

# Interrupts

Projektlabor 2015  
Hans Eberhardt

# Gliederung

1. Nutzen, Polling vs. Interrupts
2. Verschiedenen Arten eines Interrupts
  1. Software
  2. Prozessor
  3. Hardware
3. Probleme

# Nutzen, Polling vs. Interrupts

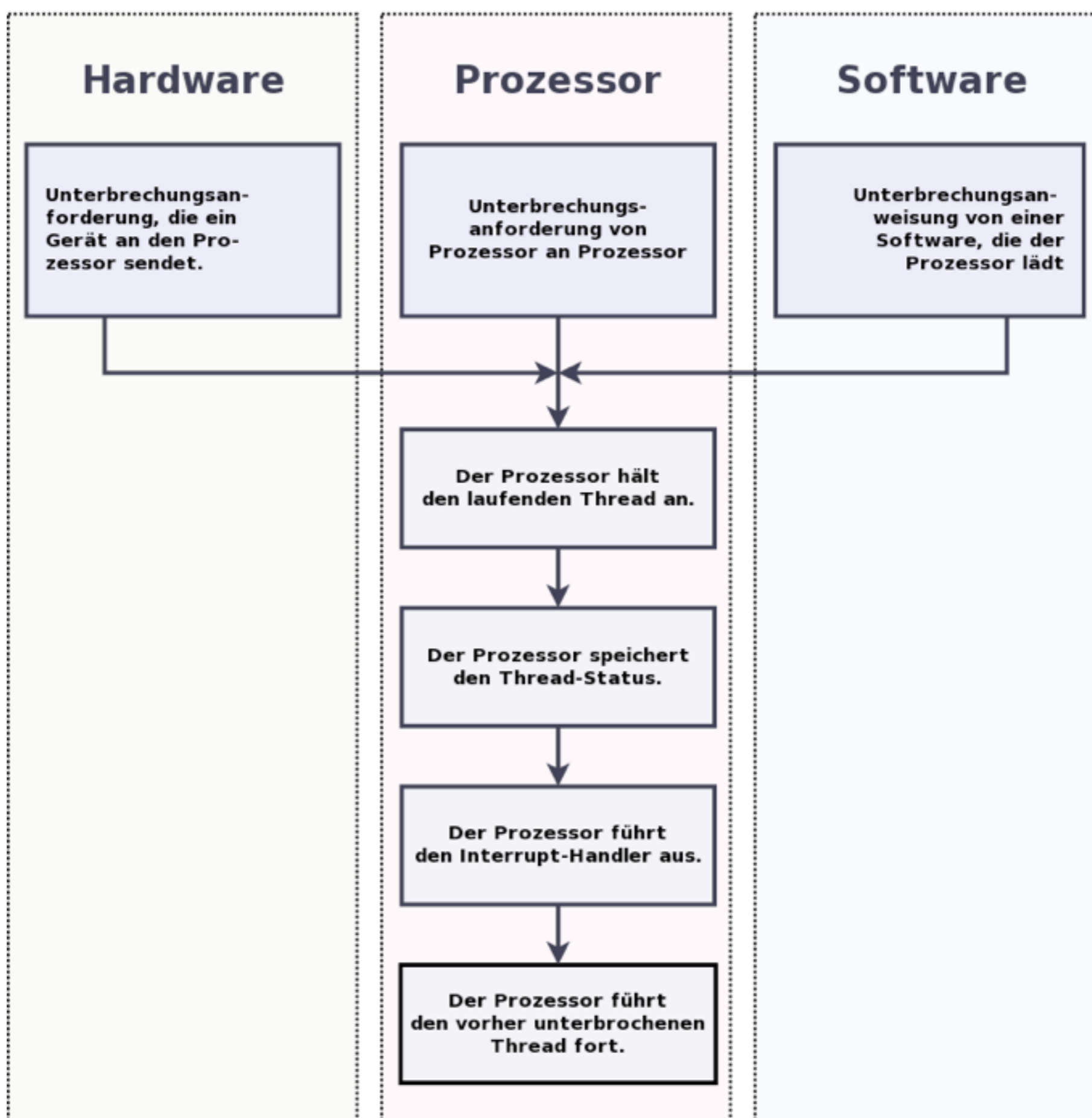
Programm mit Eingabe

Polling:

- Abfragen während Programmablauf mit if()
- Abfrage ist an vorbestimmten Stellen im Code

Interrupts:

- Sofortiges Anhalten des Hauptprogramms
- ausführen des Interrupts
- fortführen des Hauptprogramms



# Software Interrupt

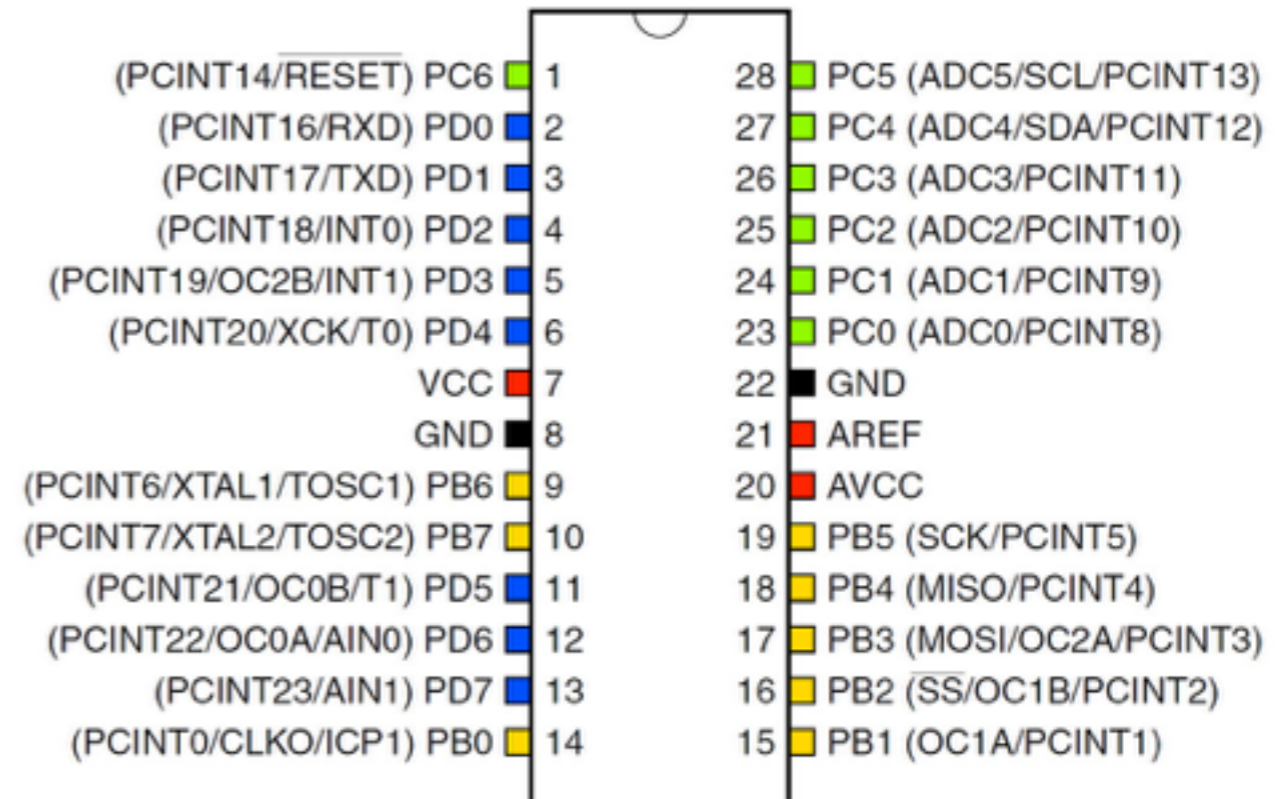
- von einem Programm gestartet
  - Antivirus check
  - automatisches Update

# Prozessor Interrupt

- Ein Prozessor stoppt einen anderen
  - Multicore PC, ausschalten
  - Aufwecken aus sleep-mode
  - “Schlaue” Sensoren
  - Real Time Clock

# Hardware Interrupt

- Von außerhalb des Prozessor
- Tastatureingabe
- Strg-Alt-Entf (NMI)
- Pegeländerung an einem Pin
- In unserem Fall...



# Problem beim Laden eines Wertes

```
int getVar (void){
    return var;
}
ISR(INT0_vect){
    var = var+1;
}

void main()
{
    var=0x00FF;
    --
    --
    --
    getVar()      ←—Interrupt
    --
    }
}
```

- Lowbyte von var wird geladen: 0xFF
- Interrupt wird ausgelöst
  - Zustand gespeichert
  - `var = var+1;` //var = 0x0100
  - Zustand geladen
- Highbyte von var wird geladen: 0x01
- `return 0x01FF`
- Fehler schwer erkennbar da nicht deterministisch

# Prioritäten

Beispiel: Medizin

- Anzeige mit Zeit
- Elektrischer Impuls zur Muskelanregung



# Zeitaufwand

- Interruptroutine muss kürzer sein als mittlere Periodendauer

```
bit flag;
```

```
ISR(INT2_vect){  
    flag=1;  
}
```

```
void main()  
{  
    #
```

```
    while(1)  
    {
```

```
        --
```

```
        --
```

```
        /* Normale Programmausführung */
```

```
        if (flag){
```

```
            flag = 0;
```

```
            /* Zeit umstellen */
```

```
        }
```

```
    }
```

```
}
```

Zwar ähnlich wie Polling aber  
ein Interrupt wird jedes mal ausgelöst

# Zwei Tricks zum Vermeiden von Fehlern

Schlüsselwort: `volatile`

Befehle `sei()` und `gci()`

# Bildquellen

- [http://upload.wikimedia.org/wikipedia/commons/c/cf/Interrupt\\_Process.PNG](http://upload.wikimedia.org/wikipedia/commons/c/cf/Interrupt_Process.PNG)
- [http://d1gsvnjtkwr6dd.cloudfront.net/large/IC-ATMEGA328-PU\\_02\\_LRG.jpg](http://d1gsvnjtkwr6dd.cloudfront.net/large/IC-ATMEGA328-PU_02_LRG.jpg)

# Informationsquellen

- <http://www.mikrocontroller.net/articles/Interrupt>
- <http://www.embedded.com/design/programming-languages-and-tools/4397803/Interrupts-short-and-simple--Part-1---Good-programming-practices>
- [http://www.linfo.org/software\\_interrupt.html](http://www.linfo.org/software_interrupt.html)
- <http://www.cs.toronto.edu/~demke/469F.06/Lectures/Lecture6.pdf>
- <http://www.webopedia.com/TERM/I/interrupt.html>

Fragen?