

Tutorium zu Computer-Engineering im SS19

Termin 5

Jakob Otto

HAW Hamburg

30. April 2019

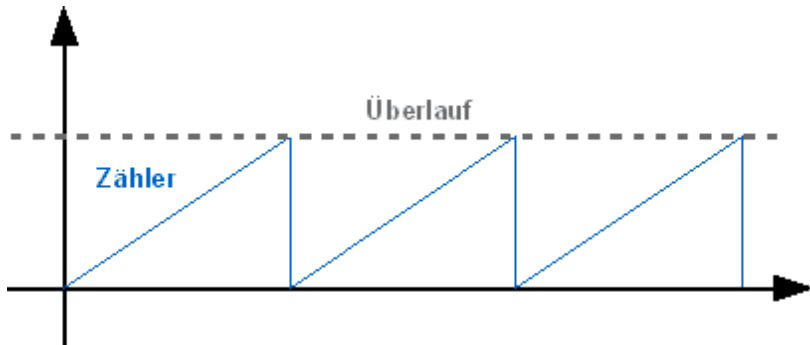
Ablauf

- Praktikum
 - ▶ Was ist ein Timer?
 - ▶ Prescaler
 - ▶ ARR/ARPE



Timer

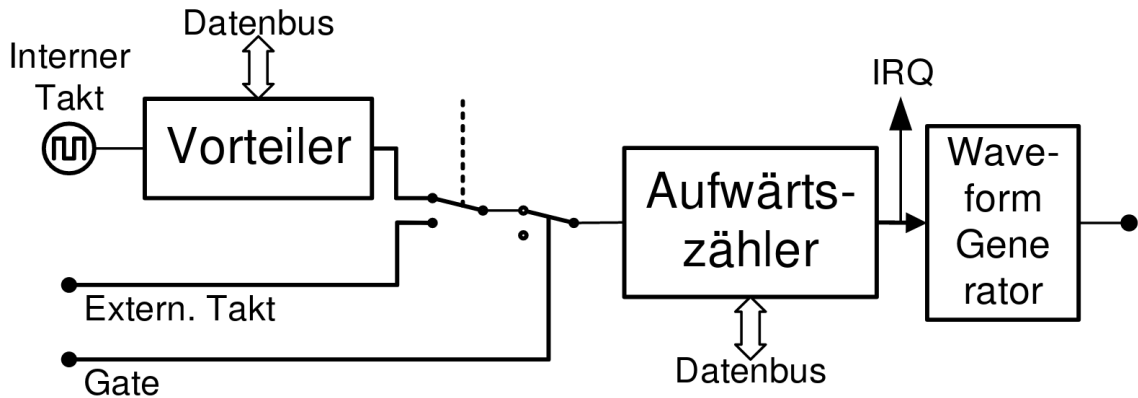
- eigentlich ein Counter
- zählt Ticks von internem Quarz/extern
- bei maximalem Zählstand kann zB IRQ ausgelöst werden
 - Timer startet dann von vorn



Prescaler

- Systemfrequenz meist zu hoch um Sinnvoll zu sein
 - Wertebereich wird zu schnell verlassen
- dafür gibt es Prescaler
- weiterer Zähler, der eingehenden takt „vorteilt“
- Auflösung wird geringer
- Wertebereich wird seltener verlassen.

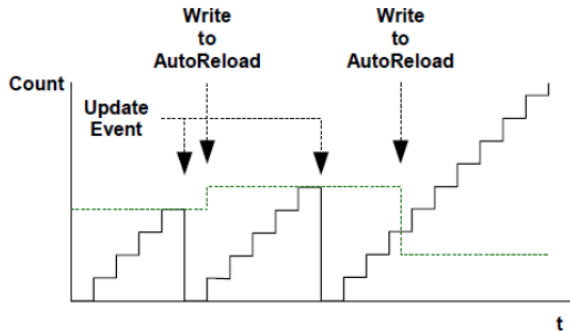
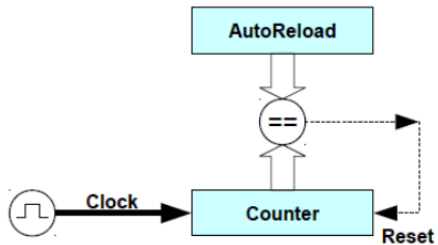
Aufbau Timer



ARR

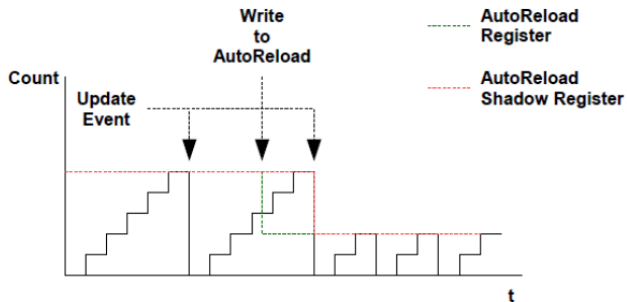
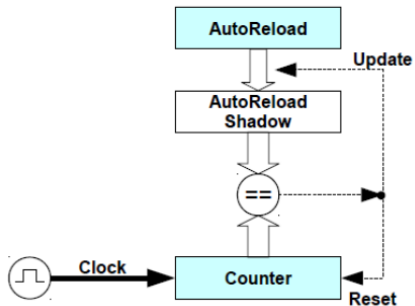
- Das ARR (AutoReloadRegister) beinhaltet einen eigenen Endwert.
- Zählstand wird mit dem Wert verglichen
- Bei erreichen:
 - ▶ reset vom Zählstand
 - ▶ prescaler → tick an timer
 - ▶ Timer → IRQ

Auto Reload Counter



- ARR kann zur laufzeit verändert werden
- Timer kann schon über den neuen Wert gezählt haben
 - zählt dann bis maxwert des Registers
- um das zu vermeiden gibts das ARPE-bit
- änderung wird durch shadow-register verzögert
 - erst bei nächstem overflow ins ARR übernommen

Auto Reload Counter mit Shadow Register



Code

```
//Timer setup
TIM1->CR1 = 0; // disable timer1
TIM1->CR2 = 0; // disable timer2
TIM1->PSC = 0; // prescaler value
TIM1->ARR = (SYS_FREQ / TIMER_FREQ) -1; //
TIM1->DIER = TIM_DIER_UIE; // enable Interrupt
TIM1->CR1 = TIM_CR1_ARPE; // Auto Reload preload enable

// enable timer
TIM1->CR1 |= TIM_CR1_CEN;
```

General-purpose timer cookbook