

# 로봇학실험3 실험보고서

2014741022 서덕현

-TIMER/COUNTER 레지스터 설정

Bit	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

1. Timer/Counter2를 쓴다는 조건이므로 데이터시트에 **8-bit Timer/Counter2 with PWM and Asynchronous Operation** 제목으로 가서 내려가보면 Timer/Counter2를 설정해주는 레지스터가 나온다.

- **Bit 7 – FOC2: Force Output Compare**

The FOC2 bit is only active when the WGM bits specify a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR2 is written when operating in PWM mode. When writing a logical one to the FOC2 bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC2 output is changed according to its COM21:0 bits setting. Note that the FOC2 bit is implemented as a strobe. Therefore it is the value present in the COM21:0 bits that determines the effect of the forced compare.

A FOC2 strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR2 as TOP.

2. 밑줄 친 내용처럼 이번 실험에서는 PWM모드를 쓰므로 Bit 7번을 0으로 설정해준다.

**Table 51.** Waveform Generation Mode Bit Description<sup>(1)</sup>

Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation	TOP	Update of OCR2	TOV2 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

3. Bit 6번과 5번은 Timer/Counter2 모드설정 해주는 것인데 여기서 우리는 Fast PWM 모드로 설정해준다.

**Table 53.** Compare Output Mode, Fast PWM Mode<sup>(1)</sup>

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on Compare Match, set OC2 at TOP (Non-Inverting).
1	1	Set OC2 on Compare Match, clear OC2 at TOP (Inverting).

1. 1(5V)에서 0(0V)로 떨어지는 설정을 해야하므로 Clear 모드로 해준다.

**Table 55. Clock Select Bit Description**

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$\text{clk}_{T2S}/(\text{No prescaling})$
0	1	0	$\text{clk}_{T2S}/8$ (From prescaler)
0	1	1	$\text{clk}_{T2S}/32$ (From prescaler)
1	0	0	$\text{clk}_{T2S}/64$ (From prescaler)
1	0	1	$\text{clk}_{T2S}/128$ (From prescaler)
1	1	0	$\text{clk}_{T2S}/256$ (From prescaler)
1	1	1	$\text{clk}_{T2S}/1024$ (From prescaler)

2. 분주비 설정은 해주는 이유가 한 클럭을 늦춰서 원하는 시간을 조정하기 위해서다. Atmega8535의 주파수는 16MHz이므로 한 주기가 62.5ns다. 그럼 1초를 만들려면 62.5ns를 16만번을 돌려야 한다. 그럼 계산하거나 프로그램 짜기 힘들어지므로 분주비 1024를 설정해서 62.5ns에 곱하면 한 주기가 64us가 나온다. 이 숫자를 15625번 돌리면 1초가 나온다. 결론적으로 분주비를 설정해주면 원하는 시간의 계산 및 프로그램 짜기가 상대적으로 수월해진다.

### Timer/Counter Register – TCNT2

Bit	7	6	5	4	3	2	1	0	
	TCNT2[7:0]								TCNT2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT2 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT2) while the counter is running, introduces a risk of missing a Compare Match between TCNT2 and the OCR2 Register.

3. TCNT는 TIMER/COUNTER의 초기값을 설정해주는 곳이다. TIMER/COUNTER2은 8bit이므로 최댓값이 255 최솟값이 0이다. 이 숫자 사이에 TCNT 값을 넣어주면 TIMER/COUNTER는 TCNT 값으로 시작해서 최댓값까지 올라가는 한 주기가 생긴다.

### Output Compare Register – OCR2

Bit	7	6	5	4	3	2	1	0	
	OCR2[7:0]								OCR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register contains an 8-bit value that is continuously compared with the counter value (TCNT2). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC2 pin.

4. OCR은 TCNT(초기값)에서 최댓값 사이에 넣어주는 숫자다. TCNT에서 최댓값으로 올라가는 도중에 OCR에 해당하는 숫자를 만나면 INTERRUPT를 발생시켜줄 수 있다. 앞에서 Clear모드로 쓴다고 설정했으므로 1(5V)에서 0(0V)로 떨어지는 INTERRUPT가 발생한다.

## Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

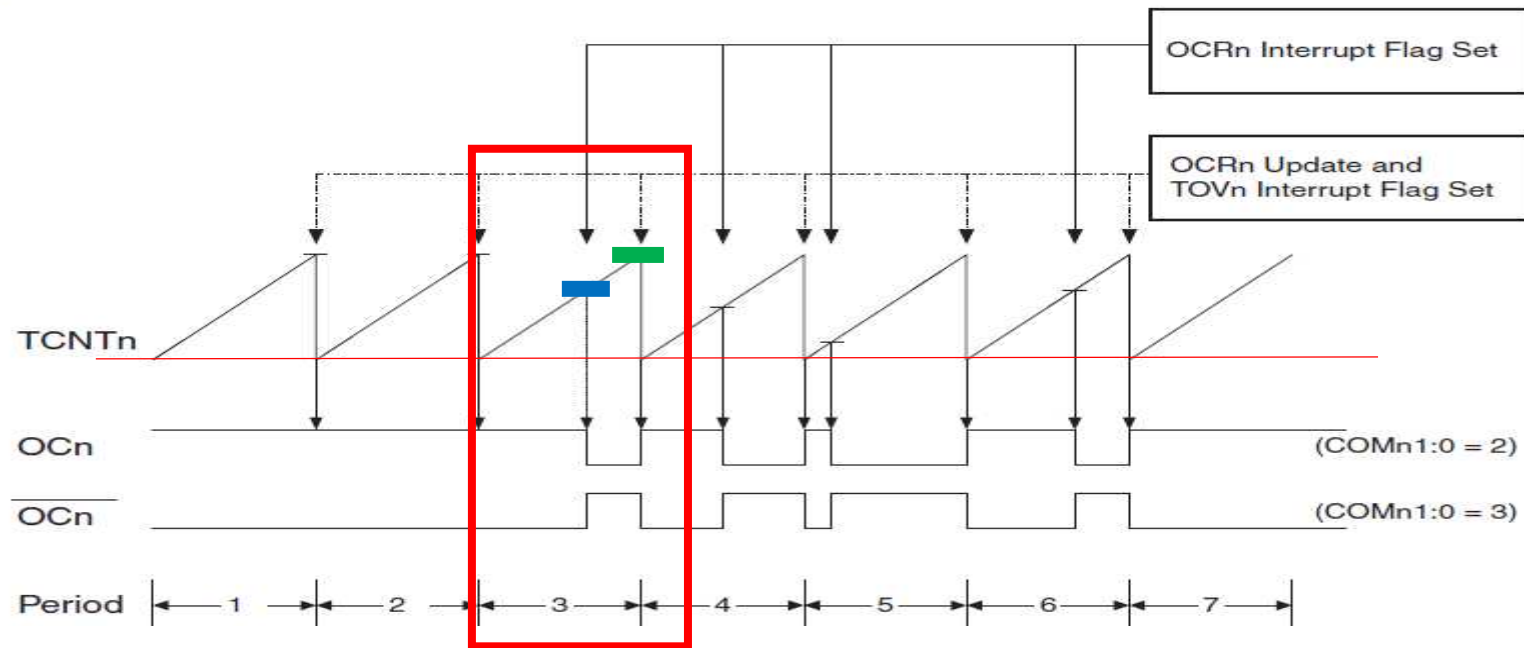
- Bit 6 – **TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is written to one and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter2 occurs (i.e., when the TOV2 bit is set in the Timer/Counter Interrupt Flag Register – TIFR).

5. TIMER/COUNTER가 최댓값을 가면 다시 TCNT(초기값)으로 떨어지게 하므로써 계속해서 주기를 발생시켜주어야 한다. 즉 최댓값에서 최솟값으로 떨어진다는 용어인 overflow를 해주는 Bit 6번 **TOIE2: Timer/Counter2 Overflow Interrupt Enable**를 1로 설정해주면 overflow가 활성화된다.



**Figure 58. Fast PWM Mode, Timing Diagram**



6. 앞에서  
설정한  
것을 이  
표로  
설명을  
하면 빨간  
줄이  
TCNT(초기  
값)이다.  
파란 줄이  
OCR이고  
초록 줄이

TOP값(최댓 값)이다. 빨간 색 상자 안을 보면 TCNT(초기값)에서 TOP값(최댓값)으로 올라가는 도중에 OCR를 만났고 앞에서 언급한 1(5V)에서 0(0V)으로 떨어지는 **Clear모드**로 인해 1(5V)에서 0(0V)으로 떨어지는 것을 확인할 수 있다. 최댓값까지 갈 시 **Timer/Counter2 Overflow Interrupt Enabled**으로 인해 다시 TCNT(초기값)로 떨어지고 0(0V)에서 1(5V)로 다시 올라간다. 이게 1024로 분주비로 설정한 64us의 한 주기다. 만약 TCNT를 131로 설정하면 TOP값(최댓값)까지 125번을 올라가므로 한 주기는  $64\mu s \times 125 = 8ms$ 가 나온다. 즉 한 주기는 8ms인 것을 알 수 있다.

## -소스 구현

```
#include <avr/io.h>
#include <avr/interrupt.h>

static int var =0; //1초 카운팅을 위한 변수
static int count =0; //LED를 왕복시키는 위한 변수

SIGNAL(TIMER2_OVF_vect){
    var++;

    if((count <=6)&&(var>=125))//LED를 왼쪽 이동 조건문
    {
        PORTC= (PORTC<<1)10x01 ;
        count++;
        var=0;
    }

    if((count>=7)&&(var>=125))//LED를 오른쪽 이동 조건문
    {
        PORTC = (PORTC>>1)10x80;
        count++;
        var=0;

        if(count>=14)//LED 왼쪽 이동 조건문으로 돌아가는 조건문
            count=0;
    }

    TCNT2=256-125; //초기값 설정
}
```

```
int main()
{
    DDRC = 0xff; //포트C를 출력으로 쓴다
    DDRD = 0xff; //TIMER2번의 OCR를 쓰기 위한 설정
    PORTC = 0xfe; //LED 한 개만 ON
    OCR2 = 200; //TIMER2번의 OCR

    TCCR2 |= (1<<WGM21)|(1<<WGM20)|(1<<COM21)|(1<<CS22)|(1<<CS21)|(1<<CS20);
    //((1<<WGM21)|(1<<WGM20) : Fast PWM 모드로 설정
    //((1<<COM21) = (1<<COM21)|(1<<COM20) : 1에서 0으로 떨어지는 clear 모드로 설정
    //((1<<CS22)|(1<<CS21)|(1<<CS20) : 분주비 1024로 설정
    TIMSK |= (1<<TOIE2); //TIMER2 오버플로우 인터럽트 ENABLE 활성화
    TCNT2 = 256-125; //초기값 설정
    sei(); //인터럽트 ENABLE

    while(1){}
```

## 1. 타이머 1초를 계산법

Atmega 8535 : 주파수 : 16MHz -> 주기 : 62.5ns

분주비 : 1024로 설정

TCNT 값 : 131로 설정(최댓값까지 125번 올라간다)

변수 var를 125이 될 때까지 LED를 이동을 안하다가 125가 되면 LED 이동한다.

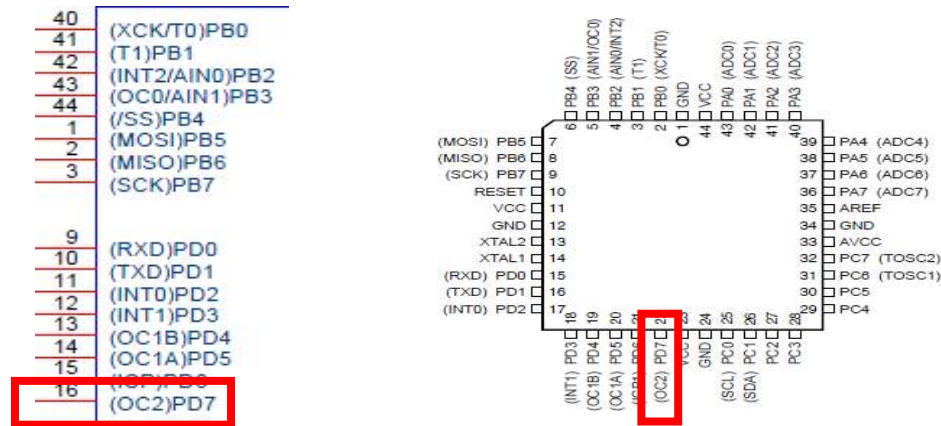
—>  $62.5\text{ns} \times 1024 \times 125 \times 125 = 1\text{s}$

## 2. 소스 설명

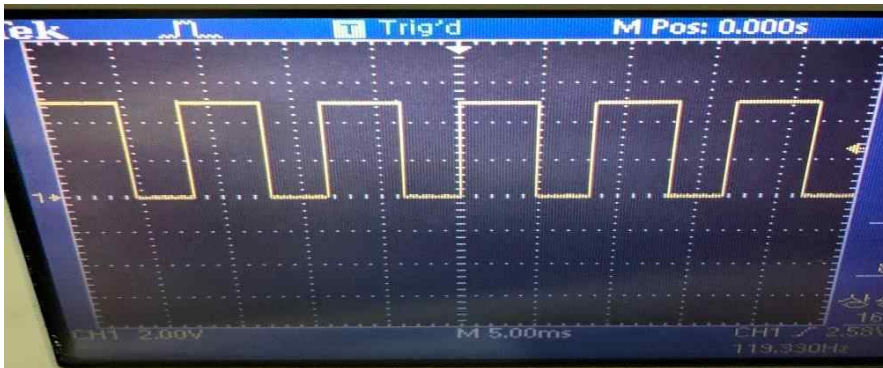
- 1) **(PORTC<<1) | 0x01**를 설정해주는 이유는 LED포트 하위비트가 왼쪽으로 이동하는데 Bit 0번 값을 알 수 없기 때문에 확실히 OFF 시켜주기 위해 설정한 것이다. 이와 똑같이 **(PORTC>>1) | 0x80**도 똑같은 맥락이다. 상위비트가 오른쪽으로 이동하는데 Bit 7번 값을 알 수 없기 때문에 확실히 OFF 하기 위한 설정이다.
- 2) **count**라는 변수로 통해 1초 지날 때마다 값을 넣어주고 7초가 되었을 때 다른 조건문으로 통해 LED를 다시 되돌아 오게 해주는 변수다. 14초가 되면 다시 0을 넣어서 LED를 다시 왼쪽으로 이동시켜주게 한다.



- TIMER/COUNTER2 작동되는지 확인하는 법



회로도나 데이터시트를 확인하면 빨간 색 상자가 TIMER/COUNTER2가 주파수가 생기는 곳이다. 오실로스코프를 작동시켜 probe를 PD7에 찍으면 아래와 같은 사진처럼 주파수가 생긴다.



(이 사진은 1초 만들기 전에 TIMER/COUNTER2가 작동이 되는지 확인하는 차 찍었던 사진이다)