

# Microcontroller Teil 6 Timer/Counter

DI (FH) Andreas Pötscher

HTL Litec

*Timer/Counter werden als Timer verwendet, um exakt zeitgesteuerte Aufgaben zu erledigen. Sie werden als Counter verwendet, um das Auftreten von Hardware-Ereignissen zu zählen.*

- ▶ Der ATmega2560 hat zwei 8-Bit-Timer/Counter und vier 16-Bit-Timer/Counter namens Timer/Counter1, 3, 4 und 5. Herzstück jedes Timer/Counter ist ein 8 bzw. 16 Bit breites Zählerstands-Register, das durch bestimmte Zähl-Impulse ständig inkrementiert wird.

- ▶ Der ATmega2560 hat zwei 8-Bit-Timer/Counter und vier 16-Bit-Timer/Counter namens Timer/Counter1, 3, 4 und 5. Herzstück jedes Timer/Counter ist ein 8 bzw. 16 Bit breites Zählerstands-Register, das durch bestimmte Zähl-Impulse ständig inkrementiert wird.
- ▶ Timer/Counter0 und Timer/Counter2 verwenden 8 Bit breite Register. Das bedeutet, dass das Zählerstandsregister Werte von 0 bis 255 annehmen kann. Solche Timer/Counter werden als 8-Bit-Timer/Counter bezeichnet.

- ▶ Der ATmega2560 hat zwei 8-Bit-Timer/Counter und vier 16-Bit-Timer/Counter namens Timer/Counter1, 3, 4 und 5. Herzstück jedes Timer/Counter ist ein 8 bzw. 16 Bit breites Zählerstands-Register, das durch bestimmte Zähl-Impulse ständig inkrementiert wird.
- ▶ Timer/Counter0 und Timer/Counter2 verwenden 8 Bit breite Register. Das bedeutet, dass das Zählerstandsregister Werte von 0 bis 255 annehmen kann. Solche Timer/Counter werden als 8-Bit-Timer/Counter bezeichnet.
- ▶ Timer/Counter1, Timer/Counter3, Timer/Counter4 und Timer/Counter5 besitzen 16 Bit breite Register. Die möglichen Werte des Zählerstandsregisters liegen daher zwischen 0 und 65535.

Der aktuelle Zählerstand ist im Register TCNTn gespeichert, wobei n die Nummer des verwendeten Timer/Counter ist. Für den Timer/Counter 1 also.

```
uint16_t cnt;  
cnt = TCNT1;
```

# Clock-Select-Logic - Auswahl der Zähl-Impulse



Figure 1: Clock-Select-Logic von Timer/Counter 1

Prescaler-Wert	Timerfrequenz $f_T$	Periodendauer $T_T$
1	16 MHz	0,0625 $\mu s$
8	2 MHz	0,5 $\mu s$
64	250 kHz	4 $\mu s$
256	62,5 kHz	16 $\mu s$
1024	15,625 kHz	64 $\mu s$



Nach einem Reset beträgt der Zählerstand aller Timer/Counter gleich 0. Angenommen, in der Initialisierungsphase des Hauptprogramms wird beim Timer/Counter1 ein Prescaler-Wert von 256 ausgewählt. Dadurch startet der Timer/Counter1 und sein Zählerstand beginnt sich von null weg zu erhöhen. Wie lange dauert es, bis ein Zählerstand von 32000 erreicht ist?

- ▶ Die Frequenz des Taktsignals, das den Zählerstand erhöht, hat eine Frequenz von
$$f_T = \frac{f_{osc}}{256} = \frac{16MHz}{256} = 62,5kHz$$

- ▶ Die Frequenz des Taktsignals, das den Zählerstand erhöht, hat eine Frequenz von  $f_T = \frac{f_{osc}}{256} = \frac{16MHz}{256} = 62,5kHz$
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{62,5kHz} = 16\mu s$

- ▶ Die Frequenz des Taktsignals, das den Zählerstand erhöht, hat eine Frequenz von  $f_T = \frac{f_{osc}}{256} = \frac{16MHz}{256} = 62,5kHz$
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{62,5kHz} = 16\mu s$
- ▶ Direkt nach dem Starten des Timer/Counter1 durch Auswahl des Prescaler-Werts von 256 beträgt der Zählerstand noch 0. Eine Zeitspanne von  $16\mu s$  später ( $T_T$ ) wird der Zählerstand von 0 auf 1

- ▶ Die Frequenz des Taktsignals, das den Zählerstand erhöht, hat eine Frequenz von  $f_T = \frac{f_{osc}}{256} = \frac{16MHz}{256} = 62,5kHz$
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{62,5kHz} = 16\mu s$
- ▶ Direkt nach dem Starten des Timer/Counter1 durch Auswahl des Prescaler-Werts von 256 beträgt der Zählerstand noch 0. Eine Zeitspanne von  $16\mu s$  später ( $T_T$ ) wird der Zählerstand von 0 auf 1
- ▶ Der Zählerstand erhöht sich genau alle  $16\mu s$  um 1. Er ist somit eine Stufenfunktion der Zeit.



Figure 2: Zählerstand am Anfang

- ▶ Insgesamt dauert es bis ein Zählerstand von 32000 erreicht wird dann  $32000 * T_T = 32000 * 16\mu s = 512ms$

- ▶ Insgesamt dauert es bis ein Zählerstand von 32000 erreicht wird dann  $32000 * T_T = 32000 * 16\mu s = 512ms$
- ▶ Bei einem Zählerstand von 32000 ist die Stufenfunktion Aufgrund der vielen Punkte nicht mehr zu sehen.



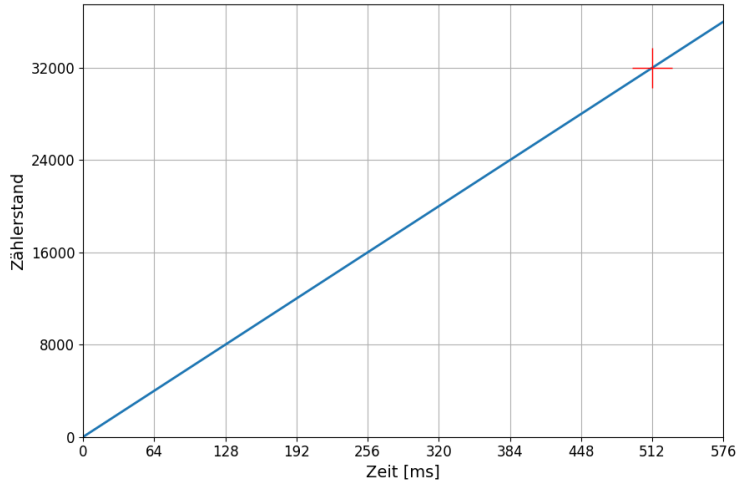


Figure 3: Nach 512 ms wird ein Wert von 32000 erreicht

Die Auswahl der Taktquelle geschieht durch das Setzen der Bits  $CS_n2$ ,  $CS_n1$  und  $CS_n0$  (CS steht für Clock Select) im Timer/Counter Control Register B ( $TCCR_nB$ ). Dabei steht das „n“ für die Nummer des jeweiligen Timer/Counters und ist entsprechend zu ersetzen.

Bit	7	6	5	4	3	2	1	0
<b>TCCR1B:</b>	—	—	—	WGM13	WGM12	CS12	CS11	CS10

CSn2	CSn1	CSn0	Taktquelle für Timer/Counter
0	0	0	keine Taktquelle der Timer ist gestoppt.
0	0	1	Prescaler 1 $f_T = 16MHz$
0	1	0	Prescaler 8 $f_T = 2MHz$
0	1	1	Prescaler 64 $f_T = 250kHz$
1	0	0	Prescaler 256 $f_T = 62,5kHz$
1	0	1	Prescaler 1024 $f_T = 15,625kHz$
1	1	0	Fallende Flanke am GPIO Pin mit der Funktion "Tn"
1	1	1	Steigende Flanke am GPIO Pin mit der Funktion "Tn"

- ▶ Timer/Counter3 soll als Timer mit einem Prescaler-Wert von 256 betrieben werden.

- ▶ Timer/Counter3 soll als Timer mit einem Prescaler-Wert von 256 betrieben werden.
- ▶ Im Register TCCR3B müssen die Bits CS32, CS31, CS30 gesetzt werden.

- ▶ Timer/Counter3 soll als Timer mit einem Prescaler-Wert von 256 betrieben werden.
- ▶ Im Register TCCR3B müssen die Bits CS32, CS31, CS30 gesetzt werden.

CSn2	CSn1	CSn0	Taktquelle für Timer/Counter
1	0	0	Prescaler 256 $f_T = 62,5kHz$

```
TCCR3B |= (0x01<<CS32);
```

```
TCCR3B &= ~( (0x01<<CS31) | (0x01<<CS30) );
```

Eine Variable vom Datentyp `uint16_t` (also ein 16-Bit breiter, vorzeichenloser Integer) kann Werte im Bereich von 0 bis  $2^{16} - 1$  annehmen, also von 0 bis 65.535.

Was passiert, wenn eine solche Variable ihren maximalen Wert hat und anschließend inkrementiert wird? Welchen Wert enthält die Variable `a` nach Ausführung des folgenden Programmcodes?

```
uint16_t a;  
a = 65535;  
a++;
```

Der Wert  $65535 + 1 = 65536$  (also  $2^{16}$ ) liegt außerhalb des darstellbaren Wertebereichs einer `uint16_t`-Variablen. Man spricht in diesem Fall von einem sogenannten Überlauf (Overflow). Die Variable erhält dann wieder den Wert 0.



Figure 4: Tacho kurz vor dem Überlauf



- ▶ Gleichzeitig wird bei einem Overflow ein Interrupt-Flag gesetzt.

- ▶ Gleichzeitig wird bei einem Overflow ein Interrupt-Flag gesetzt.
- ▶ Beim Timer/Counter1 heißt dieses Flag TOV1 (Timer Overflow 1).

- ▶ Gleichzeitig wird bei einem Overflow ein Interrupt-Flag gesetzt.
- ▶ Beim Timer/Counter1 heißt dieses Flag TOV1 (Timer Overflow 1).
- ▶ Ist zusätzlich das zugehörige Interrupt-Enable-Flag gesetzt, wird das Hauptprogramm unterbrochen und die zugehörige Interrupt Service Routine (ISR) ausgeführt – in diesem Fall die Routine mit dem Vektor `TIMER1_OVF_vect`.

- ▶ Timer/Counter1 verwendet als Clock-Source die mit einem Prescaler-Wert von 8 geteilte CPU-Taktfrequenz (16MHz). In welchem Zeitabständen erfolgen die Overflows des Zählerstand-Registers?

- ▶ Timer/Counter1 verwendet als Clock-Source die mit einem Prescaler-Wert von 8 geteilte CPU-Taktfrequenz (16MHz). In welchem Zeitabständen erfolgen die Overflows des Zählerstand-Registers?
- ▶ Der Zählerstand wird wegen des Prescaler-Werts von 8 mit einer Frequenz von  $f_T = \frac{f_{osc}}{8} = 2MHz$  inkrementiert.

- ▶ Timer/Counter1 verwendet als Clock-Source die mit einem Prescaler-Wert von 8 geteilte CPU-Taktfrequenz (16MHz). In welchem Zeitabständen erfolgen die Overflows des Zählerstand-Registers?
- ▶ Der Zählerstand wird wegen des Prescaler-Werts von 8 mit einer Frequenz von  $f_T = \frac{f_{osc}}{8} = 2MHz$  inkrementiert.
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{2MHz} = 0,5\mu s$

- ▶ Timer/Counter1 verwendet als Clock-Source die mit einem Prescaler-Wert von 8 geteilte CPU-Taktfrequenz (16MHz). In welchem Zeitabständen erfolgen die Overflows des Zählerstand-Registers?
- ▶ Der Zählerstand wird wegen des Prescaler-Werts von 8 mit einer Frequenz von  $f_T = \frac{f_{osc}}{8} = 2MHz$  inkrementiert.
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{2MHz} = 0,5\mu s$
- ▶ Die Zeitdauer zwischen zwei Overflows beträgt **65536** bzw.  $2^{16}$  mal dieser Zeitspanne (Zählerstände 0 bis **65535** bzw. 0 bis  $2^{16} - 1$ ).

- ▶ Timer/Counter1 verwendet als Clock-Source die mit einem Prescaler-Wert von 8 geteilte CPU-Taktfrequenz (16MHz). In welchem Zeitabständen erfolgen die Overflows des Zählerstand-Registers?
- ▶ Der Zählerstand wird wegen des Prescaler-Werts von 8 mit einer Frequenz von  $f_T = \frac{f_{osc}}{8} = 2MHz$  inkrementiert.
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{2MHz} = 0,5\mu s$
- ▶ Die Zeitdauer zwischen zwei Overflows beträgt 65536 bzw.  $2^{16}$  mal dieser Zeitspanne (Zählerstände 0 bis 65535 bzw. 0 bis  $2^{16} - 1$ ).
- ▶ Die Zeitdauer zwischen den Overflows  $T_{OVF}$  berechnet sich dann aus  $T_{OVF} = T_T * 2^{16} = 32,768ms$



- ▶ Timer/Counter1 verwendet als Clock-Source die mit einem Prescaler-Wert von 8 geteilte CPU-Taktfrequenz (16MHz). In welchem Zeitabständen erfolgen die Overflows des Zählerstand-Registers?
- ▶ Der Zählerstand wird wegen des Prescaler-Werts von 8 mit einer Frequenz von  $f_T = \frac{f_{osc}}{8} = 2MHz$  inkrementiert.
- ▶ Die Periodendauer ist der Kehrwert daraus:  $T_T = \frac{1}{f_T} = \frac{1}{2MHz} = 0,5\mu s$
- ▶ Die Zeitdauer zwischen zwei Overflows beträgt 65536 bzw.  $2^{16}$  mal dieser Zeitspanne (Zählerstände 0 bis 65535 bzw. 0 bis  $2^{16} - 1$ ).
- ▶ Die Zeitdauer zwischen den Overflows  $T_{OVF}$  berechnet sich dann aus  $T_{OVF} = T_T * 2^{16} = 32,768ms$
- ▶ Die Frequenz der Overflows  $f_{OVF} = \frac{1}{T_{OVF}} = \frac{1}{32,768ms} = 30,52Hz$

Blinklicht mit Timer Overflow Interrupt.

WokiWi Link