

SA Lab12-Aula2

Alberto Santos e Fábio Moraes, Turma 9 - B6

14 de Maio de 2018

1 Trabalho 2

Neste trabalho foram seguidas as regras do Grafcet hierárquico. Na página 0, correspondente ao Grafcet a implementar, as transições são calculadas conforme o valor da sua recetividade (verdadeiro ou falso). Seguidamente desliga-se as etapas a montante e liga-se a jusante. No final são calculados os outputs com base no valor lógico das etapas. Na página 1, correspondente ao Grafcet de supervisão, segue-se a mesma ordem: primeiramente calcula-se as transições e depois desliga-se etapas a montante e liga-se a jusante. As saídas são calculadas no final. Para o caso da etapa 102, é feita uma inicialização do Grafcet a implementar ou seja, este começa da etapa 0. Está incluído também um debugger que envia para a consola o panorama geral: estado das entradas, saídas, estados e transições.

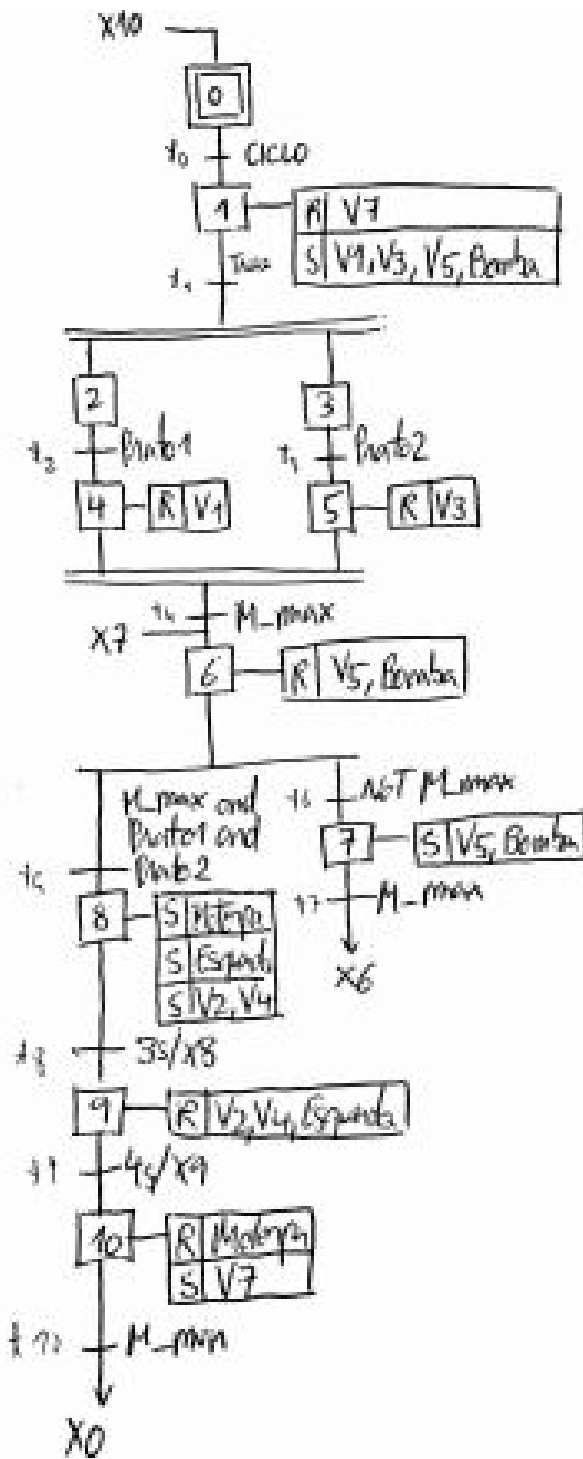


Figura 1: Grafcet a implementar

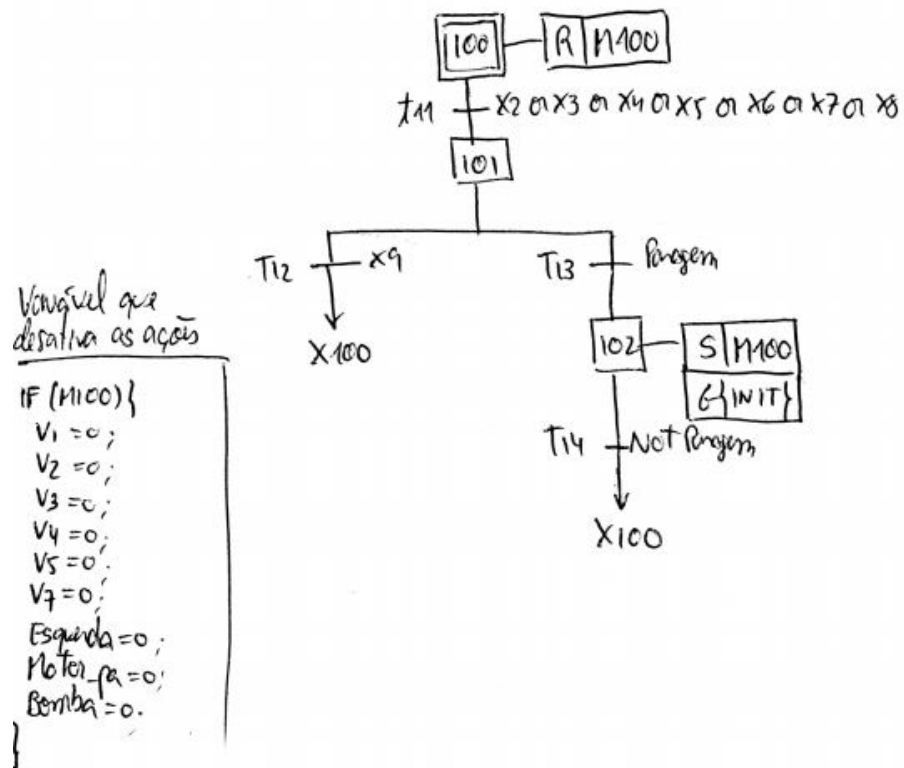


Figura 2: Grafcet de controle

```

1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <sys/types.h>
4
5 #include "timer_tools.h"
6 #include "udp_remote.h"
7 #include "s_a_hacks.h"
8
9 #include <unistd.h>
10 #include <errno.h>
11 #include <string.h>
12 #include <conio.h>
13 #include <stdbool.h>
14
15
16
17 ///////////////
18 #ifdef WIN32
19     void GOTOXY(int XPos, int YPos) { COORD Coord; Coord
        .X = XPos; Coord.Y = YPos; SetConsoleCursorPosition
        (GetStdHandle(STD_OUTPUT_HANDLE), Coord); }
20     #define CLEAR() system("cls")
21 #else
22     #define CLEAR() printf("\033[H\033[J")
23     #define GOTOXY(x,y) printf("\033[%d;%dH", (x), (y))
24 #endif
25
26
27 ///////////////
28
29
30 #define ESQUERDA out.b0
31 #define BOMBA_V5 out.b1
32 #define MOTOR_PA out.b2
33 #define V1 out.b3
34 #define V2 out.b4
35 #define V3 out.b5
36 #define V4 out.b6
37 #define V7 out.b7
38
39 #define M_max in.b0
40 #define M_min in.b1
41 #define prato1 in.b2
42 #define prato2 in.b3
43 #define paragem in.b4
44 #define ciclo in.b5

```

```

45
46
47 ms_timer timer1; // Declare a timer (milliseconds)
48 ms_timer timer2; // Declare a timer (milliseconds)
49
50 int main()
51 {
52     initialize_all();
53
54     start_timer(&timer1, 3000); // initialize timer
    for first time (milliseconds)
55     start_timer(&timer2, 4000); // initialize
    timer for first time (milliseconds)
56
57 bool x0, x1=false, x2=false, x3=false, x4=false, x5=
    false, x6=false, x7=false, x8=false, x9=false, x10=
    false, x100, x101=false, x102=false, M100=false;
58 bool t0=false, t1=false, t2=false, t3=false, t4=false,
    t5=false, t6=false, t7=false, t8=false, t9=false,
    t10=false, t11=false, t12=false, t13=false, t14=
    false;
59
60 /* ativar etapas iniciais*/
61
62 x0=true;
63 x100=true;
64
65 while (1) // infinite control cycle
66 {
67     read_all_inputs(); // global variables "in.b0" up
    to "in.b7" are updated from <<buttons>>
68     GOTOXY(1,1);
69     printf("%09.1f",cur_time/10.0);
70
71     GOTOXY(1,3);
72     printf("Entradas: M_MAX=%01d, M_MIN=%01d, PRATO1
    =%01d, PRATO2=%01d, PARAG=%01d, CICLO=%01d\n\r",
73           M_max,      M_min   ,      prato1 ,
           prato2,      paragem, ciclo);
74
75     GOTOXY(1,5);
76     printf("Saidas: ESQ=%01d B_V5=%01d MOT_PA=%01d V1
    =%01d V2=%01d V3=%01d V4=%01d V7=%01d\n\r",
77           ESQUERDA, BOMBA_V5, MOTOR_PA, V1,
           V2, V3, V4, V7);
78

```

```

79 GOTOXY(1,7);
80 printf("Estados: X0=%d x1=%d x2=%d x3=%d x4=%d x5
      =%d x6=%d x7=%d x8=%d x9=%d x10=%d timer1:%d timer2
      :%d timer1=%d\n\r",
81        x0, x1, x2, x3, x4, x5, x6, x7, x8
      , x9, x10, get_timer(&timer1), get_timer(&timer2),
      timer1);
82
83
84      t11= (x100 && (x2 || x3 || x4 || x5 || x6 || x7
      || x8)) ? 1:0;
85      t12= (x9 && x101) ? 1:0;
86      t13= (paragem && x101) ? 1:0;
87      t14= (!paragem && x102) ? 1:0;
88
89
90 /* desligar etapas a montante */
91 if(t11)
92     x100=0;
93 if(t12)
94     x101=0;
95 if(t13)
96     x101=0;
97 if(t14)
98     x102=0;
99
100 /* ligar etapas a montante */
101
102 if(t11) {
103     x101=1;
104 }
105 if(t12)
106 {
107     x100=1;
108 }
109 if(t13) {
110     x102=1;
111 }
112 if(t14) {
113     x100=1;
114 }
115 /* ligar/desligar saidas */
116 if(x100){
117     M100=0;
118 }
119 if(x102){

```

```

120     M100=1;
121     x0=1;
122 }
123
124 if(M100){ //Var. para desligar saidas
125     ESQUERDA=0;
126     BOMBA_V5=0;
127     MOTOR_PA=0;
128     V1=0;
129     V2=0;
130     V3=0;
131     V4=0;
132     V7=0;
133 }
134
135
136 /* calcular trans. disparadas*/
137 t0= (ciclo && x0) ? 1:0;
138 t1= (x1) ? 1:0;
139 t2= (prato1 && x2) ? 1:0;
140 t3= (prato2 && x3) ? 1:0;
141 t4= (M_max && x4 && x5) ? 1:0;
142 t5= (M_max && prato1 && prato2 && x6) ? 1:0;
143 t6= (!M_max && x6) ? 1:0;
144 t7= (M_max) && x7 ? 1:0;
145 t8= (get_timer(&timer1) && x8) ? 1:0;
146 t9= (get_timer(&timer2) && x9) ? 1:0;
147 t10= (M_min && x10) ? 1:0;
148
149 /* desligar etapas a montante */
150 if(t0)
151     x0=0;
152 if(t1)
153     x1=0;
154 if(t2)
155     x2=0;
156 if(t3)
157     x3=0;
158 if(t4) {
159     x4=0;
160     x5=0;
161 }
162 if(t5)
163     x6=0;
164 if(t6)
165     x6=0;

```



```

166     if (t7)
167         x7=0;
168     if (t8)
169         x8=0;
170     if (t9)
171         x9=0;
172     if (t10)
173         x10=0;
174
175     /* ligar etapas a montante */
176
177     if (t0) {
178         x1=1;
179     }
180     if (t1)
181     {
182         x2=1;
183         x3=1;
184     }
185     if (t2) {
186         x4=1;
187     }
188     if (t3)
189         x5=1;
190     if (t4) {
191         x6=1;
192     }
193     if (t5)
194     {
195         x8=1;
196         start_timer(&timer1, 3000); // initialize
timer for first time (milliseconds)
197     }
198
199     if (t6) {
200         x7=1;
201     }
202     if (t7)
203         x6=1;
204     if (t8) {
205         x9=1;
206         start_timer(&timer2, 4000); // initialize timer
for first time (milliseconds)
207     }
208     if (t9)
209         x10=1;

```

```

210     if (t10) {
211         x0=1;
212     }
213     /* ligar/desligar saidas */
214     if (x1){
215         V7=0;
216         V1=1;
217         V3=1;
218         BOMBA_V5=1;
219     }
220
221     if (x4){
222         V1=0;
223     }
224
225     if (x5){
226         V3=0;
227     }
228
229     if (x6){
230         BOMBA_V5=0;
231     }
232
233     if (x7){
234         BOMBA_V5=1;
235     }
236
237     if (x8){
238         MOTOR_PA=1;
239         ESQUERDA=1;
240         V2=1;
241         V4=1;
242     }
243
244     if (x9){
245         V2=0;
246         V4=0;
247         ESQUERDA=0;
248     }
249
250     if (x10){
251         MOTOR_PA=0;
252         V7=1;
253     }
254     fflush(stdout);
255

```

```

256     write_all_outputs(); // write global variables "
    out.b0" up to "out.b7" the <<leds>>
257
258     if (kbhit()) {
259         if (getch()==27) break;
260     }
261 }
262
263 printf("\n\r\n\r      Acabei ;) \n\r\n\r");
264     fflush(stdout);
265
266     return 0;
267 }

```