

Übungen zu Systemnahe Programmierung in C (SPiC) – Sommersemester 2018

Übung 6

Benedict Herzog
Sebastian Maier

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



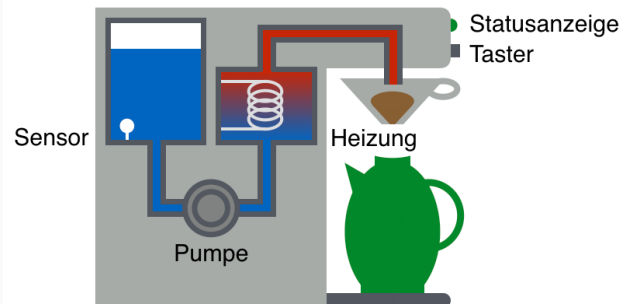
Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Hands-on: Kaffeemaschine

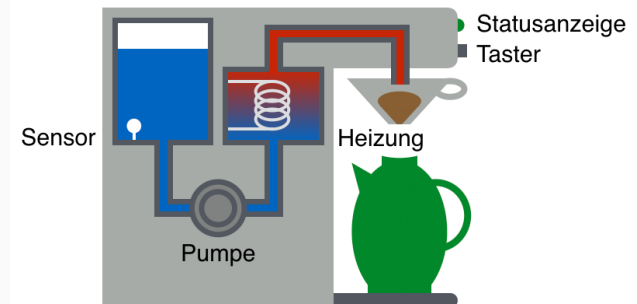
Hands-on: Kaffeemaschine (1)



■ Lernziele:

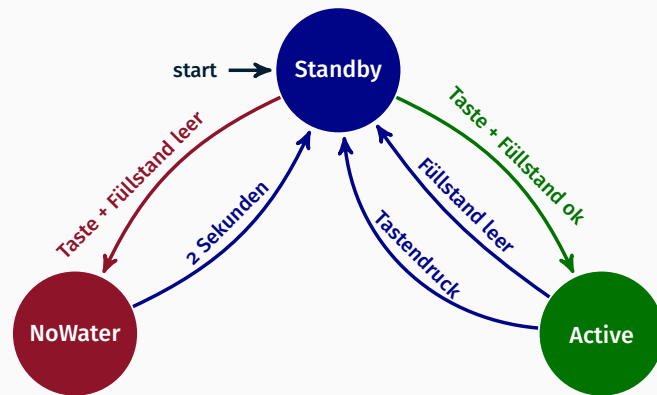
- Zustandsmaschine
- Interrupts & Schlafenlegen
- Timer bzw. Alarm

Hands-on: Kaffeemaschine (1)



■ Beschaltung:

- Pumpe & Heizung: Port D, Pin 5 (active-low)
- Taster: INTO an Port D, Pin 2 (active-low)
- Sensor: INT1 an Port D, Pin 3 (Wasser: high; kein Wasser: low)
- Statusanzeige: LEDs RED0, BLUE0 und GREEN0



- Hinweise:
 - Erkennung von Tastendruck & Füllstandsänderung durch Interrupts
 - Statusanzeige: `void setLEDState(km_state state)`
 - Wartephasen ggf. über Singleshot-Alarm realisieren
 - In Wartephasen Mikrocontroller in den Energiesparmodus versetzen

2

DDR_x hier konfiguriert man Pin i von Port x als Ein- oder Ausgang

- Bit i = 1 → Pin i als Ausgang verwenden
- Bit i = 0 → Pin i als Eingang verwenden

PORT_x Auswirkung **abhängig von DDR_x**:

- ist Pin i **als Ausgang konfiguriert**, so steuert Bit i im PORT_x Register ob am Pin i ein high- oder ein low-Pegel erzeugt werden soll
 - Bit i = 1 → high-Pegel an Pin i
 - Bit i = 0 → low-Pegel an Pin i
- ist Pin i **als Eingang konfiguriert**, so kann man einen internen pull-up-Widerstand aktivieren
 - Bit i = 1 → pull-up-Widerstand an Pin i (Pegel wird auf high gezogen)
 - Bit i = 0 → Pin i als tri-state konfiguriert

PIN_x Bit i gibt aktuellen Wert des Pin i von Port x an (nur lesbar)

3



- Interrupt Sense Control (ISC) Bits befinden sich beim ATmega328PB im External Interrupt Control Register A (EICRA)
- Position der ISC-Bits im Register durch Makros definiert

Interrupt 0		Interrupt bei	Interrupt 1	
ISC01	ISC00		ISC11	ISC10
0	0	low Pegel	0	0
0	1	beliebiger Flanke	0	1
1	0	fallender Flanke	1	0
1	1	steigender Flanke	1	1

- ATmega328PB: External Interrupt Mask Register (EIMSK)
- Die Bitpositionen in diesem Register sind durch Makros INTn definiert

4

Hands-on: Laufschrift



- Funktionsweise:
Schrittweises Anzeigen eines Textes auf der 7-Segment-Anzeige
- Lernziele:
 - Alarme & Schlafenlegen
 - Zeiger & Zeigerarithmetik
 - Zeichenfolgen in C
- Vorgehen:
 - Aktivieren eines wiederkehrenden Alarms (`sb_timer_setAlarm()`)
 - Zusammensetzen des aktuellen Teilstrings
 - Ausgabe über 7-Segment-Anzeige
 - In Wartephase Mikrocontroller in den Energiesparmodus versetzen

"hallo" $\equiv$ 

- Repräsentation von Strings
 - Zeiger auf erstes Zeichen der Zeichenfolge
 - null-terminiert: Null-Zeichen kennzeichnet Ende
- $\Rightarrow$ `strlen(s) !=` Speicherbedarf

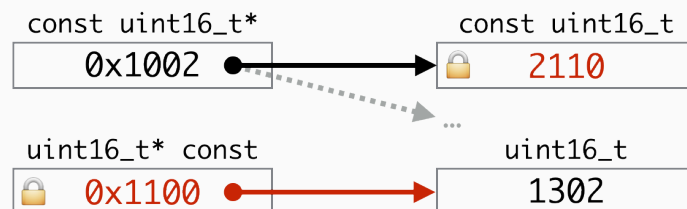
5

6

const uint8_t* vs. uint8_t* const



- `const uint8_t*`
 - ein Pointer auf einen `uint8_t`-Wert, der konstant ist
 - Wert nicht über den Pointer veränderbar
- `uint8_t* const`
 - ein **konstanter Pointer** auf einen (beliebigen) `uint8_t`-Wert
 - Pointer darf nicht mehr auf eine andere Speicheradresse zeigen



7