

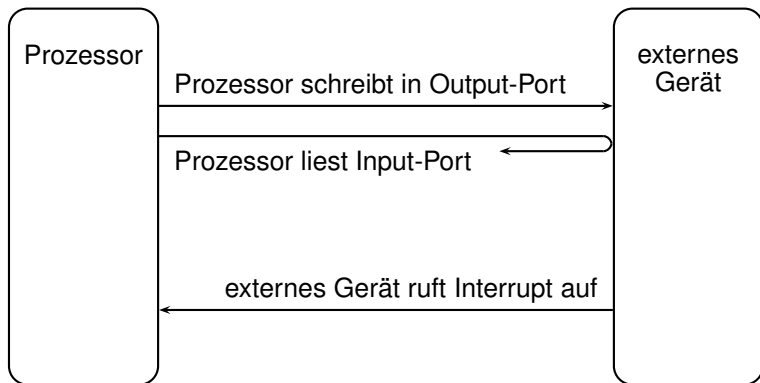
Grundlagen Rechnertechnik

Prof. Dr. Peter Gerwinski

13. Dezember 2012

5 Hardwarenahe Programmierung

Kommunikation mit externen Geräten



5.2 I/O-Ports

In Output-Port schreiben = Leitungen ansteuern

Datei: `RP6Base/RP6Base_Examples/RP6Examples_20080915/
RP6Lib/RP6base/RP6RobotBaseLib.c`

Suchbegriff: `setMotorDir`

```
void setMotorDir(uint8_t left_dir, uint8_t right_dir)
```

```
{
```

```
    /* ... */
```

```
    if(left_dir)
```

```
        PORTC |= DIR_L;
```

```
    else
```

```
        PORTC &= ~DIR_L;
```

```
    if(right_dir)
```

```
        PORTC |= DIR_R;
```

```
    else
```

```
        PORTC &= ~DIR_R;
```

```
}
```

Manipulation einzelner Bits

→ Steuerung der Motordrehrichtung

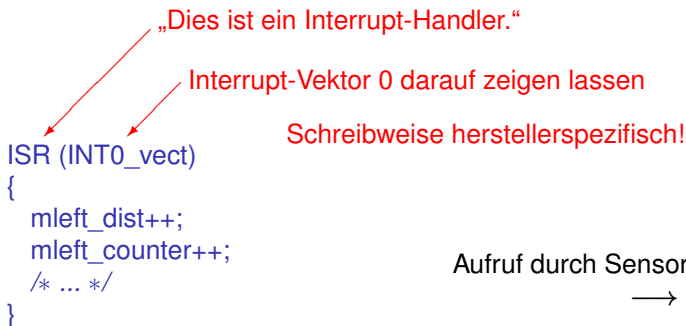
Output-Port

5.3 Interrupts

Externes Gerät ruft (per Stromsignal) Unterprogramm auf
Zeiger hinterlegen: „Interrupt-Vektor“

Datei: [RP6Base/RP6Base_Examples/RP6Examples_20080915/
RP6Lib/RP6base/RP6RobotBaseLib.c](#)

Suchbegriff: ISR



```
ISR (INT0_vect)
{
    mleft_dist++;
    mleft_counter++;
    /* ... */
}
```

„Dies ist ein Interrupt-Handler.“

Interrupt-Vektor 0 darauf zeigen lassen

Schreibweise herstellerspezifisch!

Aufruf durch Sensor an Encoder-Scheibe
→ Entfernungsmessung

5.4 volatile-Variable

```
volatile uint16_t mleft_counter;
```

 „Immer lesen und schreiben. Nicht wegoptimieren.“

```
volatile uint16_t mleft_dist;
```

```
/* ... */
```

```
ISR (INT0_vect)
```

```
{  
    mleft_dist++;  
    mleft_counter++;  
    /* ... */  
}
```

```
int main (void)
```

```
{  
    int prev_mleft_dist = mleft_dist;  
    while (mleft_dist == prev_mleft_dist)  
        /* just wait */;  
}
```

5.5 Software-Interrupts

```
mov ax, 0012
```

```
int 10
```

5.5 Software-Interrupts

mov ax, 0012 ← Parameter in Prozessorregister

int 10 ← Funktionsaufruf über Interrupt-Vektor

5.5 Software-Interrupts

`mov ax, 0012` ← Parameter in Prozessorregister

`int 10` ← Funktionsaufruf über Interrupt-Vektor

Beispiel: VGA-Grafikkarte

- Modus setzen: `mov ah, 00`
- Grafikmodus: `mov al, 12`
- Textmodus: `mov al, 03`

5.5 Software-Interrupts

`mov ax, 0012` ← Parameter in Prozessorregister
`int 10` ← Funktionsaufruf über Interrupt-Vektor

Beispiel: VGA-Grafikkarte

- Modus setzen: `mov ah, 00`
- Grafikmodus: `mov al, 12`
- Textmodus: `mov al, 03`

Verschiedene Farben: Output-Ports

- *Graphics Register*: Index `03CE`, Daten `03CF`
- Index 0: *Set/Reset Register*
- Index 1: *Enable Set/Reset Register*
- Index 8: *Bit Mask Register*
- Jedes Bit steht für Schreibzugriff auf eine Speicherbank.
- 4 Speicherbänke → 16 Farben