising edge
- TAR은 연속모드이므로 클럭 주기 측정을 위해서는 사이값을 계산해야함.