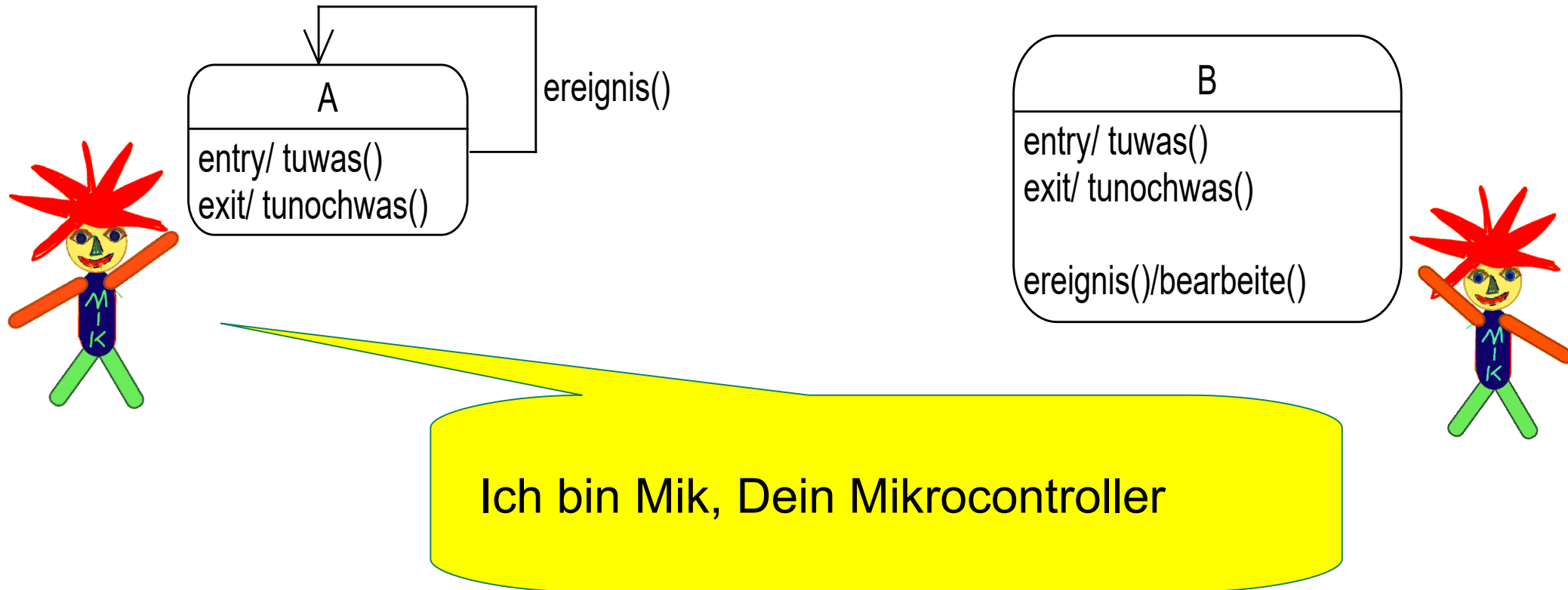
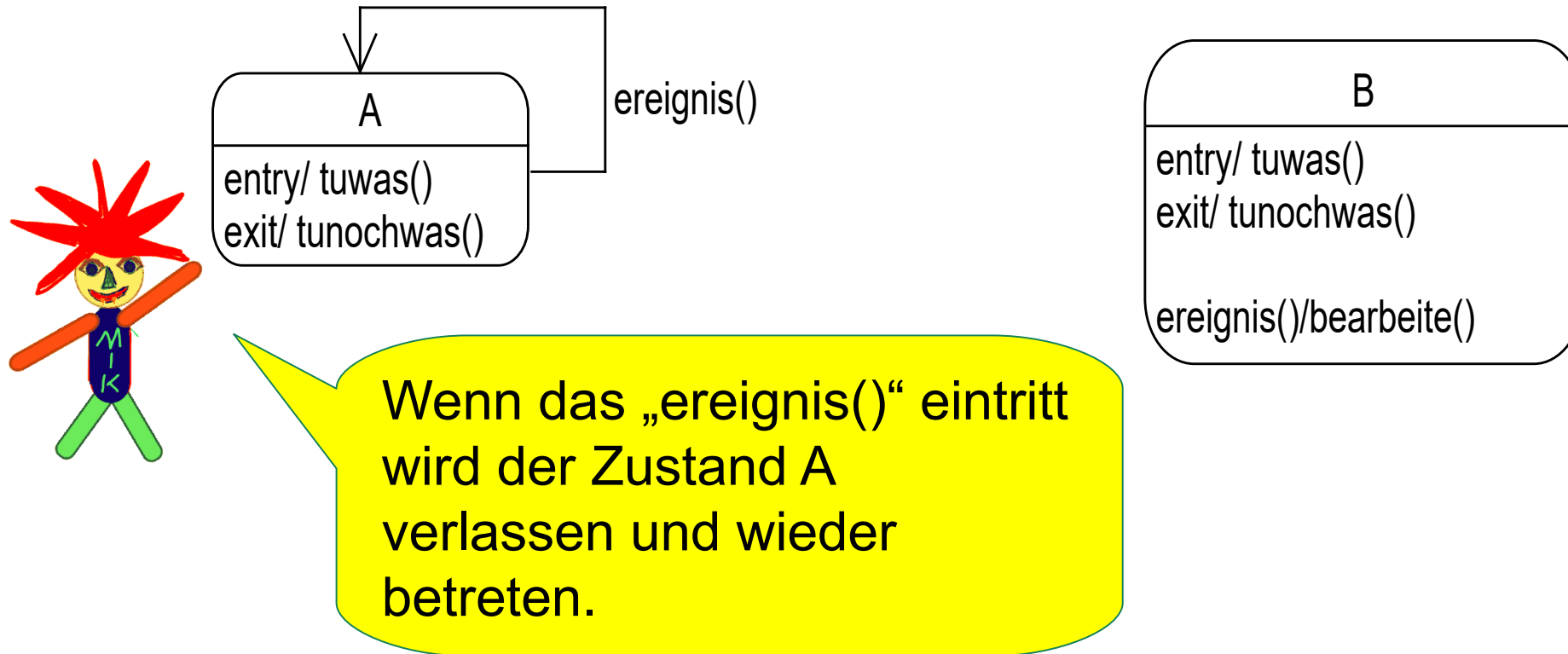


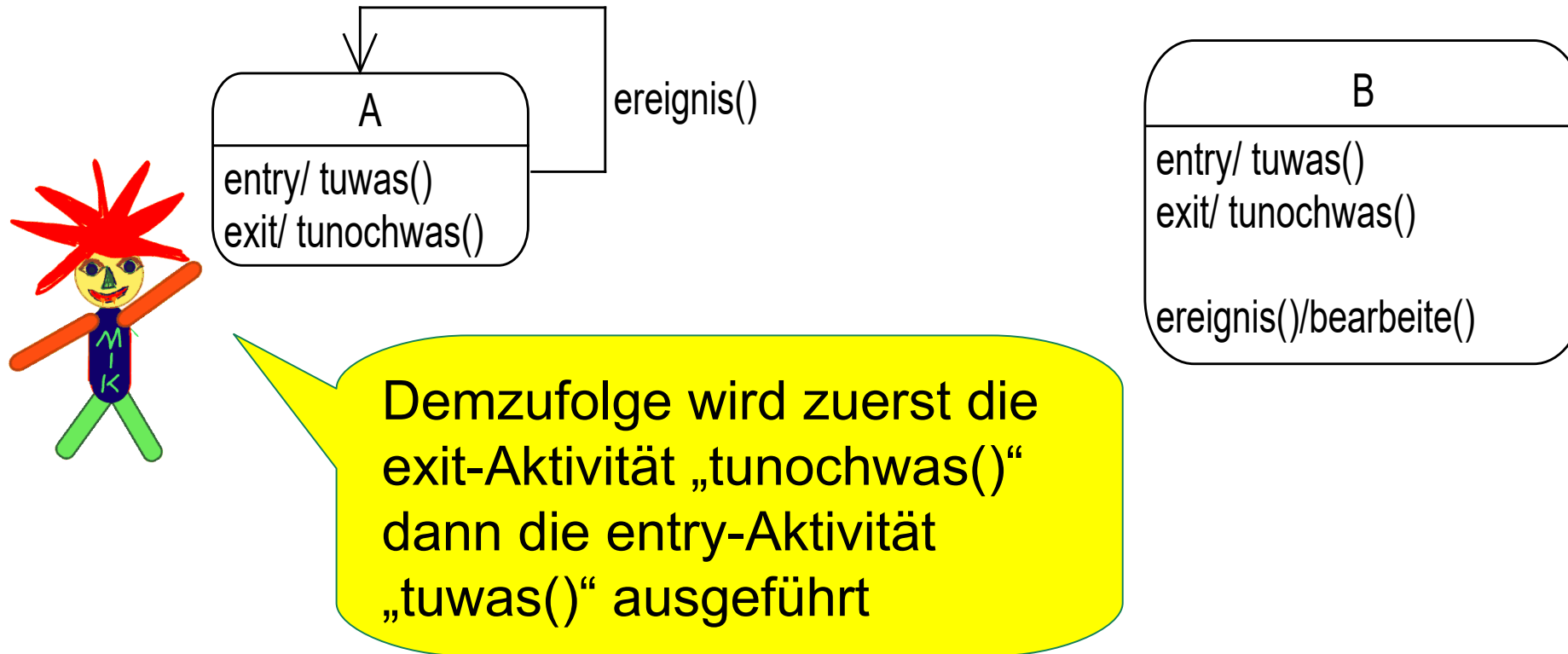
Selbsttransition vs. internes Ereignis



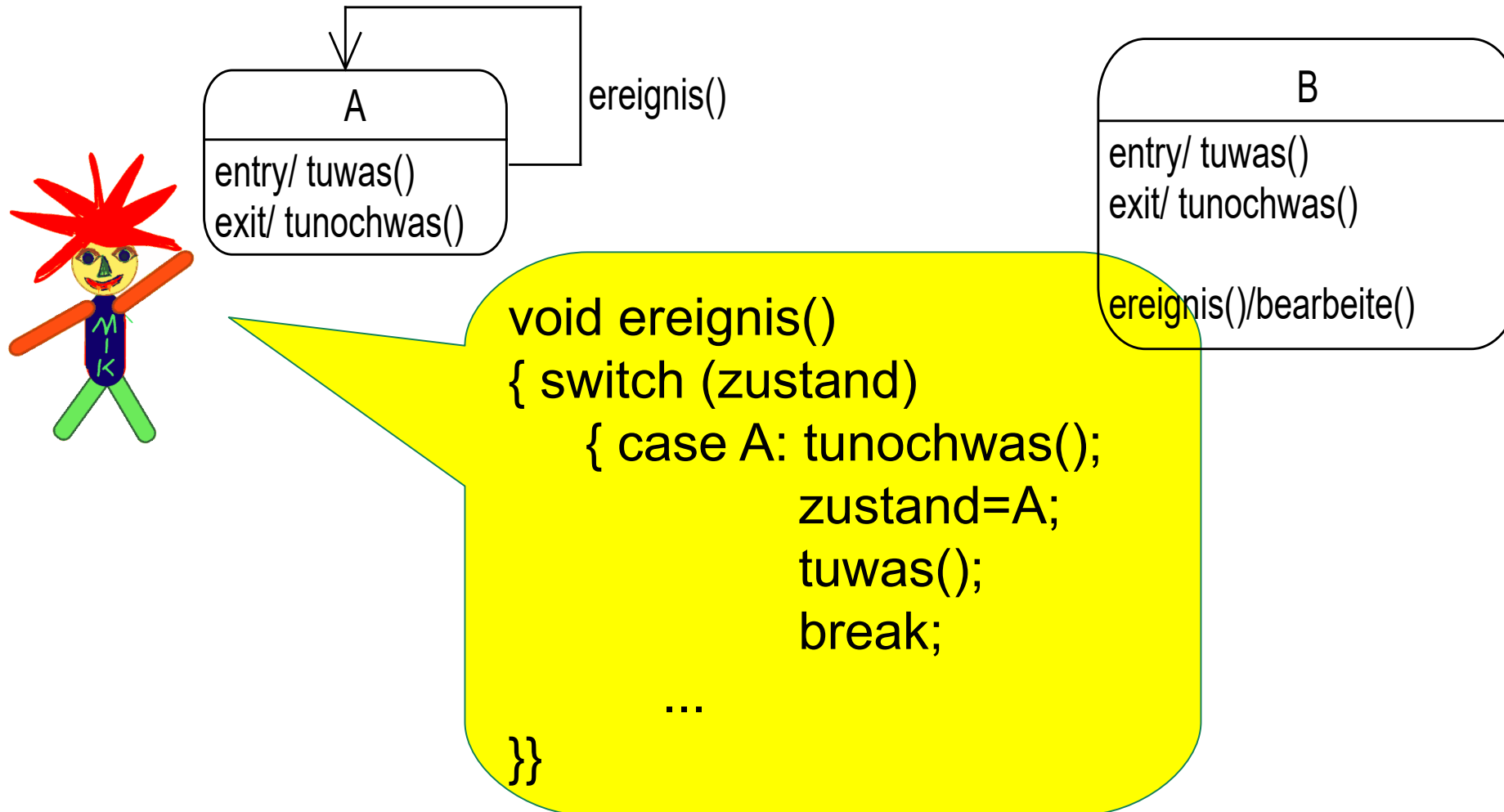
Selbsttransition vs. internes Ereignis



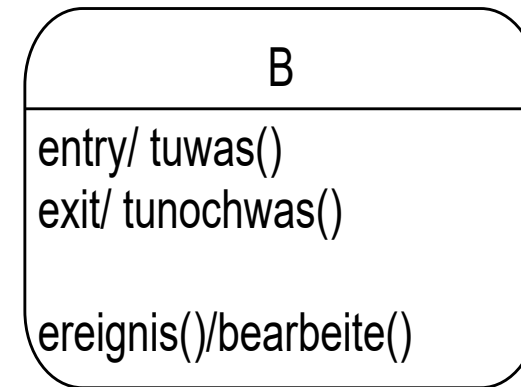
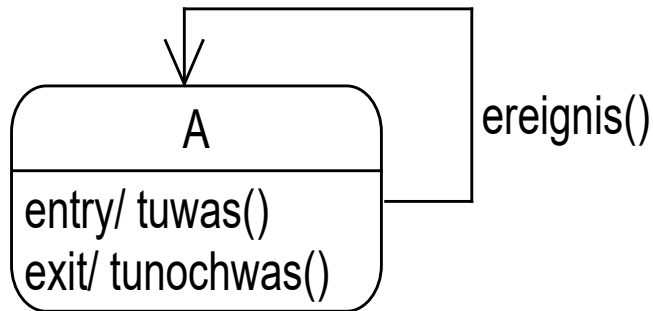
Selbsttransition vs. internes Ereignis



Selbsttransition vs. internes Ereignis

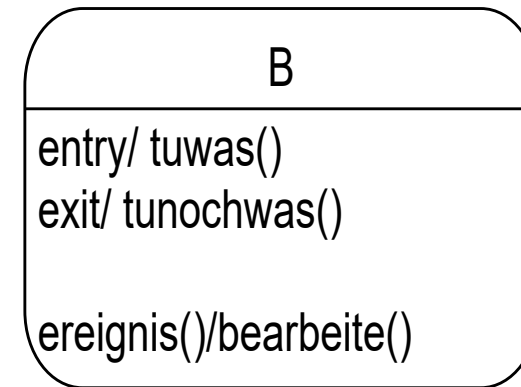
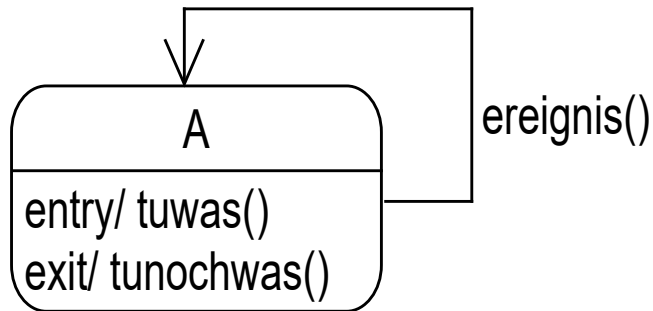


Selbsttransition vs. internes Ereignis



Wenn das „ereignis()“ eintritt
wird der Zustand B weder
verlassen noch wieder
betreten.

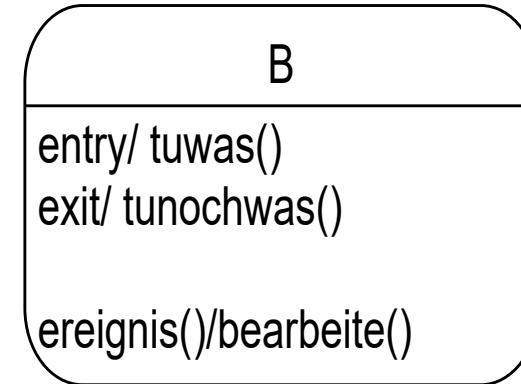
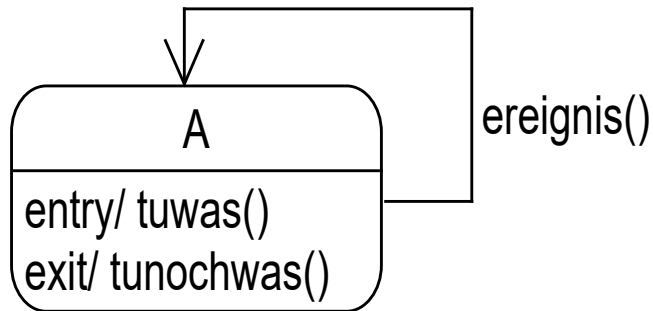
Selbsttransition vs. internes Ereignis



Demzufolge werden weder die exit-Aktivität „tunochwas()“ noch die entry-Aktivität „tuwas()“ ausgeführt



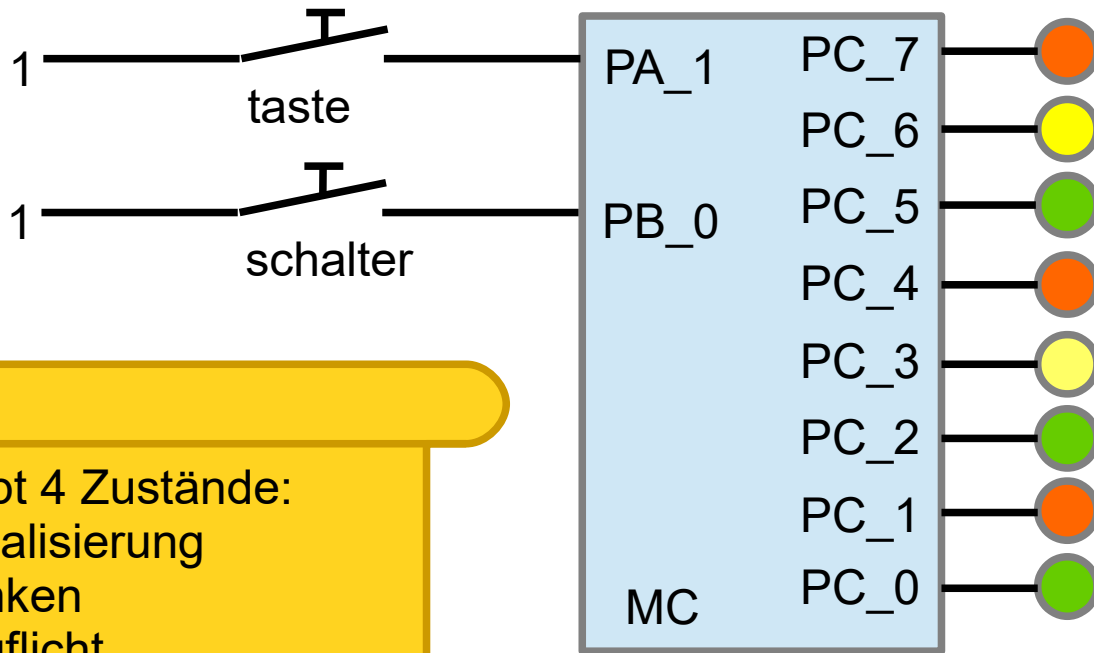
Selbsttransition vs. internes Ereignis



```
void ereignis()
{ switch (zustand)
  { case B: bearbeite()
    break;
    ...
  }
}
```



Selbsttransition vs. internes Ereignis



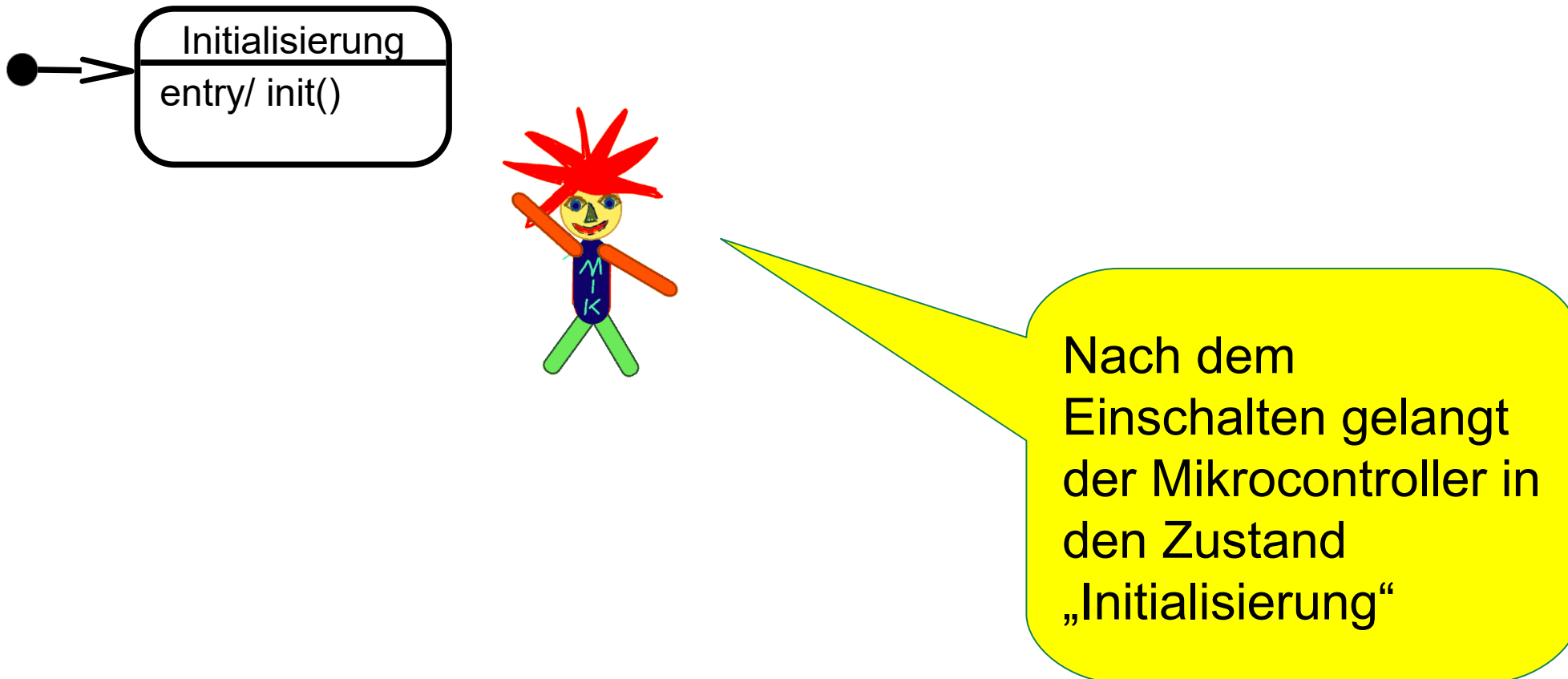
Es gibt 4 Zustände:

- Initialisierung
- Blinken
- Lauflicht
- Muster

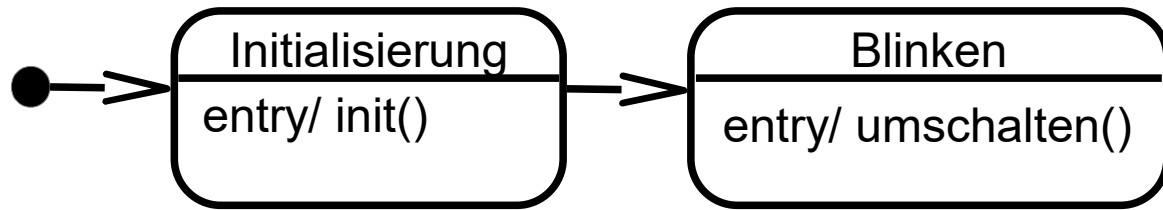


Ein Beispiel:
Effektbeleuchtung

Selbsttransition vs. internes Ereignis

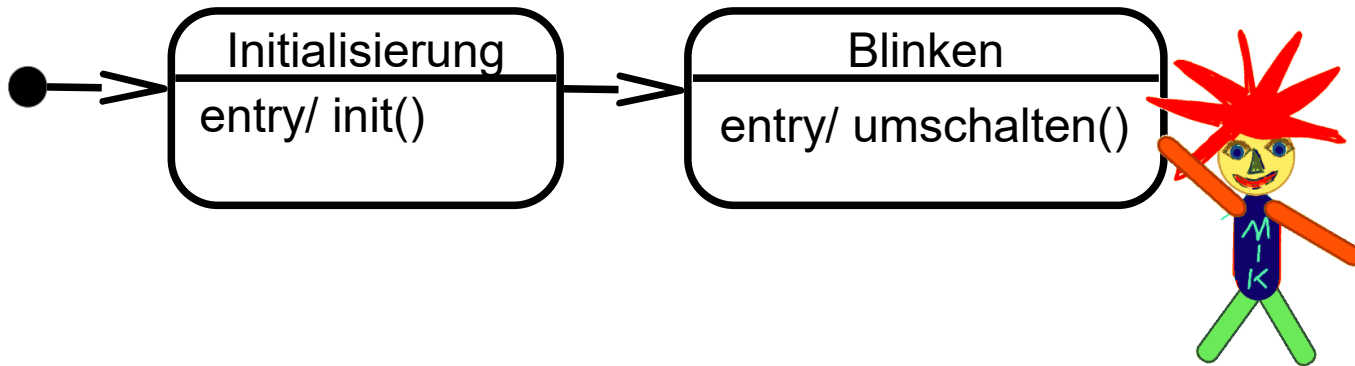


Selbsttransition vs. internes Ereignis



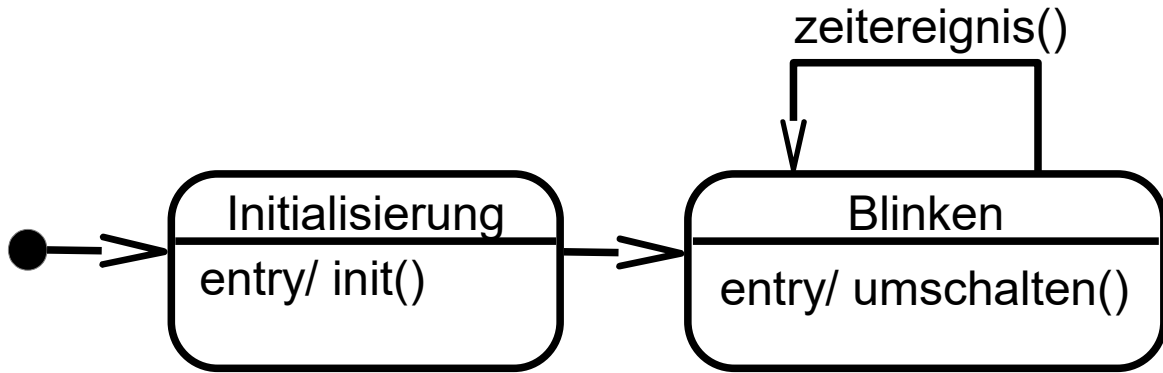
Danach wechselt der Mikrocontroller automatisch in den Zustand „Blinken“

Selbsttransition vs. internes Ereignis



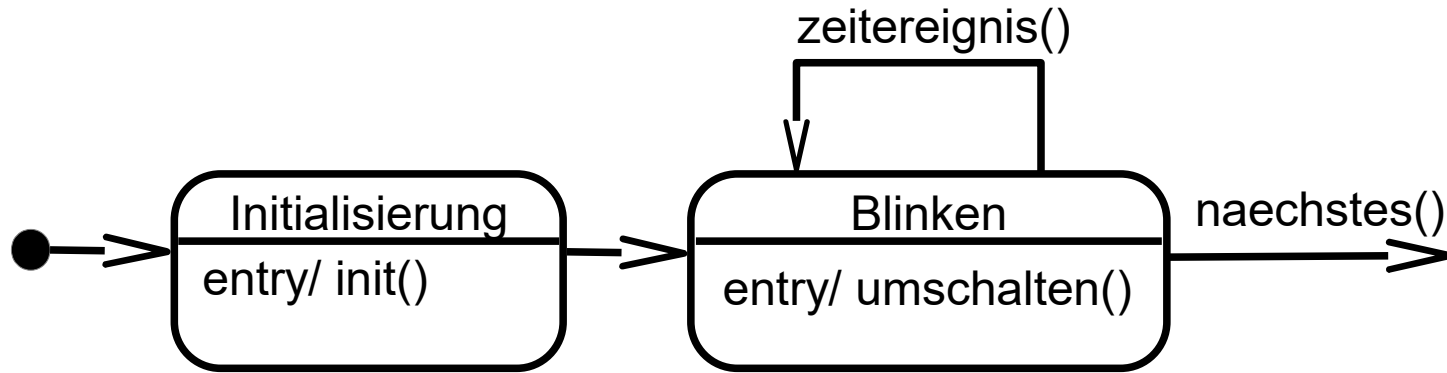
Die Operation „umschalten()“ negiert alle Bits von PortC (GBIOC)

Selbsttransition vs. internes Ereignis



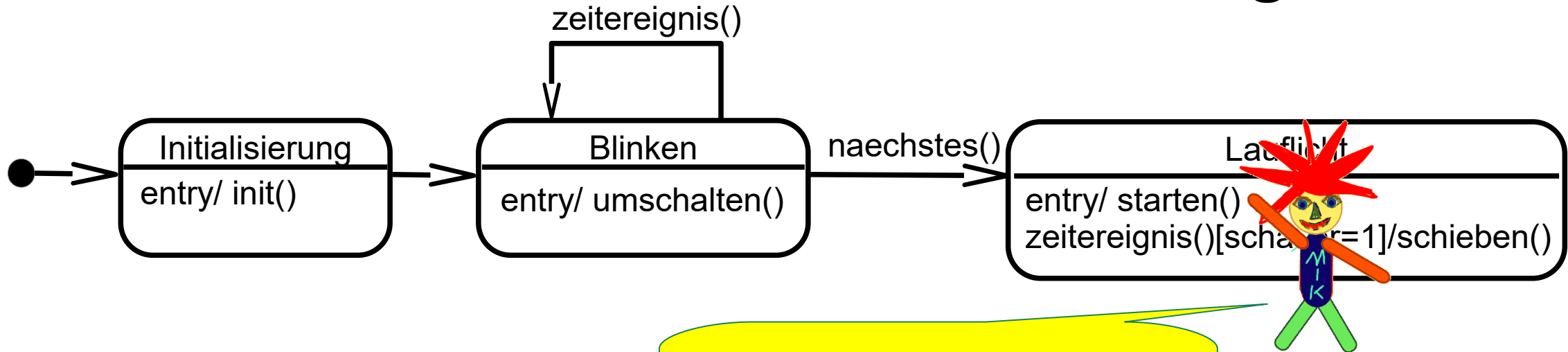
Das Zeitereignis wird zyklisch vom Timer ausgelöst und bewirkt, dass „umschalten()“ immer wieder ausgeführt wird.

Selbsttransition vs. internes Ereignis



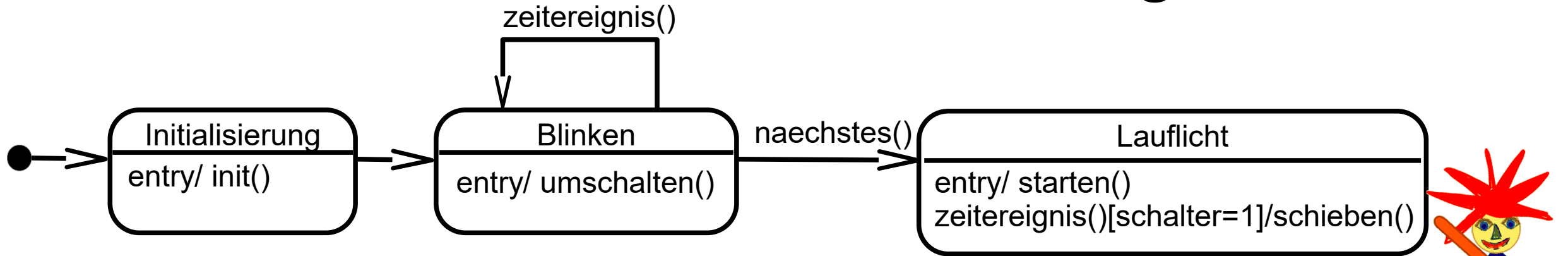
Das Ereignis „naechstes()“ wird bei Tastendruck ausgelöst.

Selbsttransition vs. internes Ereignis



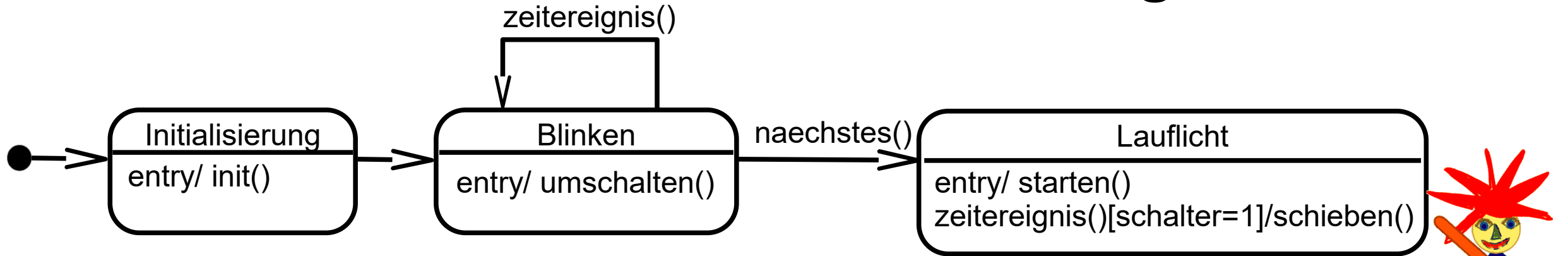
Das Lauflicht startet mit
LED an PC_0

Selbsttransition vs. internes Ereignis



Bei einem Zeitereignis wird, falls der Schalter zu ist (=1) das Licht um eine Stelle weitergeschoben (Operation „schieben“)

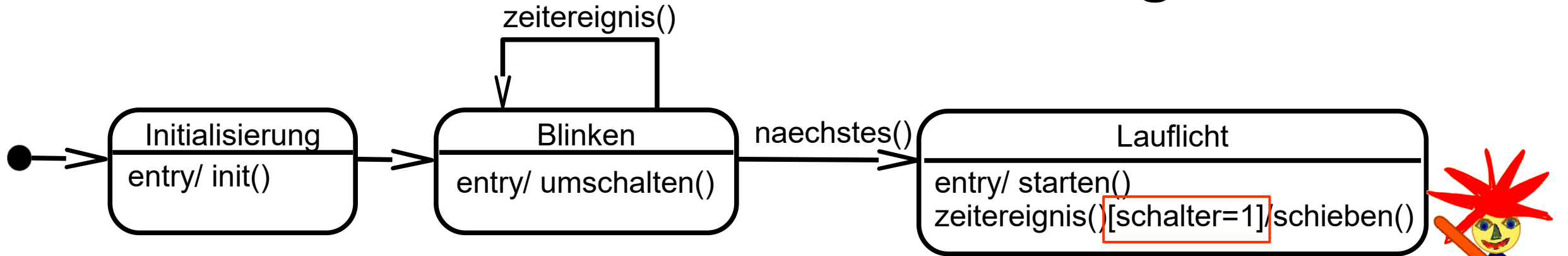
Selbsttransition vs. internes Ereignis



In diesem Fall ist ein internes Ereignis erforderlich, sonst bliebe das Licht bei LED PC_0 stehen. `entry/ starten()` würde immer wieder aufgerufen.



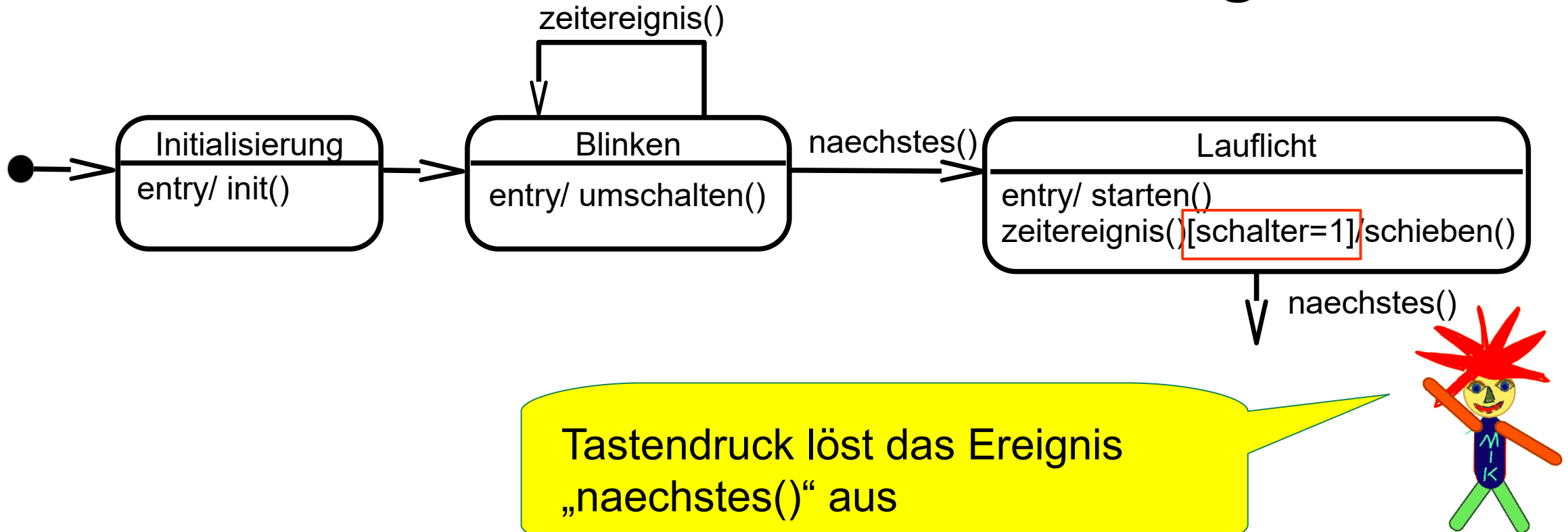
Selbsttransition vs. internes Ereignis



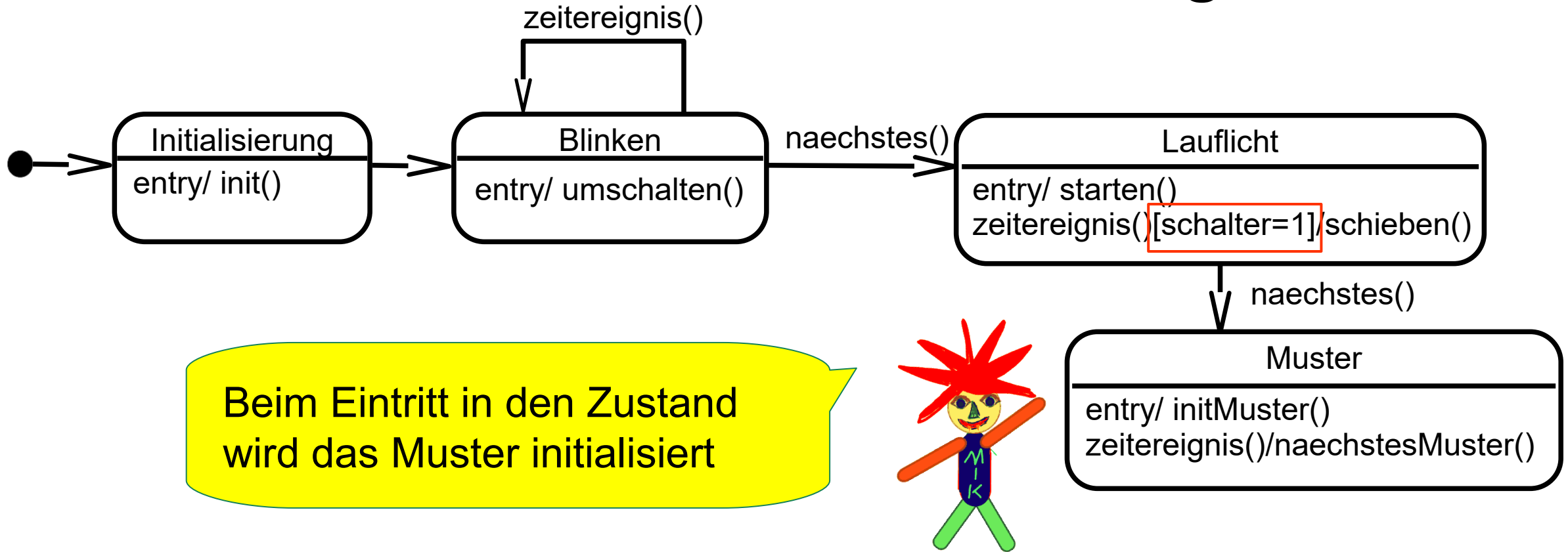
Alle Ereignisse, auch interne Ereignisse können mit Wächterbedingungen versehen sein.



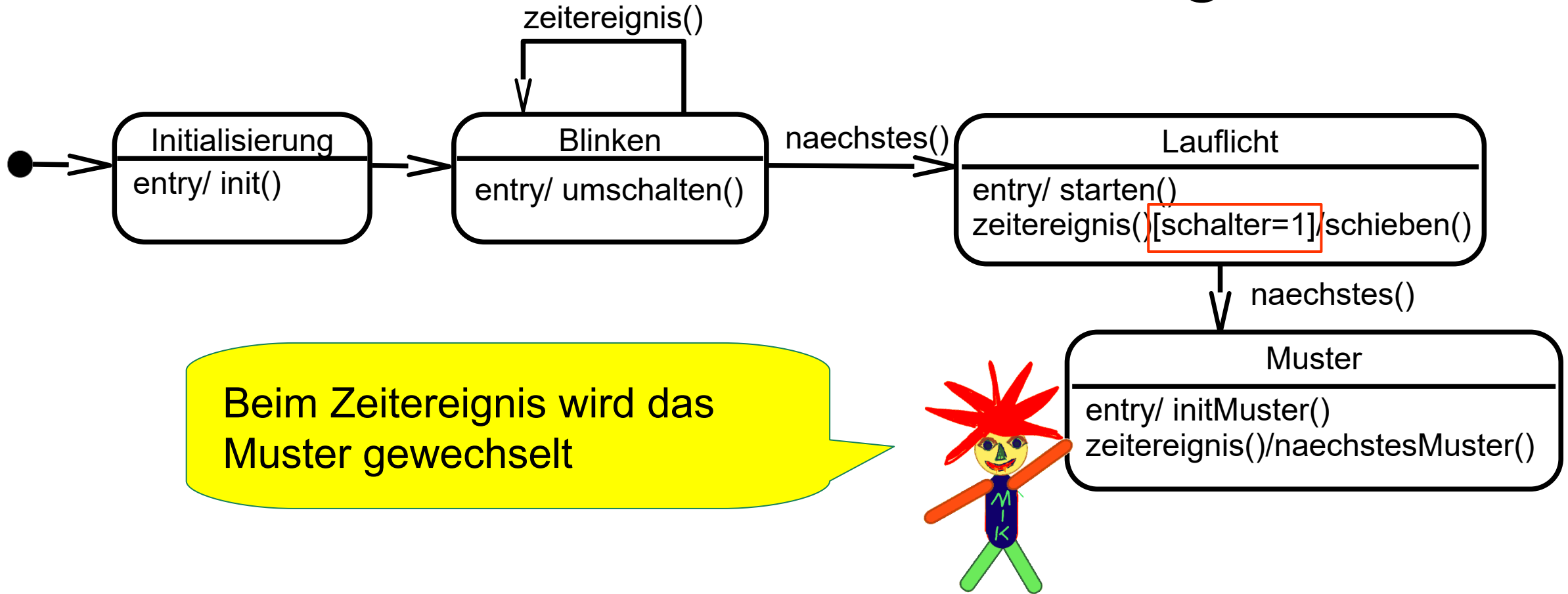
Selbsttransition vs. internes Ereignis



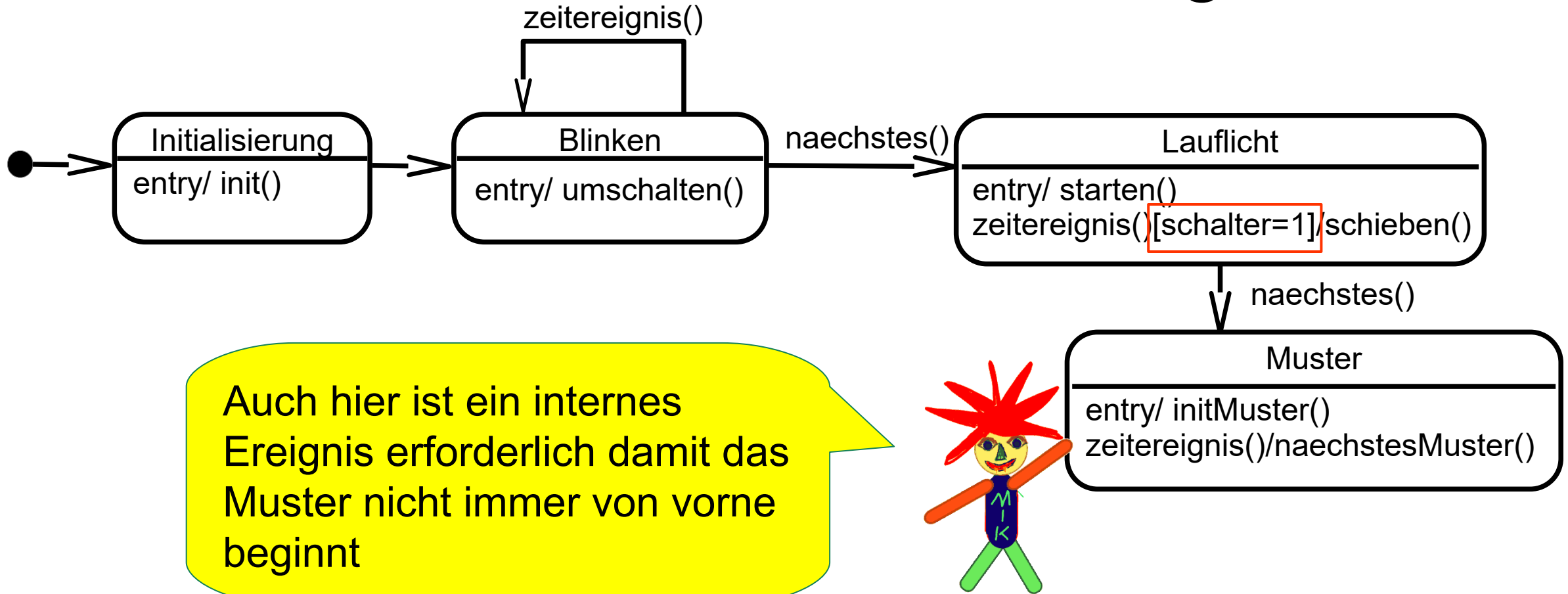
Selbsttransition vs. internes Ereignis



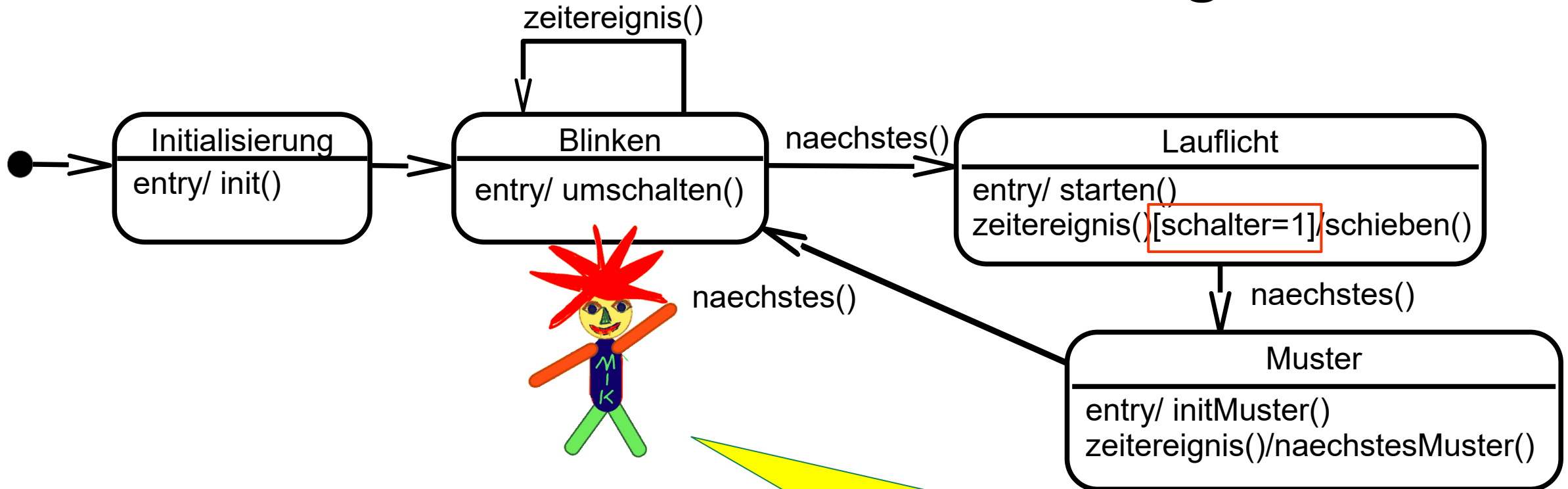
Selbsttransition vs. internes Ereignis



Selbsttransition vs. internes Ereignis

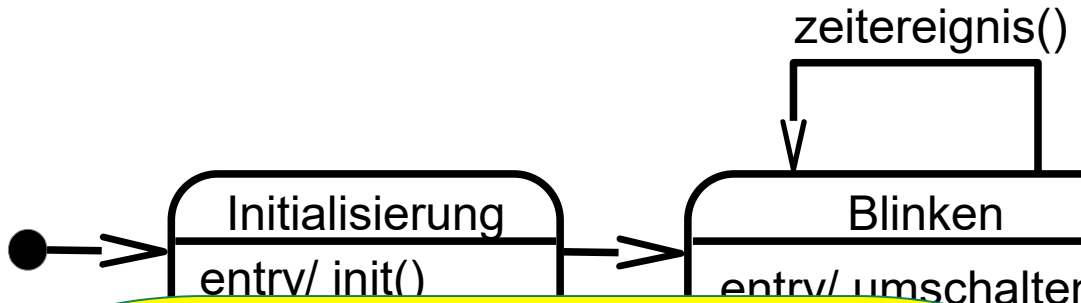


Selbsttransition vs. internes Ereignis



taste bewirkt einen
Zustandswechsel in den
Zustand Blinken

Selbsttransition v



Zustände, globale Variablen
und Ports



```
#define Init 0
```

```
#define Blinken 1
```

```
#define Lauflicht 2
```

```
#define Muster 3
```

```
int zustand=0;
```

```
int z;
```

```
int m[6]={0b00011000, 0b00100100,0b01000010,  
          0b10000001,0b01000010,0b00100100};
```

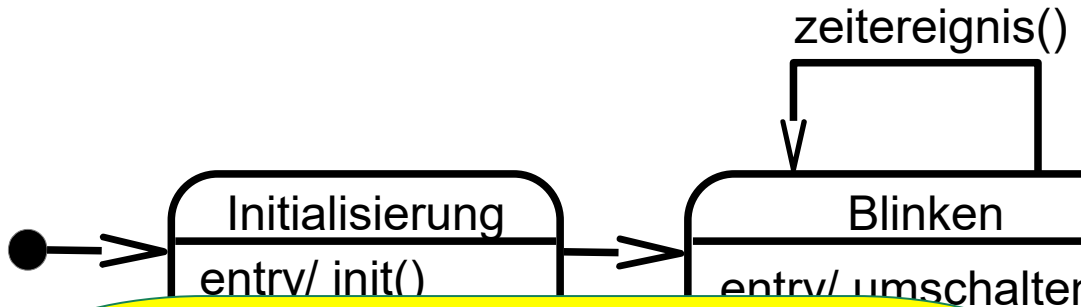
```
PortOut    lichter(PortC,0xFF);
```

```
InterruptIn taste(PA_1);
```

```
DigitalIn  schalter(PB_0);r
```

```
ster()
```

Selbsttransition v



Die Initialisierung



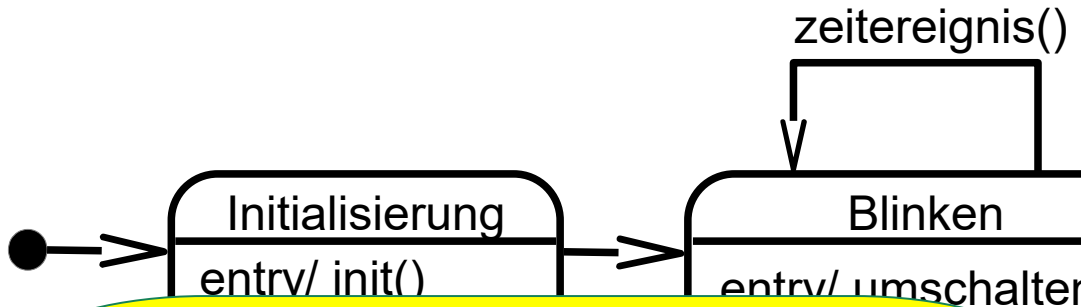
```
void init()
{
    taste.mode(PullDown);
    taste.rise(&naechstes);

    schalter.mode(PullDown);

    RCC->APB1ENR|=0b10000; //Clock Enable
    TIM6->PSC=31999; //Prescaler 1ms
    TIM6->ARR=499; //Autoreload 500*1ms = 0,5s
    TIM6->DIER=1; //UIE = 1 (Update Interrupt Enable)
    TIM6->SR=0; //UIF =0 (Update Interrupt Flag)
    TIM6->CR1=1; //CEN=1 (Counter Enable)

    /* TIM6_IRQn interrupt configuration */
    NVIC_SetVector(TIM6_IRQn, (uint32_t)&zeitereignis);
    HAL_NVIC_EnableIRQ(TIM6_IRQn);
}
```


Selbsttransition v



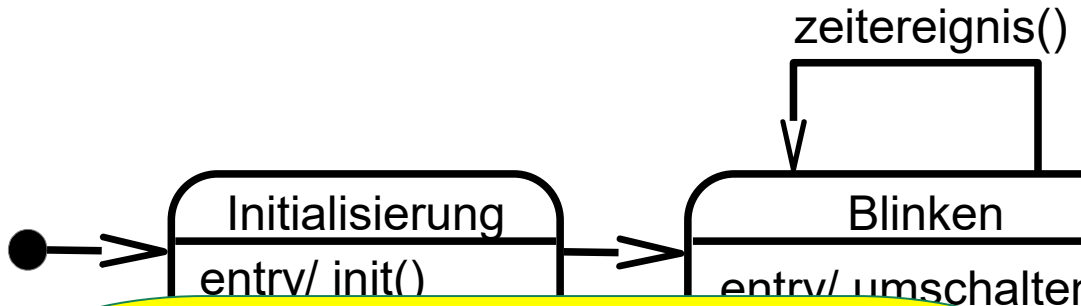
Das Hauptprogramm



```
int main()
{
    init();
    zustand=Blinken;
    umschalten();
    while (true) {

    }
}
```

Selbsttransition v

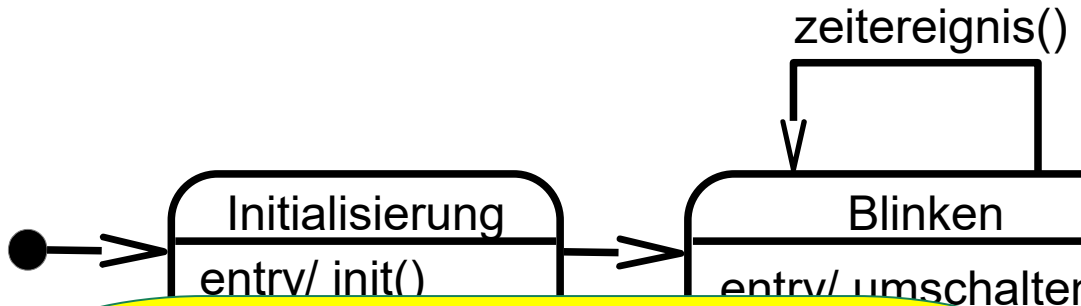


Das Zeitereignis



```
void zeitereignis()
{
    switch(zustand)
    {
        case Blinken: zustand = Blinken;
                      umschalten();
                      break;
        case Laufflicht: if (schalter==1) schieben();
                      break;
        case Muster: naechstesMuster();
    }
    TIM6->SR=0;
    HAL_NVIC_ClearPendingIRQ(TIM6_IRQn);
}
```

Selbsttransition v

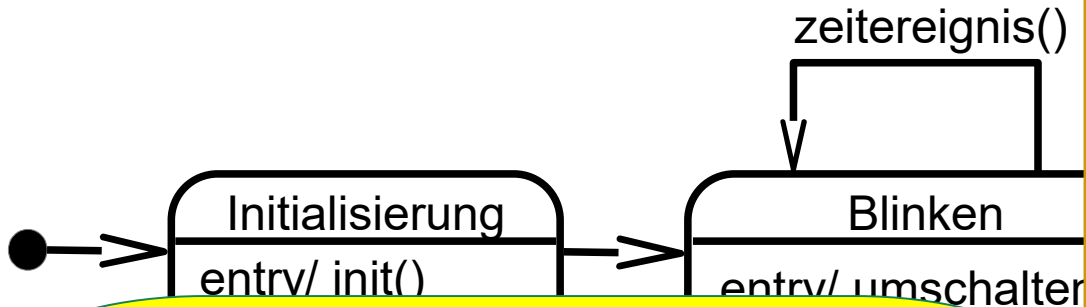


Das Tastenereignis



```
void naechstes()
{
    switch (zustand)
    {
        case Blinken: zustand=Lauflicht;
                       starten();
                       break;
        case Lauflicht: zustand=Muster;
                       initMuster();
                       break;
        case Muster: zustand=Blinken;
                    umschalten();
                    break;
    }
}
```

Selbsttransition v

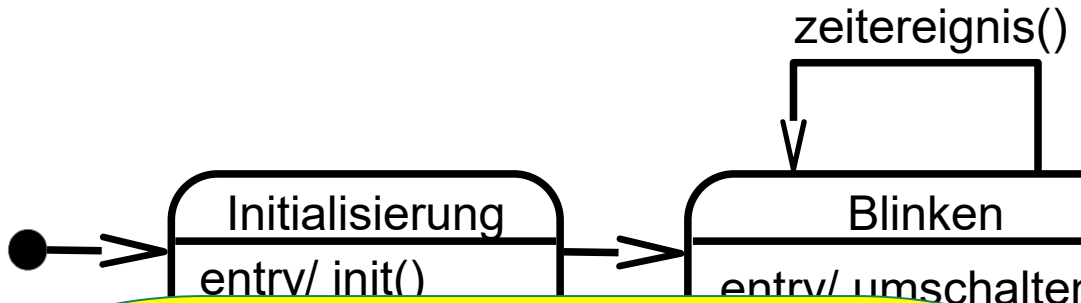


entry von Blinken

```
void umschalten()
{
    licht= ~licht;
}
```



Selbsttransition v

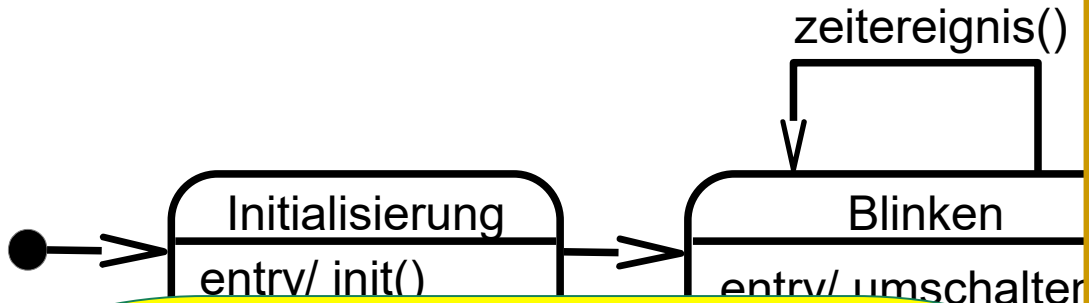


entry von Lauflicht



```
void starten()
{
    licht=0b00000001;
    TIM6->ARR=250;    //250ms
    TIM6->CNT=0;
}
```

Selbsttransition v

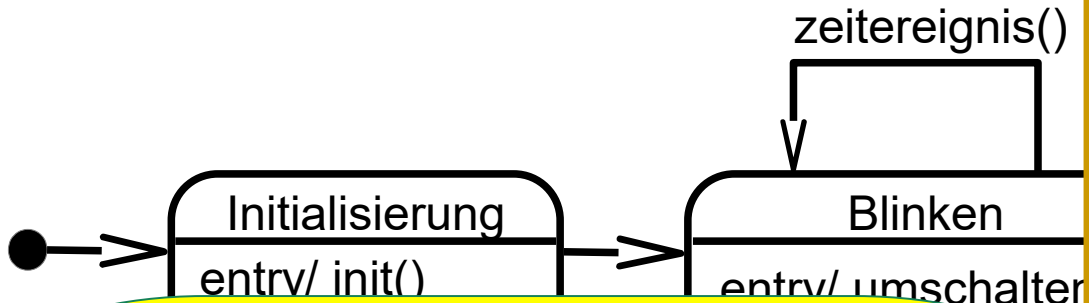


Beim Zeitereignis im Zustand
Lauflicht: schieben()



```
void schieben()
{
    licht=licht<<1;
    if (licht==0) licht=1;
}
```

Selbsttransition v

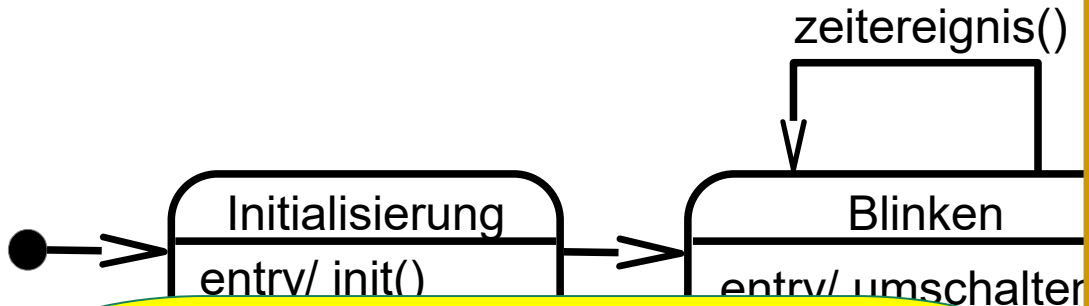


Beim Eintritt in Zustand
Muster: Muster initialisieren



```
void initMuster()
{
    z=0;
    licht=m[z];
}
```

Selbsttransition v



Im Zustand Muster beim
Zeitereignis: Muster
weitschalten



```
void naechstesMuster()
{
    z=(z+1)%6;
    licht=m[z];
}
```