

# Übung zu Betriebssystembau (Ü BS)

## Tastatursteuerung in OO-Stubs

---

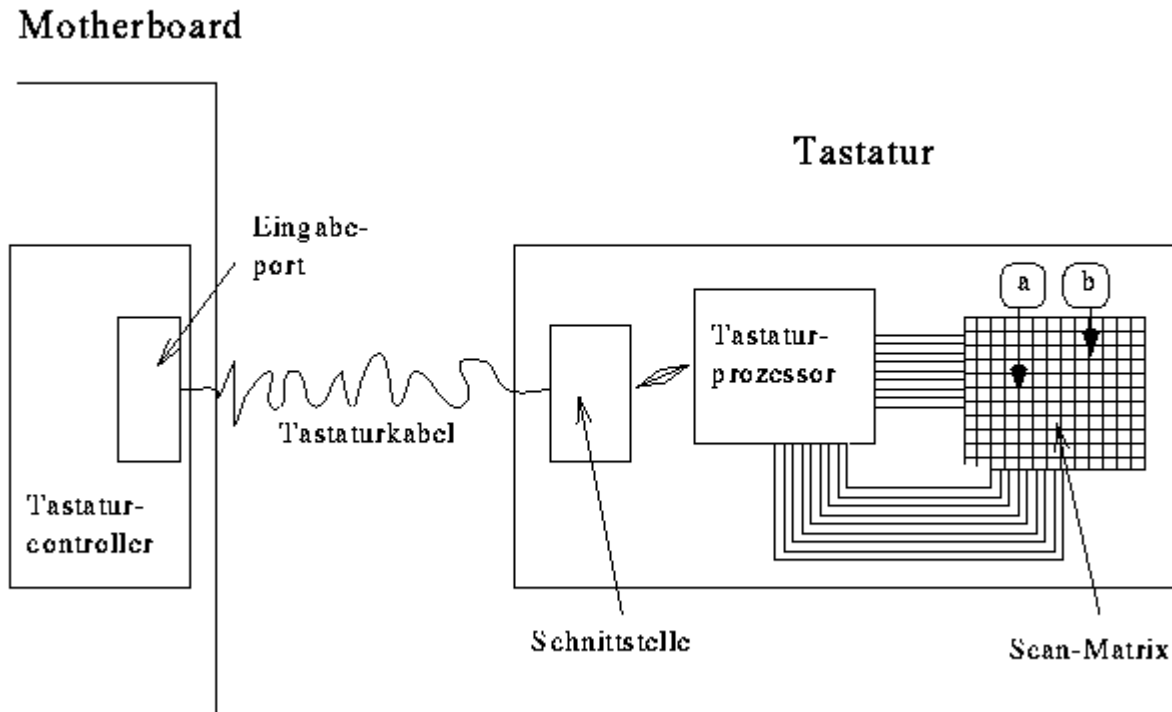
**Daniel Lohmann**

Lehrstuhl für Informatik IV

WS 06-07



# PC-Tastatur



# Tastenkodierungen

Darstellung im Programm:  
**Zeichencodes (ASCII)**

Zeichen	ASCII-Code
(	40
0	48
1	49
2	50
A	65
B	66
a	97

Darstellung in der Hardware:  
**Tastencodes**


Taste	Scan-Code
A	30
a	30
S	31
D	32
Cursor hoch	72
Cursor runter	80

- Jede Taste hat einen eindeutigen Tastencode (Scan-Code)
  - Scan-Code ist 7-Bit Zahl (max 128 Tasten)
- Tastaturcontroller sendet zusätzliche Informationen
  - Make-Code beim Drücken / Halten einer Taste
  - Break-Code beim Loslassen



# Make- und Break-Codes

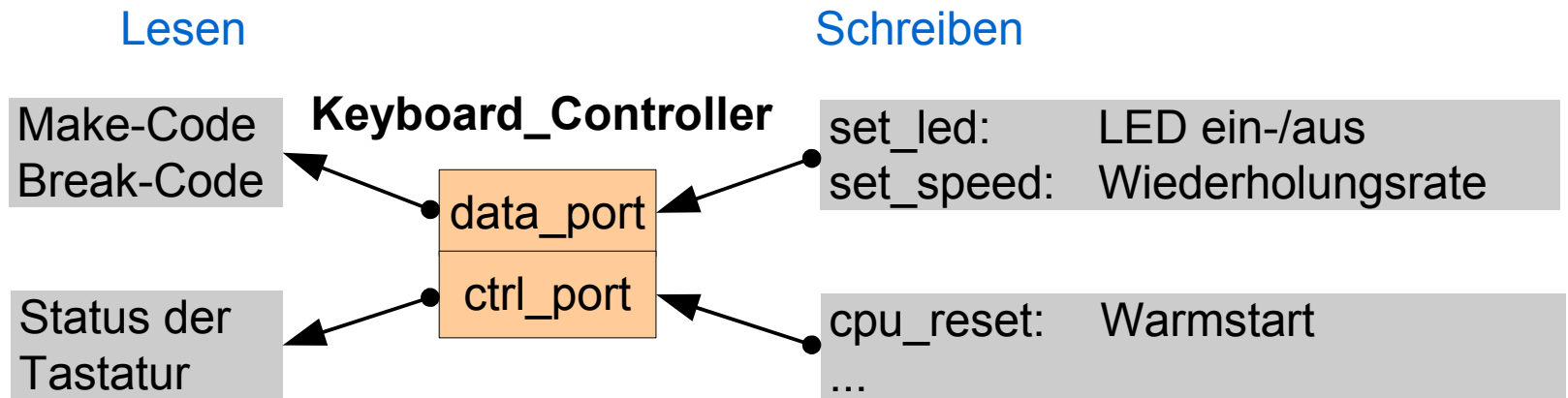
---

- Üblicherweise gilt
    - Make-Code (Taste gedrückt) = Scan-Code
    - Break-Code (Taste losgelassen) = Scan-Code + 128 (Bit 7)
  - Einige Tasten senden jedoch mehrere Codes
    - z.B. Funktionstasten (F1-F12)
    - aus historischen Gründen (XT-Tastatur)
    - bis zu drei Make/Break-Codes pro Taste
  - Eingebaute Wiederholungsfunktion
    - **Hardware** sendet zusätzliche Make-Codes, wenn eine Taste länger gehalten wird
-  Dekodierung ist mühsam
- bei OOSTubs in der Vorgabe enthalten: `bool key_decoded()`



# Datenaustausch mit der Tastatur

- Tastaturcontroller wird über zwei Ports angesprochen
  - Ein-/Ausgaberegister (data\_port) **0x60**
  - Steuerregister (ctrl\_port) **0x64**



# Status der Tastatur

---

- Statusregister stellt folgende Informationen bereit:

Bit	Maske	Name	Bedeutung
0	0x01	outb	Gesetzt, wenn ein Zeichen im Ausgabepuffer des Tastaturcontrollers zum Lesen bereit steht
1	0x02	inpb	Gesetzt, solange der Tastaturcontroller ein von der CPU geschriebenes Zeichen noch nicht abgeholt hat
5	0x20	auxb	Gesetzt, wenn der Wert im Ausgabepuffer von der Maus und nicht von der Tastatur stammt



# Status der Tastatur - Verwendung

Bit	Maske	Name	Bedeutung
0	0x01	outb	Gesetzt, wenn ein Zeichen im Ausgabepuffer des Tastaturcontrollers zum Lesen bereit steht
1	0x02	inpb	Gesetzt, solange der Tastaturcontroller ein von der CPU geschriebenes Zeichen noch nicht abgeholt hat
5	0x20	auxb	Gesetzt, wenn der Wert im Ausgabepuffer von der Maus und nicht von der Tastatur stammt

- Aktive Tastaturabfrage (nicht durch Unterbrechungen):
  - Warten, bis outb in ctrl\_port gesetzt ist
  - Make-/Break-Code vom data\_port lesen (löscht ctrl\_port.outb)
- Tastatur programmieren (set\_led, set\_speed)
  - Befehlsbyte in data\_port schreiben
  - Tastatur antwortet mit ack (0xfa), ggfs. auf Antwort warten (wie oben)
  - Datenbyte in data\_port schreiben (LED-Codes, Repeat-Rate)
  - Tastatur antwortet mit ack, ggfs. auf Antwort warten



# Tastatur programmieren

- `set_led` **0xed**, `<led_mask>` in `data_port`
- `set_speed` **0xf3**, `<config_byte>` in `data_port`

Parameter für `set_led` Befehl: (`led_mask`)

Bit 7	6	5	4	3	2	1	0
Always 0	Always 0	Always 0	Always 0	Always 0	Caps Lock	Num Lock	Scroll Lock

Parameter für `set_speed` Befehl: (`config_byte`)

Bits 5 und 6 (hex)	Verzögerung (s)
0x00	0.25
0x01	0.5
0x02	0.75
0x03	1.0

Bits 0-4 (hex)	Wiederholungsrate (cps)
0x00	30
0x02	25
0x04	20
0x08	15
0x0c	10
0x10	7
0x14	5

