

Ereignisse im Zustandsdiagramm

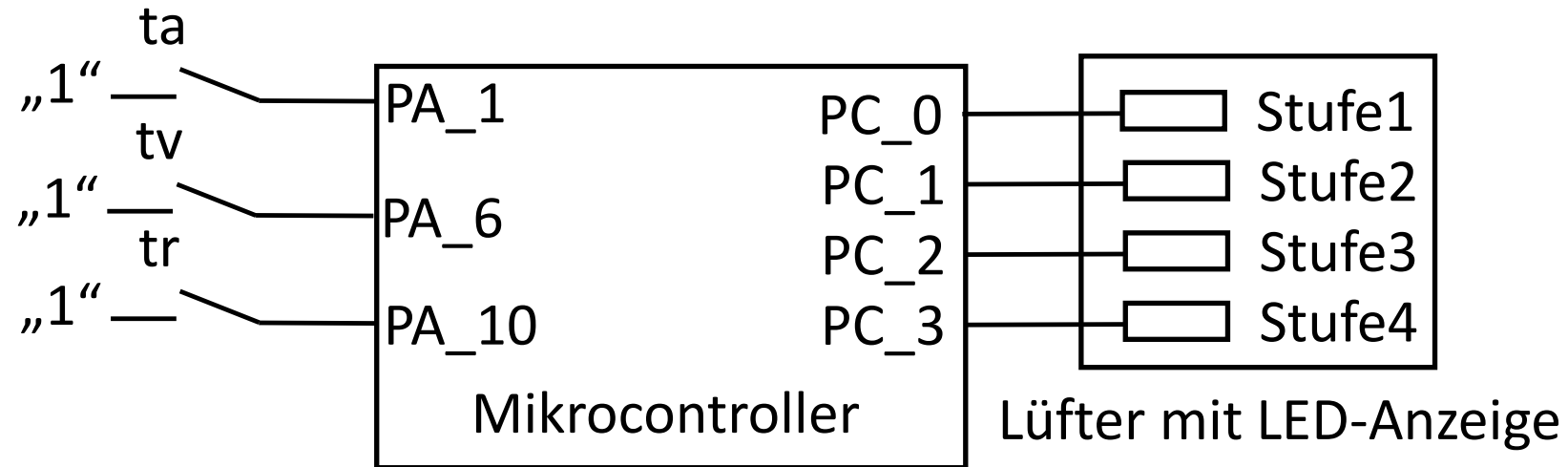
Interrupts



Ich bin Mik, Dein Mikrocontroller



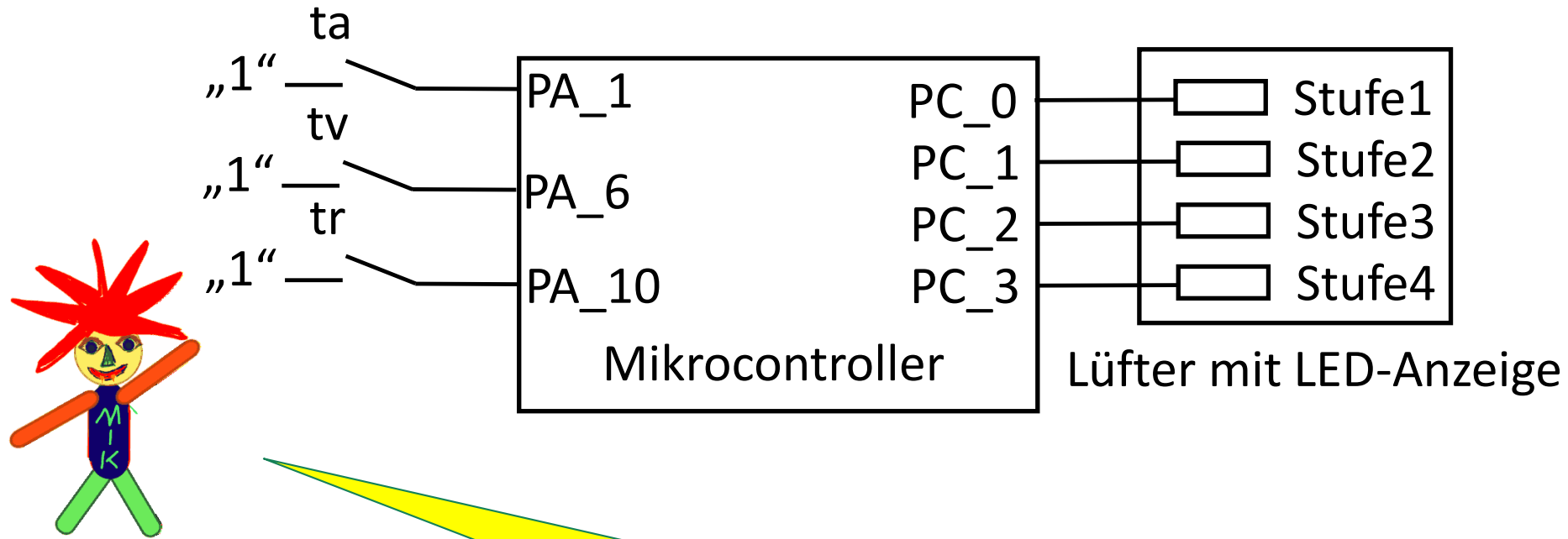
Ereignisse im Zustandsdiagramm



Die Lüftersteuerung einer Dunstabzugshaube soll programmiert werden



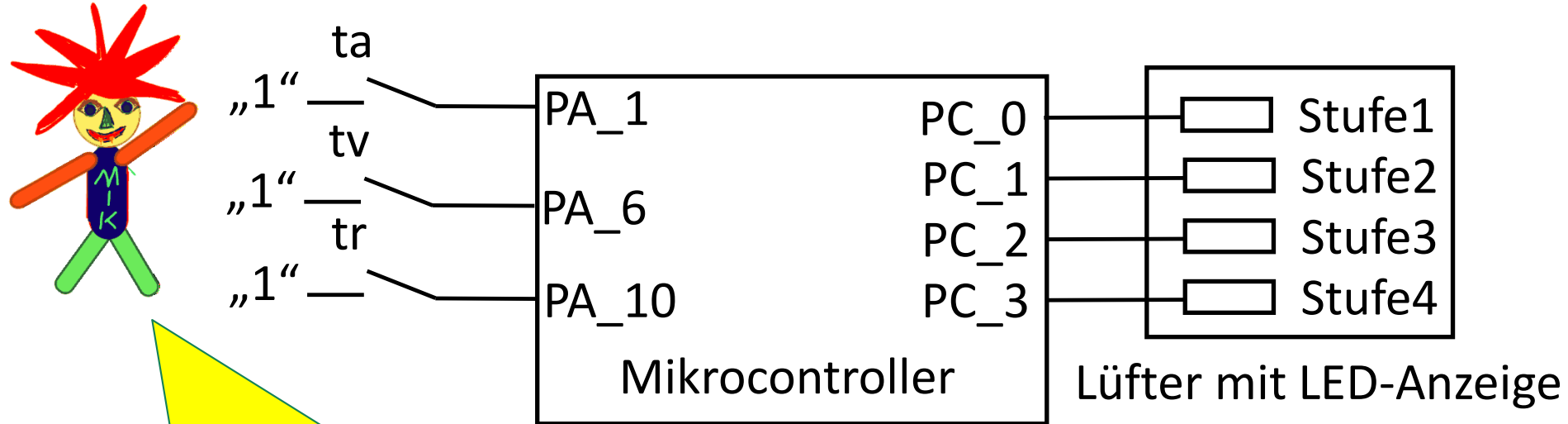
Ereignisse im Zustandsdiagramm



Die Taster benötigen PullDown und sind als Interrupteingänge zu konfigurieren



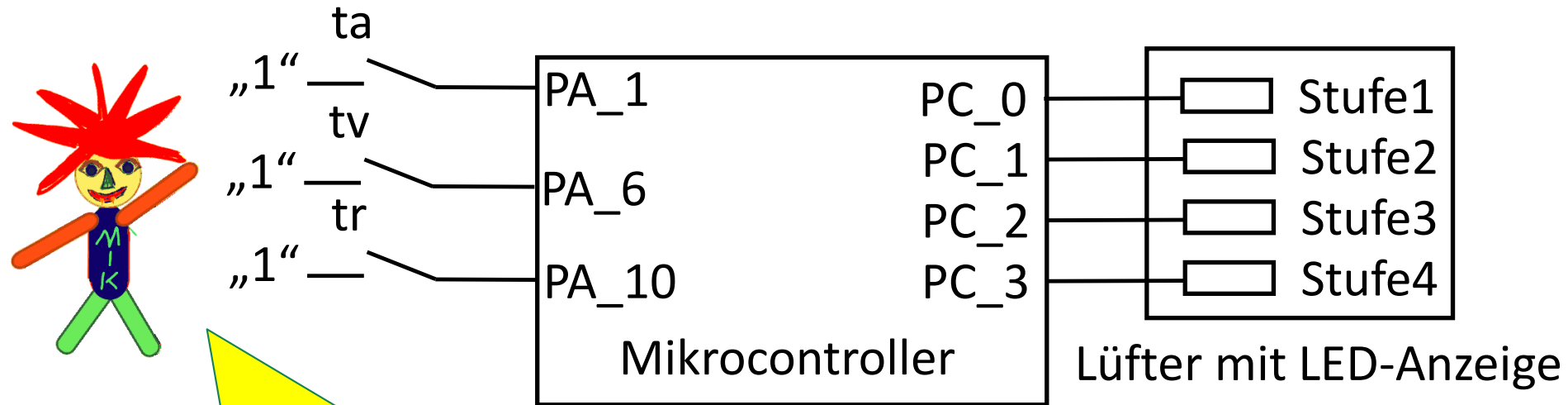
Ereignisse im Zustandsdiagramm



Taste ta schaltet den Lüfter ein oder aus.
ISR: void einaus()
Bei Aus werden alle Stufen ausgeschaltet.
Eingeschaltet wird immer in Stufe 1



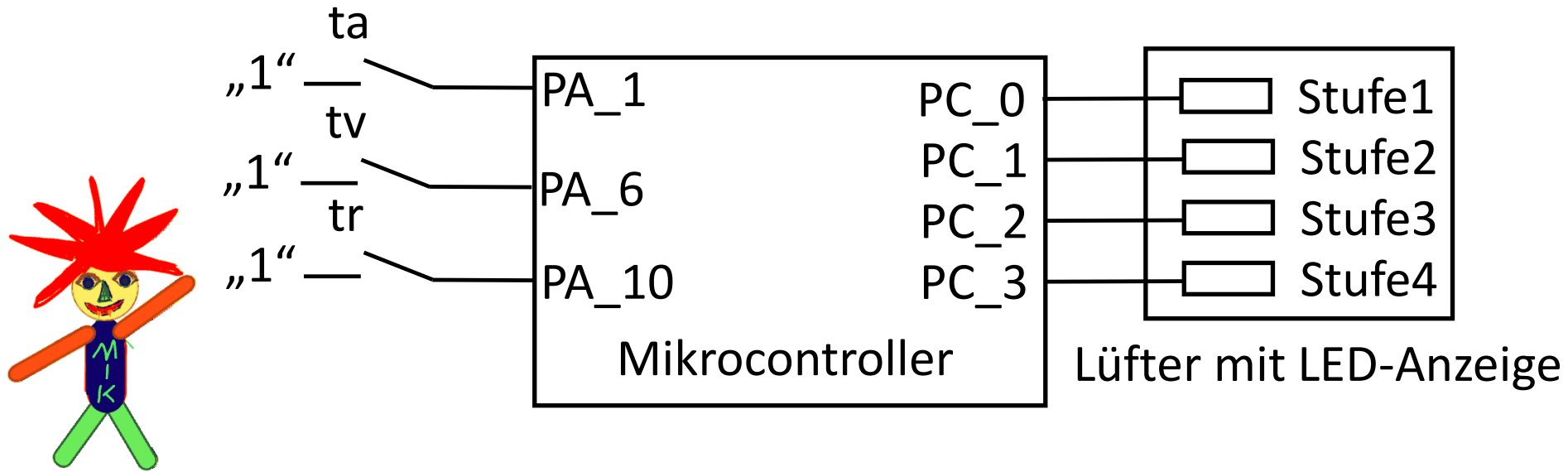
Ereignisse im Zustandsdiagramm



Taste tv schaltet den Lüfter eine Stufe hoch bis maximal Stufe 4. Wenn Stufe 4 erreicht wird bewirkt ein erneuter Druck auf Taste tv nichts mehr.
ISR: void hochschalten()



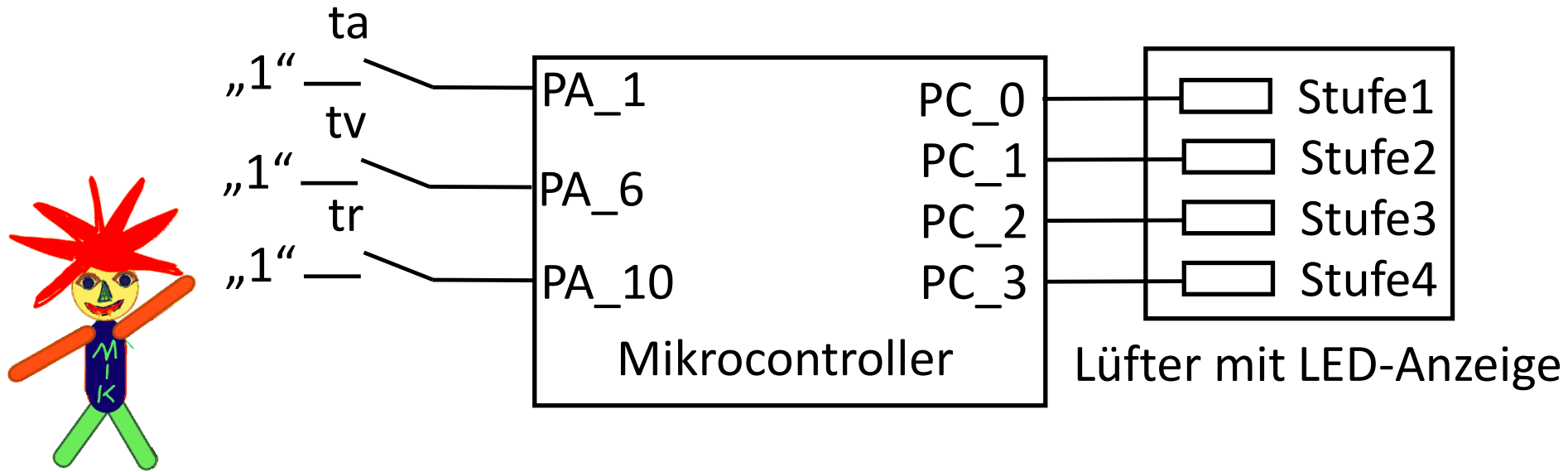
Ereignisse im Zustandsdiagramm



Taste tr schaltet den Lüfter eine Stufe herunter bis minimal Stufe 1. Wenn Stufe 1 erreicht wird bewirkt ein erneuter Druck auf Taste tr nichts mehr.
ISR: void runterschalten()



Ereignisse im Zustandsdiagramm



Wir entwerfen zuerst das Zustandsdiagramm des Mikrocontrollers.



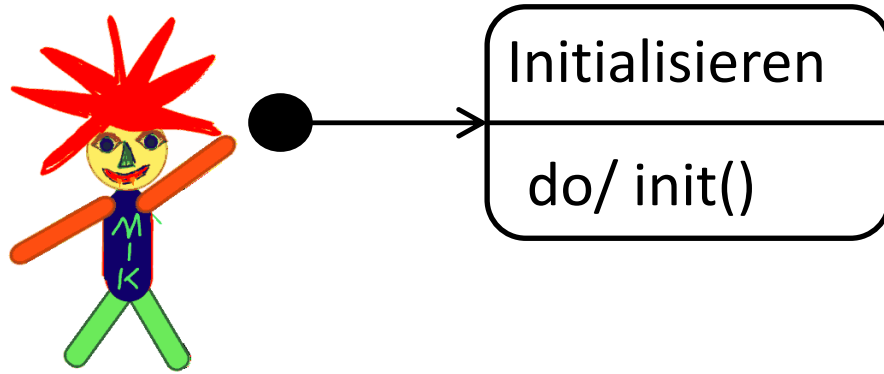
Ereignisse im Zustandsdiagramm



Das ist der Startzustand. Hier startet das Mikrocontrollerprogramm



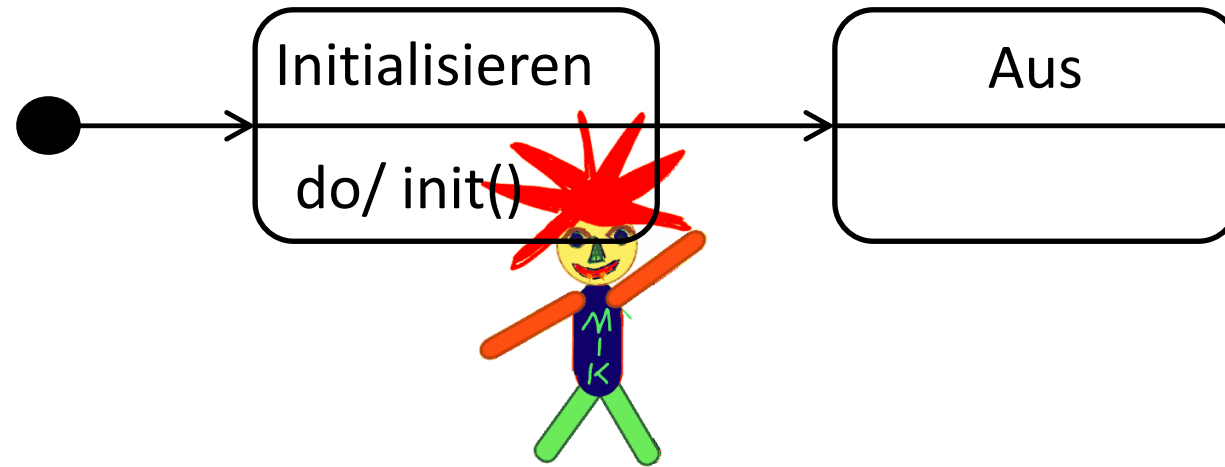
Ereignisse im Zustandsdiagramm



Das Programm wechselt in den Zustand „Initialisieren“ in dem die Ein- und Ausgänge initialisiert werden



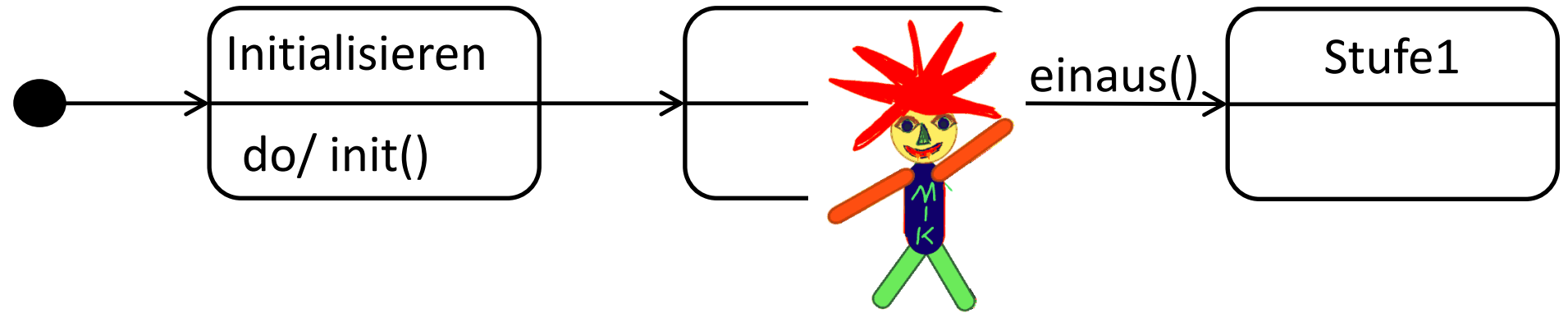
Ereignisse im Zustandsdiagramm



Danach wechselt das Programm in den Zustand „Aus“.
Der Zustandswechsel erfolgt ohne Wächterbedingung sobald die do-Aktivität „init()“ fertig ist.
Das Programm bleibt in diesem Zustand, bis Taster ta gedrückt wird.



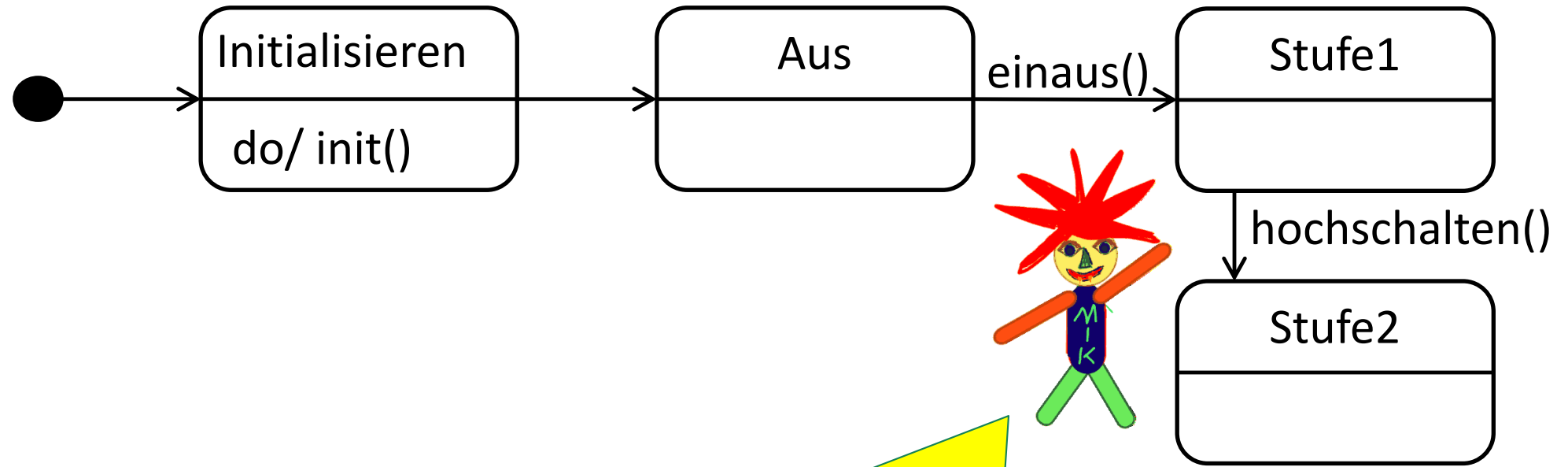
Ereignisse im Zustandsdiagramm



Taster ta gedrückt. ISR einaus() bewirkt ein Zustandswechsel nach Zustand „Stufe1“



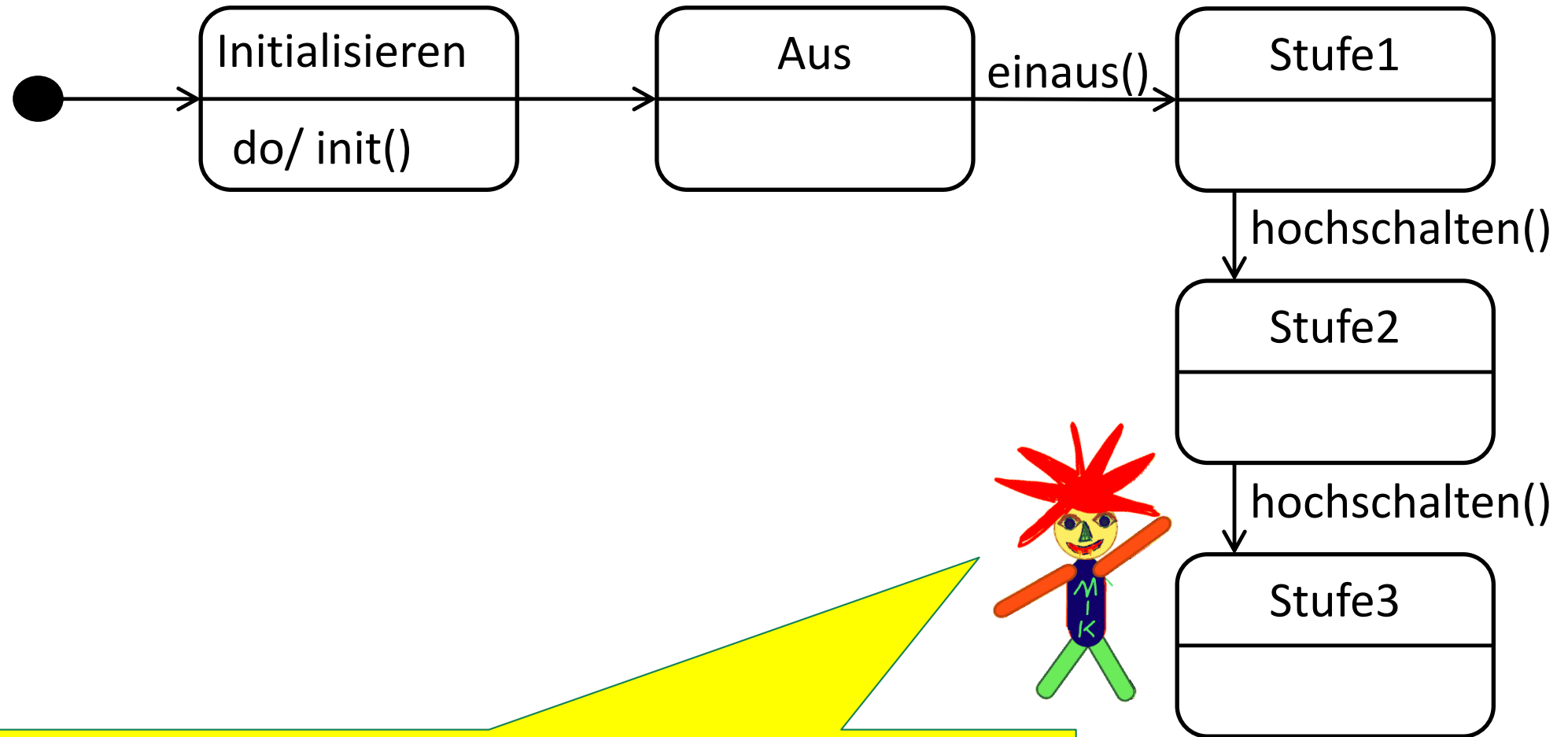
Ereignisse im Zustandsdiagramm



Taster tv gedrückt. ISR hochschalten() bewirkt ein Zustandswechsel nach Zustand „Stufe2“



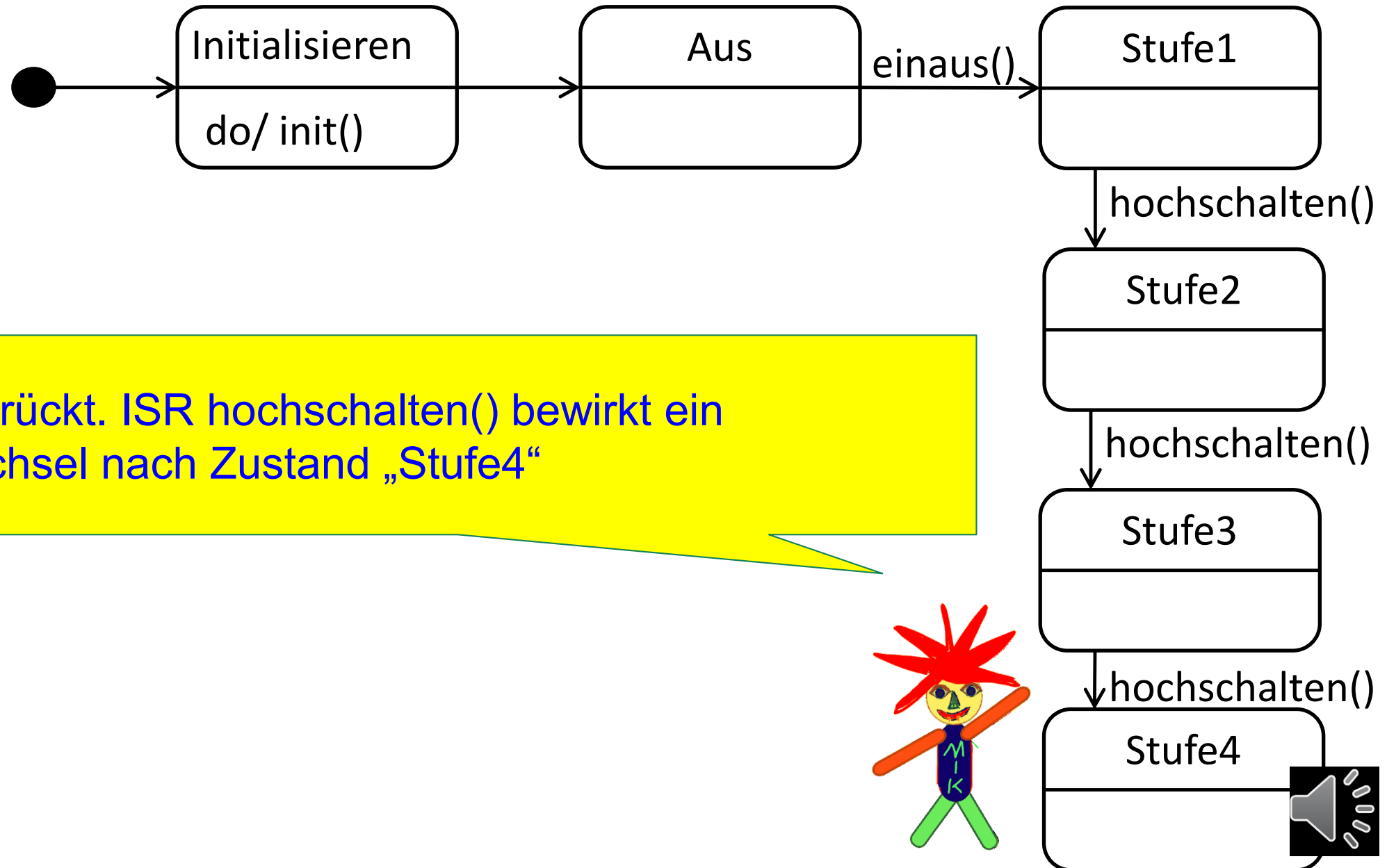
Ereignisse im Zustandsdiagramm



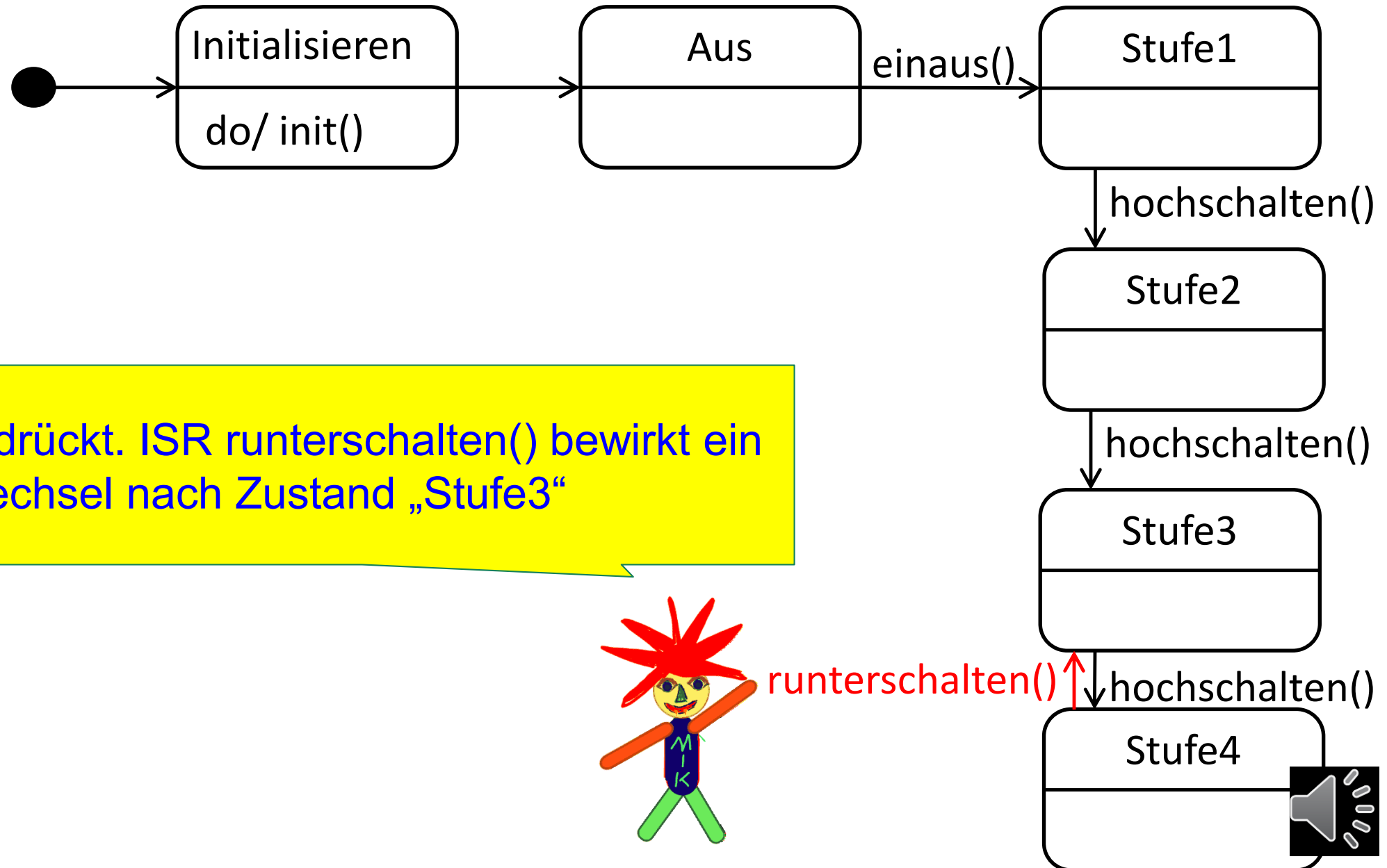
Taster tv gedrückt. ISR hochschalten() bewirkt ein Zustandswechsel nach Zustand „Stufe3“



Ereignisse im Zustandsdiagramm

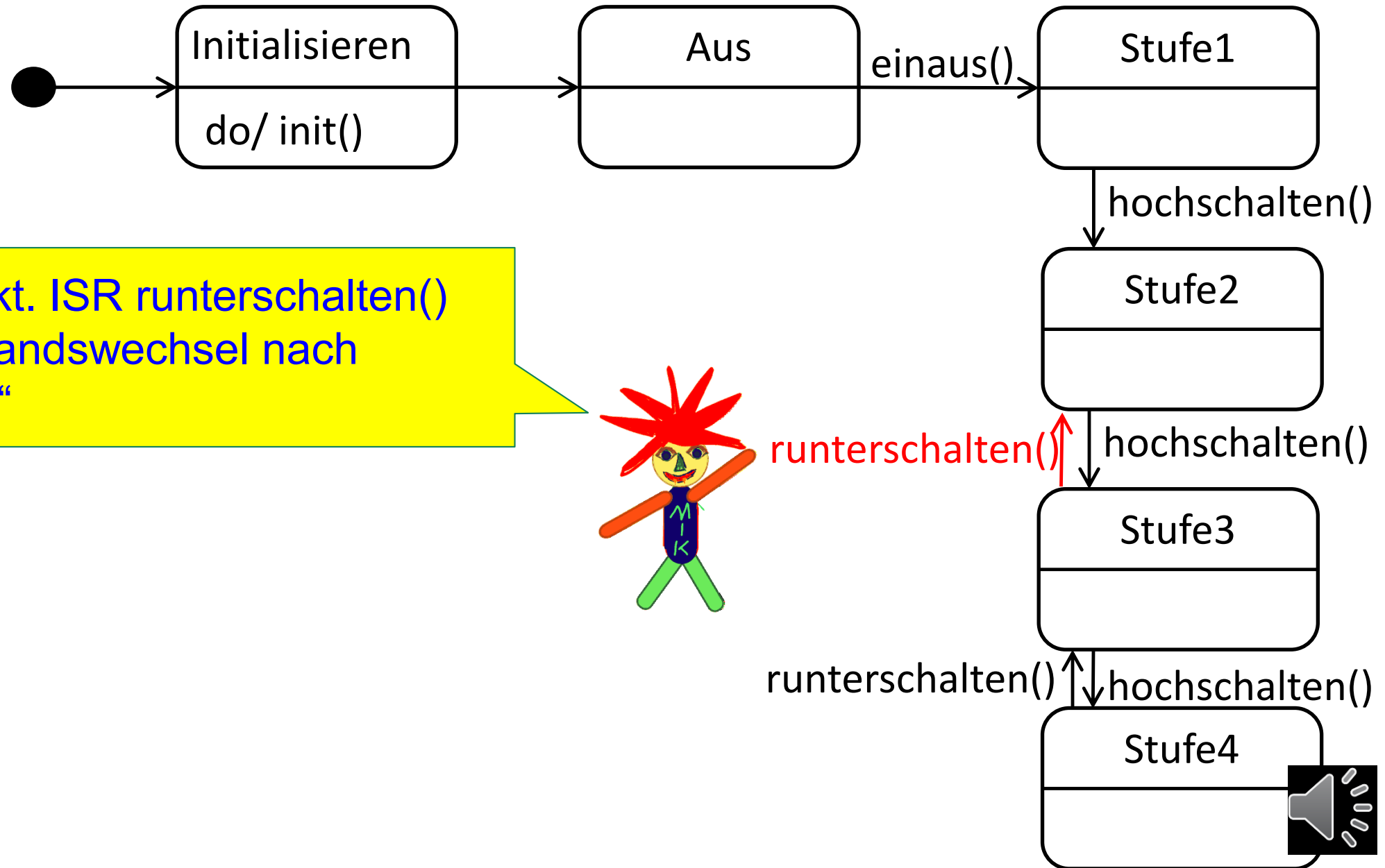


Ereignisse im Zustandsdiagramm

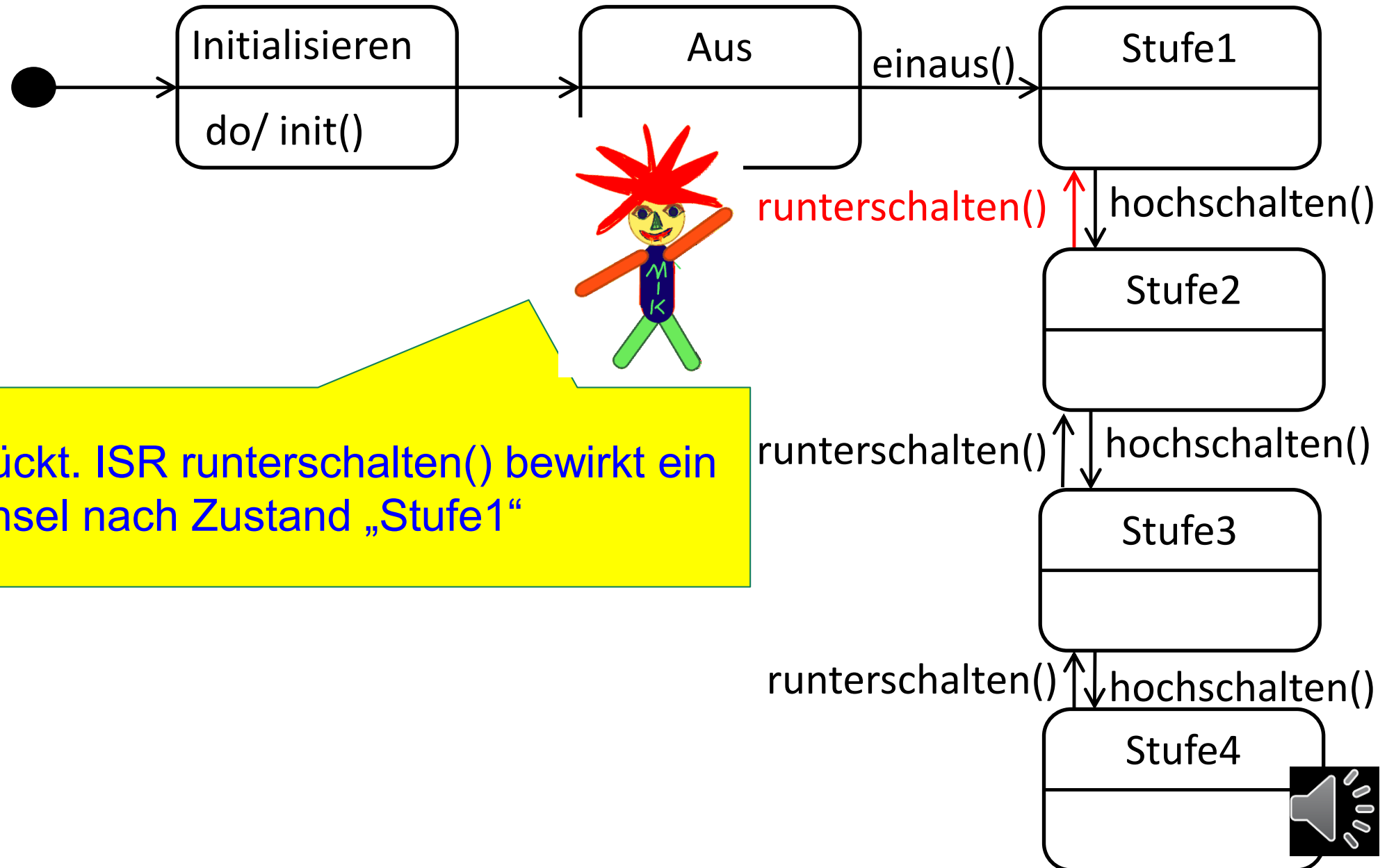


Taster tr gedrückt. ISR runterschalten() bewirkt ein Zustandswechsel nach Zustand „Stufe3“

Ereignisse im Zustandsdiagramm

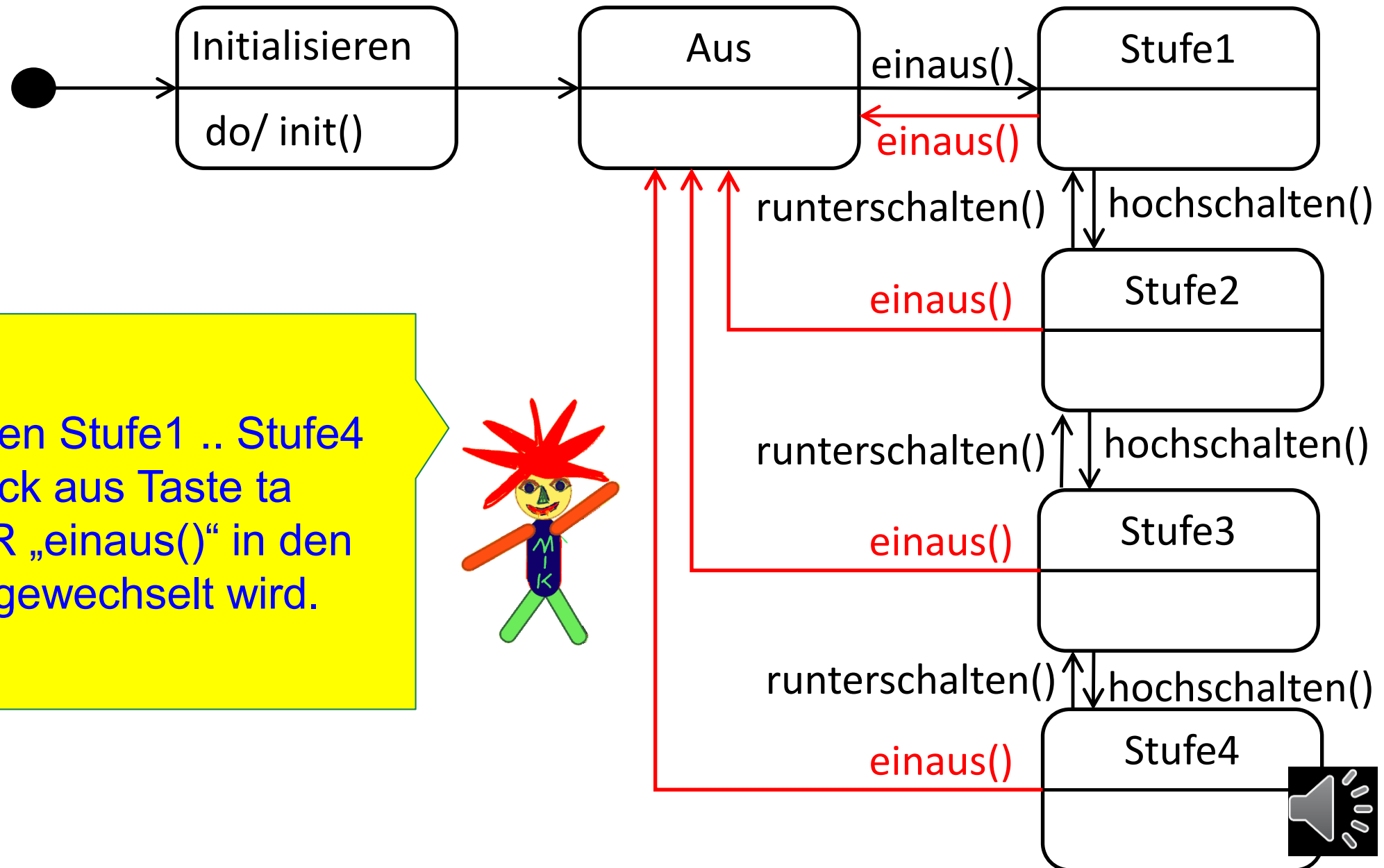


Ereignisse im Zustandsdiagramm



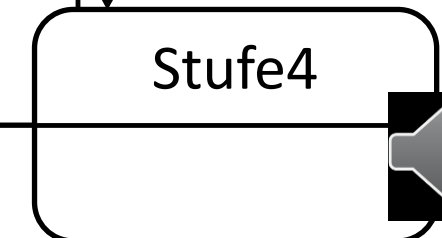
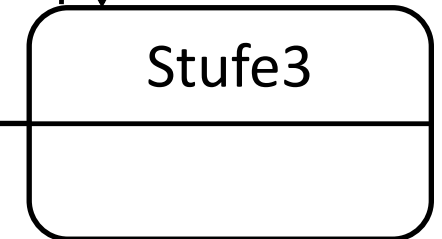
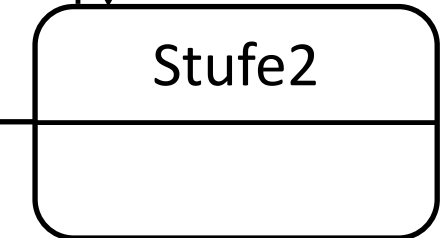
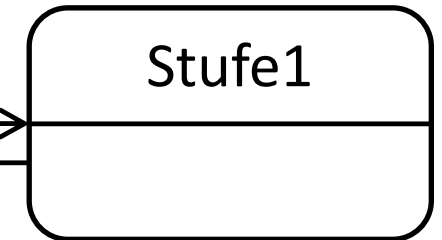
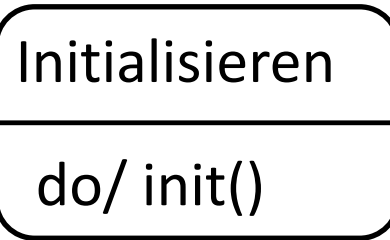
Taster tr gedrückt. ISR runterschalten() bewirkt ein Zustandswechsel nach Zustand „Stufe1“

Ereignisse im Zustandsdiagramm



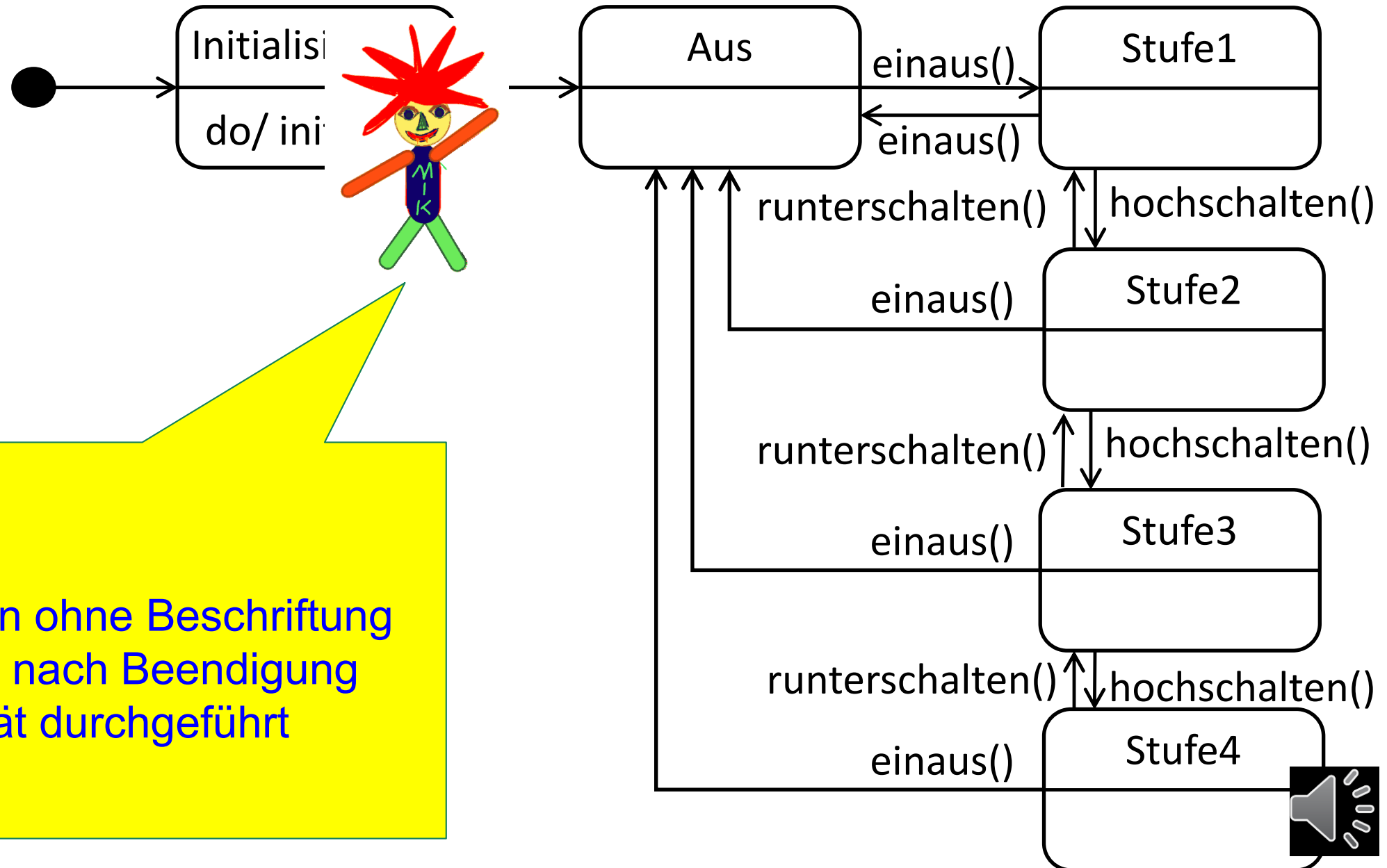
In den Zuständen Stufe1 .. Stufe4 bewirkt ein Druck aus Taste ta dass, in der ISR „einaus()“ in den Zustand „Aus“ gewechselt wird.

Ereignisse im Zustandsdiagramm



Was ist neu:
1. Der Anfang des Zustandsdiagramms wird durch den Startzustand gekennzeichnet. (ein ausgefüllter Kreis)

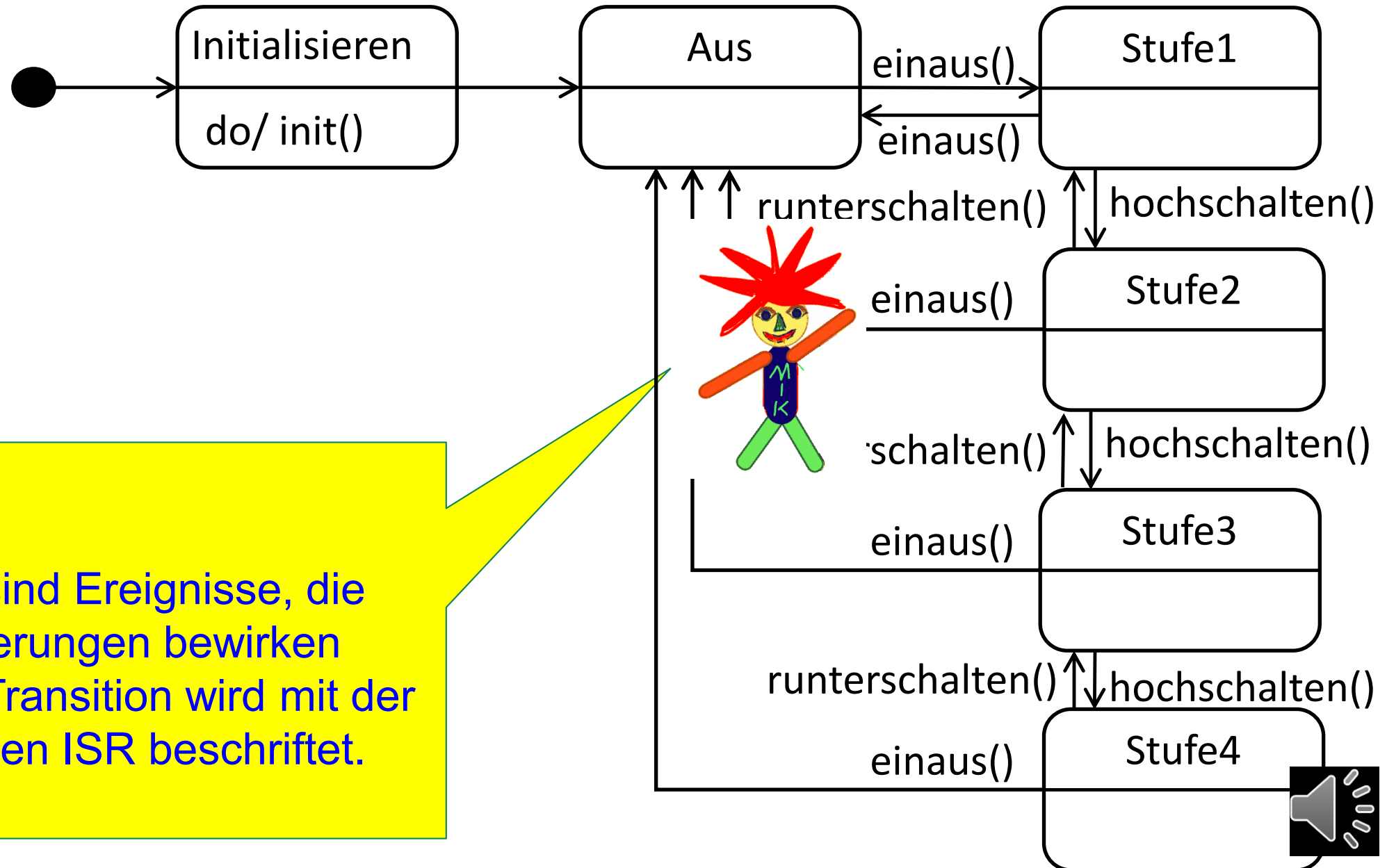
Ereignisse im Zustandsdiagramm



Was ist neu:

2. Transitionen ohne Beschriftung
werden sofort nach Beendigung
der do-Aktivität durchgeführt

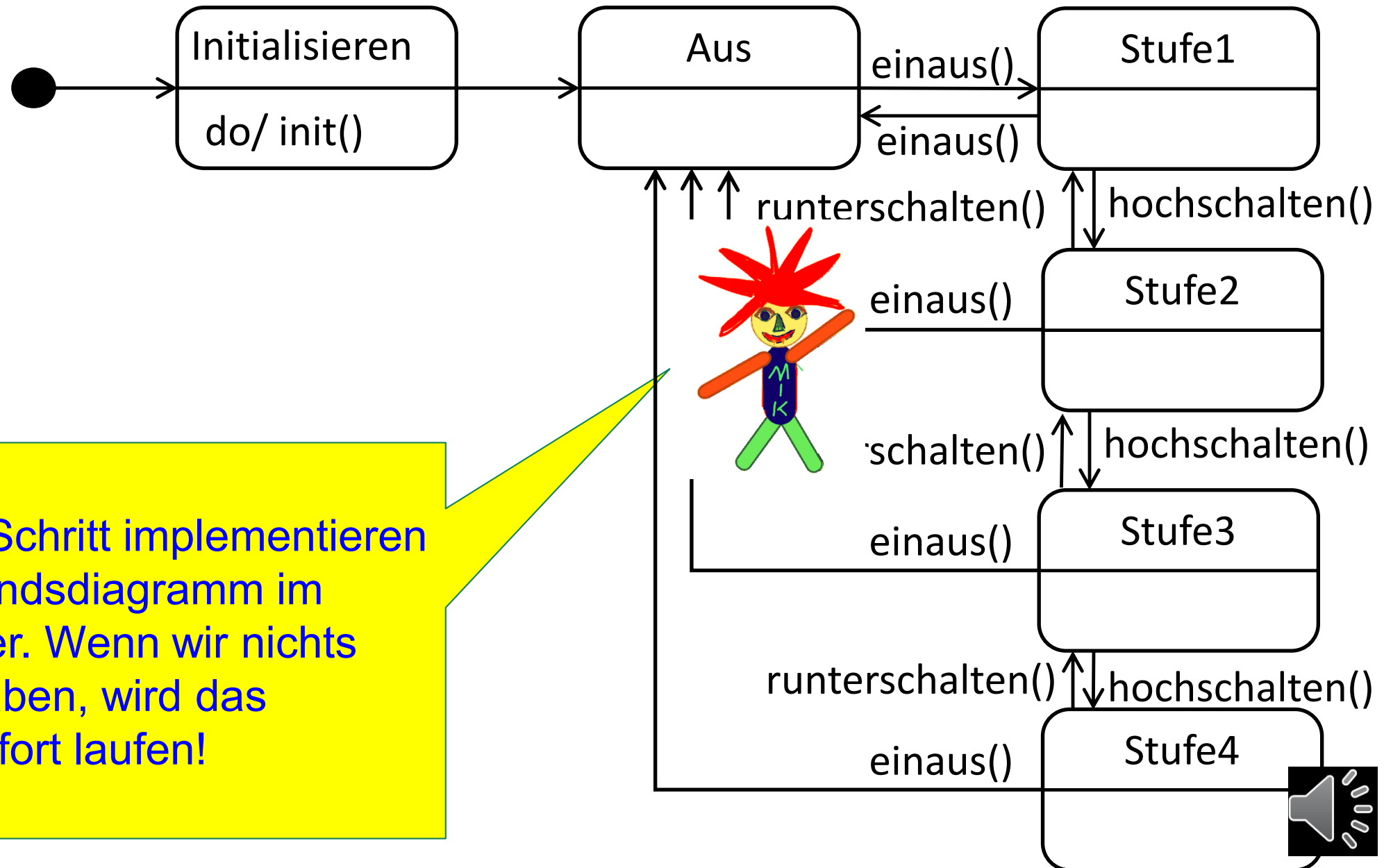
Ereignisse im Zustandsdiagramm



Was ist neu:

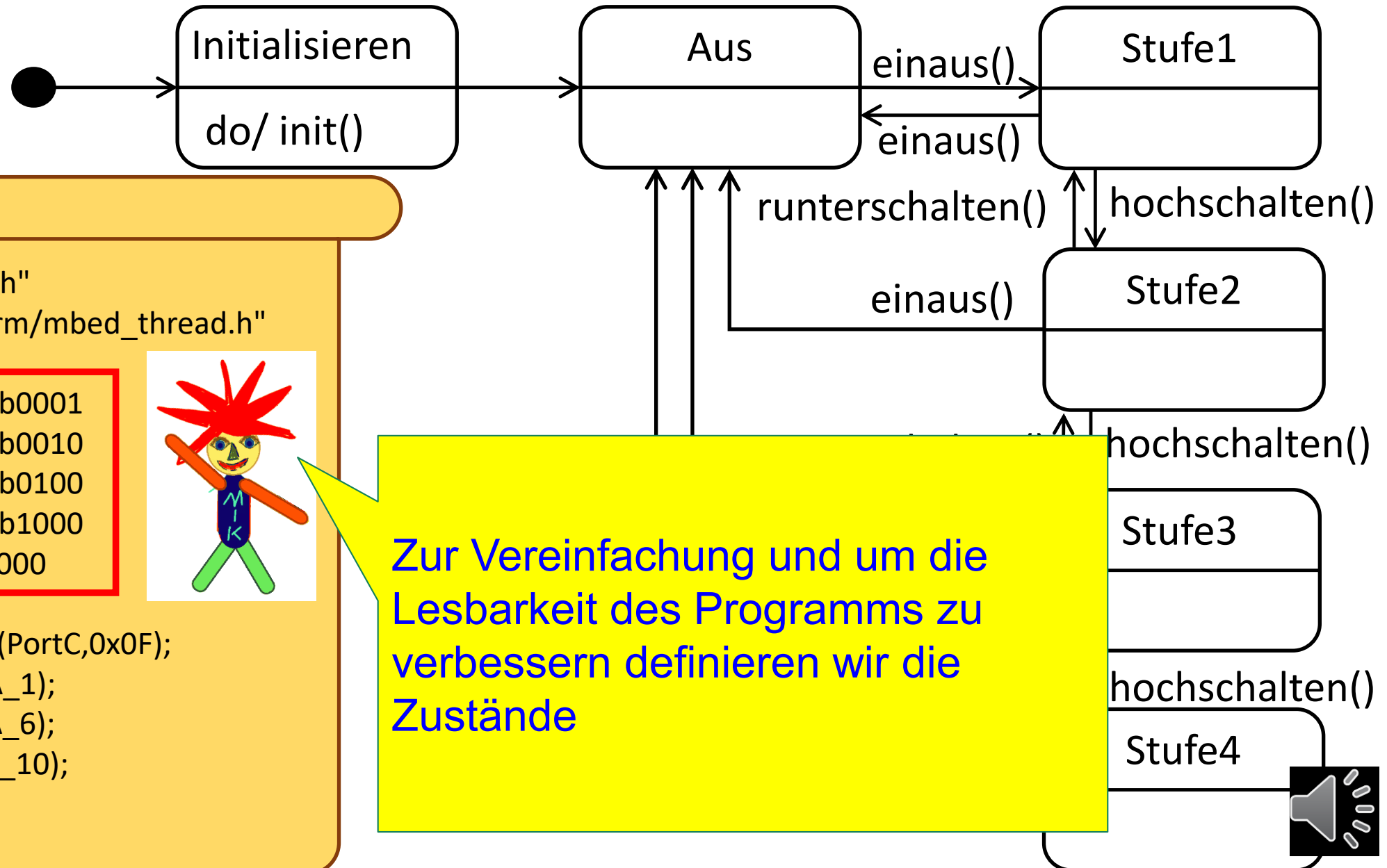
3. Interrupts sind Ereignisse, die Zustandsänderungen bewirken können. Die Transition wird mit der entsprechenden ISR beschriftet.

Ereignisse im Zustandsdiagramm



Im nächsten Schritt implementieren wir das Zustandsdiagramm im Mikrocontroller. Wenn wir nichts übersehen haben, wird das Programm sofort laufen!

Ereignisse im Zustandsdiagramm



```
#include "mbed.h"
#include "platform/mbed_thread.h"
```

```
#define Stufe1 0b0001
#define Stufe2 0b0010
#define Stufe3 0b0100
#define Stufe4 0b1000
#define Aus 0b0000
```

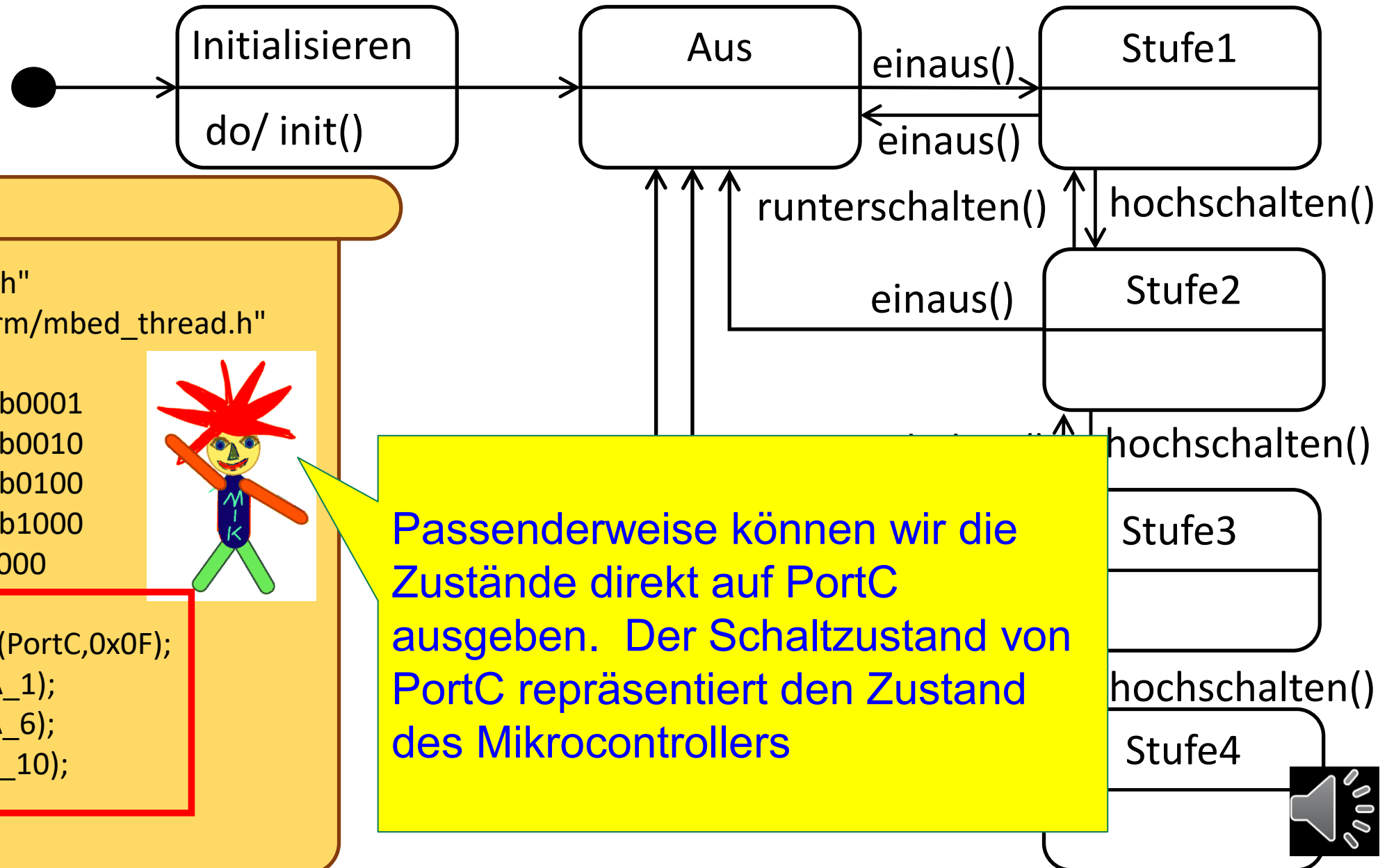


Zur Vereinfachung und um die Lesbarkeit des Programms zu verbessern definieren wir die Zustände

```
PortOut zustand(PortC,0x0F);
InterruptIn ta(PA_1);
InterruptIn tv(PA_6);
InterruptIn tr(PA_10);
```



Ereignisse im Zustandsdiagramm



```
#include "mbed.h"
#include "platform/mbed_thread.h"
```

```
#define Stufe1 0b0001
#define Stufe2 0b0010
#define Stufe3 0b0100
#define Stufe4 0b1000
#define Aus 0b0000
```



```
PortOut zustand(PortC,0x0F);
InterruptIn ta(PA_1);
InterruptIn tv(PA_6);
InterruptIn tr(PA_10);
```

Passenderweise können wir die Zustände direkt auf PortC ausgeben. Der Schaltzustand von PortC repräsentiert den Zustand des Mikrocontrollers



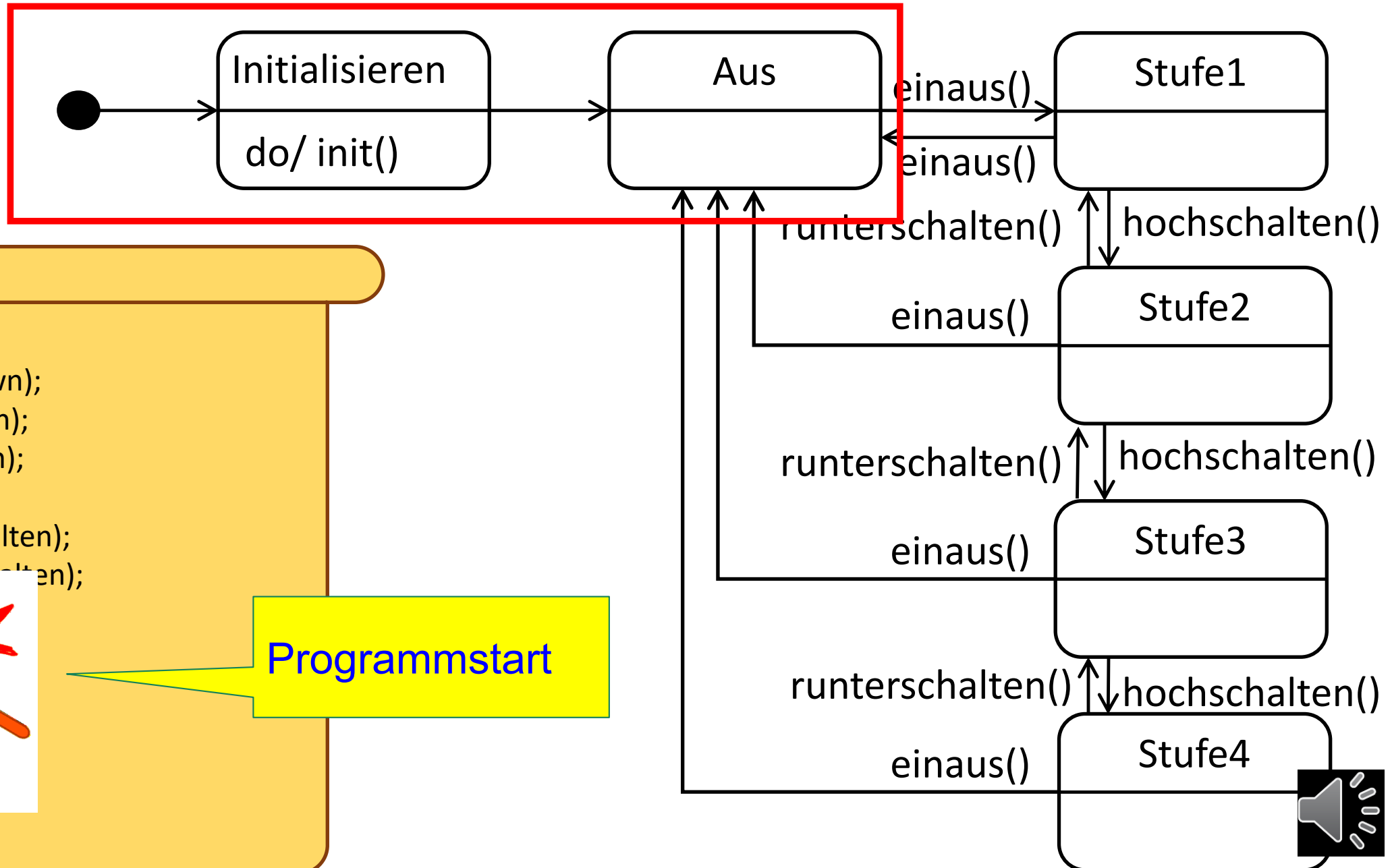
Ereignisse im Zustandsdiagramm

```
void init()
{ ta.mode(PullDown);
  tv.mode(PullDown);
  tr.mode(PullDown);
  ta.rise(&einaus);
  tv.rise(&hochschalten);
  tr.rise(&runterschalten);
}

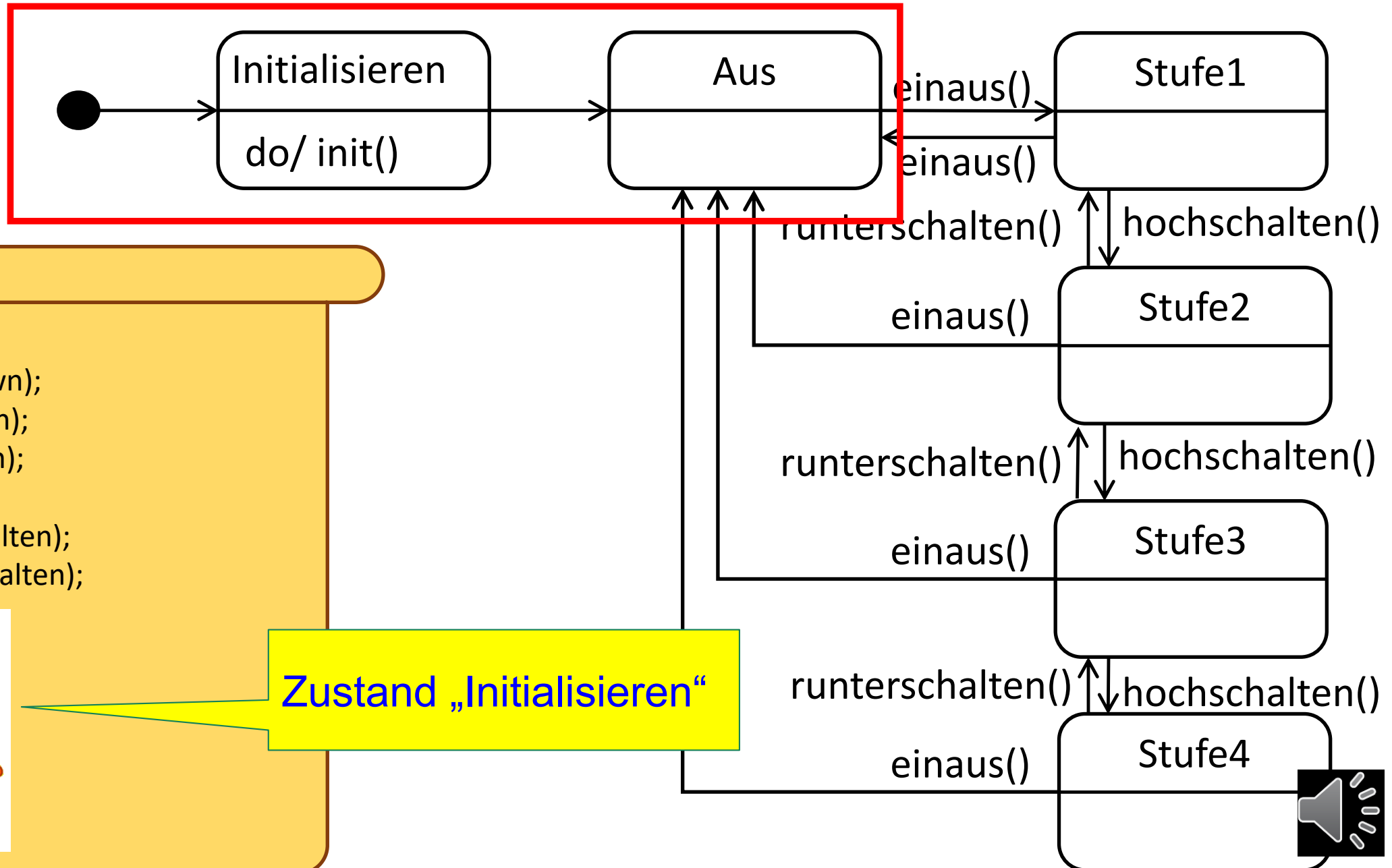
int main()
{ init();
  zustand:
  while (1)
  {
```



Programmstart



Ereignisse im Zustandsdiagramm

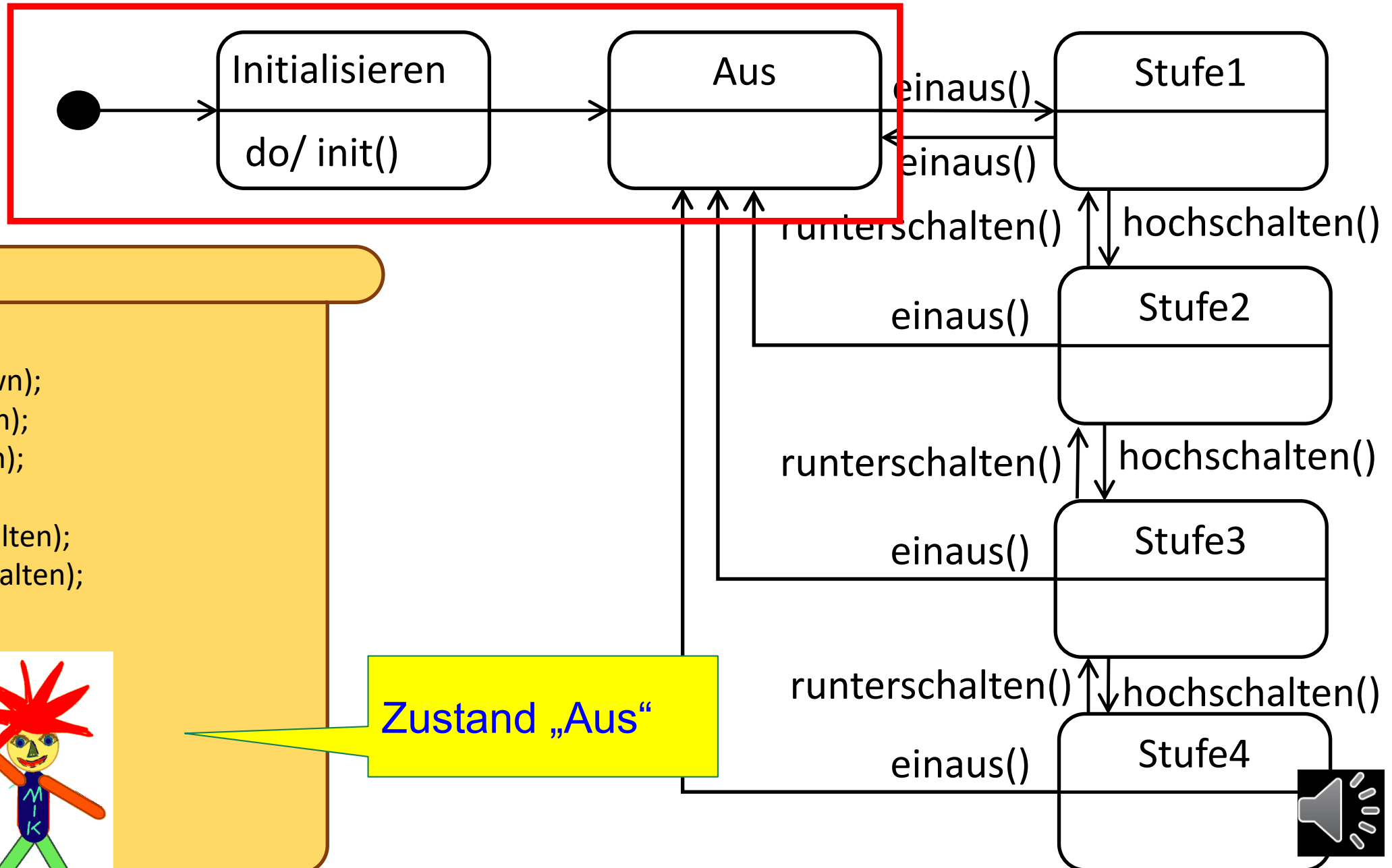


```
void init()
{ ta.mode(PullDown);
  tv.mode(PullDown);
  tr.mode(PullDown);
  ta.rise(&einaus);
  tv.rise(&hochschalten);
  tr.rise(&runterschalten);
}

int main()
{init();
  zustar
  while
}
```



Ereignisse im Zustandsdiagramm



```
void init()
{ ta.mode(PullDown);
  tv.mode(PullDown);
  tr.mode(PullDown);
  ta.rise(&einaus);
  tv.rise(&hochschalten);
  tr.rise(&runterschalten);
}

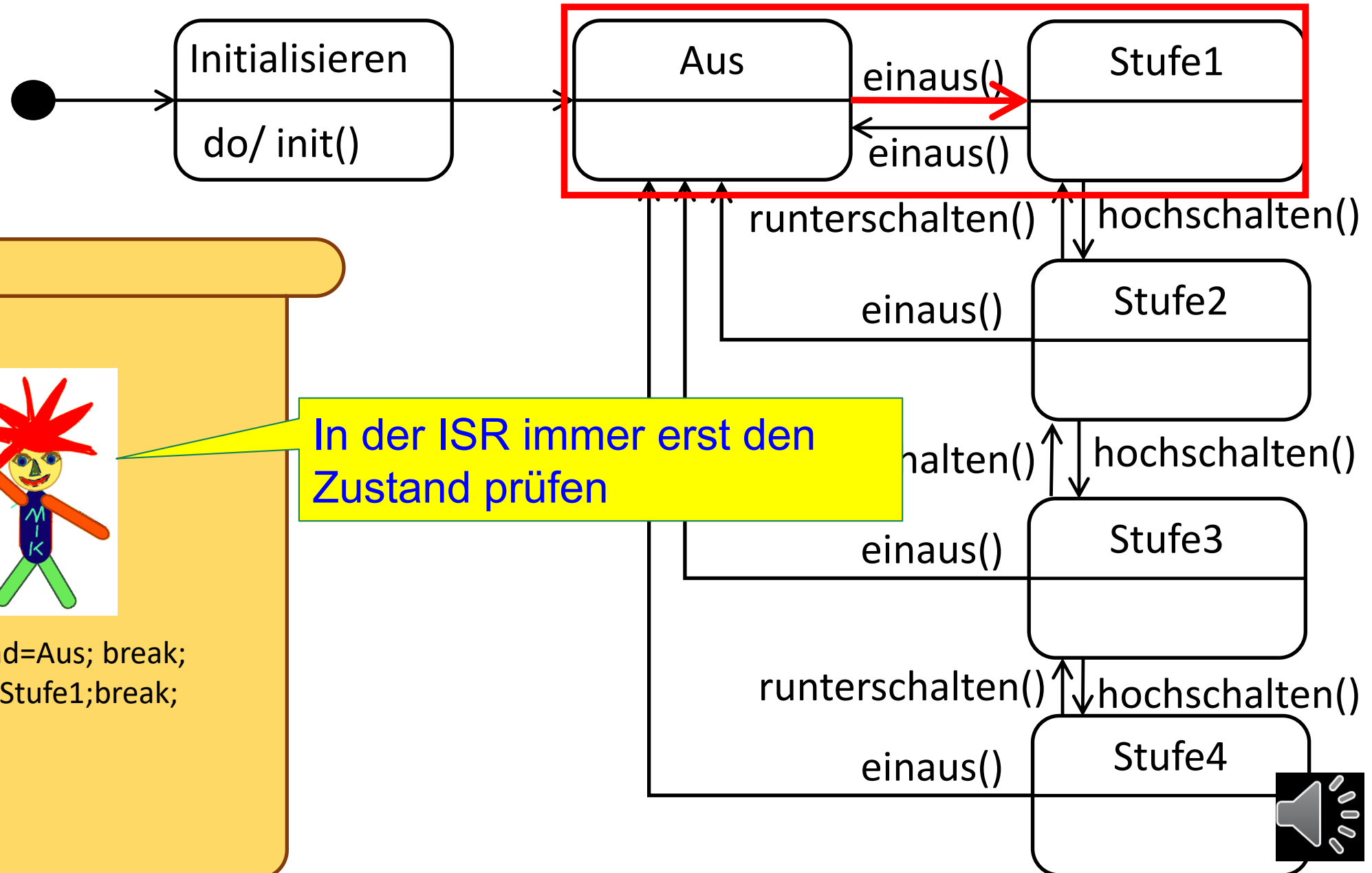
int main()
{init();
  zustand=Aus;
  while (true)
}
```



Zustand „Aus“



Ereignisse im Zustandsdiagramm



void einaus()

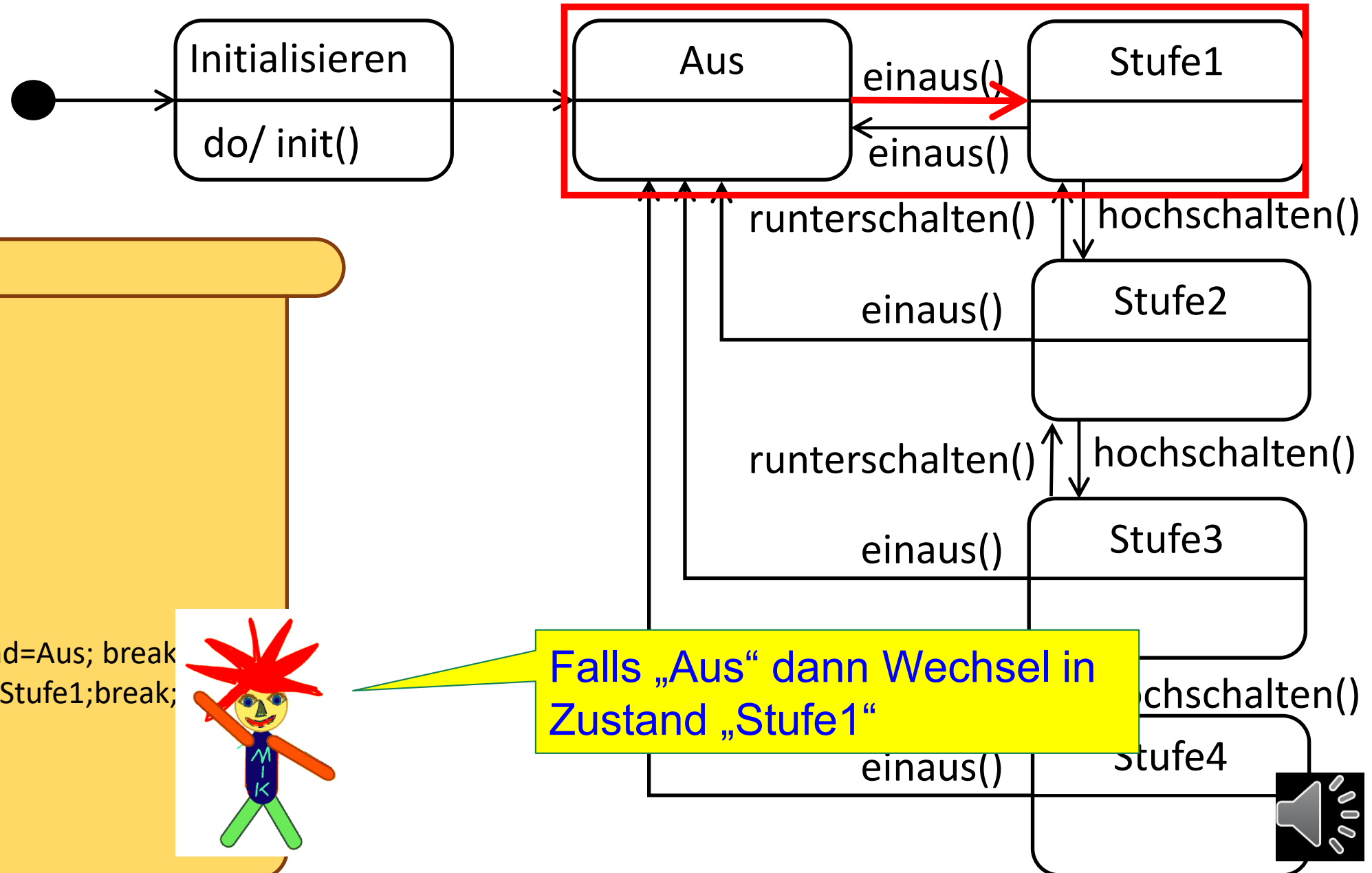
```
{
  switch(zustand)
  {
    case Stufe1:
    case Stufe2:
    case Stufe3:
    case Stufe4: zustand=Aus; break;
    case Aus: zustand=Stufe1;break;
  }
}
```



In der ISR immer erst den Zustand prüfen



Ereignisse im Zustandsdiagramm



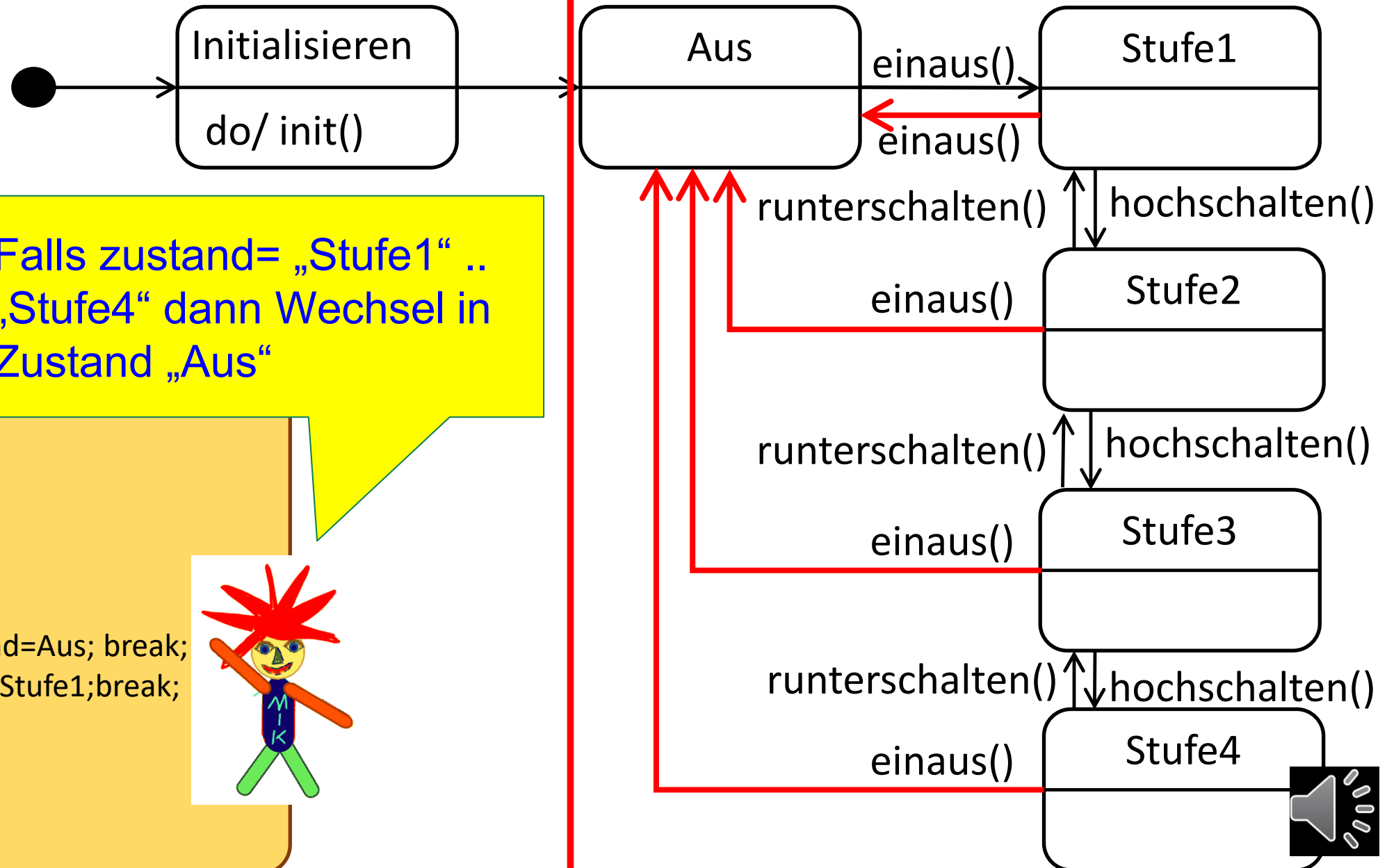
```
void einaus()
{
    switch(zustand)
    {
        case Stufe1:
        case Stufe2:
        case Stufe3:
        case Stufe4: zustand=Aus; break;
        case Aus: zustand=Stufe1; break;
    }
}
```



Falls „Aus“ dann Wechsel in Zustand „Stufe1“



Ereignisse im Zustandsdiagramm

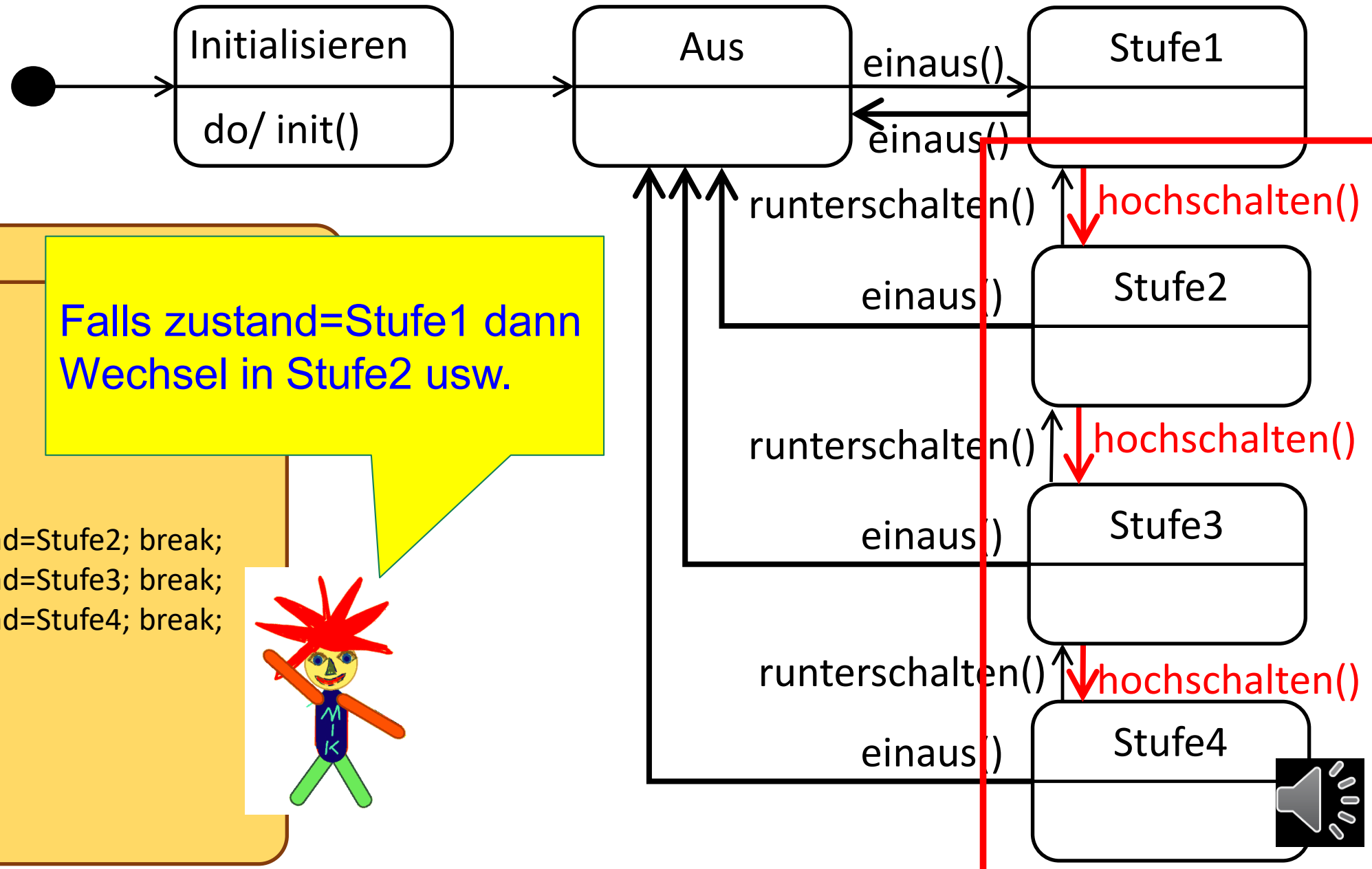


Falls zustand= „Stufe1“ .. „Stufe4“ dann Wechsel in Zustand „Aus“

```
void einaus()
{
    switch(zustand)
    {
        case Stufe1:
        case Stufe2:
        case Stufe3:
        case Stufe4: zustand=Aus; break;
        case Aus: zustand=Stufe1;break;
    }
}
```



Ereignisse im Zustandsdiagramm



Falls zustand=Stufe1 dann Wechsel in Stufe2 usw.

```
void hochschalten()
{
    switch(zustand)
    {
        case Stufe1: zustand=Stufe2; break;
        case Stufe2: zustand=Stufe3; break;
        case Stufe3: zustand=Stufe4; break;
    }
}
```



