

Materi Lab 8

Timer/Counter Interrupt

Silahkan download dan buka template Lab 8 dari Scele untuk pengerjaan lab minggu ini. Di dalamnya terdapat 2 file, yaitu “**Lab 8 Example.xml**” dan “**Lab 8 Example.asm**”.

Timer/Counter merupakan suatu jenis Interrupt, di mana AVR akan meng-interrupt program ketika suatu event timer terjadi. Jika timer/counter mencapai suatu kondisi tertentu, program akan melakukan interrupt. Timer beroperasi secara independen dari eksekusi program.

Terdapat beberapa jenis Timer/Counter Interrupt:

Addr	Source	Definition
\$003	Timer1 Capt	Timer/Counter ¹ Capture Event
\$004	Timer1 CompA	Timer/Counter ¹ Compare Match A
\$005	Timer1 CompB	Timer/Counter ¹ Compare Match B
\$006	Timer1 Ovf	Timer/Counter ¹ Overflow
\$007	Timero Ovf	Timer/Counter ⁰ Overflow
\$00E	Timero Comp	Timer/Counter ⁰ Compare

Apa bedanya Timer0 dan Timer1?

- Timer0 adalah timer 8 bit, sehingga counter-nya hanya bisa menghitung dari 0 sampai 255.
- Timer1 adalah timer 16 bit, sehingga counter-nya bisa menghitung lebih banyak yaitu dari 0 sampai 65 535.

Apa bedanya event Timer Overflow, Compare dan Capture?

- Event Timer Overflow berarti bahwa counter-nya telah menghitung sampai max value, dan akan di-reset ke 0 pada timer clock cycle berikutnya. Ketika timer overflow, interrupt akan terjadi.
- Event Timer Compare berarti bahwa value dalam counter timer akan dibandingkan dengan value Output Compare Register (OCRx). Jika valuenya sama, interrupt akan terjadi.
- Event Timer Capture berarti ada perubahan signal pada AVR input pin. Perubahan signal ini mengakibatkan value counter timer value di-read dan disimpan dalam Input Capture Register (ICRx), lalu interrupt akan terjadi.

Perhatikan kodingan dari “**Lab 8 Example.asm**”. Pada template tersebut, kita menggunakan TIMER0 OVF (Timer/Counter0 Overflow), tapi Anda boleh mengubahnya sesuka Anda. Anda boleh memakai Timer1, dan boleh memakai Overflow atau Compare. Tapi untuk tutorial lab ini hanya akan dijelaskan mengenai Timer Overflow.

```
.org $07      ; When Timer1 overflows, jump to ISR_TOV0  
rjmp ISR_TOV0
```

Di bagian “**RESET and INTERRUPT VECTORS**”, kita harus setup interrupt vector Timer/Counter yang ingin dipakai dulu. Pada template, kita menggunakan TIMER1 OVF (Timer/Counter1 Overflow), yang memiliki interrupt vector \$06.

Berikut adalah cara meng-activate Timer/Counter Overflow Interrupt:

1. Setting the values of TCCR0/TCCR1A/TCCR1B
2. Setting the values of TIFR
3. Setting the clock in TIMSK
4. Activate global Interrupt using SEI instruction

```
; Setup Overflow Timer0  
SET_TIMER:  
    ; Timer speed = clock/8 (set CS01 in TCCR0)  
    ldi r16, (1<<CS01)  
    out TCCR0, r16  
  
    ; Execute an internal interrupt when Timer0 overflows  
    ldi r16, (1<<TOV0)  
    out TIFR, r16  
  
    ; Set Timer0 overflow as the timer  
    ldi r16, (1<<TOIE0)  
    out TIMSK, r16  
  
    ; Set global interrupt flag  
    sei
```

TCCR merupakan Timer/Counter Control Register yang digunakan untuk mengatur mode clock cycle settings untuk timer, seperti mengatur prescaler dan lain-lain. Dengan register ini, kita bisa mengatur 2 hal:

1. Frekuensi dari Clock Source melalui CS02, CS01, CS00 bits.
2. Mode dari timer. Untuk template ini kita menggunakan normal mode di mana timer menghitung dari 0 sampai max value (255).

Terdapat 3 jenis TCCR, yaitu TCCR0/TCCR1A/TCCR1B. TCCR0 digunakan untuk Timer0, sementara TCCR1A dan TCCR1B digunakan untuk Timer1. Template ini menggunakan Timer0, sehingga register yang dipakai adalah TCCR0.

Berikut adalah 8 bit dalam TCCR0:

TCCR0							
D7	D6	D5	D4	D3	D2	D1	D0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

Berikut adalah 3 bit yang mengatur prescaler dan frekuensi clock Timer0:

D2	D1	D0	Clock Source
CS02	CS01	CS00	Freq
0	0	0	No Clock (Stopped)
0	0	1	Clk
0	1	0	Clk/8
0	1	1	Clk/64
1	0	0	Clk/256
1	0	1	Clk/1024
1	1	0	Clk/T0-Falling edge
1	1	1	Clk/T0-Rising Edge

Berikut adalah 2 bit yang digunakan untuk mengatur mode timer:

D6	D3	PWM
WGM00	WGM01	Mode
0	0	Normal
0	1	CTC (Clear timer on compare match)
1	0	PWM (Phase correct)
1	1	Fast PWM

Jika menggunakan Timer1, karena ada 16-bit, jadi harus dibagi menjadi 2 bagian dari TCCR, yaitu TCCR1A dan TCCR1B. Berikut adalah 8 bit TCCR1A:

TCCR1A							
D7	D6	D5	D4	D3	D2	D1	D0
COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10

Dan 8 bit TCCR1B:

TCCR1B							
D7	D6	D5	D4	D3	D2	D1	D0
ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

Berikut adalah 3 bit yang mengatur prescaler dan frekuensi clock Timer1:

D2	D1	D0	Clock Source
CS12	CS11	CS10	Freq
0	0	0	No Clock (Stopped)
0	0	1	Clk

0	1	0	Clk/8
0	1	1	Clk/64
1	0	0	Clk/256
1	0	1	Clk/1024
1	1	0	Clk/T1-Falling edge
1	1	1	Clk/T1-Rising Edge

Template “**Lab 8 Example.asm**” ini menggunakan Timer0 prescaler 8, berarti bit CS02-CS01-CS00 nya harus diubah menjadi 0-1-0. Kalian boleh menggunakan prescaler berapa pun, dari 8 sampai 1024. Lebih besar prescaler yang dipakai, lebih lambat timer akan berjalan.

“**ldi r16, (1<<CS01)**” akan me-set value dalam bit CS01 menjadi 1, lalu me-load nya ke dalam temporary register r16. “**out TCCR0, r16**” lalu akan menyimpan data dari temporary register r16 tersebut dalam TCCR0.

- **Timer/Counter Interrupt Mask Register (TIMSK)** → penentu jenis timer
- **Timer/Counter Interrupt Flag Register (TIFR)** → penanda timer event

Timer Interrupt Mask Register (TIMSK) dan Timer Interrupt Flag Register (TIFR) digunakan untuk mengatur interrupt mana yang "valid" untuk digunakan, dengan mengatur bit nya dalam TIMSK dan menentukan interrupt mana yang sedang terjadi (TIFR).

Berikut adalah bit yang mengatur jenis interrupt dalam TIMSK:

TIMSK							
D7	D6	D5	D4	D3	D2	D1	D0
-	-	TICIE1	OCIE1A	OCIE1B	TOIE1	-	-

- **TOIE1: Timer Overflow Interrupt Enable (Timer 1)** → Jika bit ini di-set dan jika global interrupts di-enable, program pointer akan jump ke Timer Overflow 1 interrupt vector begitu terjadi Timer 1 Overflow.

- **OCIE1A: Output Compare Interrupt Enable 1 A** → Jika bit ini di-set and jika global Interrupts di-enable, program pointer akan jump ke Output Compare A Interrupt vector begitu terjadi Compare Match event.
- **TICIE1: Timer 1 Input Capture Interrupt Enable** → Jika bit ini di-set dan jika global Interrupts di-enable, program pointer akan jump ke Input Capture Interrupt vector begitu terjadi Input Capture event.
- **TOIE0: Timer Overflow Interrupt Enable (Timer 0)** → Sama seperti TOIE1, tapi untuk 8-bit Timer 0.

Berikut adalah bit yang menyimpan Interrupt Flag dalam register TIFR:

TIFR							
D7	D6	D5	D4	D3	D2	D1	D0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0

- **Bit 7 - OCF2:** Timer/Counter2 Output Compare Match Interrupt Enable Flag
- **Bit 6 - TOV2:** Timer/Counter2 Overflow Interrupt Enable Flag
- **Bit 5 - ICF1:** Timer/Counter1 Input Capture Interrupt Enable Flag
- **Bit 4 - OCF1A:** Timer/Counter1 Output CompareA Match Interrupt Enable Flag
- **Bit 3 - OCF1B:** Timer/Counter1 Output CompareB Match Interrupt Enable Flag
- **Bit 2 - TOV1:** Timer/Counter1 Overflow Interrupt Enable Flag
- **Bit 1 - OCF0:** Timer/Counter0 Output Compare Match Interrupt Enable Flag
- **Bit 0 - TOV0:** Timer/Counter0 Overflow Interrupt Enable Flag

TIFR tidak terlalu diperlukan untuk mengontrol dan menggunakan timer. Register tersebut menyimpan Timer Interrupt Flags yang sesuai dengan enable bits in TIMSK. Jika sebuah Interrupt tidak di-enable, kode Anda bisa mengecek TIFR untuk menentukan apakah sebuah interrupt sudah terjadi, lalu clear interrupt flags. Clearing interrupt flags biasanya dilakukan dengan memasukkan value 1 ke dalamnya.

Dalam template “**Lab 8 Example.asm**” ini, kita menggunakan Timer0 Overflow, sehingga kita lakukan kode berikut untuk setup TIFR:

```
; Execute an internal interrupt when Timer0 overflows
ldi r16, (1<<TOV0)
out TIFR, r16
```

“**ldi r16, (1<<TOV0)**” akan me-set value bit TOV0 menjadi 1 lalu di-load ke temporary register r16, dan “**out TIFR, r16**” akan meng-output value tersebut ke TIFR. Sehingga flag untuk Timer/Counter0 Overflow Interrupt akan disiapkan.

Kita lalu melakukan kode berikut untuk setup TIMSK:

```
; Set Timer0 overflow as the timer  
ldi r16, (1<<TOIE0)  
out TIMSK, r16
```

“**ldi r16, (1<<TOIE0)**” akan me-set value bit TOIE0 menjadi 1 lalu di-load ke temporary register r16, dan “**out TIMSK, r16**” akan meng-output value tersebut ke TIMSK. Sehingga jenis timer yang dipakai, yaitu Timer0 Overflow Interrupt, akan di-enable.

Setelah semua itu, kita harus memanggil command **SEI**, yang akan meng-enable global interrupt. Command tersebut akan set value Global Interrupt flag (I) dalam SREG (status register) menjadi 1.

Berikut adalah program yang akan di-execute begitu timer event terjadi. Dalam contoh template “**Lab 8 Example.asm**”, nama fungsinya adalah ISR_TOV0:

```
; Program executed on timer interrupt  
ISR_TOV0:  
    push r16  
    in r16, SREG  
    push r16  
    in r16, PORTB      ; Read Port B  
  
    com r16            ; Invert value in Port B  
  
    out PORTB, r16     ; Write to Port B  
    pop r16  
    out SREG, r16  
    pop r16  
  
    reti
```

Silahkan buka dan coba jalankan, lalu pelajari isi dari file **Contoh Timer-Counter** di Scele. Di dalamnya terdapat latihan atau contoh penggunaan berbagai jenis timer berbeda, mulai dari Timer1 Overflow sampai Compare juga. Jika ingin menggunakan jenis timer yang berbeda dari template, silahkan pakai contoh dari file-file tersebut.