Timer - Einführung

Ein Timer ist ein bestimmtes Register im Mikrocontroller, das hardwaregesteuert fortlaufend um 1 erhöht (oder erniedrigt) wird.

Anstatt also Befehle im Programm vorzusehen, die regelmäßig ausgeführt werden und ein Register inkrementieren, erledigt dies der Mikrocontroller ganz von alleine. Dazu nutzt der Timer dem Systemtakt und so die Genauigkeit des Quarzes, um ein Register regelmäßig und vor allen Dingen unabhängig vom restlichen Programmfluss hochzählen zu lassen.
Bei bestimmten Zählerständen wird dann eine Aktion ausgeführt. Einer der 'bestimmten Zählerstände' ist zum Beispiel der **Overflow**.

Der höchste Zählerstand, den ein 8-Bit-Timer erreichen kann, $2^8 - 1 = 255$. Beim nächsten Inkrementierschritt tritt ein Überlauf (engl. Overflow) auf, der den Timerstand wieder zu 0 werden lässt.

Der Controller kann so konfiguriert werden, dass beim Auftreten des Timer-Overflows ein **Interrupt** ausgelöst wird.

Der Vorteiler

Wenn der Quarzoszillator z.B, mit 4 MHz schwingt, dann würde auch der Timer 4 Millionen mal in der Sekunde erhöht werden.

Da der Timer jedes mal bis 255 zählt, bevor ein Overflow auftritt, heißt das auch, dass in einer Sekunde 4000000 / 256 = 15625 Overflows vorkommen!

Um diese Raten zu verzögern, gibt es den Vorteiler. Er kann auf die Werte 1, 8, 64, 256 oder 1024 eingestellt werden.

Bsp: Prescaler: 1024 Takt 4 MHz → wie viele Overflows/s

$\frac{4 \cdot 10^6 Hz}{1024 \cdot 256} = 15{,}25$ Overflows/s $\Rightarrow T = \frac{1}{15{,}25} = 65{,}6\,ms$

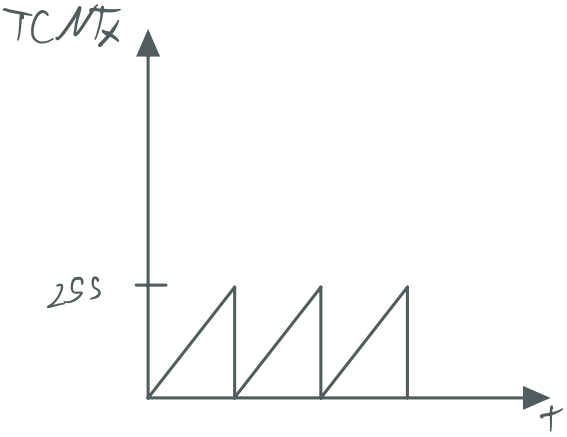
Die Betriebsmodi

Die AVR-Timer können in unterschiedlichen Betriebsmodi betrieben werden.
Diese sind:

- Normaler Modus
- CTC Modus
- PWM

Normaler Modus (Normal Mode)

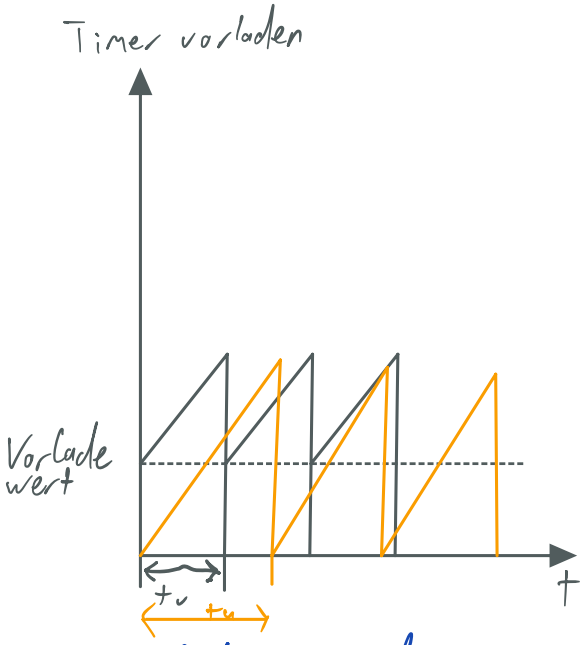
Der einfachste Betriebsmodus ist der normale Modus.
Die Zählrichtung des Timers ist immer aufsteigend, bis zum Überlauf, - dann fängt der Zähler wieder bei 0 an.
Der Überlauf kann einen Interrupt(Timer-Overflow) auslösen.



Der Zähler des Timers ist in dem Register TCNTx gespeichert, wobei x für den verwendeten Timer steht. Soll z.B. Timer0 -> TCNT0.

Wie lange es braucht, bis der Zähler einen Overflow auslöst, ist von der Taktfrequenz des Controllers, dem eingestellten Prescaler-Wert und von der Timerauflösung abhängig.

Um die Anzahl der Schritte bis zum Überlauf zu verändern, kann nach einem Überlauf der Zähler vorgeladen werden:



tv .... Periode vorgeladen
tu ... Periode unvorgeladen
Durch das Vorladen wird die Periodendauer des Timers verkürzt

Vorladen durch direktes schreiben von Vorladewert auf TCNTx Register nach Überlauf

z.B: Timer 0, Vorladewert = 100
TCNT0 = 100,

Bsp: Timer 0; Quarzfrequenz = 8 MHz; Prescaler = 64

Wunsch: Overflow alle 1 ms

↳ Vorladewert!

Timer 0 = 8 bit

$f_{Timer} = \frac{8\,MHz}{64} = 125\,kHz \;\hat{=}\; 8\,\mu s$

$\frac{1\,ms}{8\,\mu s} = 125$ Steps

Vorladewert = 256 − 125 = 131

Bsp: Die Blinkfrequenz einer LED soll 0,5 Hz sein.

Timer 0, f_q = 1 MHz

Eingangsperiodendauer des Timers: $1\,\mu s \cdot 1024 \approx 1024\,\mu s \approx 1\,ms$

$\frac{1}{1\,MHz} = \frac{1}{f_q}$

Preload = 0 ⇒ Overflow alle $1024\,\mu s \cdot 256$ Steps = 250 ms

↳ alle 250 ms einen Overflow

Softwaremäßig kann nun eine Countervariable "count" bei jedem Overflow erhöht werden.

Beim erreichen von count == 4 wird die LED getoggelt und count wieder auf 0 gesetzt.