RUHR-UNIVERSITÄT BOCHUM

RUB

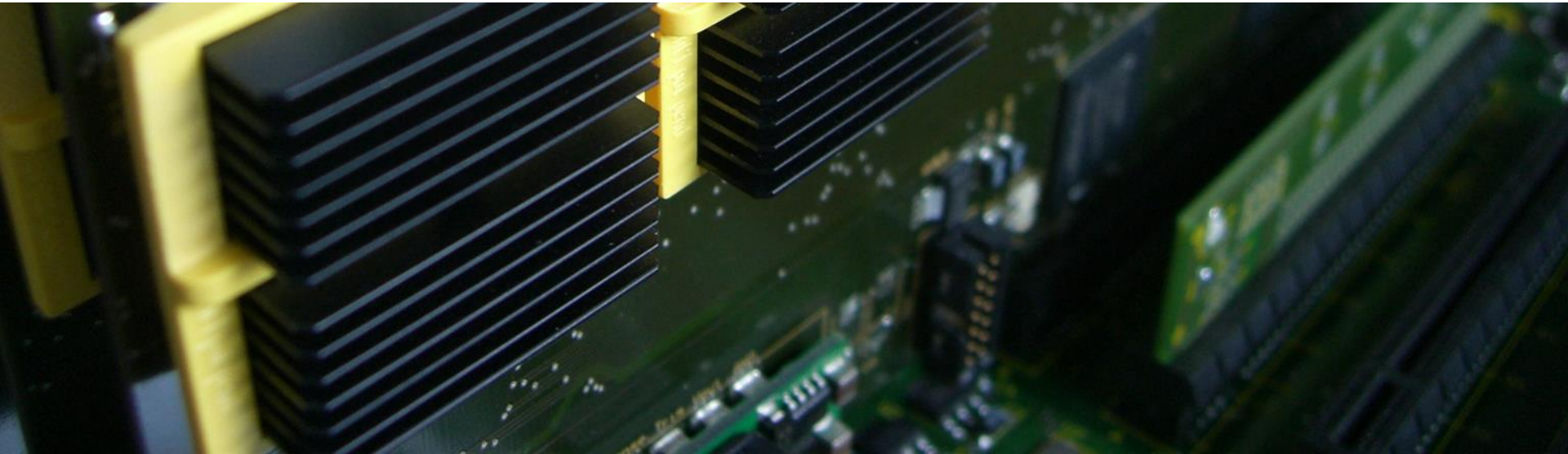
Eingebettete Prozessoren

SS 2014

Übung 11: PWM

Dipl.-Ing. Thomas Pöppelmann
Arbeitsgruppe Sichere Hardware
Horst Görtz Institut für IT-Sicherheit

10.07.2013



Agenda

1. Besprechung Übung 10
2. Konfiguration der H-Brücke und PWM Einheit

1. Besprechung Übung 10

Lösung Aufgabe 1)a)

1)a) Geben Sie die notwendigen Assemblerbefehle an, um den Timer 2 des ATMega8 mit dem Vorteiler CLK/64 zu aktivieren. Es soll weiterhin ein Interrupt eingerichtet werden, der ausgelöst wird, sobald der Timer die Hälfte seiner maximalen Zählschritte erreicht

- ; **Output Compare Match** Interrupt des **Timer2 aktivieren**
  LDI R16, (1<<OCIE2)
  OUT TIMSK, R16

  ; Setze das **Output Compare Match Register auf 128**
  LDI R16, 128
  OUT OCR2, R16

  ; **Timer 2 aktivieren** (d.h. Bit(s) für **PRESCALER CLK/64** setzen => Einstellung **CS22:CS21:CS20 = 100)**
  LDI R16, (1<<CS22)
  OUT TCCR2, R16

  ; aktiviere globale Interrupts
  SEI

(Interrupt Timer/Counter2 Compare Match : TIMER2_COMP)

Lösung Aufgabe 1)b)

1)b) Der Timer 1 des ATMega8 besitzt 16-Bit Register u.a. TCNT1, OCR1A/B und ICR1, die jedoch aufgrund von Sparmaßnahmen individuell nur mit jeweils 8 Bit an den Datenbus angebunden sind. Welche Konvention muss beim Schreiben und Lesen von 16-Bit Werten bei dieser Einschränkung eingehalten werden?

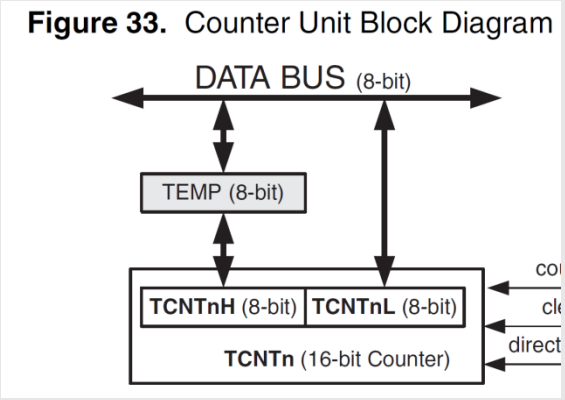
- **Schreiboperation mit höherwertigem Byte beginnen** (z.B. TCNT1 auf 0x1FF):

  LDI R17, 0x01
  LDI R16, 0xFF
  OUT TCNT1H,R17
  OUT TCNT1L,R16

  Leseoperation beginnt mit niederwertigem Byte, dann das höherwertige Byte

  IN R16, TCNT1L
  IN R17, TCNT1H

  – **Grund:** Schreib/Leseoperation muss atomar sein



Lösung Aufgabe 1)c)

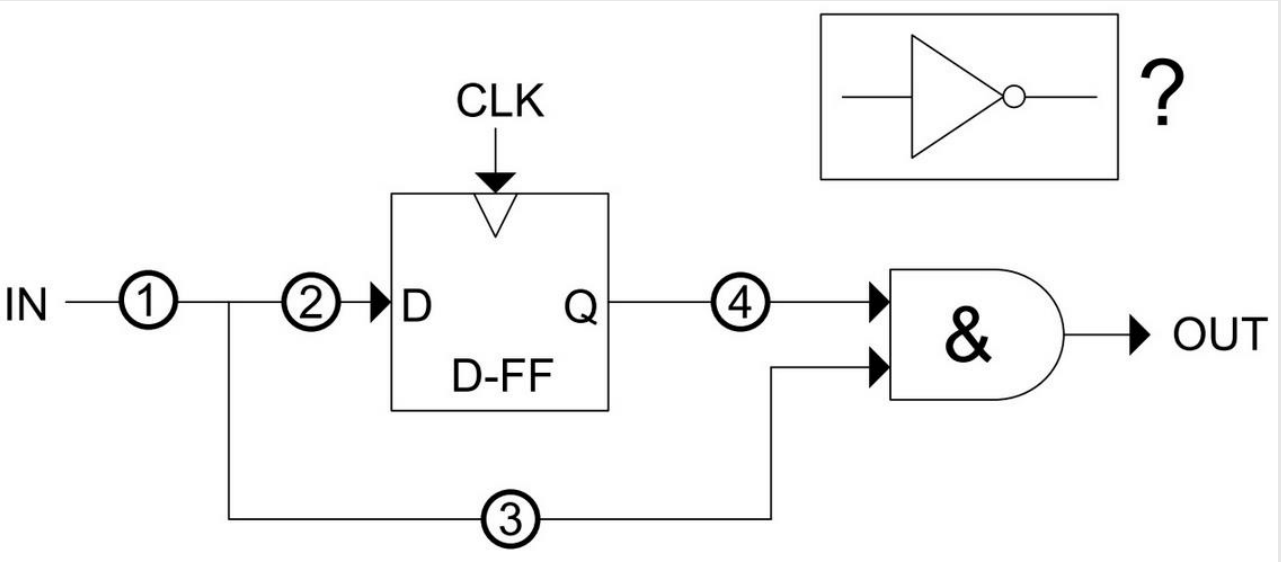
1)c) Nehmen Sie an, dass sowohl Timer 0 als auch Timer 1 vom AVR gleichzeitig verwendet werden. Nun setzt ein Programm den Timer 1 sowie den zugehörigen Vorteiler per Reset auf Null zurück. Welche Implikation hat das für Timer 0?

- **Der Timer 0 und der Timer 1 teilen sich den Prescaler**. **Wird der Prescaler also von Timer 1 zurückgesetzt, gilt dies auch für Timer 0**
- Die Einstellungen der Prescaler können pro Timer unterschiedlich sein
  - Abgriffe CS0x, CS1x
- Prescaler ist „free running“. Ungenauigkeit beim Einschalten der Timer

Aufgabe 2

- **2)** Die folgende Abbildung zeigt einen fast vollständigen Flankendetektor (bestehend aus D-Flipflop und UND-Gatter), in dem ein Inverter aber noch fehlt.

  An welcher Position 1 bis 4 muss der Inverter eingesetzt werden, damit die Schaltung als Detektor für eine fallende Taktflanke fungiert? Spielen Sie die Signalverläufe aller Möglichkeiten durch und geben Sie diese in ihrer Lösung an. Legen Sie dann kurz in Ihrer Lösung in Textform dar, mit welcher Konfiguration eine sinnvolle Funktion erreicht wird und wie sich das Ausgangssignal jeweils gegenüber des Eingangssignals verhält.



Lösung Aufgabe 2

- (Position 3 ist die gesuchte Position des Inverters, um fallende Flanken zu detektieren.)
- Zusammenfassung der einzelnen Positionen:
  - **Inverter an Pos 1**: dieser Fall produziert kein sinnvolles Ergebnis. Durch den Inverter vor dem Flipflop und dem UND-Gatter wird das Eingangssignal invertiert, wobei die logische Null am Ausgang einen Takt länger als beim Eingangssignal anliegt:
  - **Inverter an Pos. 2**: Fall für den Flankendetektor einer steigenden Flanke
  - **Inverter an Pos. 3**: Fall für den Flankendetektor einer fallenden Flanke
  - **Inverter an Pos. 4**: Dies ist eine alternative Bauform für den Flankendetektor der steigenden Flanke. Es ist hierbei jedoch zu beachten, dass diese Schaltung im Ausgangssignal die Gatterverzögerung von zwei Bauelementen (nämlich Inverter und UND-Gatter) aufweist (d.h. der Ausgang ist etwas mehr phasenverschoben gegenüber dem Taktsignal). Die Schaltung mit Inverter an Position 2 hat hingegen nur eine Verzögerung von einem Gatter (UND-Gatter) im Ausgangssignal, ist also zu bevorzugen.

Hinweis zur Klausur

- Musterklausur (WS09/10) im Blackboard verfügbar
  - Klausur
  - Musterlösung/Lösungshinweise
  - Anhang
- Technische Probleme (Zugriff) hoffentlich gelöst

- Die Klausur fragt sowohl theoretisches Wissen/Fähigkeiten aus der Vorlesung als auch praktisches Wissen/Fähigkeiten aus der der Übung ab
- Keine Herausgabe weiter Klausuren
- Übungen/Hausaufgaben sind eine sehr gute Vorbereitung auf die Klausur (wie immer)
- Ausro-Führerschein ist ebenfalls sehr gute Vorbereitung
  - Möglichst selbstständig lösen
  - Mehr dazu in der Vorlesung am 18.07.2014 und im Blackboard

- Die Prüfung wird letztmalig im WS 2016/2017 angeboten.
- **Schriftliche Prüfung am 21.08.2014**

(Hinweis: Bitte zur Vorbereitung den Papier-Anhang benutzen – in der Klausur steht keine PDF-Suche zur Verfügung)

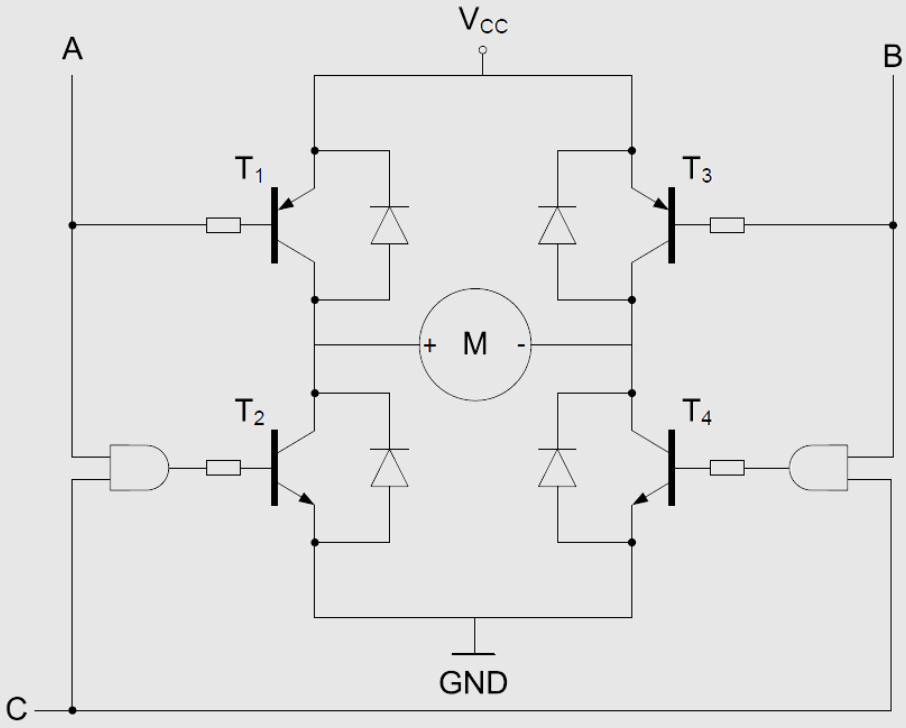
Rückblick: Timer/Counter

- Berechnen und Implementieren einer Zeitbasis
  - Generell: Bestimme Anzahl volle Timerdurchläufe (*N*) und ein partieller Durchlauf (*R*)
  - **Frage:** Warum läuft der Timer genau, auch wenn wir in der Interrupt Routine im letzten Durchlauf einige Zeit brauchen, bis wir den Wert *255-R* in das Timerregister geschrieben haben?
  - **Antwort**: Der Timer/Prescaler läuft nebenläufig und unabhängig vom Prozessor. Bevor der Timer nach einem Overflow das erste mal inkrementiert wird, muss erst der Prescaler durchlaufen werden. Bei Prescalerwert x=256 bleiben so also ~256 Takte Zeit, um die Konstante *255-R* für den partiellen Durchlauf in das Timerregister zu laden ohne das Probleme mit der Genauigkeit auftreten.

2. Konfiguration der H-Brücke und PWM Einheit

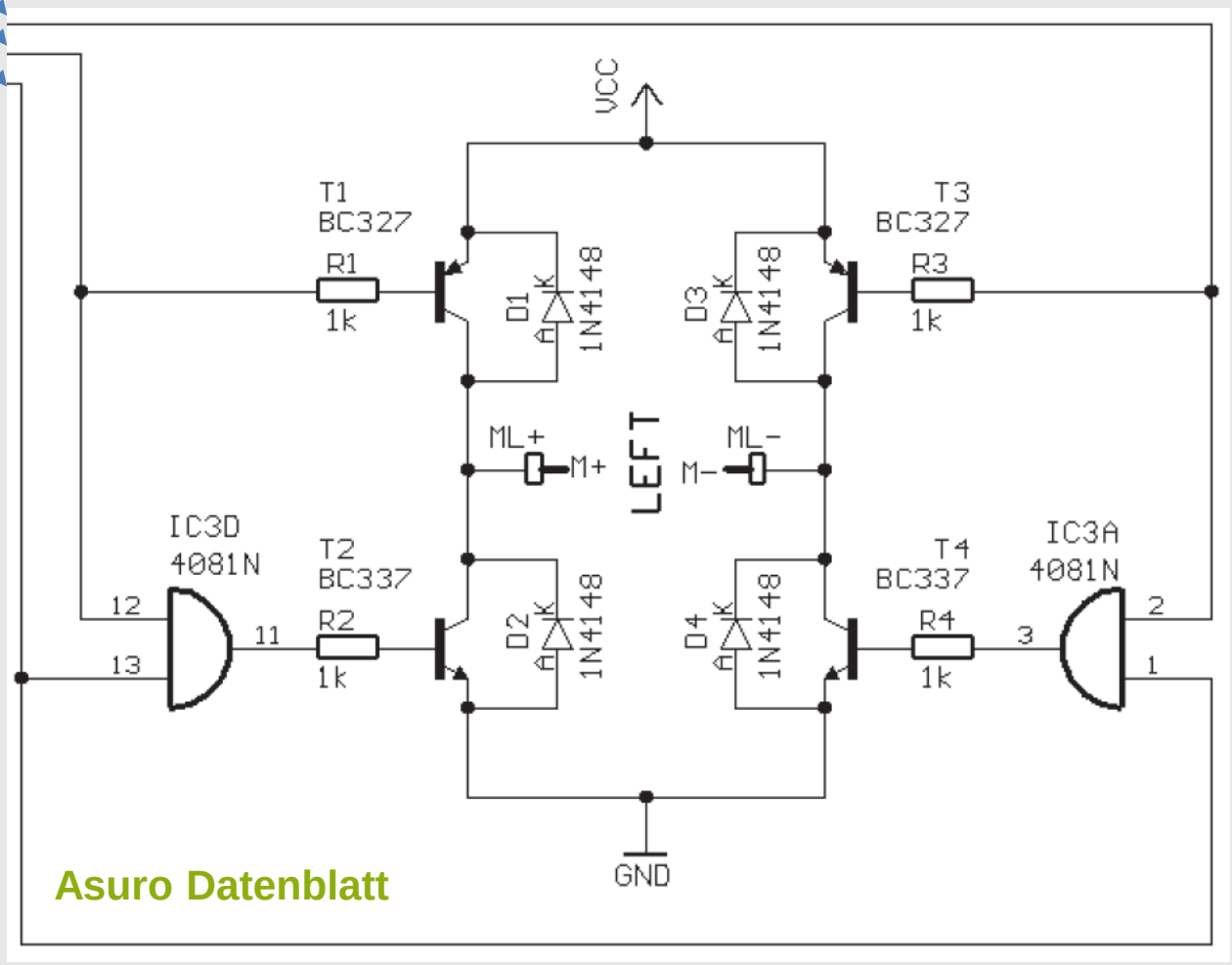
H-Brücke

- Prinzip der H-Brücke wurde in der Vorlesung behandelt
  - Ermöglicht Normalbetrieb (vorwärts), Umkehr der Polarität (rückwärts), Leerlauf und Bremsen
  - Freilaufdioden schützen vor schädlichen Spannungsspitzen
  - H-Brücke(n) des Asuros: Siehe Asuro Manual S. 74
  - PNP leitet falls B=0
  - NPN leitet falls B=1



H-Brücke: Asuro

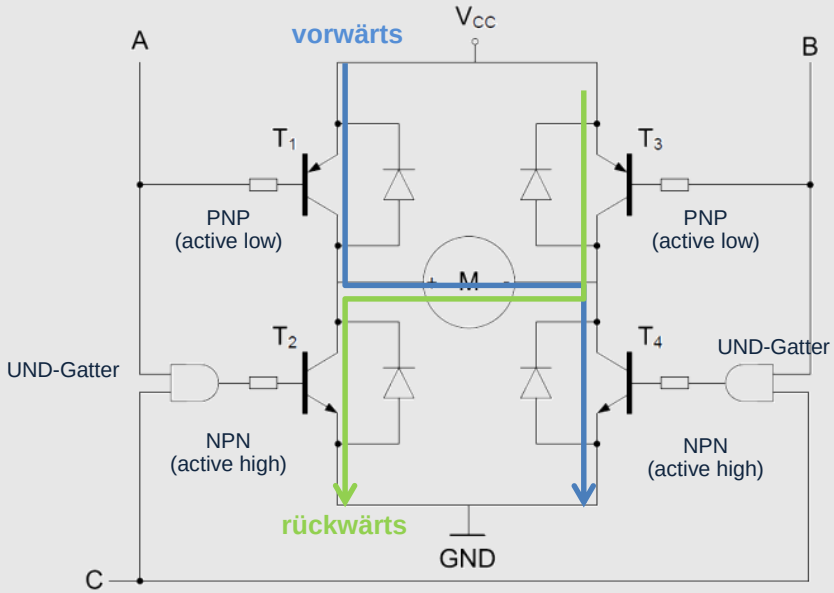
- An welchem Port sind diese Signale angeschlossen?



Vorwärts fahren

Beispiel - Vorwärts: A = 0, B = 1, C = 1 (Vcc am Plus-Pol), d.h. die Transistoren T1 und T4 leiten, Rückwärts A = 1, B =0, C = 1 (Vcc am Minus-Pol)

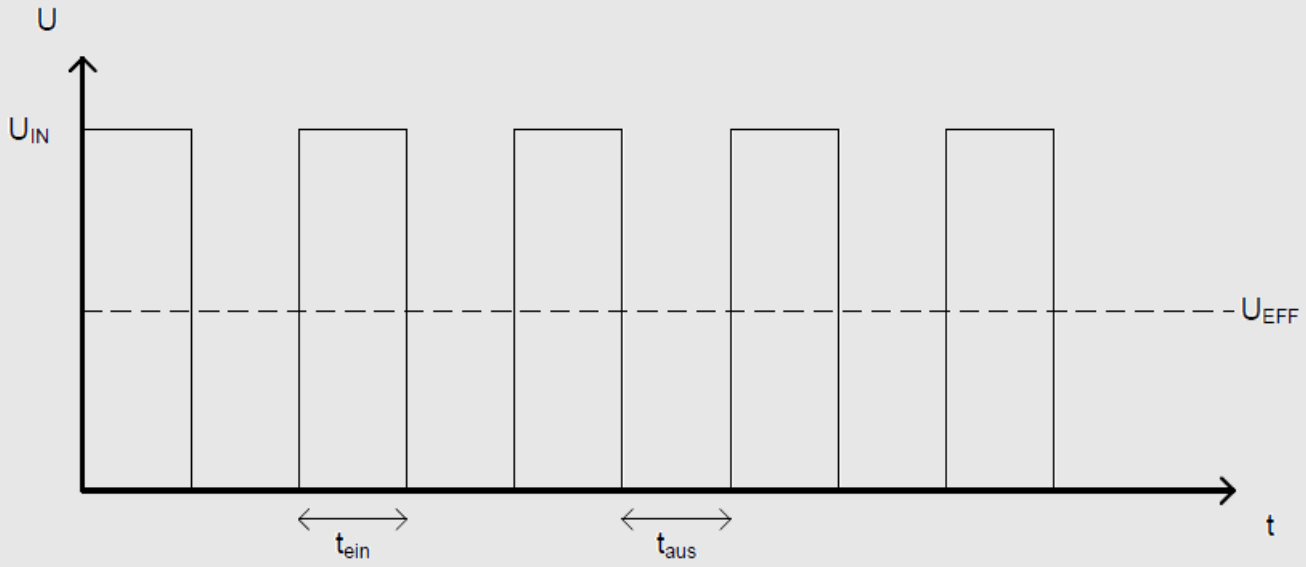
```
.include "m8def.inc"

;Motor LEFT    vorwärts     ;A=PD4=0; B=PD5=1;  C=PB1=1
;Motor RIGHT rückwärts   ;A=PB5=1; B=PB4=0;   C=PB2=1

Ldi r16, (1<<PD4|1<<PD5)        ;Ausgänge – Pull-Ups aktivieren
Out DDRD, r16
Ldi r16, (1<<PB1|1<<PB5|1<<PB4|1<<PB2)
Out DDRB, r16

Ldi r16, (1<<PD5)                    ;vorwärts fahren
Out PORTD, r16
Ldi r16, (1<<PB1|1<<PB2|1<<PB5)
Out PORTB, r16
….
```



(Tut der Pfeil der Basis weh, handelt's sich um PNP.)

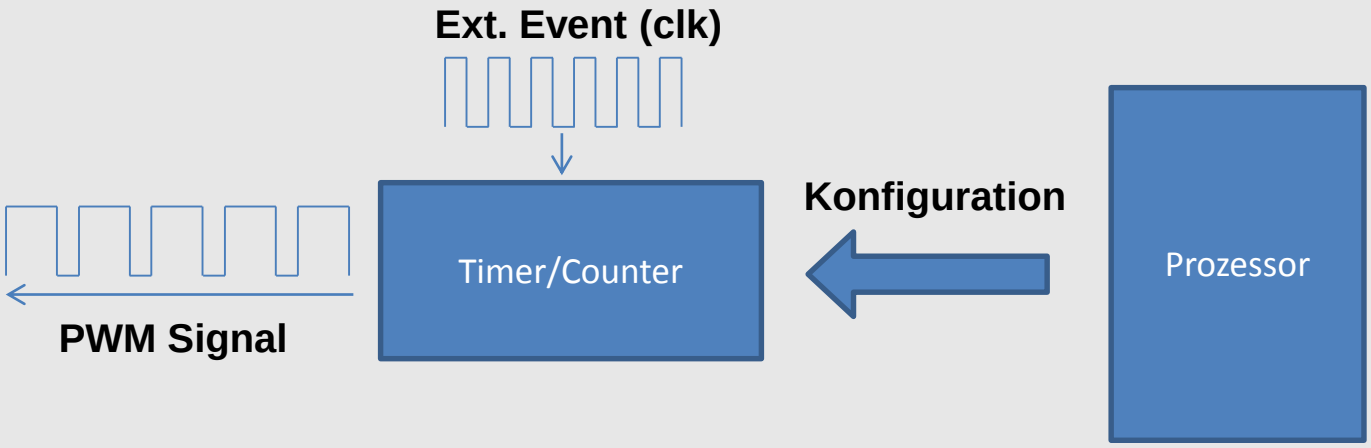
PWM

- Prinzip der Pulsweitenmodulation (PWM) wurde in der Vorlesung behandelt
  - Drehzahl ist in etwa proportional zur Spannung
  - Leitung wird mit Spannungsimpulsen mit einer genau definierten Länge belegt
  - Duty Cycle: $d = \frac{t_{ein}}{t_{ein}+t_{aus}}$ (Prozentsatz der Zeit im aktiven Zustand)
  - Effektivwert der Spannung: $U_{eff} = U_{IN} * d = U_{IN} \frac{t_{ein}}{t_{ein}+t_{aus}}$
  - Einfacher zu realisieren als die am Motor anliegende Spannung zu verringern



Realisierung der PWM auf dem ATMega

- Timer kann je nach Konfiguration das PWM Signal erzeugen
- Timer ist nebenläufig in separater Hardware realisiert
- PWM Funktion ist nur mit *Timer1* (2 Kanäle) und *Timer2* (1 Kanal) realisierbar
- Prozessor kann schlafen, während die PWM Einheit arbeitet



Vergleich der Timer im ATMega 8 uC

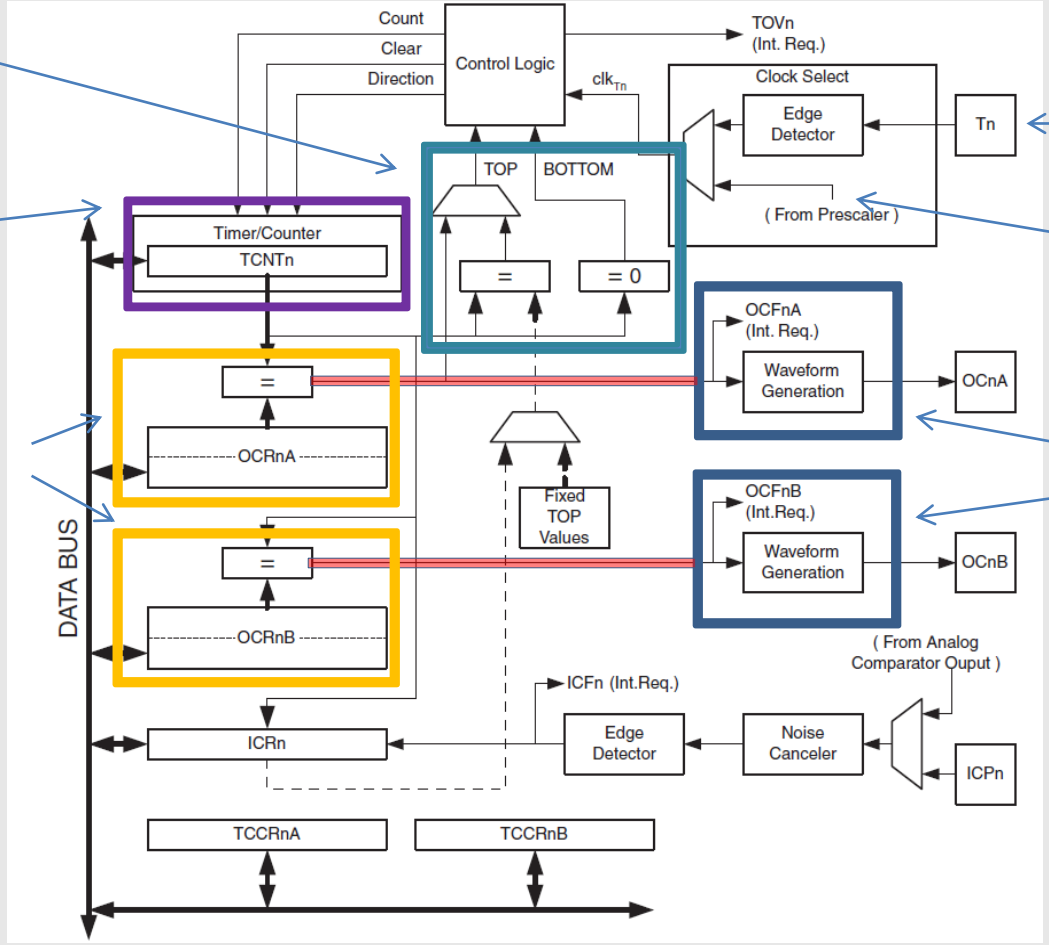
	Timer 0	Timer 1	Timer 2
Quellen	CLK, pos./neg. Flanke T0 (PD4)	CLK, pos./neg. Flanke T1 (PD5), Analogvergleich	CLK, Ext. Takt
Prescaler	*Prescaler von Timer 1*	CLK/*x* *x*= {*1,8,64,256,1024*}	CLK/*x* *x*= {1,8,32,64,128,256, 1024}
Zähler	8 Bit, ++, R/W	16 Bit, ++, --, clr, R/W	8 Bit, ++, --, clr, R/W
Auswertung	= 0xFF	= 0x00, = refA	= 0x00, = 0xFF, = refB
Ausgabe	Interrupt.: Overflow	Interrupt : Capture, Overflow, Compare1, Compare2 **PWM1A, PWM1B**	Interrupt.: Overflow, Compare **PWM2**

Aufbau Timer1

Timer Events

Wert des Timers wird im TCNT Register gespeichert

Output Compare Register (OCR) triggert Waveform generator



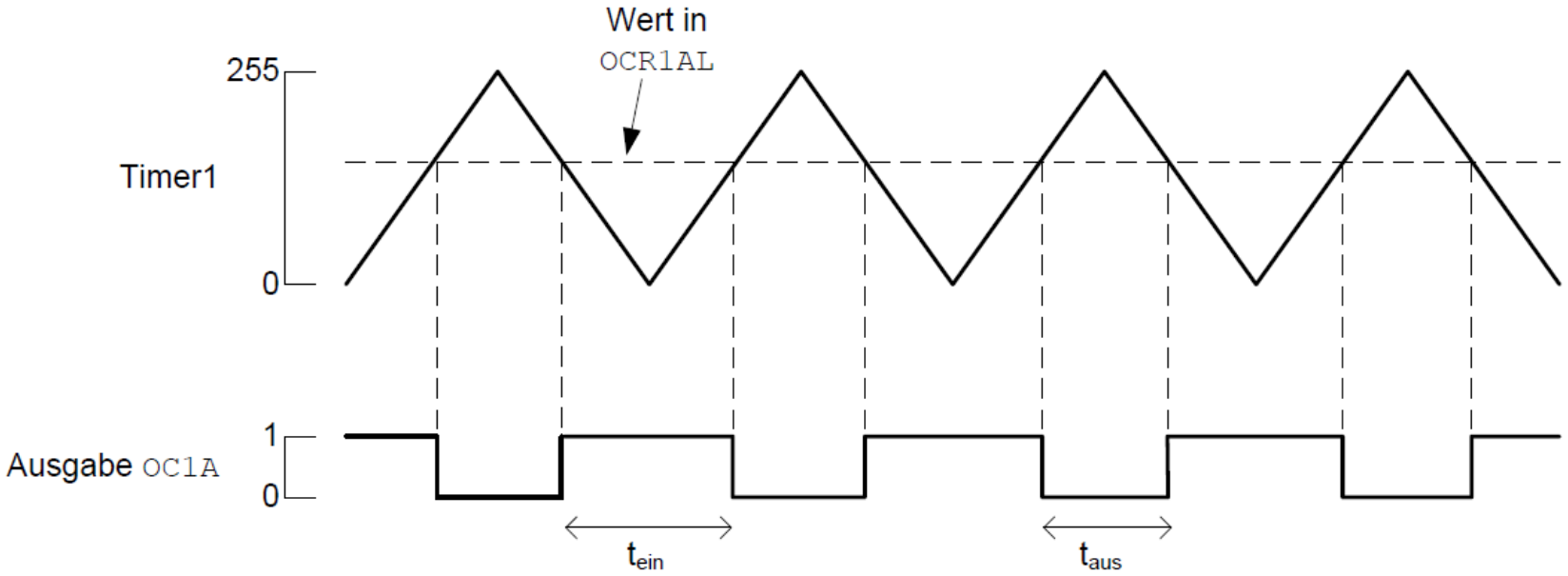
Externer Trigger

Interner Trigger (Prescaler)

Waveform Generator

Timer arbeitet “unabhängig“ vom Prozessor. Wird über den Datenbus kofiguriert.

- Timer zählt Bottom-Top-Bottom …
- Ausgabe OC1A wird umgekehrt, wenn der Wert des Timers gleich dem Wert in OCR1AL ist



PWM Konfiguration

- Die beiden PWM-Kanäle A und B des Timer 1 steuern den linken und den rechten Motor des ASURO.
- Vorgehen
  - Wir benötigen drei Ausgänge jeweils für den linken und rechten Motor (im DDRx Register setzen):
    - A = PD4 (links) bzw. PB4 (rechts)
    - B = PD5 (links) bzw. PB5 (rechts)
    - C = OC1Ab= PB1 (links) bzw. OC1Bb= PB2 (rechts)
  - Die Einstellungen WGM10:1, COM1A1:COM1A0=10 und COM1B1:COM1B0=10 im TCCR setzen:
    - **WGM10**: Verwendung des Mode 1: 8-Bit Phase Correct PWM Modus für beide Kanäle (Einstellung durch die entsprechenden WGM-Bits im TCCR1A/TCCR1B).
    - **COM1x1:COM1x0**: Lösche OC1A/OC1B bei Compare Match beim hochzählen, setze OC1A/OC1B bei Compare Match when heruntergezählt wird
  - Timer 1 aktivieren durch setzen des Prescalers (der Vorteiler des Zählers sollte auf mindestens CLK=8 im TCCR1B gesetzt werden)
  - Die aktuelle Geschwindigkeitseinstellung (d.h. der duty cycle) der Motoren wird jeweils als 8-Bit Wert in das OC-Register für Kanal A/B geschrieben

Exkurs: Sleep Mode

- Begrenzte Batterielebensdauer erfordert die Möglichkeit die CPU sowie Peripheriekomponenten manuell bei Nichtgebrauch in unterschiedlich tiefe Schlafzustände zu schicken
  - Aufwecken über externe Events
  - Kein Busy-Waiting benutzen (Stromverschwendung)
- *SLEEP* Instruktion
- Asuro Datasheet: 9. Power Management and Sleep Modes

Hausaufgabe Übung 11

- 1) Duty Cylce und Effektivspannung berechnen
- 2) Fragen zum Sleep Mode (einfach)
- 3) Motorsteuerung und PWM des Asuro konfigurieren
  - Macros
    - BACKWARD
    - TURNLEFT
    - TURNRIGHT
    - BRAKE
  - Geschwindigkeit mittels PWM einstellen

(Hinweis: H-Brücke und PWM ggf. erst im separaten Projekten testen und dann in Antwortdatei integrieren)

Hausaufgabe Übung 12

- Nächsten Donnerstag 17.07.2014: Letzte Übung Eingebettete Prozessoren im Sommersemester 2014 bzw. für immer in dieser Form ☹☹☹
- Hausaufgabe zur Übung 12
  - Abgabe bis 24.07.2014
  - Freischaltung der Übung schon in den nächsten Tagen
    - Wenig Zeit zwischen Vorlesungszeit und Klausuren. Besser planbar.
    - Material aus dem letzten Jahr wird bereitgestellt
  - Stoff (ADC) wird Freitag (morgen) in der Vorlesung behandelt
  - Übung am Donnerstag 17.07.2014 behandelt den ADC ebenfalls wie gewohnt