

# SA Lab12-Aula1

Alberto Santos e Fábio Moraes, Turma 9 - B6

30 de Abril de 2018

# 1 Trabalho 1

Neste trabalho foram seguidas as regras do Grafcet hierárquico. Na página 0, correspondente ao G1, as transições t0 e t1 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, G2, 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 60, é feita uma inicialização do G1 ou seja, este começa da etapa 10. Está incluído também um debugger que envia para a consola o panorama geral: estado das entradas, saídas, estados e transições.

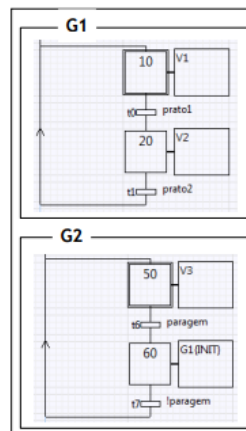


Figura 1: Grafcet a implementar

```

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) {
20     COORD Coord;
21     Coord.X = XPos;
22     Coord.Y = YPos;
23     SetConsoleCursorPosition(GetStdHandle(
24         STD_OUTPUT_HANDLE), Coord);
25 }
26 #define CLEAR() system("cls")
27 #else
28 #define CLEAR() printf("\033[H\033[J")
29 #define GOTOXY(x,y) printf("\033[%d;%dH", (x), (y))
30 #endif
31
32 //////////////////////////////////////////////////
33
34
35 #define V1          out.b3
36 #define V2          out.b4
37 #define V3          out.b5
38
39 #define prato1      in.b2
40 #define prato2      in.b3
41 #define paragem     in.b4
42
43
44
45 ms_timer timer1; // Declare a timer (milliseconds)

```

```

46
47 int main() {
48     initialize_all();
49     start_timer(&timer1, 5000); // initialize timer for
        first time (miliseconds)
50
51     bool x10, x20=false, x50, x60=false, t0=false, t1=
        false, t6=false, t7=false;
52
53     /* ativar etapas iniciais*/
54
55     x10=true;
56     x50=true;
57
58     while (1) { // infinite control cycle
59         read_all_inputs(); // global variables "in.b0" up
            to "in.b7" are updated from <<buttons>>
60
61
62
63         /* ZONA DO DEBUGGER */
64         GOTOXY(30,1);
65         printf("DEBUGGER\n\n\n");
66         GOTOXY(1,5);
67         printf("%09.1f", cur_time/10.0);
68
69         GOTOXY(1,7);
70         printf("Entradas: PRATO1=%01d, PRATO2=%01d, PARAG
            =%01d\n\r",
71             prato1 ,      prato2,      paragem);
72
73         GOTOXY(1,10);
74         printf("Saidas: V1=%01d V2=%01d V3=%01d \n\r",
75             V1,      V2,      V3);
76         GOTOXY(8,12);
77         printf("estados: x10=%d x20= %d x50= %d x60=
            %d\n\r",
78             x10,      x20,      x50,      x60);
79
80         GOTOXY(8,14);
81         printf("transicao: t0=%d t1= %d t6= %d t7= %d
            \n\r",
82             t0,      t1,      t6 ,      t7);
83         /* ***** */
84
85         /*******/

```

```

86  /**      *****PAGINA 1*****      **/
87  /*******/
88  /* calcular trans. disparadas*/
89  t6= (paragem) ? 1:0;
90  t7= (!paragem) ? 1:0;
91
92  /* desligar etapas a montante */
93  if(t6) {
94      x50=0;
95      x20=0;
96  }
97  if(t7)
98      x60=0;
99
100 /* ligar etapas a montante */
101
102 if(t6) {
103     x10=1;
104     x60=1;
105 }
106 if(t7)
107     x50=1;
108
109 /* ligar/desligar saidas */
110 if(x50)
111     V3=1;
112
113 if(x60)
114     V3=0;
115
116 /*******/
117 /**      *****PAGINA 0*****      **/
118 /*******/
119
120 /* calcular trans. disparadas*/
121 t0= (prato1) ? 1:0;
122 t1= (prato2) ? 1:0;
123
124 /* desligar etapas a montante */
125 if(t0)
126     x10=0;
127 if(t1)
128     x20=0;
129
130 /* ligar etapas a montante */
131 if(t0)

```

```

132     x20=1;
133     if(t1)
134         x10=1;
135
136     /* ligar/desligar saidas */
137     if(x10) {
138         V1=1;
139         V2=0;
140     }
141     if(x20) {
142         V2=1;
143         V1=0;
144     }
145
146     fflush(stdout);
147
148     write_all_outputs(); // write global variables "
149     out.b0" up to "out.b7" the <<leds>>
150
151     if (kbhit()) {
152         if (getch()==27) break;
153     }
154 }
155
156 printf("\n\r\n\r          Acabei ;) \n\r\n\r");
157 fflush(stdout);
158
159 return 0;
160 }

```