
Lab 5

Timer/Counter

Timer/Counter

Timer/ Counter	Number of Bits	Timer	Counter	Compare	Capture	Waveform
0	8	O	X	O	X	O
1	16	O	O	O	O	O
2	8	O	O	O	X	O
3	16	O	O	O	O	O

Sample Code: timer0.c

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

ISR(TIMER0_OVF_vect)
{
    PORTA ^= 1<<PA0;
}

int main(void)
{
    DDRA = 1<<PA0;

    TCCR0 |= 1<<CS02 | 1<<CS01; //prescaler 256
    TIMSK |= 1<<TOIE0;

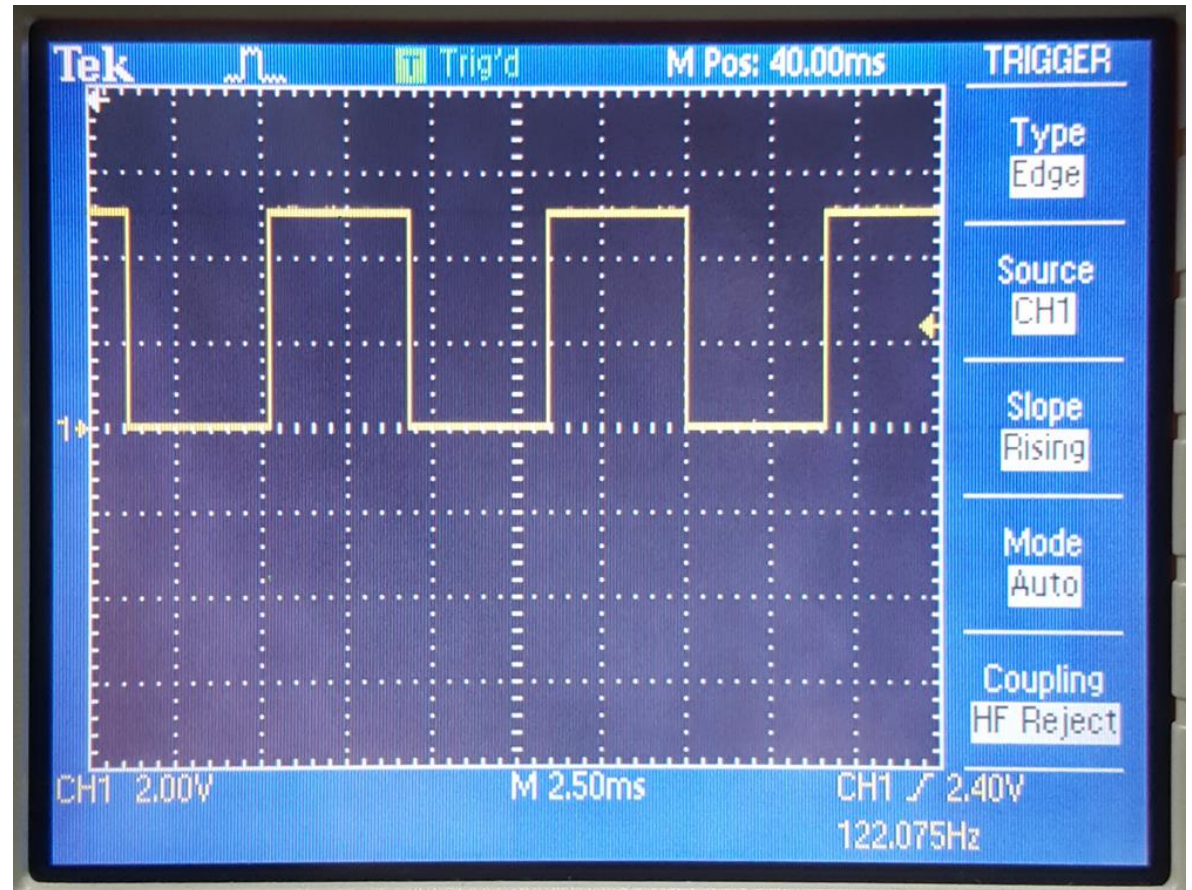
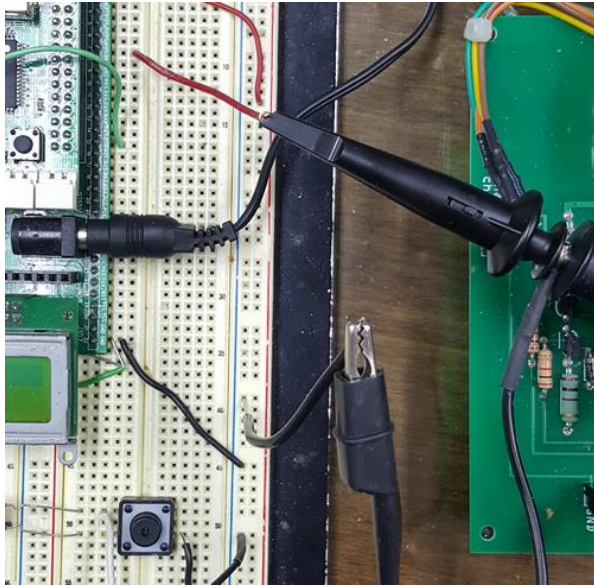
    sei(); /* enable interrupt */
    while(1);
    return 0;
}
```

Exercise 1: Square Wave

- Timer0.c 프로그램을 실행한다.
- 오실로스코프를 이용하여 PA0에서 출력되는 square wave의 주파수를 측정한다.
- Prescaler의 값을 다음과 같은 값으로 바꾼 후, 오실로스코프를 이용하여 PA0에서 출력되는 square wave의 주파수를 측정한다.

Prescaler: 8, 64, 128

- 각 prescaler의 값(8,64,128,256)에 대해서 측정된 주파수 값이 나오는 이유를 보고서에 설명한다.



16-bit Timer/Counter Overview

TCNTn : *timer/counter register*

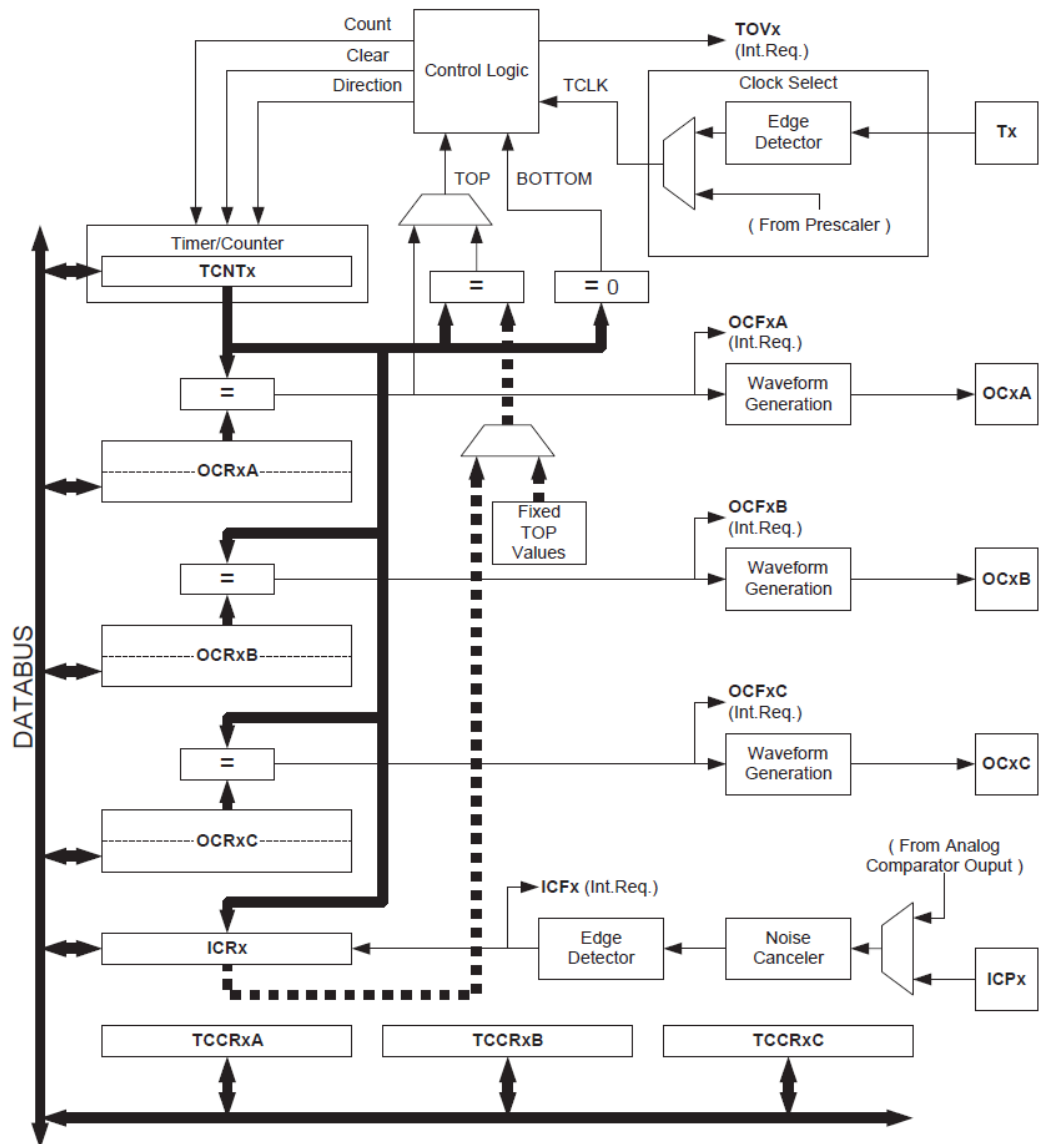
- 16-bit counter itself
- holds the present value of counter

OCRnA/B/C : *output compare register*

- always compared against TCNTn

TCCRnA/B/C : *timer/counter control register*

- determines the mode of operation



n: Timer/Counter # (1/3)

x: OCU channel # (A/B/C)

16-bit Timer/Counter1,3

Control Register A (TCCR1A, TCCR3A)

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	TCCR1A
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	7	6	5	4	3	2	1	0	
	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
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 7:6 – COMnA1:0: Compare Output Mode for Channel A
- Bit 5:4 – COMnB1:0: Compare Output Mode for Channel B
- Bit 3:2 – COMnC1:0: Compare Output Mode for Channel C

Timer/Counter1,3

Control Register A (TCCR1A, TCCR3A)

Table 61. Waveform Generation Mode Bit Description

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation ⁽¹⁾	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Timer/Counter1,3 Control Register B (TCCR1B, TCCR3B)

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 2:0 – CSn2:0: Clock Select

Table 62. Clock Select Bit Description

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}/1$ (No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

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	

Note: This register contains interrupt control bits for several Timer/Counters, but only Timer1 bits are described in this section. The remaining bits are described in their respective timer sections.

- **Bit 5 – TICIE1: Timer/Counter1, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Input Capture interrupt is enabled. The corresponding interrupt vector ([See “Interrupts” on page 60.](#)) is executed when the ICF1 flag, located in TIFR, is set.

- **Bit 4 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A Match Interrupt is enabled. The corresponding interrupt vector (see “Interrupts” on page 60) is executed when the OCF1A flag, located in TIFR, is set.

- **Bit 3 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B Match Interrupt is enabled. The corresponding interrupt vector (see “Interrupts” on page 60) is executed when the OCF1B flag, located in TIFR, is set.

- **Bit 2 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 overflow interrupt is enabled. The corresponding interrupt vector (see “Interrupts” on page 60) is executed when the TOV1 flag, located in TIFR, is set.

Extended Timer/Counter Interrupt Mask Register (ETIMSK)

Bit	7	6	5	4	3	2	1	0	
	–	–	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C	ETIMSK
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 0 – OCIE1C: Timer/Counter1, Output Compare C Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare C Match Interrupt is enabled. The corresponding interrupt vector (see “Interrupts” on page 60) is executed when the OCF1C flag, located in ETIFR, is set.

Timer/Counter Modes

- Normal Mode
- Clear Timer on Compare Match (CTC) Mode
- PWM Modes

Normal Mode

- $WGM_{n3:0}=0$
- Counting up
- No counter clear

Sample Code: timer1.c

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

int int_counter;

ISR(TIMER1_OVF_vect)
{
    int_counter++;
    if (int_counter==50)
    {
        int_counter=0;
        PORTD ^= 1<<PD4;PORTA ^= 1<<PA0;
    }
    TCNT1 = 64911; /* Timer value 10msec */
}
```

```
int main(void)
{
    cli(); /* disable interrupt */
    MCUCR = 0x00;
    MCUCR = 0x01; /* IVCE = 1 */
    MCUCR = 0x00; /* IVSEL = 0 */
    DDRD = 0xF0; /* PORT PD4~PD7 output mode */
    DDRA = 1 << PA0;

    TCCR1A = 0x00;
    TCCR1B = (1 << CS12); /* clk / 256 */
    TCCR1C = 0x00;
    TIFR = 0x00;
    ETIFR = 0x00;
    TCNT1 = 64911; /* Timer value 10msec */
    TIMSK = 1 << TOIE1; /* enable Timer1 interrupt */
    ETIMSK = 0x00;
    sei(); /* enable interrupt */
    PORTD=0xF0;
    int_counter=0;
    while(1);
}
```

Control Register Setting

TCCR1B = (1 << CS12); /* clk / 256 */

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 62. Clock Select Bit Description

CSn2	CSn1	CSn0	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	clk _{I/O} /1 (No prescaling)
0	1	0	clk _{I/O} /8 (From prescaler)
0	1	1	clk _{I/O} /64 (From prescaler)
1	0	0	clk _{I/O} /256 (From prescaler)
1	0	1	clk _{I/O} /1024 (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge
1	1	1	External clock source on Tn pin. Clock on rising edge

Timer/Counter Interrupt Mask Register

TIMSK = 1 << TOIE1; /* enable Timer1 interrupt */

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 2 – TOIE1: Timer/Counter1, Overflow Interrupt Enable

Timer Value

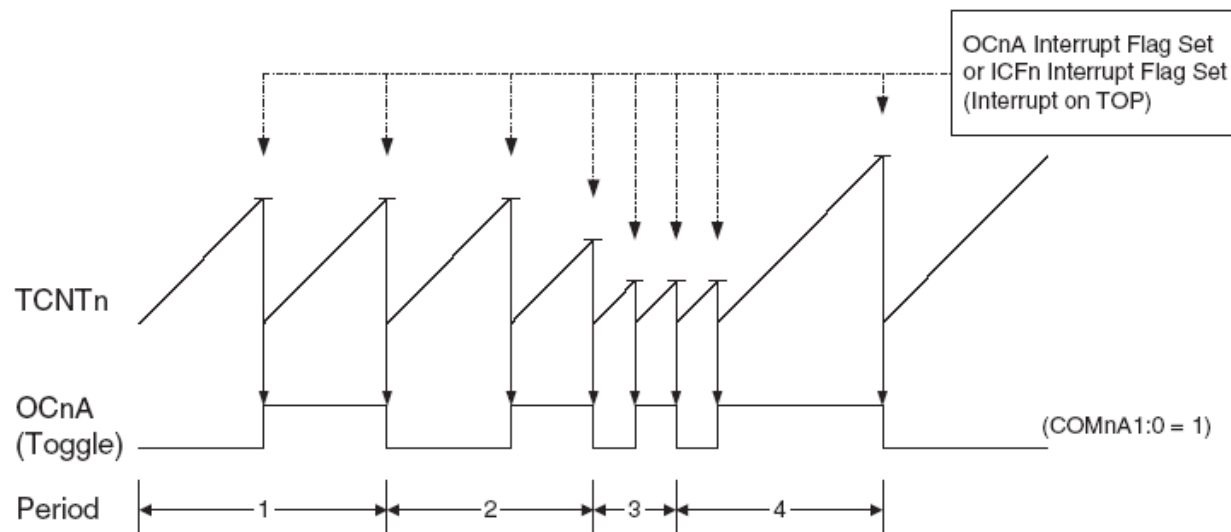
- $16\text{MHz}/256=62500\text{Hz}$
- $2^{16}=65536$
- $10\text{msec}:100\text{Hz}$
- $62500\text{Hz}/100\text{Hz}=625$
- $65536-625=64911$

```
TCNT1 = 64911; /* Timer value 10msec */
```

Clear Timer on Compare (CTC) Mode

- $WGMn3:0=4$
- $OCRnA$: compare register
- The counter is cleared to zero when matches $OCRnA$

Figure 51. CTC Mode, Timing Diagram



Sample Code: timer2.c

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

int int_counter;

ISR(TIMER1_COMPA_vect)
{
    int_counter++;
    if (int_counter==50)
    {
        int_counter=0;
        PORTD ^= 1<<PD4;PORTA ^= 1<<PA0;
    }
}
```

```
int main(void)
{
    cli(); /* disable interrupt */
    MCUCR = 0x00;
    MCUCR = 0x01; /* IVCE = 1 */
    MCUCR = 0x00; /* IVSEL = 0 */
    DDRD = 0xF0; /* PORT PD4~PD7 output mode */
    DDRA = 1 << PA0;

    TCCR1A = 0x00;
    TCCR1B = (1 << CS12) | (1 << WGM12); /* clk / 256 CTC Mode */
    TCCR1C = 0x00;
    TIFR = 0x00;
    ETIFR = 0x00;
    OCR1A = 624;
    TIMSK = 1 << OCIE1A; /* enable Timer1 Compare Match A interrupt */
    ETIMSK = 0x00;
    sei(); /* enable interrupt */
    PORTD=0xF0;
    int_counter=0;
    while(1);
}
```

Control Register Setting

TCCR1B = (1 << CS12) | (1<< WGM12); /* clk / 256 CTC Mode */

Table 61. Waveform Generation Mode Bit Description

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation ⁽¹⁾	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Interrupt Mask Register

TIMSK = 1 << OCIE1A; /* enable Timer1 Compare Match A interrupt */

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 4 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable

Interrupt Vector

SIGNAL(SIG_OUTPUT_COMPARE1A)

{

lom128.h

```
/* Timer/Counter1 Compare Match A */
```

```
#define TIMER1_COMPA_vect
```

_VECTOR(12)

```
#define SIG_OUTPUT_COMPARE1A
```

_VECTOR(12)

Timer Value

- $16\text{MHz}/256=62500\text{Hz}$
- $2^{16}=65536$
- $10\text{msec}:100\text{Hz}$
- $62500\text{Hz}/100\text{Hz}=625$
- 0~624: 625 pulses

OCR1A = 624; /* Timer value 10msec */

Exercise2: Stop Watch(1 단계)

- 1/100 초 단위로 측정할 수 있는 stop watch 를 만든다.
- 분,초,1/100초를 각 2자리로 character LCD에 다음과 같이 나타낸다. 분과 초는 0부터 59까지, 1/100초는 0부터 99까지 나타낸다.

12:34:56

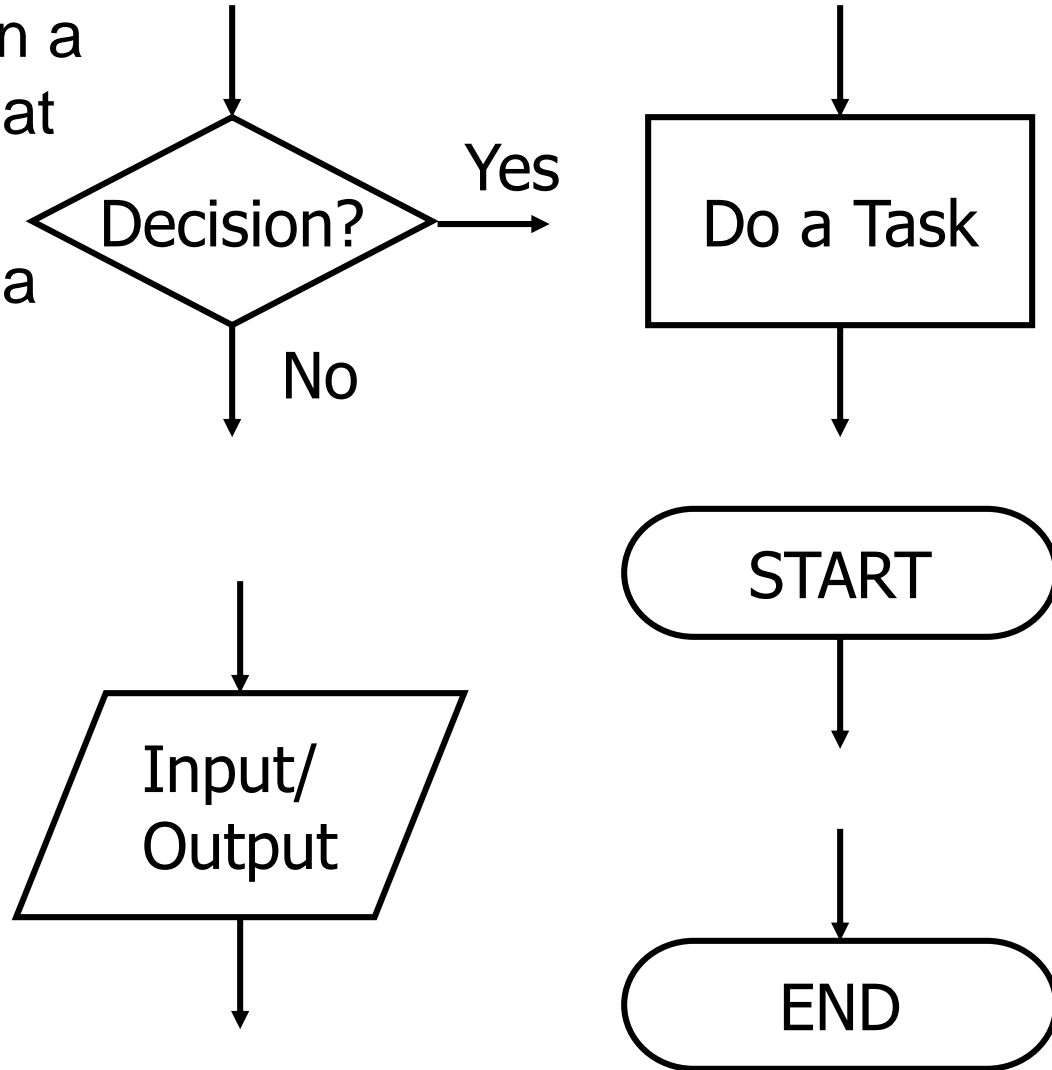
- 프로그램이 시작되면 00:00:00 부터 시작하여 즉시 stop watch가 동작 되도록 한다.

Exercise3: Stop Watch(2단계)

- 앞의 Exercise에서 만든 stop watch에 버튼 입력 기능을 추가한다.
- 처음 시작은 00:00:00을 디스플레이하고 버튼 키 입력을 기다린다.
- 버튼을 누르면 시작하고, 다시 버튼을 누르면 정지 한다. 정지 상태에서 버튼을 누르면 시작한다. 시간은 정지 상태 값에 이어서 나타낸다. 다시 버튼을 누르면 정지한다.
- 이 동작을 무한 반복한다.

Flowchart

- Shows flow of control in a processing activity (what gets done)
- Used to show steps in a process, including decision-making
- Does not scale well: becomes confusing if larger than a page



Flowchart Example

