

```

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 condition are true, the new phase can be set. It is possible to programm
26 // more then 1 transition. In such a case it is necassery to keep the priority in mind.
27 //
28 /*=====
29
30
31 void UserFSM()
32 {
33
34     // START
35     //

```

```
36 // Doe de initialisatie voor het programma.
37 // Breng de spindel naar rechts tot het eindcontact, zodat het programma
38 // weet waar de beker houder is. Ga start daarna de status INIT, NORMAL enn CIP.
39 //
40 if (MachineState == FiniteState("START"))
41 {
42     // Move the cup transport to the left to find the zero position...
43     // Now we know where we are in the machine
44     Spindle.setSpeed(-200);
45     Spindle.runSpeed();
46     CupPresent = false;
47     NewCup = true;
48     PrepareDrink = false;
49     //UpdateServo("HATCH", 90);
50     //UseHMISerial = false;
51     if (DigLow("ZERO"))
52     {
53         Spindle.stop();
54         Spindle.disableOutputs();
55         Spindle.setCurrentPosition(0);
56         TransitionToState("INIT");
57         TransitionToState("NORMAL");
58         TransitionToState("CIP");
59     }
60 };
61
62 // NORMAL
63 //
64 // Door de toetsen combinatie F1 en F2, 5 seconden in te houden en onder
65 // voorwaarde dat we géén drankjes maken; start dan nu de service mode.
66 //
67 if (MachineState == FiniteState("NORMAL"))
68 {
69     Service = 0;
70     //if (DigLow("F1") && DigLow("F2") && DigLow("F3") && !Service) if (Timer(5000, timService)) TransitionToState
```

```
    ("SERVICE");
71     if (DigLow("F1") && DigLow("F2") && !PrepareDrink)
72     {
73         if (Timer(5000, timService)) TransitionToState("SERVICE");
74     }
75     else CancelTimer(timService);
76 }
77
78 // SERVICE
79 //
80 // Laat de LEDS in een looplicht oplichten. Het is nu mogelijk de betreffende
81 // pompjes en de water klep te bedienen buiten de besturing om.
82 // Leid het water naar een riool punt!
83 // Wanneer 3 seconden op F1 en F2 wordt gedrukt, wordt de service mode
84 // weer verlaten.
85 //
86 if (MachineState == FiniteState("SERVICE"))
87 {
88     Service = 1;
89     if (Timer(50, timServiceMode)) ServiceLED = ServiceLED + 1;
90     if (ServiceLED == 6) ServiceLED = 1;
91     switch (ServiceLED)
92     {
93     case 1:
94         Activate("ACKD1");
95         break;
96     case 2:
97         Activate("ACKD2");
98         break;
99     case 3:
100        Activate("ACKD3");
101        break;
102    case 4:
103        Activate("ACKSTART");
104        break;
```

```
105     case 5:
106         Activate("READY");
107         break;
108     }
109     if (DigLow("F1") && DigHigh("F2") && DigHigh("F3") && DigHigh("START")) { Activate("&D1"); Activate("ACKD1"); }
110     if (DigHigh("F1") && DigLow("F2") && DigHigh("F3") && DigHigh("START")) { Activate("&D2"); Activate("ACKD2"); }
111     if (DigHigh("F1") && DigHigh("F2") && DigLow("F3") && DigHigh("START")) { Activate("&D3"); Activate("ACKD3"); }
112     if (DigHigh("F1") && DigHigh("F2") && DigHigh("F3") && DigLow("START")) { Activate("&DW"); Activate("ACKSTART"); }
113     //if (DigLow("F1") && DigLow("F2") && DigLow("F3")) if (Timer(3000, timService)) TransitionToState("NORMAL");
114     if (DigLow("F1") && DigLow("F2"))
115     {
116         if (Timer(3000, timService)) TransitionToState("NORMAL");
117     }
118     else CancelTimer(timService);
119 };
120
121 // INIT
122 //
123 // Na het initialiseren van het programma, brengt de besturing de beker houder op de
124 // laad/los positie van de machine.
125 // Start nu de state CHOICE (keuze maken) en CUPPRESSENT (check op "beker aanwezig").
126 //
127 if (MachineState == FiniteState("INIT"))
128 {
129     // Move to the load/unload position
130     if (Timer(2000, timStartup))
131     {
132         Spindle.setSpeed(350);
133         Spindle.runToNewPosition(200);
134         Spindle.setCurrentPosition(0);
135         Spindle.disableOutputs();
136         TransitionToState("CHOICE");
137         TransitionToState("CUPPRESSENT");
138     }
139 };
```

```
140
141 // CHOICE
142 //
143 // Wacht op de keuze van de gebruiker voor één van de drie smaken.
144 // Wanneer er een keuze is gemaakt EN er is een beker aanwezig EN
145 // de machine zit niet in de servide mode; Start dan status MIX.
146 //
147 if (MachineState == FiniteState("CHOICE"))
148 {
149     if (CupPresent && !Service) if (Timer(400, timAanUit)) AanUit = !AanUit; else if (AanUit)
150     {
151         Activate("ACKD1");
152         Activate("ACKD2");
153         Activate("ACKD3");
154     }
155     if (DigRising("F1"))
156     {
157         LiquirNo = 1;
158         TimeLiquir = 2200;
159         PositionLiquir = 6750;
160     }
161     if (DigRising("F2"))
162     {
163         LiquirNo = 2;
164         TimeLiquir = 2400;
165         PositionLiquir = 4850;
166     }
167     if (DigRising("F3"))
168     {
169         LiquirNo = 3;
170         TimeLiquir = 2200;
171         PositionLiquir = 2850;
172     }
173     if ((DigRising("F1") || DigRising("F2") || DigRising("F3")) && CupPresent && !Service) TransitionToState("MIX");
174     if (!CupPresent) if (Timer(150, timAanUit)) AanUit = !AanUit; else if (AanUit) Activate("READY");
```

```
175     };
176
177     // MIX
178     //
179     // Wacht op de bevestiging van de keuze (knop "START") OF op het drukken op een
180     // keuze knop. Wanneer er weer op een keuze knop wordt gedrukt, gaat de status
181     // terug naar CHOICE. Bij een bevestiging, sluit het toegangshekje en start de
182     // besturing de status GOTO-LIQUIR, ondervoorwaarde dat er een nieuwe beker
183     // staat EN deze op de laad/los plaats staat.
184     //
185     if (MachineState == FiniteState("MIX"))
186     {
187         Activate("ACKD" + String(LiquirNo));
188         if (Timer(400, timAanUit) AanUit = !AanUit; else if (AanUit) Activate("ACKSTART");
189         if (!CupPresent) if (Timer(150, timAanUit) AanUit = !AanUit; else if (AanUit) Activate("READY");
190         if ((DigRising("START")) && CupPresent && NewCup)
191         {
192             //UpdateServo("HATCH", 1);
193             GoMix = true;
194             PrepareDrink = true;
195             Activate("ACKSTART");
196             CupPresent = false;
197             //UseHMISerial = true;
198             UpdateText("MUZIEK", "---" );
199             //UseHMISerial = false;
200             TransitionToState("GOTO-LIQUIR");
201         }
202         if (DigRising("F1") || DigRising("F2") || DigRising("F3")) TransitionToState("CHOICE");
203     };
204
205     // GOTO-LIQUIR
206     //
207     // Stop bij de juiste positie voor één van de drie
208     // limonade vulpunten. Bij aankomst start de status LIQUIR.
209     //
```

```
210     if (MachineState == FiniteState("GOTO-LIQUIR"))
211     {
212         Activate("ACKD" + String(LiquirNo));
213         Activate("ACKSTART");
214         UpdateText("MUZIEK", "F" + String(LiquirNo));
215         if (Timer(200, timWait))
216         {
217             Spindle.setSpeed(350);
218             Spindle.runToNewPosition(PositionLiquir);
219             Spindle.disableOutputs();
220             TransitionToState("LIQUIR");
221         }
222     };
223
224     // LIQUIR
225     //
226     // Sart de pomp voor het limonade smaak voor bepaalde tijd.
227     // Ga daarna naar status GOTO-WATER.
228     //
229     if (MachineState == FiniteState("LIQUIR"))
230     {
231         Activate("ACKD" + String(LiquirNo));
232         Activate("ACKSTART");
233         Activate("&D" + String(LiquirNo));
234         if (Timer(TimeLiquir,timLimo)) TransitionToState("GOTO-WATER");
235     };
236
237     // GOTO-WATER
238     //
239     // Wacht eerst 4 seconden, om de limonade te laten uitdruppen en
240     // ga dan terug naar de laad/los positie. Bij aankomst start
241     // de water dosering in status WATER.
242     //
243     if (MachineState == FiniteState("GOTO-WATER"))
244     {
```

```
245     Activate("ACKD" + String(LiquirNo));
246     Activate("ACKSTART");
247     if (Timer(4000, timWait))
248     {
249         Spindle.setSpeed(350);
250         Spindle.runToNewPosition(0);
251         Spindle.disableOutputs();
252         TransitionToState("WATER");
253     }
254 };
255
256 // WATER
257 //
258 // Start de water dosering voor bepaalde tijd door openen van de afsluiter.
259 // Daarna starten de states READY en NEWCUP.
260 //
261 if (MachineState == FiniteState("WATER"))
262 {
263     Activate("ACKSTART");
264     Activate("ACKD" + String(LiquirNo));
265     Activate("&DW");
266     if (Timer(1500, timWater))
267     {
268         NewCup = false;
269         TransitionToState("READY");
270         TransitionToState("NEWCUP");
271     }
272 };
273
274 // READY
275 //
276 // Hekje open sturen. Start status CHOICE en CUPPRESENT.
277 //
278 if (MachineState == FiniteState("READY"))
279 {
```

```
280     PrepareDrink = false;
281     //UpdateServo("HATCH", 90);
282     if (Timer(150, timAanUit)) AanUit = !AanUit; else if (AanUit && !Service) Activate("READY");
283     if ((Timer(5000, timGameOver)) && NewCup)
284     {
285         GoMix = false;
286         TransitionToState("CHOICE");
287         TransitionToState("CUPPRESSENT");
288     }
289 };
290
291 // NEWCUP
292 //
293 // Controle of de beker is wegehaald van de laad/los plaats.
294 // Dit is een voorwaarde (NewCup) voor het opnieuw starten van een cyclus.
295 // Hiermee wordt voorkomen dat een gevulde beker weer de machine
296 // in gaat.. en ja daar wordt alles niet skjinner van!
297 //
298 if (MachineState == FiniteState("NEWCUP"))
299 {
300     PollUltra = true;
301     if (DistanceLonger("PRESENT", 8))
302     {
303         PollUltra = false;
304         NewCup = true;
305         TransitionToState("END");
306     }
307 };
308
309 // CUPPRESSENT
310 //
311 // In deze status wordt gecontroleerd of er een beker op de laad/los plek
312 // aanwezig is (CupPresent). Check d.m.v. een ultrasone meting.
313 //
314 if (MachineState == FiniteState("CUPPRESSENT"))
```

```
315     {
316         PollUltra = true;
317         if (Distance("PRESENT") != 0) if (DistanceBetween("PRESENT", 4, 7)) CupPresent = true; else CupPresent = false;
318         if (GoMix)
319             {
320                 PollUltra = false;
321                 TransitionToState("END");
322             }
323     };
324
325     // CIP
326     //
327     // Elke 2 minuten, wanneer er geen drankjes worden gemaakt, zal
328     // een afsluiter worden open gestuurd door CIPCYLE om de aanvoerslang
329     // te verversen.
330     //
331     if (MachineState == FiniteState("CIP"))
332     {
333         if (Timer(120000, timWaitCIP))
334         {
335             if (!PrepareDrink)
336             {
337                 TransitionToState("CIPCYLE");
338                 TransitionToState("CIP");
339             }
340             else CancelTimer(timWaitCIP);
341         }
342     };
343
344     // CIPCYLE
345     //
346     // Afsluiter voor de CIP gaat open voor 10 seconden. Daarna wordt
347     // de status gestopt. Het kan zijn dat iemand ondertussen weer
348     // een drankje wil maken; de CIP afsluiter wordt nu ook dichtgestuurd.
349     //
```

```
350     if (MachineState == FiniteState("CIPCYCLE"))
351     {
352         Activate("&CIP");
353         if (Timer(10000, timCIP)) TransitionToState("END");
354         if (PrepareDrink)
355         {
356             TransitionToState("END");
357             CancelTimer(timCIP);
358         }
359     };
360
361     /*=====
362     /*=====
363     /*=====  END USER APPLICATION =====  END USER APPLICATION =====
364     /*=====
365     /*=====
366 }
367
```