RUHR-UNIVERSITÄT BOCHUM

RUB

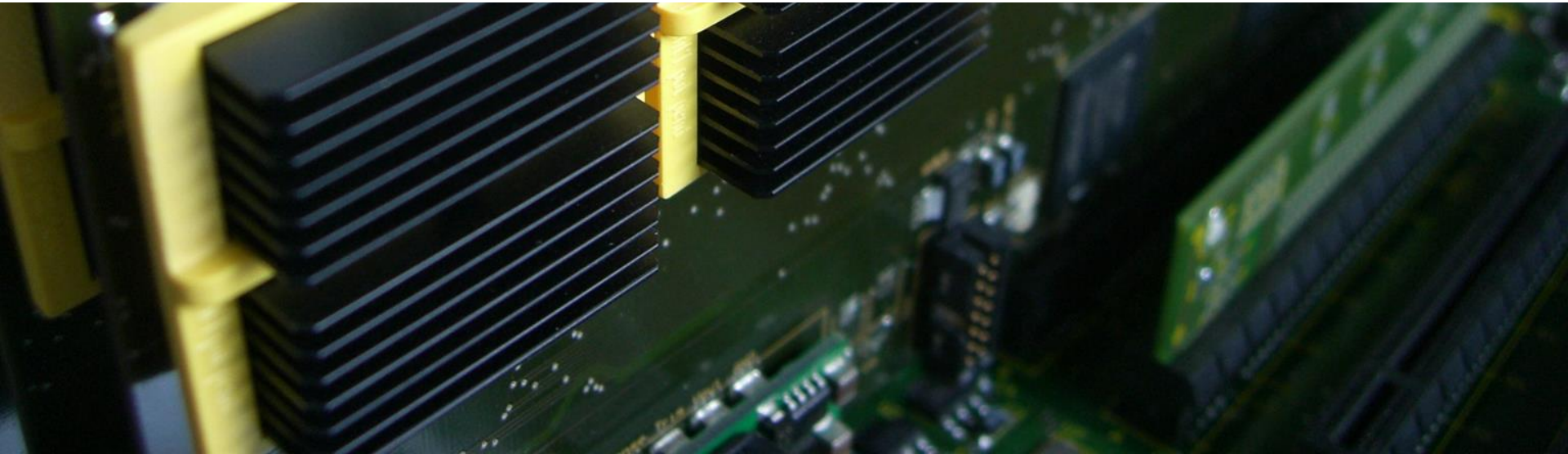
Eingebettete Prozessoren

SS 2014

Übung 10: Timer/Counter

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

03.07.2014



Agenda

1. Besprechung Übung 9
2. Timer/Counter

1. Besprechung Übung 9

Übung 9

- (**Aufgabe 2**) Erklären Sie, warum manchmal auf das Sichern des Statusregisters, bei durch rcall aufgerufenen Funktionen, verzichtet werden kann
  - Der Programmierer weiß, wann rcall aufgerufen wird und kann sich an den entsprechenden Stellen absichern und ggf. selber Register sichern. Warnung: Vorgehensweise kann Programme fehleranfällig machen.
- (**Aufgabe 2**) Warum muss in der Regel das Statusregister in einer Interruptserviceroutine immer gesichert werden? Geben Sie ein Beispiel an, in dem sich ein Programm abhängig davon wann ein Interrupt ausgelöst wird anders verhält
  - Interrupts können an jeder Stelle im Hauptprogramm ausgeführt werden (nicht vom Programmierer beeinflusst)
  - Beispiel im AVRStudio (*bsp_uebung9_lsg_sreg.asm*)

Übung 9

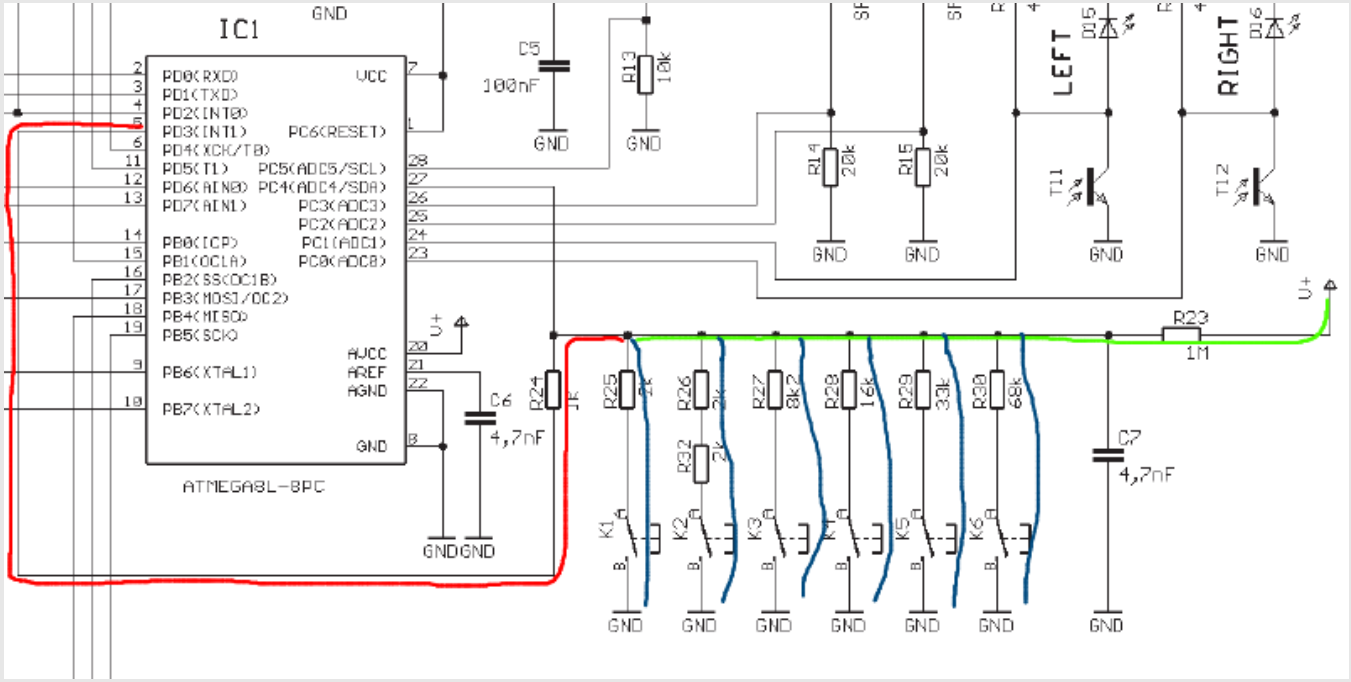
- (**Aufgabe 3a**) Gibt es Software-Interrupts?
  - Für SW-Interrupts kommen die externen Interrupts INT0, INT1 in Frage. Dabei muss allerdings jeweils der externe Pin (INT0 => PD2 und INT1 => PD3) als Ausgang konfiguriert sein. Dann kann der Programmierer entsprechend den Pin-Ausgang (und damit den Eingang des Interupts) per Software setzen und diesen so nach Bedarf auslösen.

- (**Aufgabe 4a**) Welche Einstellung sollten Sie für den Interrupt wählen, um die Front-LED bei Tasterdruck ein- und beim Loslassen wieder auszuschalten?
  - Auslösung bei Flankenwechsel (d.h. 0->1, 1->0). Dann kann bei jeder Auslösung der LED-Zustand einfach gewechselt (=toggle z.B. mit XOR) werden, und die Lampe wird bei steigender Flanke ein- und bei fallender Flanke auszuschalten.

Frage zu Übung 9

- **Aufgabenstellung Übung 9:** Der Pin des ATMega8, auf dem der Interrupt INT1 implementiert ist, sollte dabei als Eingang ohne Pull-up Widerstand konfiguriert sein, wobei eine logische Null bei einem gedrückten Schalter und die logische Eins bei einen geöffneten Schalter am externen Pinn des INT1 anliegt.
  - Warum liegt kein undefinierter Zustand an – der Pull-up ist ja deaktiviert?

Frage zu Übung 9

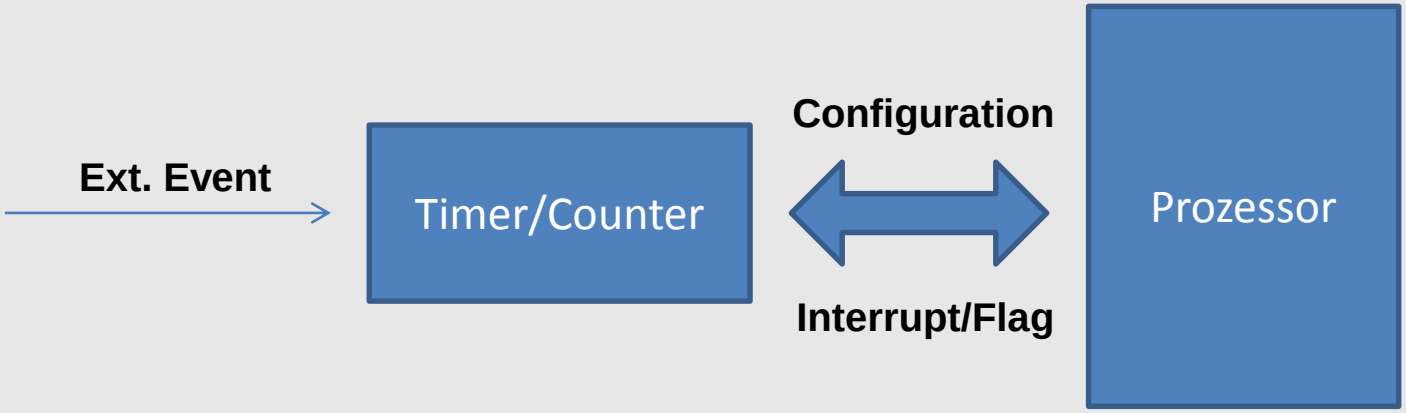


- Die Beschaltung stellt sicher, dass bei geöffnetem Schalter U+ anliegt und bei geschlossenem Schalter der Pin auf GND gezogen wird
- Es treten keine undefinierten Zustände auf
- Exkurs: Mit Hilfe des ADC lässt sich in den nächsten Übungen der gedrückte Schalter bestimmen

2. Timer/Counter

2. Timer/Counter

- Anwendung von Timer/Counter
  - Zählen von externen Events: Tastendrücke/Flanken
  - Zeitbasis/Uhr: z.B. Kaffeemaschine/Ampel
- Timer/Counter sind separate Hardwarebausteine, die weitestgehend unabhängig vom Prozessor agieren
- Interaktion mit dem Hauptprogramm erfolgt i.d.R. über Interrupts [Exkurs: Strom sparen mit dem sleep mode]



Klassifikation/Konfiguration

- **Quellen:** Systemtakt, externe Taktgeneratoren oder Pins (z.B. Taster) an
- **Vorteiler (engl. Prescaler):** Erhöht die Zählerauflösung durch Zählen nur jedes x-ten Ereignisses/Taktes (z.B. x = 8; 32; 64; 128; 256; 1024)
- **Zähler:** Bitbreite(8 Bit, 16 Bit), Schrittweite (+1; +2; +4), Richtung (++/--) und Startwert t_0
- **Vergleich:** Überlauf des Zählerwertes, Vergleich gegen Null oder einer frei wählbaren Konstante
- **Ausgabe:** Interrupt, Setzen eines Flags, externe Pins, PWM-Einheit

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

Definition Zeitbasis

- **Zeitbasis** = Kleinste Zeiteinheit mit der präzise gearbeitet werden kann
- Beispiel: Zeitbasis 100 ms
  - Alle hundert Millisekunden kann ein Event ausgelöst werden, wie z.B.
    - Software Interrupt
    - Register Inkrement
    - Direkte Aktion in der ISR
  - Mit Zeitabständen von weniger als 100 ms kann nicht zuverlässig gearbeitet werden
  - Wenn Vielfache von 100 ms benötigt werden, müssen die Events gezählt werden

Auswahl Zeitbasis

- Zeitbasis muss gemäß dem zu lösenden Problem gewählt werden
  - Jede Sekunde soll eine LED blinken
    - Zeitbasis z.B. 1 s oder 100 ms
  - Alle 10 ms muss ein Wert gespeichert werde
    - Zeitbasis z.B. 10 ms oder 1 ms
- Nicht jede Zeitbasis kann problemlos mit jedem Timer erzeugt werden (Skript 4.3.4)
- Timer Interrupts möglichst selten aufrufen und möglichst simpel halten
  - Möglichst große Vorteiler – entlastet den Prozessor (sleep mode)
  - „runde“ Ticks – genauer und einfacher zu programmieren

Definition: Optimaler Vorteiler

- Aufgabe: Bestimmen Sie den **optimalen** Vorteiler p
  - Der Vorteiler p ist **optimal**, wenn das größtmögliches p gewählt wurde, für welches die Anzahl der verbleibender Ticks ganzzahlig ist.
  - Welche Werte für p gewählt werden können hängt vom Timer ab (siehe Datenblatt)
  - Ob verbleibende nicht ganzzahlige Ticks auftreten hängt von der Zeitbasis, dem Prescaler und der gewählten Clock-Frequenz ab

- Bitte genau lesen
  - Bestimmen Sie den optimalen Vorteiler

    vs.
  - Bestimmen Sie alle Vorteiler und begründen Sie, welcher optimal ist

Timer Berechnung

- Tafel
  - Timer Berechnung (siehe Skript)
- Beispiel im AVR Studio
  - Siehe timer.asm

State Machine

- State Machine/Zustandsmaschine
  - Zustände, Zustandsübergängen und Aktionen
- Implementierung
  - Speichern des Zustands in einem Register
  - Callback-Funktionen
- Für die Übung [Zustand in Register]: Zustandsübergang wird durch Ablauf einer Zeitspanne getriggert
  1. Warten auf ein Event der Zeitbasis
  2. Ausführen der Aktion, die zum Zustand gehört, der im Register gespeichert ist
  3. Update des Zustandregisters auf nächsten Zustand

Übung 10

- 1) Fragen zum Timer: siehe Skript/Datenblatt/Vorlesung
- 2) Flankendetektor: Ausprobieren bzw. Skript/Vorlesung. Bitte alle Konfigurationen durchspielen
- 3) Timer Berechnung + State Machine: Berechnung der Zeitbasis wie in der Übung/Skript + Programmieren einer State Machine