

Heizungsregelung

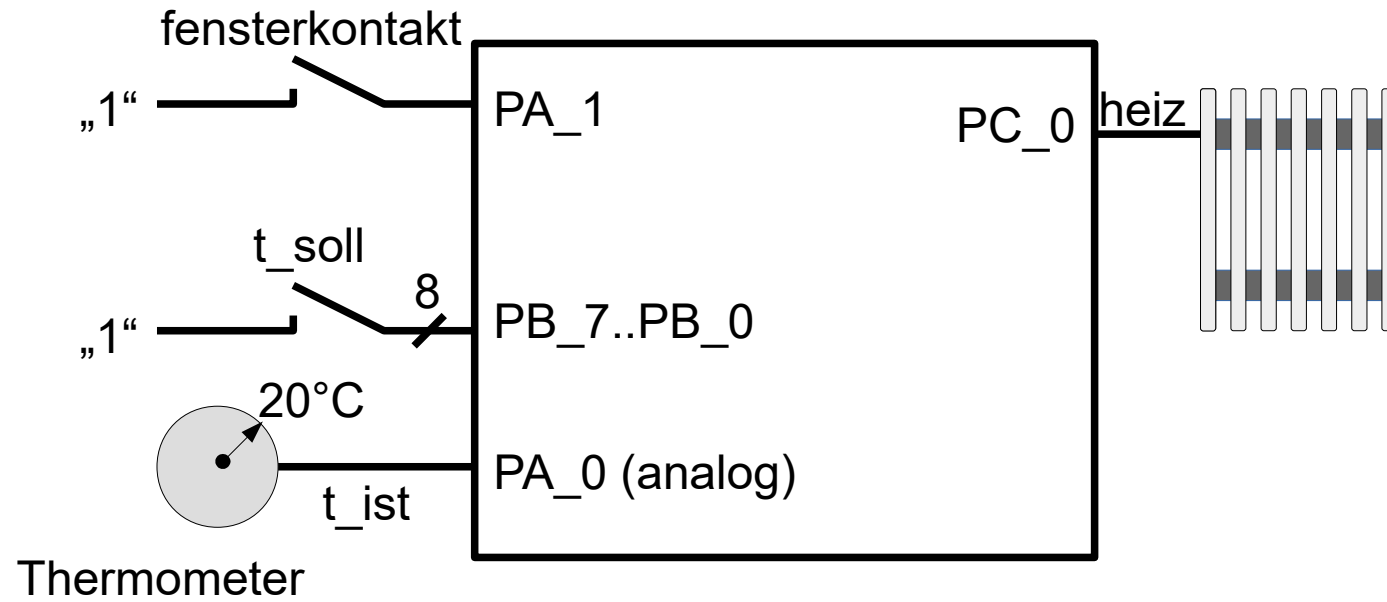
Mit Interrupt und Bedingung



Ich bin Mik, Dein Mikrocontroller

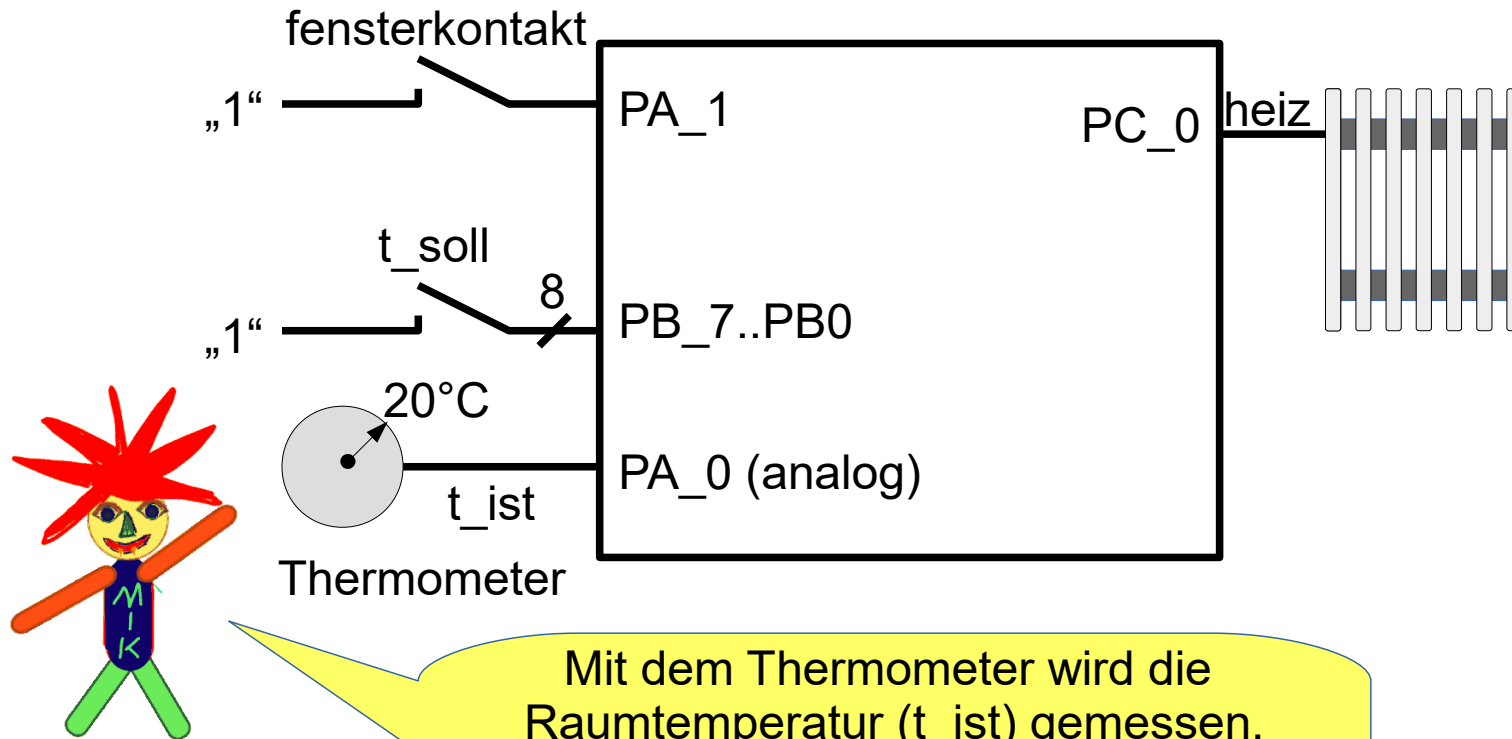


Heizungsregelung



Die Raumtemperatur soll geregelt werden.

Heizungsregelung

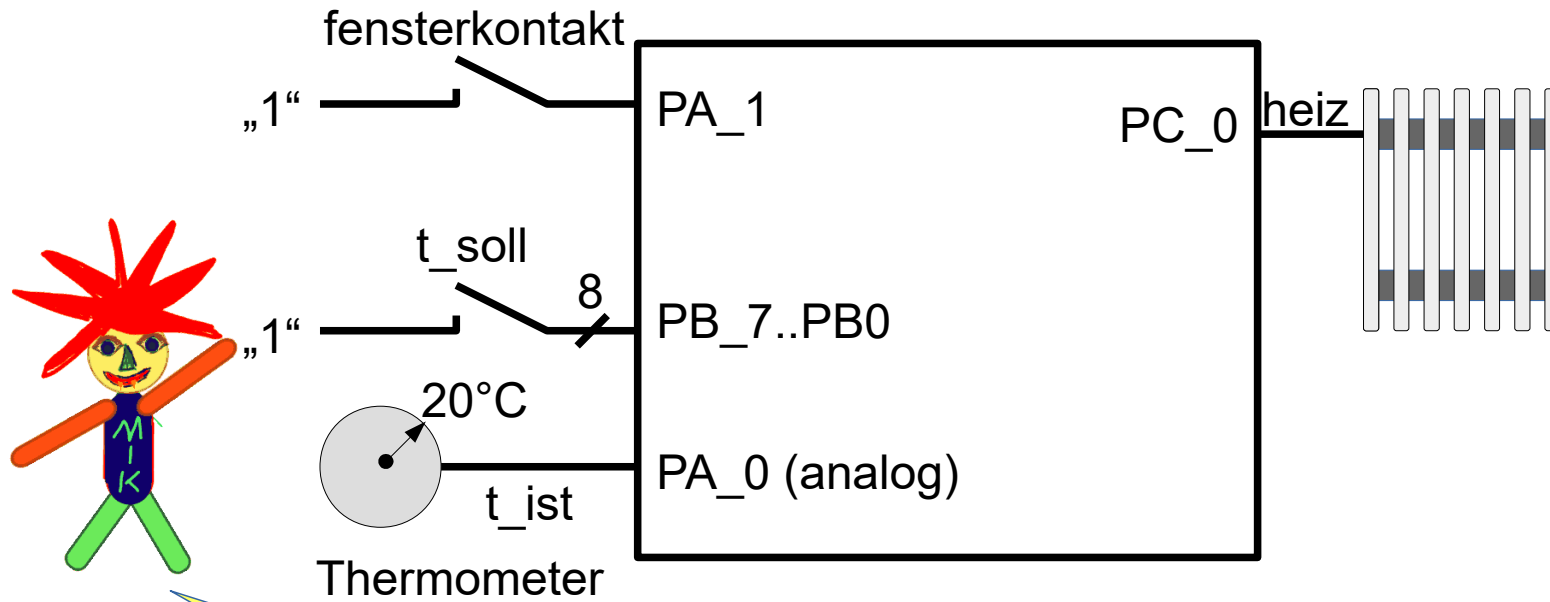


Mit dem Thermometer wird die Raumtemperatur (t_{ist}) gemessen.

Programmierung:

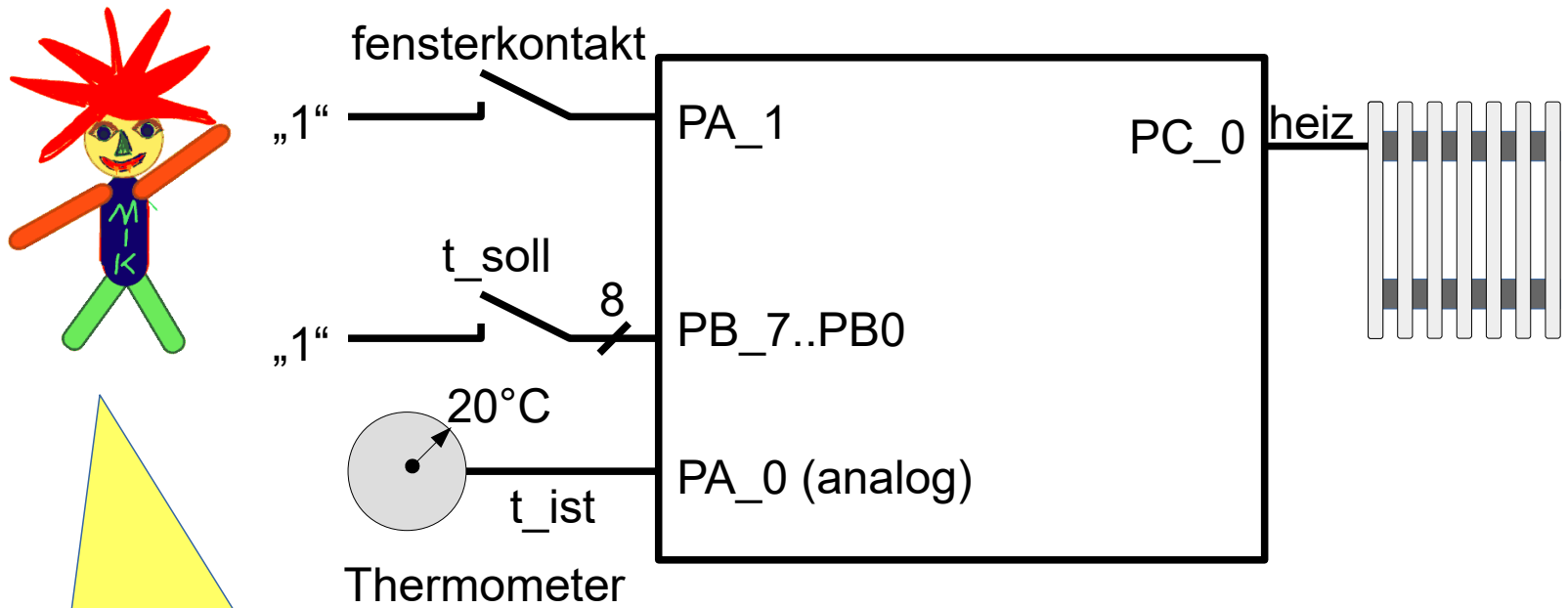
```
#define t_ist Thermometer*50  
HSAnalogIn Thermometer(PA_0);
```

Heizungsregelung



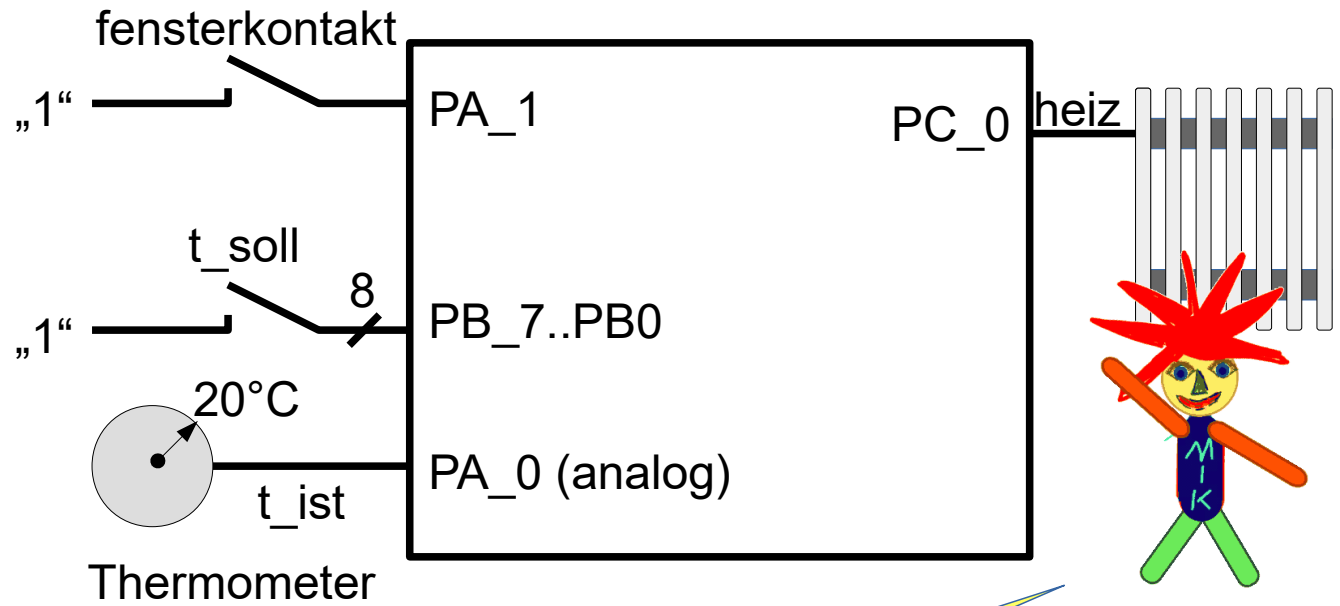
Die Wunsch- oder Solltemperatur t_{soll} wird an 8 DIP-Schaltern eingestellt:
`PortIn t_soll(PortB,0xFF);`

Heizungsregelung



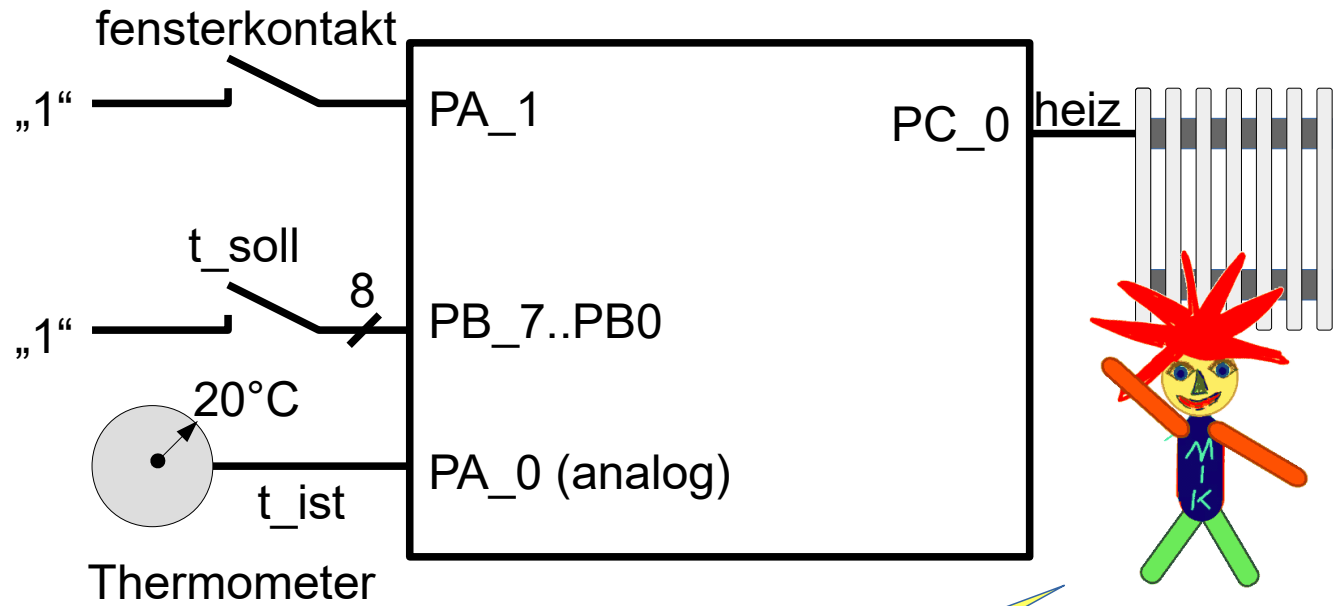
Wenn das Fenster offen ist
(fensterkontakt=0)
Wird die Heizung ausgeschaltet (heiz=0)

Heizungsregelung



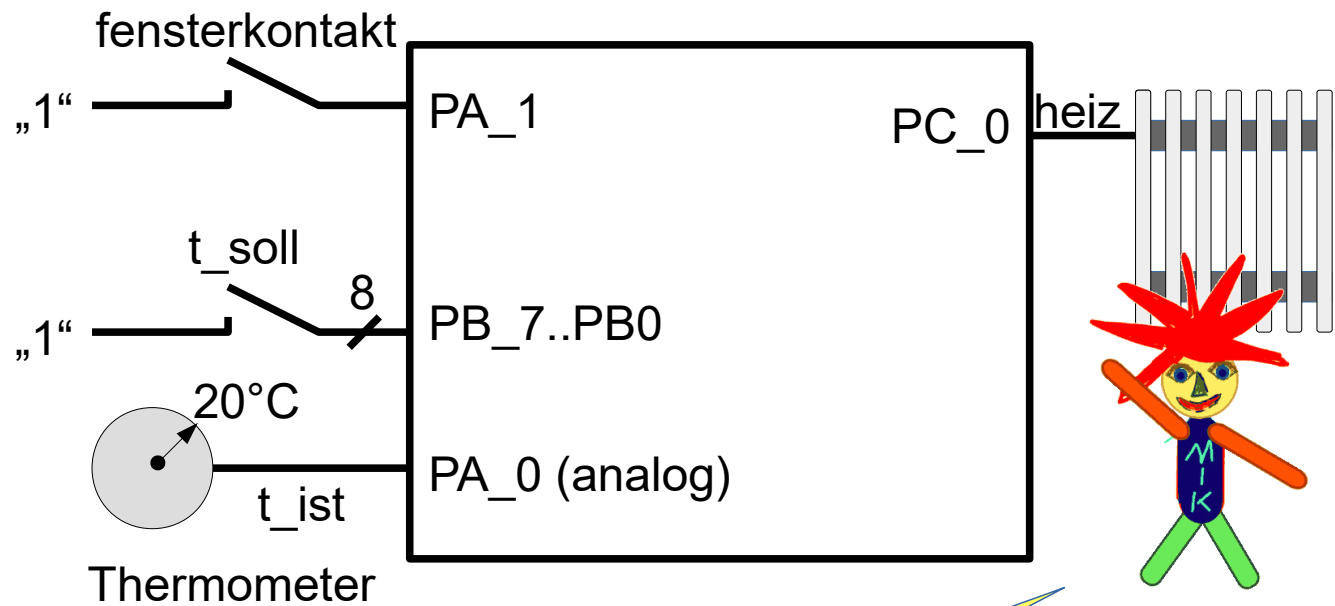
heiz schaltet die Heizung:
heiz=1; //Heizung Ein
heiz=0; //Heizung Aus

Heizungsregelung



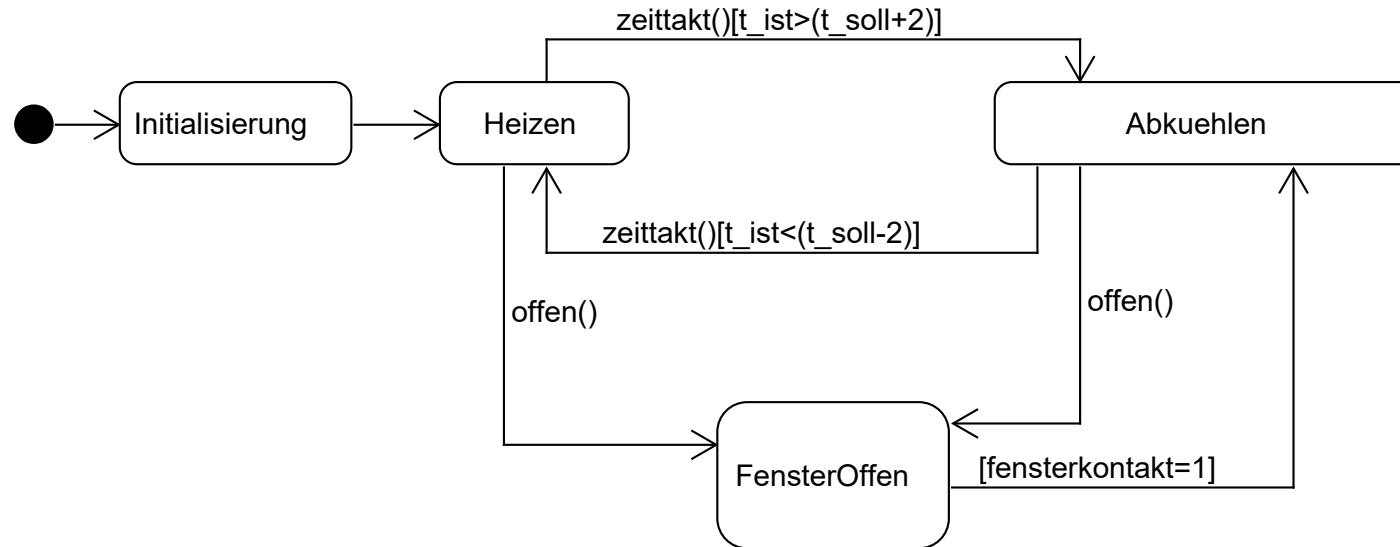
Die Temperatur soll auf $\pm 2^\circ\text{C}$ genau geregelt werden:
wenn $t_{\text{ist}} > (t_{\text{soll}} + 2) \Rightarrow \text{heiz} = 0$
wenn $t_{\text{ist}} < (t_{\text{soll}} - 2) \Rightarrow \text{heiz} = 1$

Heizungsregelung



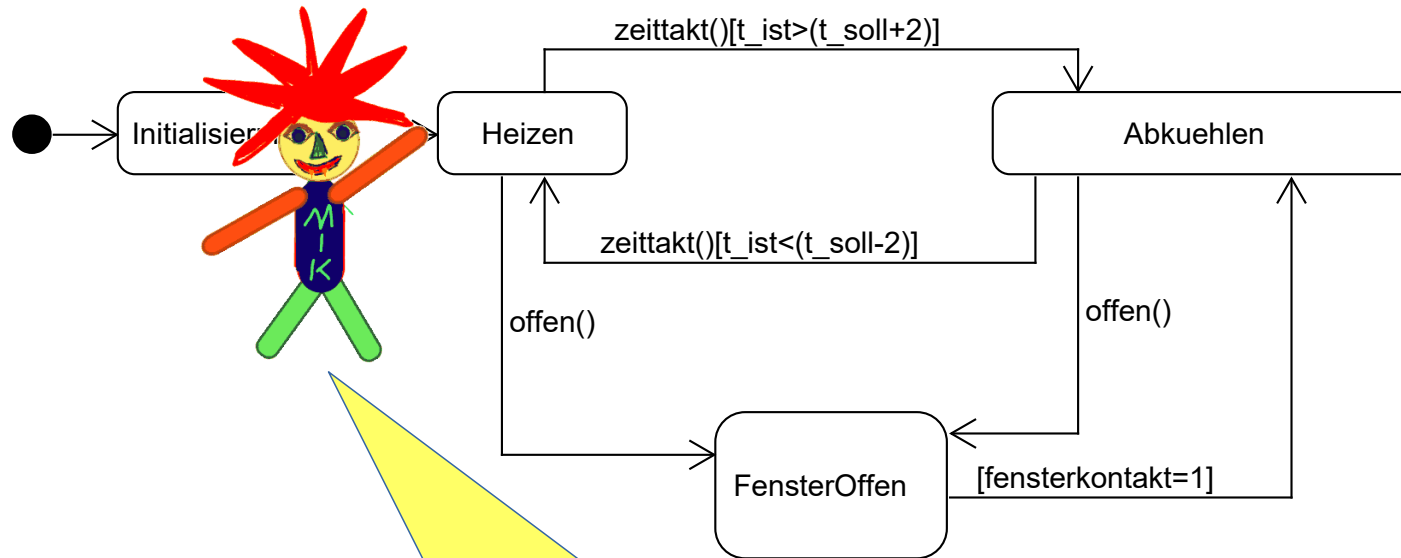
Damit die Heizung nicht zu oft schaltet sollen, mit einem Timer, alle 1 Sekunde die Temperaturen geprüft werden und „heiz“ entsprechend geschaltet werden.

Heizungsregelung



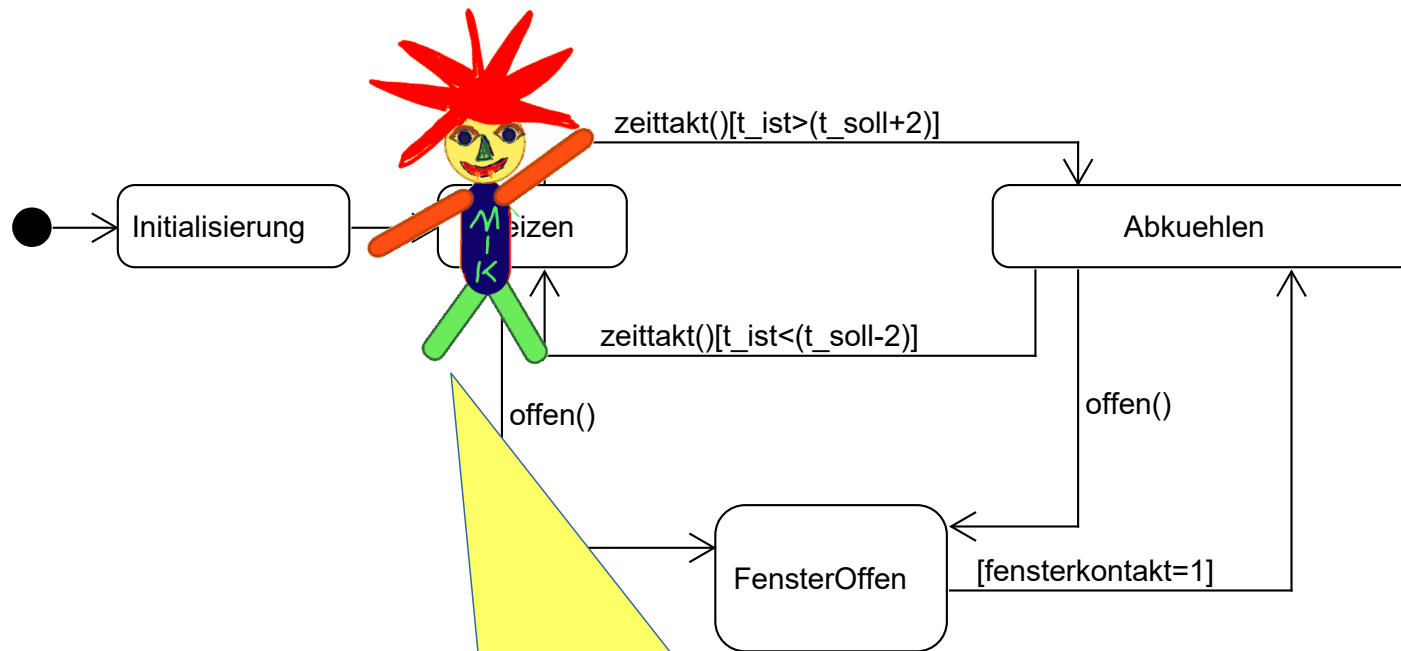
Ein Kollege hat sich bereits ein Zustandsdiagramm ausgedacht. Wir müssen es nur noch programmieren.

Heizungsregelung



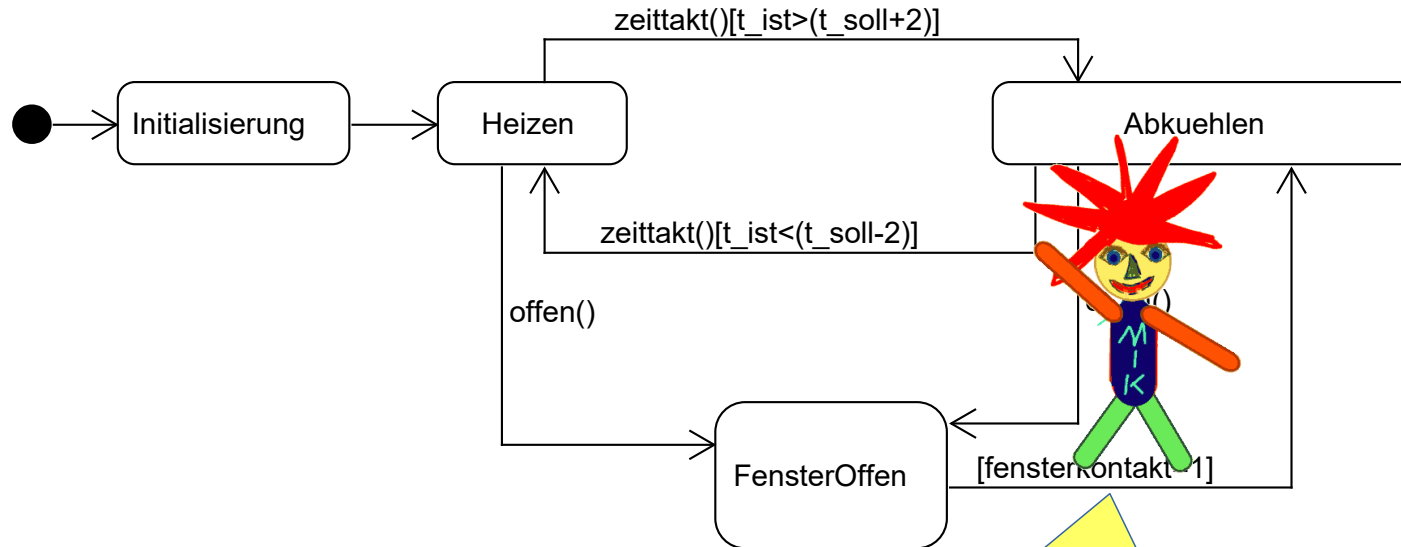
Nach der Initialisierung kommen wir in den Zustand Heizen.

Heizungsregelung



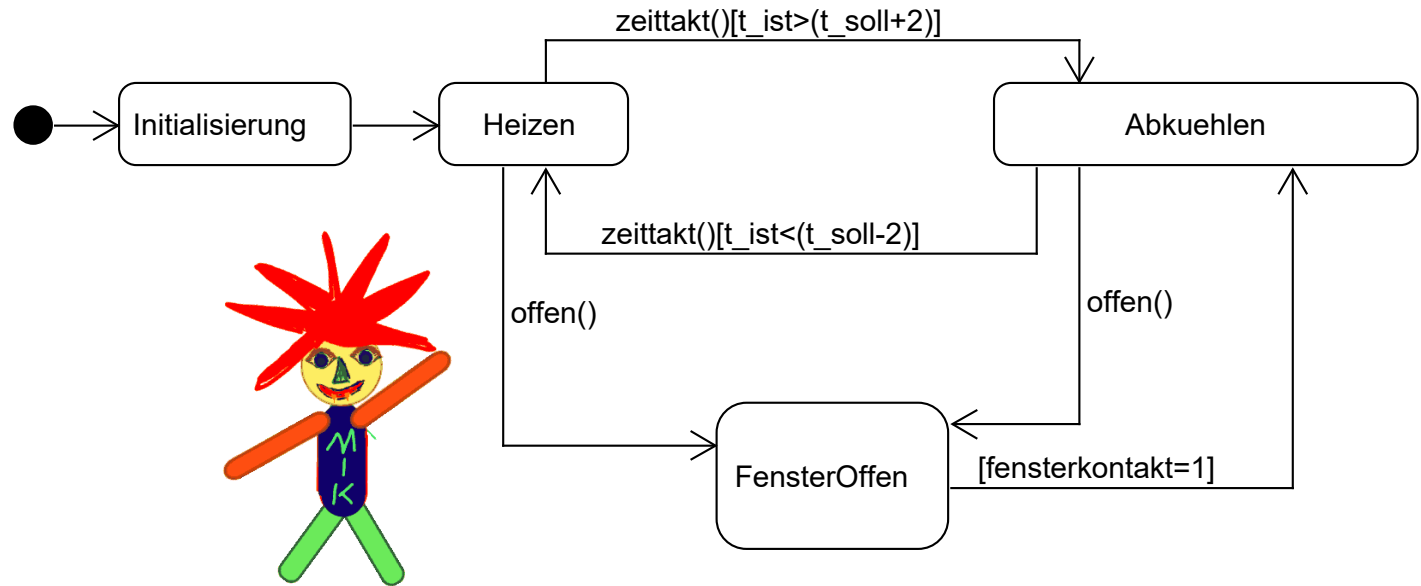
Wenn beim `zeittakt()` die Ist-Temperatur größer als die Soll-Temperatur + 2°C ist, wird in den Zustand „Abkuehlen“ gewechselt.

Heizungsregelung



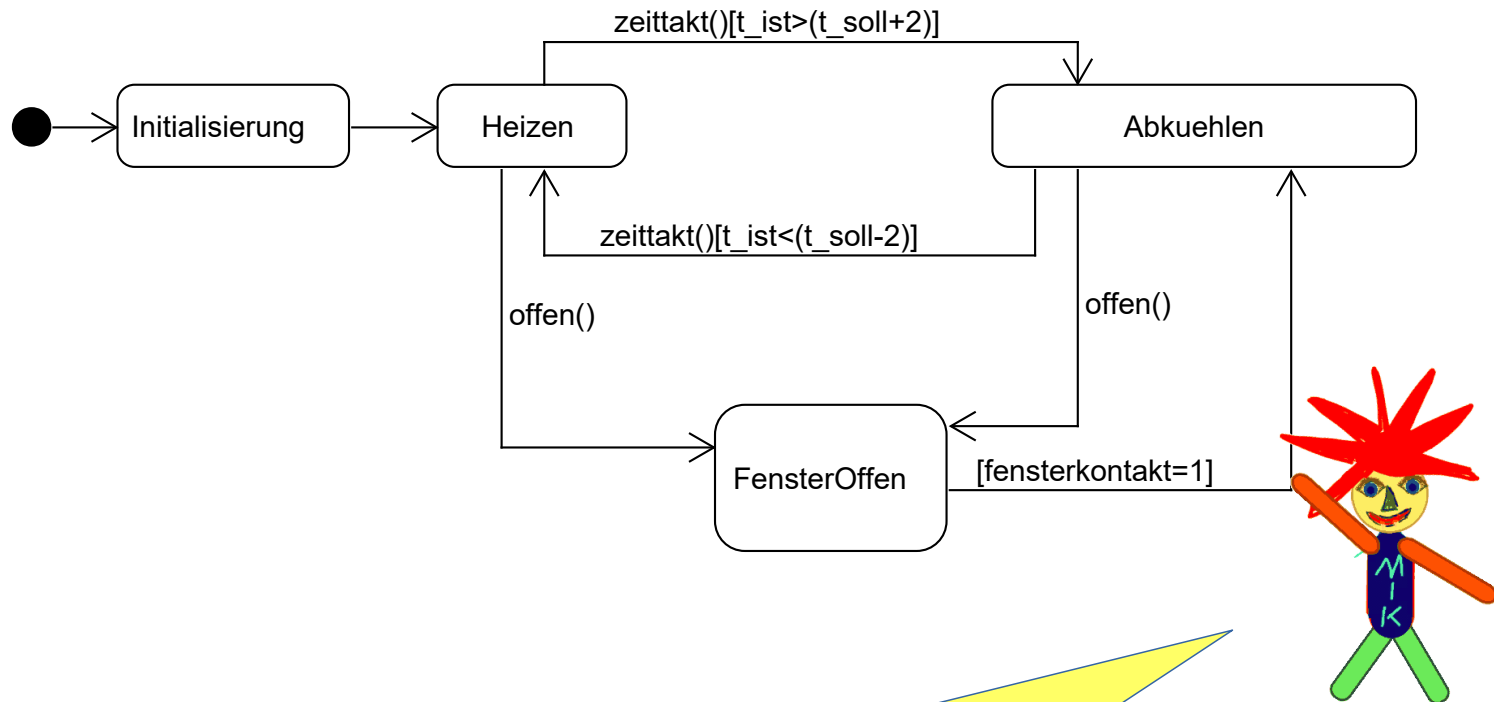
Wenn im Zustand „Abkuehlen“
beim `zeittakt()` die Ist-Temperatur
kleiner als die Soll-Temperatur -2°C ist,
wird in den Zustand „Heizen“ gewechselt.

Heizungsregelung



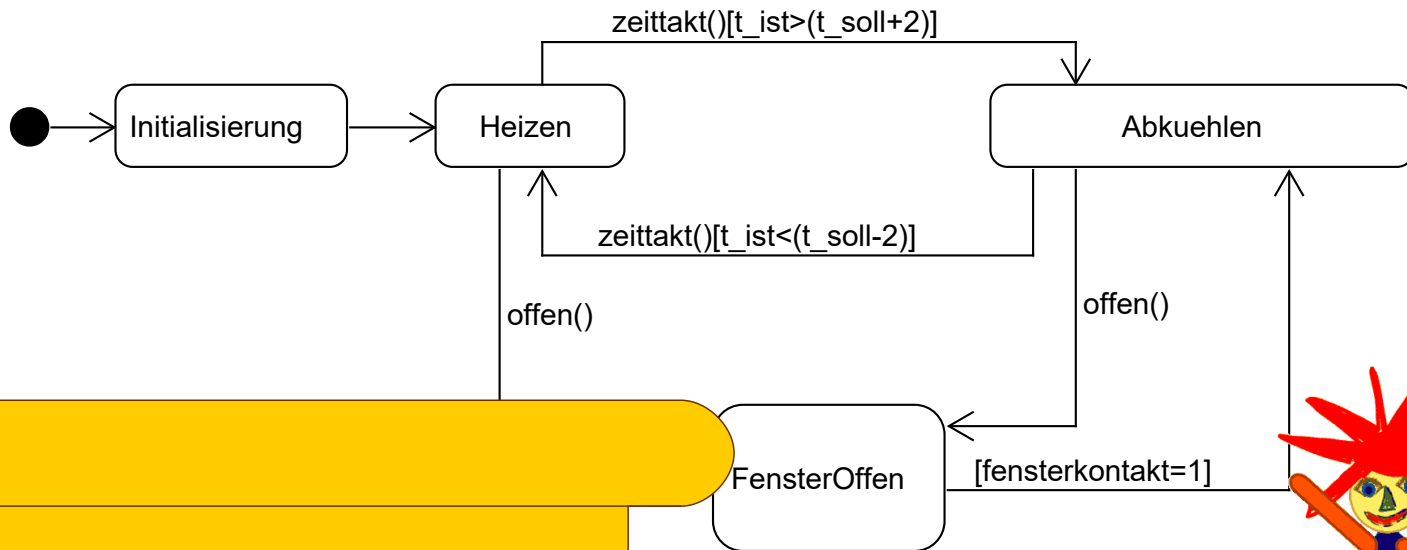
Wenn das Fenster geöffnet wird, wechselt das Programm in den Zustand „FensterOffen“

Heizungsregelung



Solange das Fenster offen ist ($fensterkontakt=0$)
bleibt der Mikrocontroller im
Zustand „FensterOffen“

Heizungsregelung



```
#define Heizen 0b01
#define Akuehlen 0b00
#define FensterOffen 0b10
```

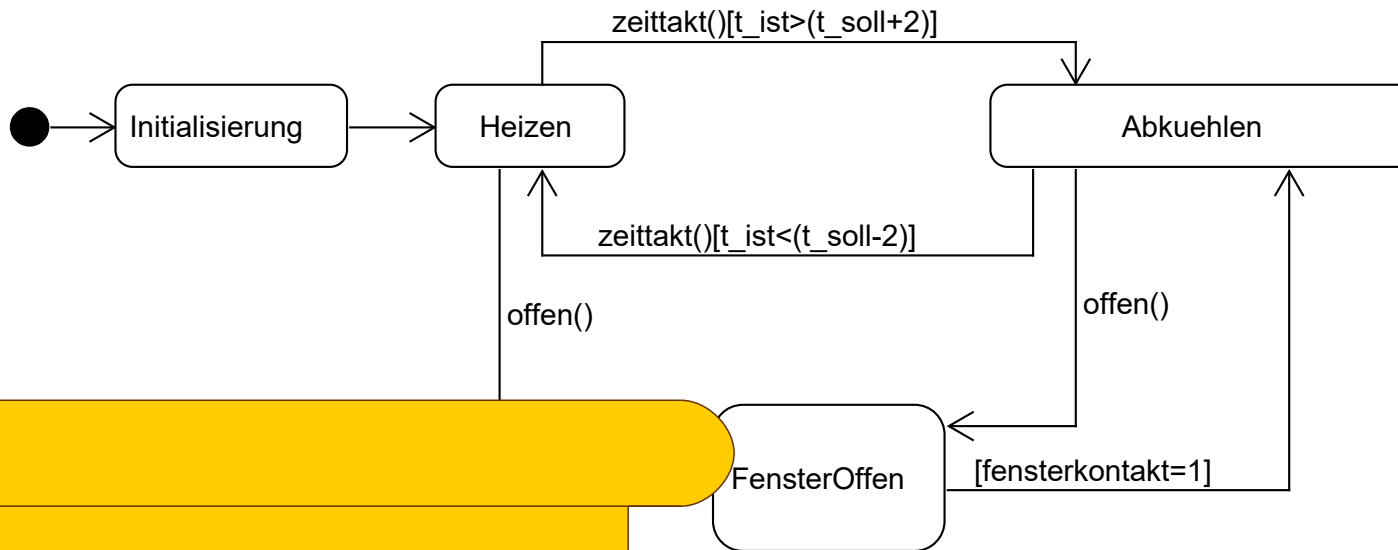
```
#define t_ist (int)(thermometer*50)
```

```
PortOut zustand(PortC,0b11);
HSAAnalogIn thermometer(PA_0);
PortIn t_soll(PortB,0xFF);
InterruptIn fensterkontakt(PA_1);
```

Wir definieren die Zustände
und die Anschlüsse:



Heizungsregelung



```
#define Heizen 0b01
#define Akuehlen 0b00
#define FensterOffen 0b10
```

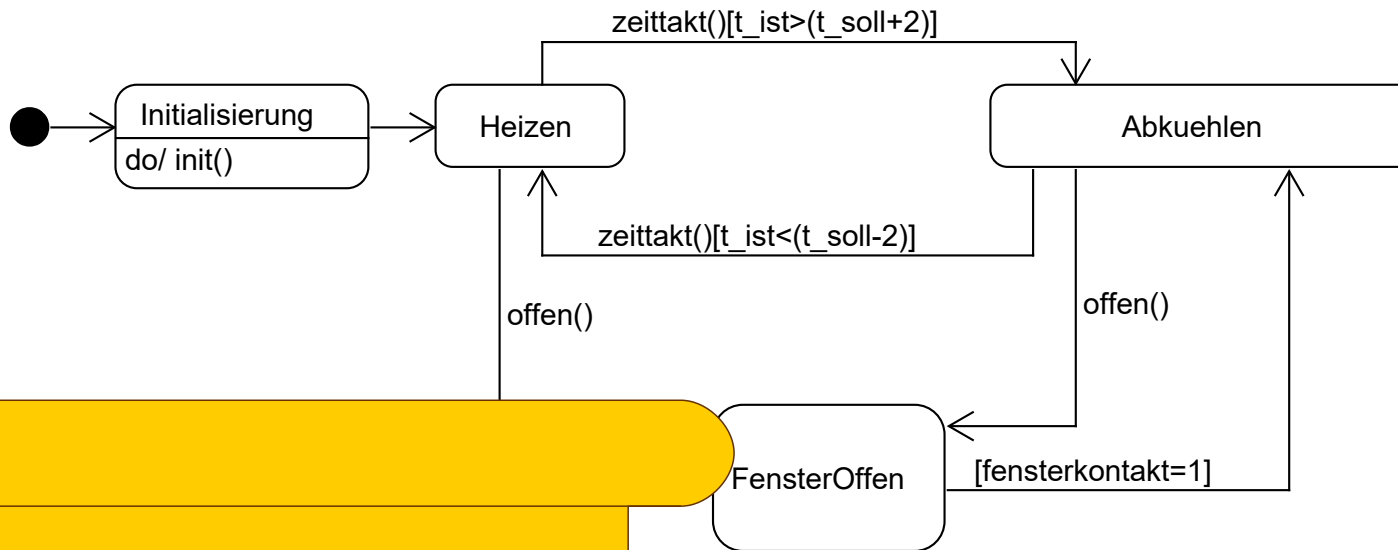
```
#define t_ist (int)(thermometer*50)
```

```
PortOut zustand(PortC,0b11);
HSAAnalogIn thermometer(PA_0);
PortIn t_soll(PortB,0xFF);
InterruptIn fensterkontakt(PA_1);
```



Wir verwenden PortC direkt als Zustandsvariable. D.h. Bei Zustandswechsel müssen wir uns nicht mehr um die Ausgabe kümmern

Heizungsregelung



```
#define Heizen 0b01
#define Abkuehlen 0b00
#define FensterOffen 0b10
```

```
#define t_ist (int)(thermometer*50)
```

```
PortOut zustand(PortC,0b11);
HSAAnalogIn thermometer(PA_0);
PortIn t_soll(PortB,0xFF);
InterruptIn fensterkontakt(PA_1);
```



Wir verwenden PortC direkt als Zustandsvariable. D.h. Bei Zustandswechsel müssen wir uns nicht mehr um die Ausgabe kümmern

Heizungsregelung

Die Initialisierung



Initialisierung
do/ init()

Heizen

Abkuehlen

zeittakt()[t_ist>(t_soll+2)]

zeittakt()[t_ist<(t_soll-2)]

kontakt=1]

```
void init()
```

```
{
```

```
    t_soll.mode(PullDown);
```

```
    fensterkontakt.mode(PullDown);
```

```
    fensterkontakt.fall(&offen);
```

```
    RCC->APB1ENR |= 0b10000; //TIM6
```

```
    TIM6->PSC=31999; //1ms
```

```
    TIM6->ARR=1000; //1s
```

```
    TIM6->CNT=0; //Counter startet bei 0
```

```
    TIM6->DIER=1; //UIE=1
```

```
    TIM6->SR=0; //UIF=0
```

```
    TIM6->CR1=1; //CEN=1
```

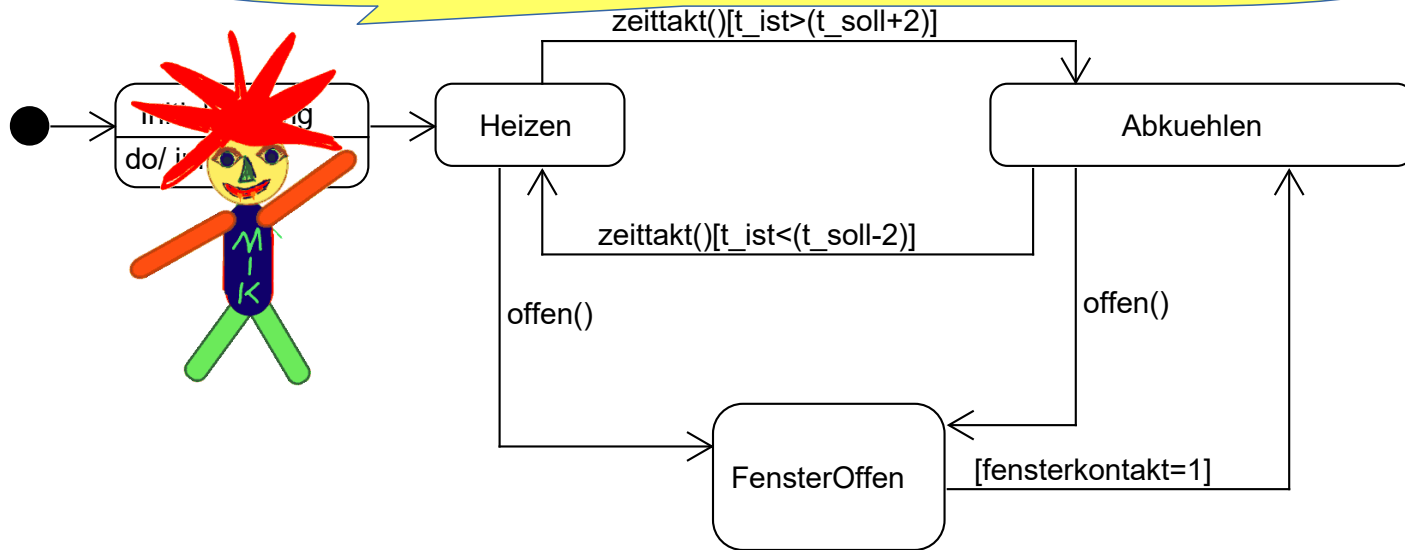
```
    NVIC_SetVector(TIM6_IRQn, (uint32_t)&zeittakt);
```

```
    HAL_NVIC_EnableIRQ(TIM6_IRQn);
```

```
}
```

Heizungsregelung

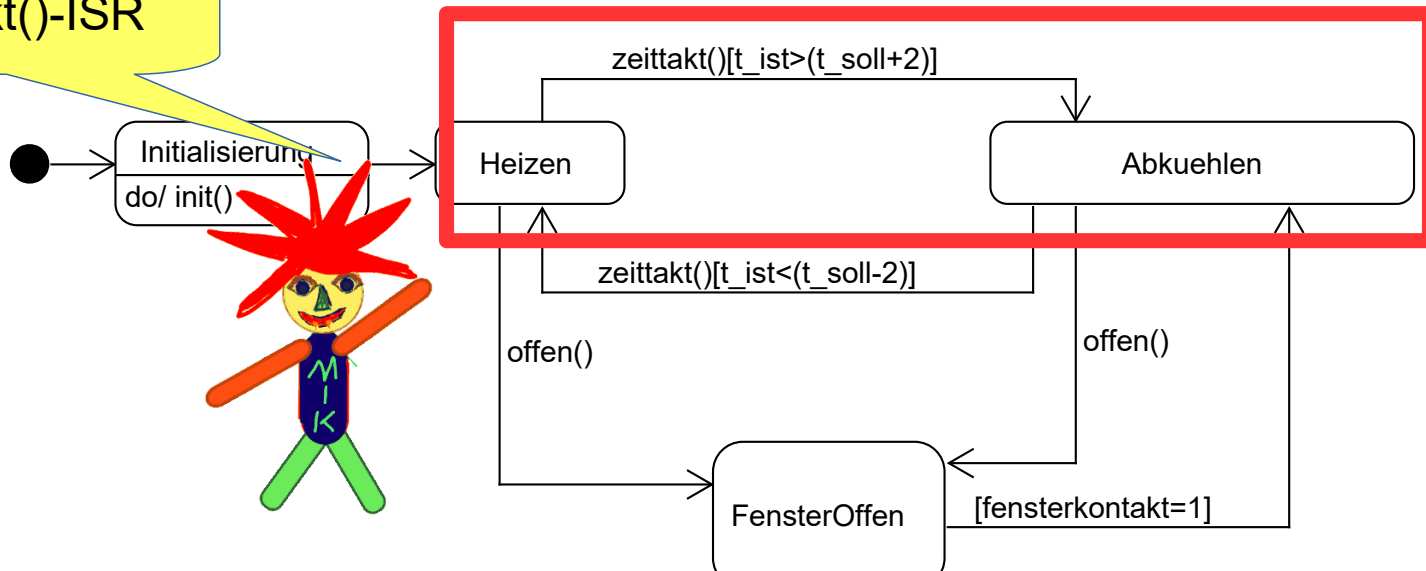
Nach der Initialisierung wechseln wir in Zustand Heizen



```
void main(void)
{
    init();
    zustand=Heizen;
    ...
}
```

Heizungsregelung

Die zeittakt()-ISR



```
void zeittakt()
```

```
{
```

```
    switch (zustand)
```

```
    {
```

```
        case Heizen: if (t_ist > (t_soll + 2)) zustand = Abkuehlen; break;
```

```
        case Abkuehlen: if (t_ist < (t_soll - 2)) zustand = Heizen; break;
```

```
    }
```

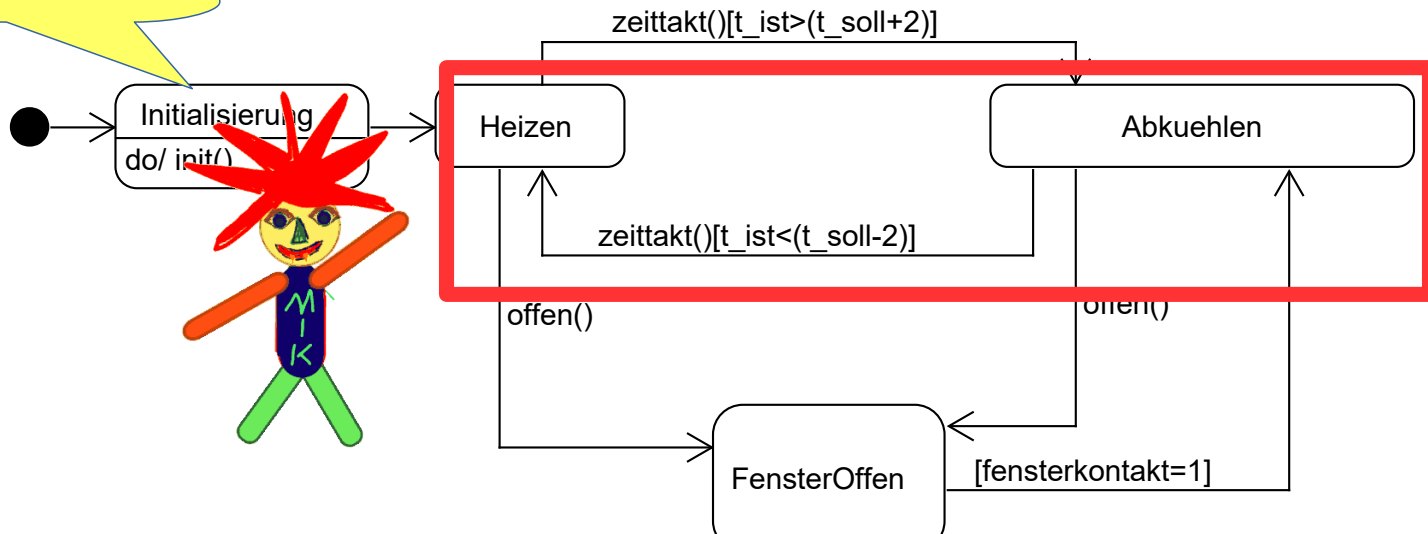
```
    TIM6->SR=0;
```

```
    HAL_NVIC_ClearPendingIRQ(TIM6_IRQn);
```

```
}
```

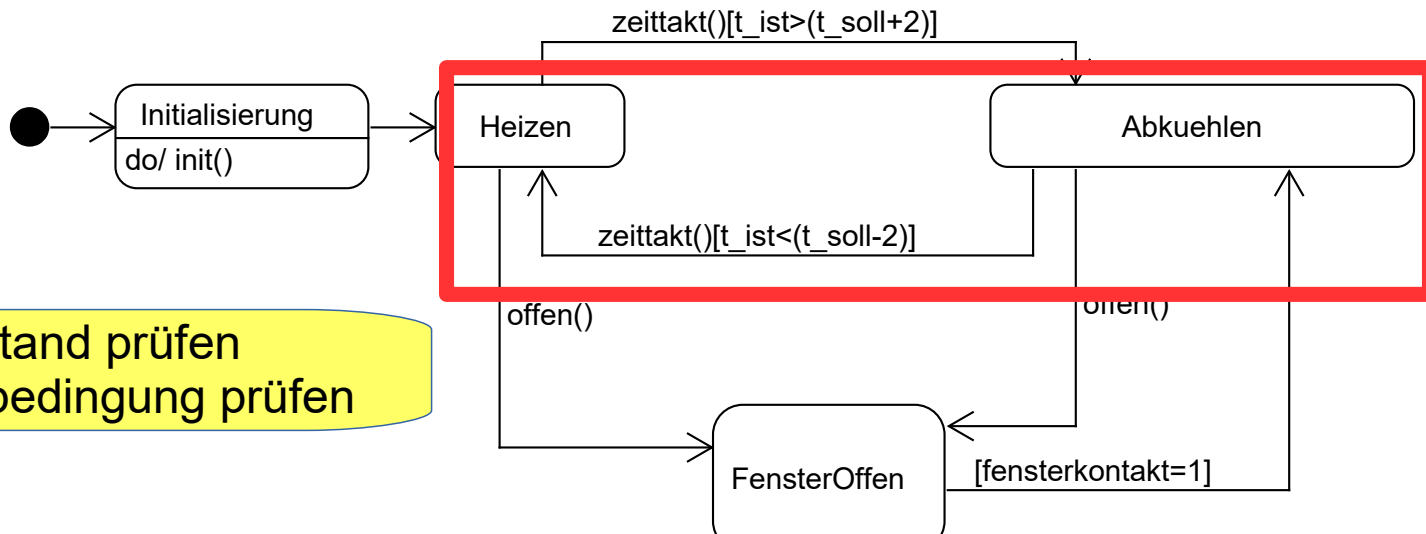
Heizungsregelung

Die zeittakt()-ISR



```
void zeittakt()
{
    switch (zustand)
    {
        case Heizen: if (t_ist > (t_soll+2)) zustand=Abkuehlen; break;
        case Abkuehlen: if (t_ist < (t_soll-2)) zustand=Heizen; break;
    }
    TIM6->SR=0;
    HAL_NVIC_ClearPendingIRQ(TIM6_IRQn);
}
```

Heizungsregelung



1. Zustand prüfen
2. Wächterbedingung prüfen

```
void zeittakt()
```

```
{
```

```
    switch (zustand)
```

```
    {
```

```
        case Heizen: if (t_ist > (t_soll + 2)) zustand = Abkuehlen; break;
```

```
        case Abkuehlen: if (t_ist < (t_soll - 2)) zustand = Heizen; break;
```

```
        TIM6->SR=0
```

```
        HAL_NVIC_ClearPendingIRQ(TIM6_IRQn);
```

```
    }
```

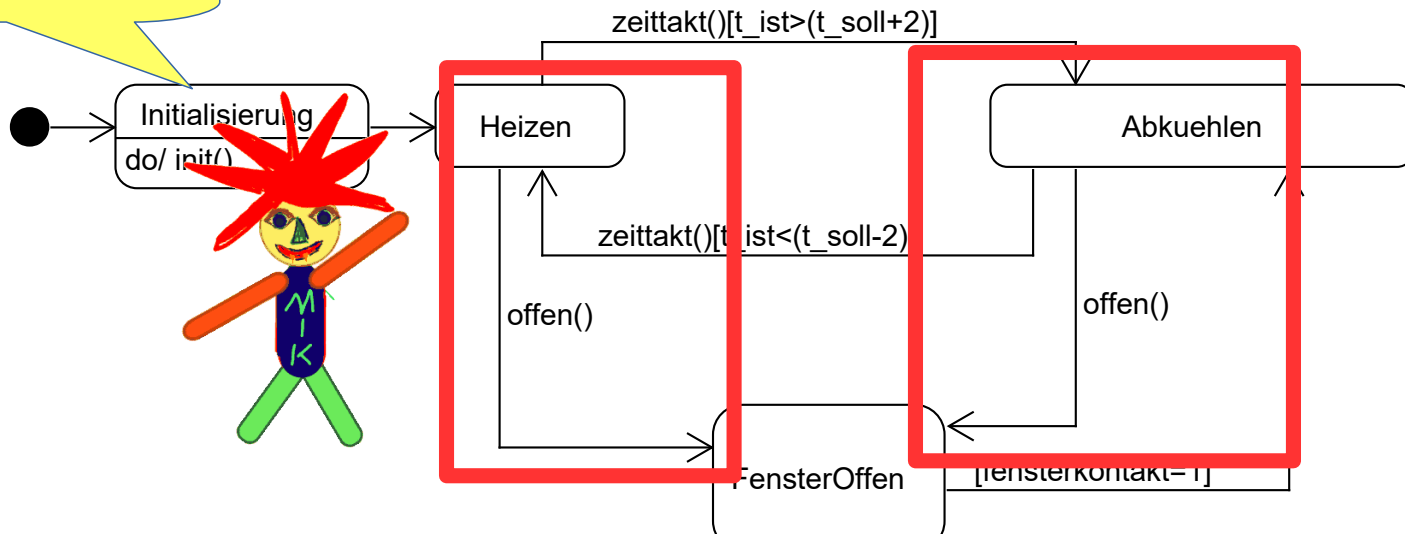


1.

2.

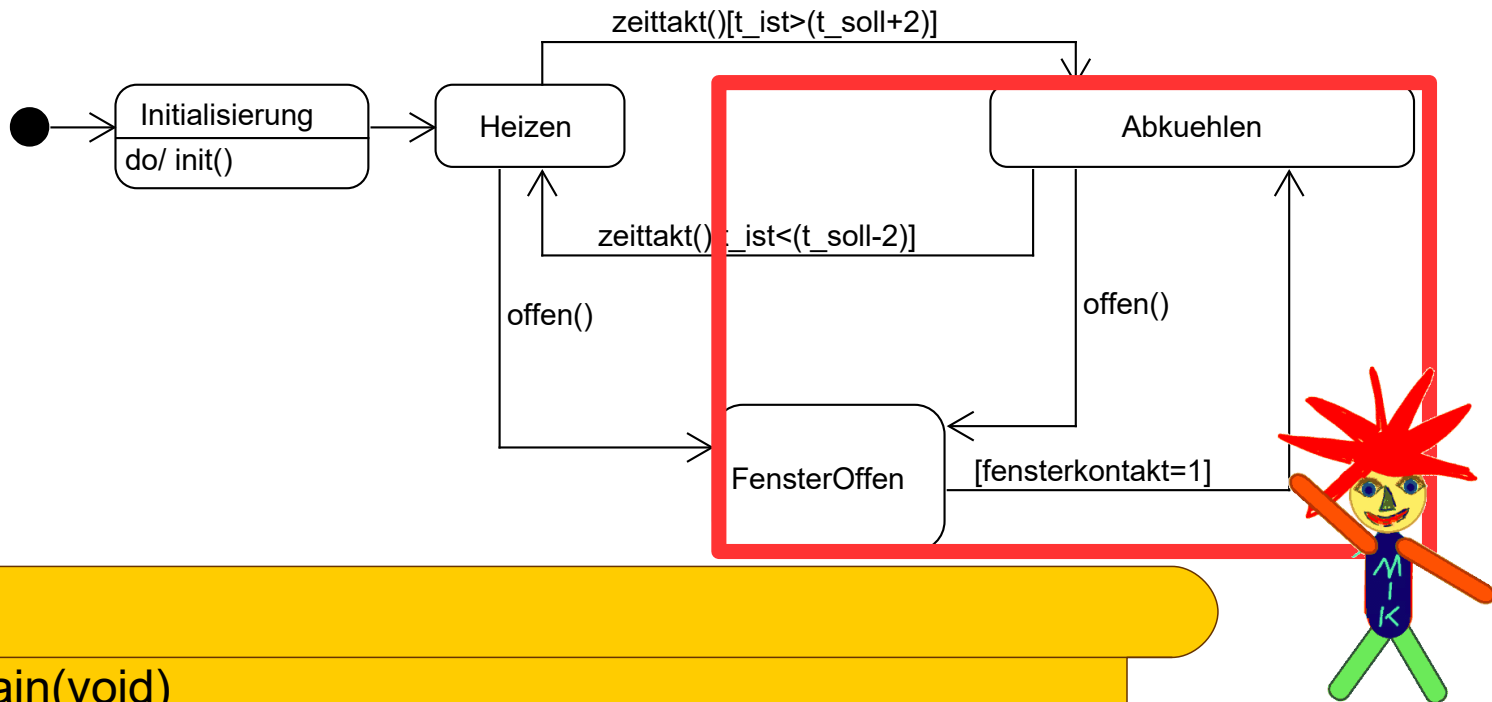
Heizungsregelung

Die `offen()`-ISR



```
void offen()
{
    switch (zustand)
    {
        case Heizen:
        case Abkuehlen: zustand=FensterOffen; break;
    }
}
```

Heizungsregelung



```
void main(void)
```

```
{
```

```
    init();
```

```
    zustand=Heizen;
```

```
    while(true){
```

```
        if (zustand==FensterOffen) {
```

```
            if (fensterkontakt==1) {zustand=Abkuehlen;}
```

```
        }
```

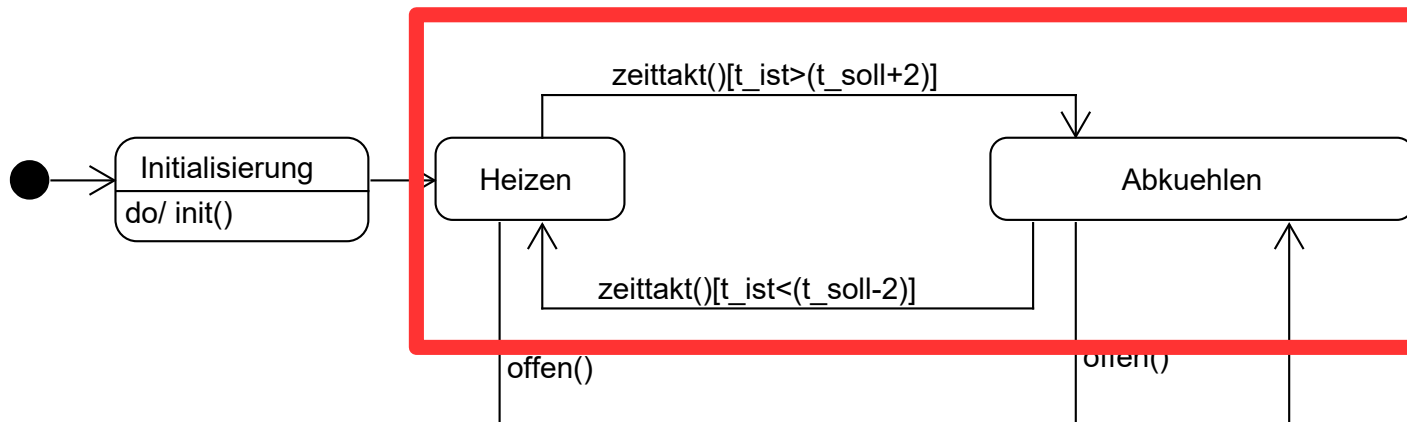
```
    }
```

```
}
```

Wächterbedingung
prüfen im
Hauptprogramm

Heizungsregelung

Was ist neu??



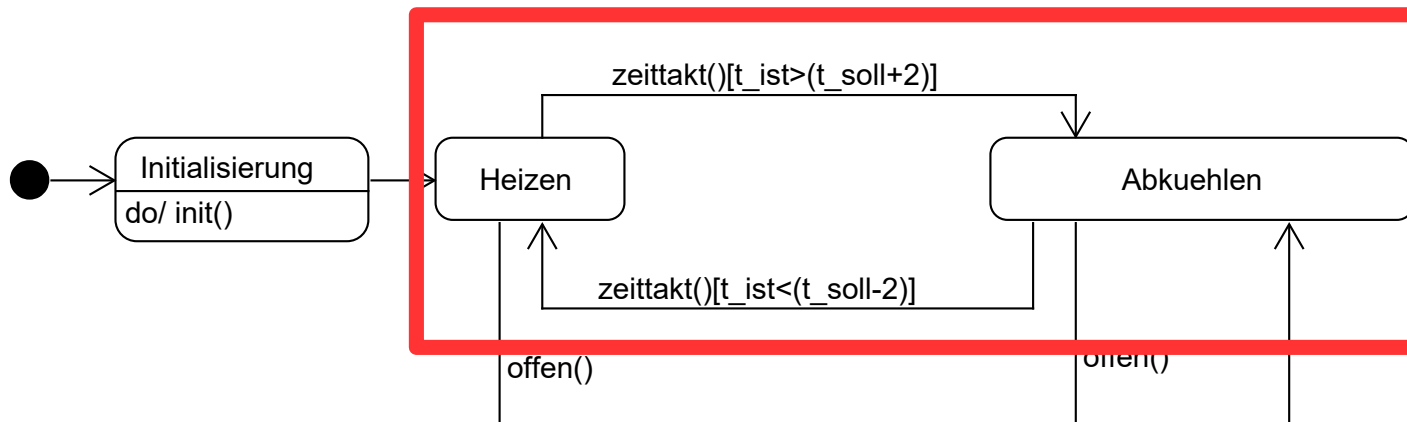
Ein Ereignis (Interrupt) kann mit einer Wächterbedingung versehen sein!!

In der ISR:

```
switch(zustand){  
  case Zustand1: if (wächterbedingung) {zustand=Zustand2;};break;  
  ..  
}
```



Heizungsregelung



Ein Ereignis (Interrupt) kann mit einer Wächterbedingung versehen sein!!

In der ISR:

```
switch(zustand){  
  case Zustand1: if (wächterbedingung) {zustand=Zustand2;};break,  
  ..  
}
```



1. Zustand prüfen
2. Wächterbedingung prüfen
dann handeln

