

```
1 // UserFiniteStateMachine.h
2
3 /*=====
4 /* Application code
5 /*
6 /* Name: UserFSM()
7 /* Version:
8 /* Date:
9 /* Author:
10 /*
11 /* Short description:
12 /*
13 /*
14 /*=====
15 /*=====
16 /*===== START USER APPLICATION ===== START USER APPLICATION =====
17 /*=====
18 /*=====
19 //
20 // State machine
21 //
22 // Each phase exists of 2 parts, e.g. the phase actions and transition conditions.
23 // In the phase actions, all activated tags are mentioned. Deactivation is not needed, deactivation is
24 // done at the end of the main loop. In the conditions section, all conditions for the transitions to another
25 // phase are mentioned. When all conditions are true, the new phase can be set. It is possible to programm
26 // more then 1 transition. In such a case it is necessary to keep the code order in mind.
27 //
28 // Arduino Finite State Machine
29 // Copyright(C) 2018 Jelle Siemonsma
30 //
31 // This program is free software : you can redistribute it and/or modify
32 // it under the terms of the GNU General Public License as published by
33 // the Free Software Foundation, either version 3 of the License, or
34 // (at your option) any later version.
35 //
```

```

36 // This program is distributed in the hope that it will be useful,
37 // but WITHOUT ANY WARRANTY; without even the implied warranty of
38 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
39 // GNU General Public License for more details.
40 //
41 // < https://www.gnu.org/licenses/>.
42 //
43 /*=====
44
45
46 void UserFSM()
47 {
48
49     if (MachineState == FiniteState("START"))
50     {
51         ActHour = hour();
52         TransitionToState("SET");
53         if (AnaHigher("LDR1", Twilight)) TransitionToState("DAYLIGHT");
54         else
55             if ((AnaLower("LDR1", Twilight) && (ActHour > 12)) TransitionToState("TO-EVENING");
56             else
57                 if ((AnaLower("LDR1", Twilight) && (ActHour <= 12)) TransitionToState("MORNING");
58         UpdateMarker("DARK", String(Dark));
59         UpdateMarker("TWILIGHT", String(Twilight));
60     };
61
62     if (MachineState == FiniteState("DAYLIGHT"))
63     {
64         if (Rst) TransitionToState("END");
65         ActHour = hour();
66         if (AnaLower("LDR1", Twilight) && (ActHour > 12)) TransitionToState("TO-EVENING");
67     };
68
69     if (MachineState == FiniteState("TO-EVENING"))
70     {

```

```
71     if (Rst) TransitionToState("END");
72     if (AnaHigher("LDR1", Twilight + 10))
73     {
74         TransitionToState("DAYLIGHT");
75         CancelTimer(timStable);
76     }
77     if (Timer(30000, timStable)) TransitionToState("EVENING");
78 };
79
80 if (MachineState == FiniteState("EVENING"))
81 {
82     if (Rst) TransitionToState("END");
83     if (AnaLower("LDR1", Dark) && (ActHour > 12)) TransitionToState("TO-NIGHT");
84 };
85
86 if (MachineState == FiniteState("TO-NIGHT"))
87 {
88     if (Rst) TransitionToState("END");
89     if (AnaHigher("LDR1", Dark + 10 ))
90     {
91         TransitionToState("EVENING");
92         CancelTimer(timStable);
93     }
94     if (Timer(30000, timStable))
95     {
96         TransitionToState("NIGHT");
97         RdHour = random(21, 22);
98         RdMinute = random(30, 59);
99         UpdateMarker("RANDOM-OFF", String(RdHour) + ":" + String(RdMinute));
100    }
101 };
102
103 if (MachineState == FiniteState("NIGHT"))
104 {
105     if (Rst) TransitionToState("END");
```

```
106     if ((hour() >= 22) || ((hour() >= 21) && (minute() >= RdMinute)))
107     {
108         TransitionToState("TO-MORNING");
109         TransitionToState("CHKMOV");
110     }
111     Activate("K1");
112 };
113
114 if (MachineState == FiniteState("CHKMOV"))
115 {
116     if (Rst) TransitionToState("END");
117     ActHour = hour();
118     if (!DelayTimer(600000, delK1)) Activate("K1");
119     if (DigRising("MOV1")) RestartDelayTimer(delK1);
120     if (ActHour == 7) TransitionToState("END");
121 };
122
123 if (MachineState == FiniteState("TO-MORNING"))
124 {
125     if (Rst) TransitionToState("END");
126     ActHour = hour();
127     if (ActHour == 7) TransitionToState("MORNING");
128 };
129
130 if (MachineState == FiniteState("MORNING"))
131 {
132     if (Rst) TransitionToState("END");
133     if (!DelayTimer(600000, delK1)) Activate("K1");
134     if (DigRising("MOV1")) RestartDelayTimer(delK1);
135     ActHour = hour();
136     if (ActHour == 10) TransitionToState("DAYLIGHT");
137 };
138
139 if (MachineState == FiniteState("SET"))
140 {
```

```
141     if (DigLow("SETREF"))
142     {
143         TransitionToState("SETREF");
144         NumberMeas = 0;
145         TotalDark = 0.0;
146         UpdateMarker("DARK", String(Dark));
147         UpdateMarker("TWILIGHT", String(Twilight));
148     }
149     if (Rst) Rst = !Rst;
150 };
151
152 if (MachineState == FiniteState("SETREF"))
153 {
154     if (DelayTimer(4000, timSetRef1))
155     {
156         if (!DelayTimer(8000, timSetRef2))
157         {
158             if (Timer(100, timSetRef4)) AanUit = !AanUit; else if (AanUit) Activate("MeasActive");
159
160             if (Timer(500, timSetRef3))
161             {
162                 TotalDark = TotalDark + AnaValue("LDR1");
163                 NumberMeas = NumberMeas + 1;
164             }
165         }
166     } else
167     {
168         Dark = int(TotalDark / NumberMeas);
169         Twilight = Dark + 50;
170         UpdateMarker("DARK", String(Dark));
171         UpdateMarker("TWILIGHT", String(Twilight));
172         CancelTimer(timSetRef1);
173         CancelTimer(timSetRef2);
174         CancelTimer(timSetRef3);
175         TransitionToState("SET");

```

```
176         TransitionToState("START");
177     }
178 }
179 else
180 {
181     Activate("MeasActive");
182     Rst = true;
183 }
184 };
185
186
187
188
189
190 // *%END USER APPLICATION Generation directive dont remove!=====
191 // *=====
192 // *=====
193 }
194
```