FSM-Implementierung in Aseba

```
# If the robot is bumping, this behavior performs a set sequence of
# actions. First it backs up for some time (1s), and then it
# spins in place for some time (1s), then goes forward.

var state = START

# proximity sensor event
onevent prox
        if state == START then
                if  prox.horizontal[2] > 3000 then
                        timer.period[0] = 1000 # 1s
                        state = BACKUP
                end

        elseif state ==  BACKUP then
                motor.left.target = -200
                motor.right.target = -200

        elseif state == SPIN  then
                motor.left.target = 200
                motor.right.target = -200

        elseif state == FORWARD then
                motor.left.target = 200
                motor.right.target = 200
                if prox.horizontal[2] > 3000 then
                        timer.period[0] = 0
                        timer.period[0] = 1000 # 1s
                        state = BACKUP
                end
         end

# timer 0 event
onevent timer0
        timer.period[0] = 0 # stop timer

        if state == BACKUP then
                state = SPIN
                timer.period[0] = 2000 # 2s

        elseif state == SPIN then
                state = FORWARD
                timer.period[0] = 1000   # 1s

        elseif state == FORWARD then
                state  = START
        end
```

Routinen werden *bei Auftreten von Ereignissen* durchgeführt (ereignisbasierte Architektur), z.B. die Aktualisierung der Werte der Infrarotsensoren;
der Zustand wird in Variable state gehalten (Anfangszustand: state := START)
Schnittstelle zum Roboter durch vorhandene Variablen (z.B. motor.left.target),
andere verbreitete Programmiersprachen für Roboter: C/C++, Python, …

Vorteil: beliebige Algorithmen (inkl. Regler) implementierbar