7

Timer/Counters

In veel applicaties moet even gewacht worden zodat bijvoorbeeld elke minuut een meting kan worden gedaan. Het meten van deze tijd (wachten) is op te lossen door middel van de bekende wachtlus (zie paragraaf **??**). De processor kan dan echter geen andere taken uitvoeren en dat is in veel situaties wél gewenst. Denk hierbij aan Real Time systemen en besturingssystemen. Beter is deze tijd hardwarematig te meten. Een *timer* kan dan uitkomst bieden. Een timer is niets anders dan een teller die op iedere klokpuls, bijvoorbeeld van de systeemklok, wordt verhoogd. Na een bepaald aantal klokpulsen heeft de timer de hoogste stand bereikt en begint weer opnieuw. Er wordt dan een signaal aan de processor afgegeven. Dit signaal genereert, indien geactiveerd, een interrupt, waardoor de processor zijn tijdelijk hoofdprogramma verlaat en een Interrupt Service Routine gaat uitvoeren.

Dezelfde schakeling kan ook gebruikt worden om extern aangeboden pulsen te tellen. Denk hierbij aan het tellen van het aantal fietsen dat over een fietspad rijdt. We noemen zo'n teller een *counter*. Technisch gezien is er geen verschil tussen de hardware van een timer en een counter. Het verschil is dus alleen wat de klokbron is.

De ATmega32 heeft drie timer/counters: een 8-bits Timer/Counter 0 (T/C0), een 16-bits Timer/Counter 1 (T/C1) en een 8 bits Timer/Counter 2 (T/C2). T/C0 en T/C1 hebben gemeen dat ze kunnen worden geklokt op een intern kloksignaal en op externe pulsen. T/C2 kan worden geklokt op een intern kloksignaal en een externe oscillator. Alle timer/counters hebben diverse interrupt-mogelijkheden en zogenoemde pulsbreedte gemoduleerde signalen genereren (PWM-signalen). Het is goed om te weten dat de alle timers min of meer dezelfde mogelijkheden hebben.

In dit hoofdstuk behandelen we de drie timer/counters. Van elke timer/counter worden de instelmogelijkheden behandeld, zoals de werkmodi, interruptgeneratie en signaalvormgeneratie. We laten de mogelijkheden zien aan de hand van een aantal programmavoorbeelden, zowel in assembler als C.

7.1 Timer/Counter 0

Timer/Counter 0 is de eerste van de drie timer/counters van de ATmega32 die besproken worden. Het is een 8-bits timer/counter wat inhoudt dat het kan tellen tussen 0 en 255. Als de timer/counter intern opgewekte klokpulsen telt, spreken we van een timer. Bij het tellen van extern opgewekte klokpulsen spreken we van een counter.

7.1.1 Werkmodi

Timer/Counter 0 heeft vier werkmodi:

- Normal mode: Timer/Counter 0 telt van 0 tot en met 255 en begint dan weer bij 0. Zowel interne ale externe klokpulsen kunnen geteld worden. Deze modus wordt besproken in paragraaf 7.1.3.
- CTC mode: Timer/Counter telt van 0 tot een van te voren opgegeven maximale waarde en begint dan weer op 0. Zowel interne als externe klokpulsen kunnen geteld worden. Deze modus wordt besproken in paragraaf 7.1.4.
- Fast PWM mode: Timer/Counter telt van 0 tot 255 en begint dan weer op 0. Deze modus wordt voornamelijk gebruikt om een signaalvorm op één van de externe pinnen van de ATmega32 te genereren. Deze modus wordt besproken in paragraaf 7.1.6.
- Phase correct PWM mode: Timer/Counter telt omhoog van 0 tot en met 255 en telt dan weer omlaag tot en met 0. Deze modus wordt voornamelijk gebruikt om een signaalvorm op één van de externe pinnen van de ATmega32 te genereren. Deze modus wordt besproken in paragraaf 7.1.7.

In alle werkmodi is het mogelijk om interrupts te genereren. Elke interrupt is gekoppeld aan een eigen interruptvector.

Bij het programmeren van Timer/Counter 0 worden diverse I/O-registers gebruikt. Ten eerste is er de Timer/Counter Register (TCNT0). Hierin is de huidige telwaarde opgeslagen. Ten tweede is er het Output Compare Register (OCR0). Dit register wordt gebruikt bij de CTC-modus en de beide PWM-modi. Om T/C0 van de juiste instellingen te voorzien, is er het Timer/Counter Control Register (TCCR0). Hiermee stellen we onder andere de werkmodus in.

De vier werkmodi kunnen worden ingesteld met de WGM01- en WGM00-bits in het TCCR0-register. WGM staat voor *Waveform Generation Mode*. De indeling van deze twee bits is te zien in figuur 7.1. Merk op dat de bits niet netjes naast elkaar in het register staan. Let daar op bij het programmeren van dit register.

7	6	5	4	3	2	1	0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figuur 7.1: *De WGM01- en WGM00-bits in het TCCR0 register.*

In tabel 7.1 is te zien hoe de bits ingesteld moeten worden voor de diverse werkmodi.

Tabel 7.1: *De vier werkmodi van T/C0.*

WGM01	WGM00	Modus
0	0	Normal mode
0	1	Phase Correct PWM mode
1	0	CTC mode
1	1	Fast PWM mode

7.1.2 Klokbronnen

Al eerder is aangegeven dat T/C0 kan tellen op een intern kloksignaal of op een extern kloksignaal. Aangezien de interne klokfrequentie vrij groot is en het aantal timer-bits klein, zal T/C0 zeer snel op het maximum zitten en genereert zodanig heel vaak een *roll-over*. Vaak willen we de roll over op een veel lagere frequentie laten plaatsvinden. Een *prescaler* biedt dan uitkomst. Een prescaler is een digitale schakeling die het aangeboden kloksignaal 'deelt' door een getal, meestal een macht van 2. Hierdoor telt T/C0 op een (veel) lagere frequentie. De prescaler van T/C0 heeft vier mogelijke delers: 8, 64, 256 en 1024. Staat de prescaler bijvoorbeeld op 1024 ingesteld dan wordt TCNT0 om de 1024 klokpulsen met één verhoogd. Dat betekent dat er eens in de $1024 \times 256 = 262144$ klokpulsen een roll-over plaatsvindt.

Naast interne klokpulsen kan T/C0 ook externe klokpulsen tellen. Eigenlijk telt TC/0 op *flanken*. Zo kan T/C0 op opgaande of neergaande flanken tellen. Het externe kloksignaal moet worden aangeboden aan ingang T0 (pin 1 op een 40-pin PDIP). De prescaler kan niet ingeschakeld worden bij gebruik van externe klokpulsen.

Natuurlijk kan T/C0 ook "uit" staan. T/C0 is dan gestopt met tellen. Dit is de situatie direct na een reset. Bij elkaar levert dit acht mogelijke klokbronnen op. Ook "uit" noemen we een klokbron en de teller kan ook direct met de systeemklok gevoed worden. Dit is met drie bits aan te geven. Dit zijn de CS0-bits in register TCCR0. De indeling van de CS0-bits zijn te vinden in figuur 7.2. De betekening van de bitcombinaties zijn te vinden in tabel 7.2.

7	6	5	4	3	2	1	0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

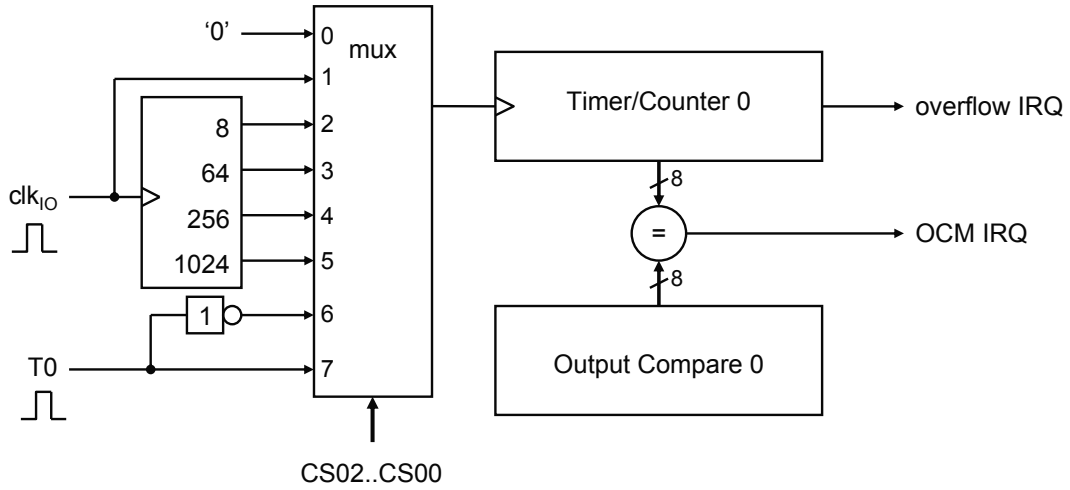
Figuur 7.2: *De CS02-, CS01 en CS00-bits in het TCCR0-register.*

In figuur 7.3 is een vereenvoudigde weergave te zien van T/C0 met de klokbronselectie en normal- en CTC-modi. Signaal CLK_{IO} is het kloksignaal dat afkomstig van de *Clock Control Unit*. De frequentie van deze klok is identiek aan de systeemfrequentie, maar kan afgeschakeld worden door de processor in slaapstand te brengen. De klok wordt

Tabel 7.2: *Klokbronselectie voor Timer/Counter 0.*

CS02	CS01	CS00	operatie
0	0	0	geen werking, klok is gestopt
0	0	1	$CLK_{IO}/1$, geen prescaler
0	1	0	$CLK_{IO}/8$, van prescaler
0	1	1	$CLK_{IO}/64$, van prescaler
1	0	0	$CLK_{IO}/256$, van prescaler
1	0	1	$CLK_{IO}/1024$, van prescaler
1	1	0	externe klokbron op de T0-pin, neergaande flank
1	1	1	externe klokbron op de T0-pin, opgaande flank

aan de prescaler aangeboden die het kloksignaal “deelt” door een aantal vaste waarden. Als klokbron kan ook een extern signaal dienen dat van de T0-pin afkomstig is. Er kan zowel op de opgaande als neergaande flank geklokt worden. Al deze bronnen worden aangeboden aan een multiplexer die één van de klokbronnen selecteert.



Figuur 7.3: *Eenvoudige voorstelling van de normal en CTC-modus van Timer/Counter 0.*

De code in listing 7.1 zorgt ervoor dat T/C0 ingesteld wordt in CTC-modus met een prescalerwaarde van 1024 als klokbron.

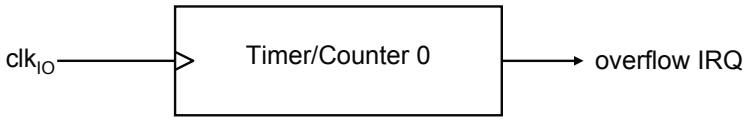
```
1       ldi   r16,0b00001101   ; select CTC mode, prescaler 1024
2       out   TCCR0,r16
```

Listing 7.1: *Selectie van CTC-modus en prescaler 1024.*

7.1.3 Normal mode

Een simpele voorstelling van T/C0 is te zien in figuur 7.4. T/C0 wordt geklokt door het I/O-kloksignaal clk_{IO}. Op elke opgaande flank van de klok wordt de telstand van de

teller met één verhoogd. De telstand wordt bijgehouden in het Timer/Counter Register (TCNT0). Als TCNT0 op 255 staat, is de volgende telstand 0. T/C0 heeft dan een roll-over gemaakt. We spreken dan van een *timer overflow*. Het is mogelijk om dan een interrupt te genereren. Merk op dat kloksignaal clk_{IO} uitgezet kan worden door de ATmega32 is slaapstand te brengen (sleep mode).



Figuur 7.4: *Eenvoudige voorstelling van de Normal modus van Timer/Counter 0.*

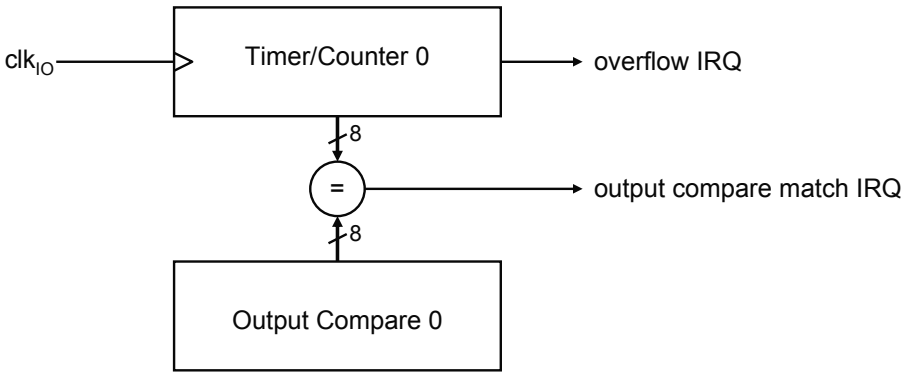
Het TCNT0-register is door de gebruiker te lezen en te schrijven. Als de software een waarde schrijft in TCNT0, dan gaat T/C0 verder tellen vanaf deze waarde. Het is dan eenvoudig om te bepalen of de teller een bepaalde waarde heeft. In listing 7.2 is te zien hoe TCNT0 eerst geladen wordt met 0 en dan gewacht wordt totdat TCNT0 de waarde 123 heeft bereikt.

```
1       ldi  r16,0         ; load TCNT0 with 0
2       out  TCNT0,r16
3
4 wait: in   r16,TCNT0     ; wait for TCNT0 to reach 123
5       cpi  r16,123
6       brne wait
```

Listing 7.2: *Lezen van de tellerwaarde van T/C0.*

7.1.4 Clear Timer on Compare Match Mode (CTC)

In deze modus telt T/C0 vanaf 0 tot en met een opgegeven maximum en begint dan weer op 0. Het opgegeven maximum wordt aangegeven door het Output Compare Register (OCR0). Dit register moet voor aanvang van het tellen door de gebruiker geladen worden met een waarde. Het is mogelijk om op een *compare match* een interrupt te genereren. Een eenvoudige voorstelling van de CTC-modus is te zien in figuur 7.5.



Figuur 7.5: *Eenvoudige voorstelling van de CTC-modus van Timer/Counter 0.*

Zoals te zien is, kan in de CTC-modus een interrupt gegenereerd worden als de waarde in het TCNT0-register en OCR0-register gelijk zijn. Dit wordt de Output Compare Match interrupt genoemd. Merk op dat de interrupt pas gegenereerd wordt als TCNT0 weer op 0 staat.

In listing 7.3 is te zien hoe het OCR0-register wordt ingesteld op de waarde 143. Tevens wordt de CTC-modus geselecteerd in het TCCR0-register. Merk op dat de teller na 144 klokpulsen weer op 0 staat. De teller telt immers van 0 t/m 143.

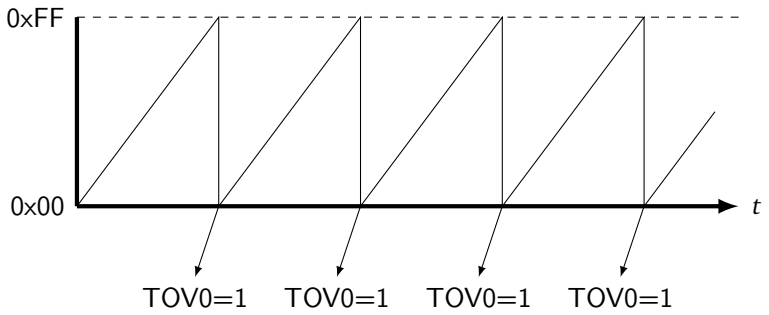
```
1       ldi   r16,143          ; load OCR0 with 143
2       out   OCR0,r16
3
4       ldi   r16,0b00001000   ; select CTC mode
5       out   TCCR0,r16
```

Listing 7.3: *Instellen van het OCR0-register van T/C0*

7.1.5 Interrupts: Timer overflow en Output Compare Match overflow

Een taak die vaak voorkomt in de ATmega32, is het periodiek uitvoeren van een stukje code. Te denken valt aan een temperatuurmeting die elke minuut moet plaatsvinden. Natuurlijk kunnen we een wachtlus gebruiken, maar de controller kan dan niets anders doen. We kunnen T/C0 gebruiken om een stukje tijd af te meten.

In de normal-modus telt T/C0 van 0 t/m 255. Als T/C0 een roll-over heeft, dus van telstand 255 naar telstand 0 gaat, wordt op telstand 0 de Timer/Counter 0 Overflow Flag (TOV0) op 1 gezet. In figuur 7.6 is dit grafisch uitgebeeld. Elke keer als de teller weer op 0 staat, wordt de TOV-vlag op 1 gezet. De TOV0-vlag is terug te vinden in het Timer/Counter Interrupt Flag Register (TIFR).



Figuur 7.6: *Tijddiagram van de TCNT0-waarde met de momenten van een timer overflow.*

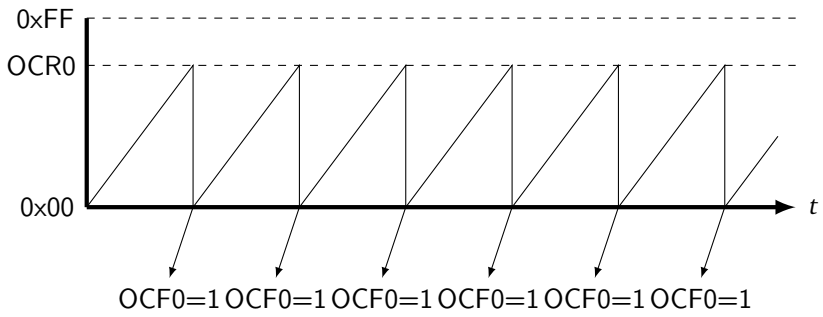
De TOV0-vlag is op twee manieren op 0 te krijgen: door hardware bij het uitvoeren van een timer overflow interrupt en door software door het schrijven van een 1 in het TIFR-register. De frequentie van de timer overflow is te berekenen met de formule:

$$f_{ov} = \frac{f_{clk}}{\text{prescaler} \times 256} \tag{7.1}$$

Bij een frequentie van 3,6864 MHz en een prescalerwaarde van 1024 is de overflow-frequentie dus 14,0625 Hz. Merk op dat bij deze klokfrequentie de overflow-frequentie niet lager te realiseren is.

7	6	5	4	3	2	1	0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figuur 7.7: *Het TIFR-register.*



Figuur 7.8: *Tijddiagram van de TCNT0-waarde met de momenten van een output compare match overflow.*

7.1.6 Fast-PWM mode

Volgt nog.

7.1.7 Phase Correct PWM mode

Volgt nog.

7.2 De registers van Timer/Counter 0

De besturing van Timer/Counter 0 is vastgelegd in het TCCR0-register, zie figuur 7.9. Dit register kan zowel gelezen als geschreven worden.

7	6	5	4	3	2	1	0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figuur 7.9: *TCCR0 register.*

Bit 7 – FOC0: Force Output Compare
De FOC0-bit is alleen actief als we de WGM0*x*-bits non-PWM-modus specificeren. Bij het specificeren van één van de twee PWM-modi moet deze bit altijd als 0 geschreven worden.

Als deze bit met een 1 geladen wordt, zal de signaalvormgenerator een directe compare match krijgen en zal de OC0-uitgang veranderen zoals is vastgelegd in de COM0*x*-bits.

Bits 3,6 – WGM0*x* signaalvormgeneratorbits
Deze bits bepalen de telvolgorde van de Timer/Counter, de maximale telwaarde en het type van de signaalvormgenerator. Zie tabel 7.3.

Tabel 7.3: *Signaalvormgeneratie*

WGM01	WGM00	Modus	Top	Update OCR0	TOV0-vlag
0	0	Normaal	255	direct	255
0	1	Phase Correct PWM	255	255	0
1	0	CTC	OCR0	direct	255
1	1	Fast PWM	255	0	255

Bits 5,4 – COM0*x* Compare Match Output Mode
Deze bits bepalen het gedrag van de Output Compare pin (OC0). Merk op dat de Data Direction (DDR) bit van de OC0-pin op 1 gezet moet worden om de uitgang te activeren. Tabel 7.4 geeft de functionaliteit van de COM0*x*-bits weer in non-PWM-modus (Normal en CTC).

Tabel 7.4: *Compare Match uitgang, non-PWM-modus.*

COM01	COM00	Beschrijving
0	0	Normale poortwerking, OC0 is gedeactiveerd
0	1	Verander OC0 op een compare match (toggle)
1	0	Zet OC0 op 0 op een compare match
1	1	Zet OC0 op 1 op een compare match

Tabel 7.5 geeft de functionaliteit van de COM0*x*-bits weer in Fast PWM-modus.

Tabel 7.5: *Compare Match uitgang, Fast PWM-modus.*

COM01	COM00	Beschrijving
0	0	Normale poortwerking, OC0 is gedeactiveerd
0	1	Niet gebruikt
1	0	Zet OC0 op 0 op een compare match, zet op 1 op TCNT = 255
1	1	Zet OC0 op 1 op een compare match, zet op 0 op TCNT = 255

Tabel 7.6 geeft de functionaliteit van de COM0*x*-bits weer in Phase Correct PWM-modus.

Bits 2,1,0 – CS0*x* klokselectiebits
In tabel 7.7 is te zien hoe een klok geselecteerd kan worden.

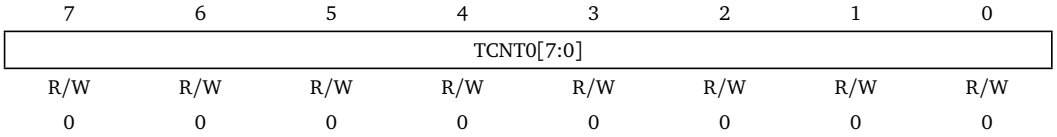
Het TCNT0-register is een 8-bits register en bevat de tellerwaarde van Timer/Counter 0, zie figuur 7.10. Dit register kan zowel gelezen als geschreven worden. Bij een schrijfactie telt Timer/Counter 0 verder vanaf de geschreven waarde.

Tabel 7.6: *Compare Match uitgang, Fast PWM-modus.*

COM01	COM00	Beschrijving
0	0	Normale poortwerking, OC0 is gedeactiveerd
0	1	Niet gebruikt
1	0	Zet OC0 op 0 bij Compare Match bij omhoog tellen, zet OC0 op 1 bij omlaag tellen
1	1	Zet OC0 op 1 bij Compare Match bij omhoog tellen, zet OC0 op 0 bij omlaag tellen

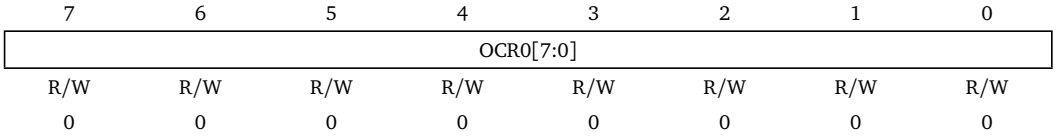
Tabel 7.7: *Klokselectie voor Timer/Counter 0.*

CS02	CS01	CS00	operatie
0	0	0	geen werking, klok is gestopt
0	0	1	$CLK_{IO}/1$, geen prescaler
0	1	0	$CLK_{IO}/8$, van prescaler
0	1	1	$CLK_{IO}/64$, van prescaler
1	0	0	$CLK_{IO}/256$, van prescaler
1	0	1	$CLK_{IO}/1024$, van prescaler
1	1	0	externe klokbron op de T0-pin, neergaande flank
1	1	1	externe klokbron op de T0-pin, opgaande flank



Figuur 7.10: *Het TCNT0 register.*

Het OCR0-register bevat een 8-bits waarde en wordt continue vergeleken met de waarde in het TCNT0-register, zie figuur 7.11. Als de twee register gelijk zijn aan elkaar kan een interrupt gegenereerd worden of kan een golfvorm gegenereerd worden op de OC0-uitgang.



Figuur 7.11: *Het OCR0 register.*

De indeling van het TIMSK-register is te zien in figuur 7.12. Voor Timer/Counter 0 zijn alleen de bits OCIE0 (bit 1) en TOIE0 (bit 0) van belang.

7	6	5	4	3	2	1	0
OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figuur 7.12: *Het TIMSK register.*

Bit 1 – OCIE0: Timer/Counter 0 Output Compare Match Interrupt Enable
Een logische 1 in deze bit zorgt ervoor dat de compare match interrupt geactiveerd wordt. Een interrupt wordt pas daadwerkelijk uitgevoerd als de I-bit in het statusregister ook een logische 1 is.

Bit 0 – TOIE0: Timer/Counter 0 Overflow Interrupt Enable
Een logische 1 in deze bit zorgt ervoor dat de overflow interrupt geactiveerd wordt. Een interrupt wordt pas daadwerkelijk uitgevoerd als de I-bit in het statusregister ook een logische 1 is.

De indeling van het TIFR-register is te zien in figuur 7.13. Voor Timer/Counter 0 zijn alleen de bits OCF0 (bit 1) en TOV0 (bit 0) van belang.

7	6	5	4	3	2	1	0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figuur 7.13: *Het TIFR register.*

Bit 1 – OCF0: Output Compare Flag 0
Deze bit wordt op een logische 1 gezet als de waarde van het OCR0-register gelijk is aan de waarde in het TCNT0-register (compare match). Dit gebeurt op de volgende Timer/Counter 0 klokpuls. Deze bit wordt op een logische 0 gezet als de bijbehorende interrupt wordt uitgevoerd. Als alternatief voor het op 0 zetten kan een 1 worden geschreven naar deze bit.

Bit 0 – TOV0: Timer/Counter 0 Overflow Flag
Deze bit wordt op een logische 1 gezet als Timer/Counter 0 een overflow veroorzaakt. Dit gebeurt op de volgende Timer/Counter 0 klokpuls. Deze bit wordt op 0 gezet als de bijbehorende interrupt wordt uitgevoerd. Als alternatief voor het op 0 zetten kan een 1 worden geschreven naar deze bit. In fasecorrect PWM-modus wordt deze bit op 1 gezet als Timer/Counter 0 van telrichting verandert op 0.

In figuur 7.14 is het SFIOR-register te zien.

7	6	5	4	3	2	1	0
ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PRS10
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

Figuur 7.14: *Het SFIOR register.*

Bit 0 – PRS10: Prescaler reset Timer/Counter 1 en Timer/Counter 0
Wanneer deze bit met een 1 geschreven wordt, zal de prescaler van Timer/Counter 1 en Timer/Counter 0 worden gereset. Nadat de operatie is voltooid, wordt deze bit door de hardware op 0 gezet. Merk op dat de prescaler door zowel Timer/Counter 1 als Timer/Counter 0 gebruikt wordt. Het beïnvloedt de werking van beide Timer/Counters. Deze bit leest altijd als een 0.