

# S&A - Lab12 Aula 1 - I005

Bancada 9

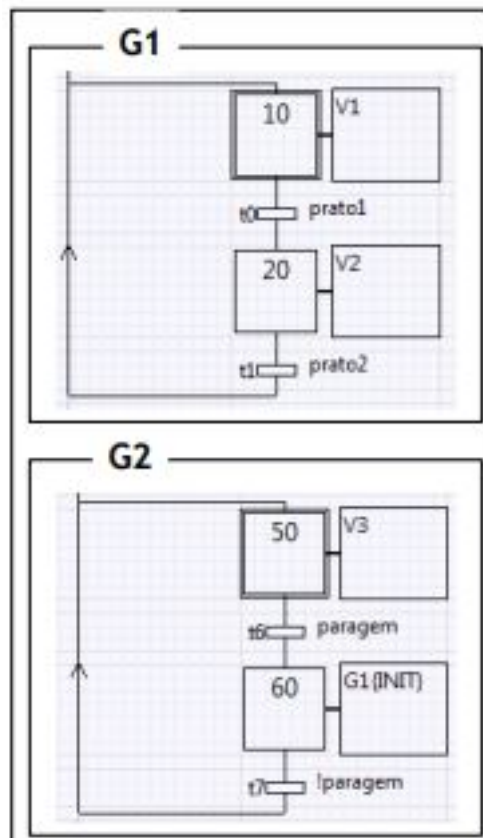
Gonçalo Santos e Rafael Moura, Turma 1

May 4, 2018

## Contents

1	Grafkets	1
2	Funcionalidades conseguidas	1
3	Código	2

## 1 Grafkets



## 2 Funcionalidades conseguidas

- Os 2 Grafkets implementados, com:
  - Etapas
  - Transições

- Saídas associadas a etapas
- Hierarquia (feita com a ordem correta dos outputs)
- Debug com impressão do estado de ambos os Grafets

### 3 Código

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>

#include "timer_tools.h"
#include "udp_remote.h"
#include "s_a_hacks.h"

#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <conio.h>

#ifdef WIN32
    void GOTOXY(int XPos, int YPos) { COORD Coord; Coord.X = XPos; Coord.Y = YPos; SetConsoleCursorPosition(
    #define CLEAR() system("cls")
#else
    #define CLEAR() printf("\033[H\033[J")
    #define GOTOXY(x,y) printf("\033[%d;%dH", (x), (y))
#endif

#define ESQUERDA out.b0
#define BOMBA_V5 out.b1
#define MOTOR_PA out.b2
#define V1 out.b3
#define V2 out.b4
#define V3 out.b5
#define V4 out.b6
#define V7 out.b7

#define M_MAX in.b0
#define M_MIN in.b1
#define PRATO1 in.b2
#define PRATO2 in.b3
#define PARAGEM in.b4
#define CICLO in.b5

#define MAX_STATES 128
#define MAX_TRANSITIONS 128

int States[MAX_STATES];
int Transitions[MAX_TRANSITIONS];
```

```

ms_timer timer1;

int main()
{
    initialize_all();
    start_timer(&timer1, 5000);

    States[10] = 1;
    States[50] = 1;

    while (1)
    {
        read_all_inputs();

        GOTOXY(1,1);
        printf("%09.1f",cur_time/10.0);

        GOTOXY(1,3);
        printf("Entradas: M_MAX=%01d, M_MIN=%01d, PRATO1=%01d, PRATO2=%01d, PARAG=%01d, CICLO=%01d\n\r",
               M_MAX,      M_MIN  ,      PRATO1  ,      PRATO2,      PARAGEM, CICLO);

        GOTOXY(1,5);
        printf("Saidas: ESQ=%01d B_V5=%01d MOT_PA=%01d V1=%01d V2=%01d V3=%01d V4=%01d V7=%01d\n\r",
               ESQUERDA, BOMBA_V5, MOTOR_PA, V1, V2, V3, V4, V7);

        // Deactivate outputs
        V7 = 0;
        V4 = 0;
        V3 = 0;
        V2 = 0;
        V1 = 0;

        // Set transitions
        if(States[10] && PRATO1) {
            Transitions[0] = 1;
        }
        if(States[20] && PRATO2) {
            Transitions[1] = 1;
        }

        if(States[50] && PARAGEM) {
            Transitions[6] = 1;
        }
        if(States[60] && !PARAGEM) {
            Transitions[7] = 1;
        }
        ///// FINAL DO CÁLCULO DAS TRANSIÇÕES /////

        // Deactivate above steps
        if(Transitions[0]) {
            States[10] = 0;
        }
        if(Transitions[1]) {

```

```

    States[20] = 0;
}

if(Transitions[6]) {
    States[50] = 0;
}
if(Transitions[7]) {
    States[60] = 0;
}

// Activate next steps
if(Transitions[0]) {
    States[20] = 1;
}
if(Transitions[1]) {
    States[10] = 1;
}

if(Transitions[6]) {
    States[60] = 1;
}
if(Transitions[7]) {
    States[50] = 1;
}

// Special output for hierarchy
if(States[60]) {
    States[10] = 1;
    States[20] = 0;
}

// Activate outputs
if(States[10]) {
    V1 = 1;
}
if(States[20]) {
    V2 = 1;
}

if(States[50]) {
    V3 = 1;
}

printf("Grafcet G1\n");
printf("State 10: %d\nState 20: %d\n\n", States[10], States[20]);

printf("Grafcet G2\n");
printf("State 50: %d\nState 60: %d\n\n", States[50], States[60]);

// Reset all transitions
{
    int i;
    for(i = 0; i < MAX_TRANSITIONS; ++i) {

```

```

        Transitions[i] = 0;
    }
}

fflush(stdout);

write_all_outputs();

if (kbhit()) {
    if (getch()==27) break;
}

}

printf("\n\r\n\r      Acabei ;) \n\r\n\r");
fflush(stdout);

return 0;
}

```