

Automatização de Processos

ANO LECTIVO: 2019/2020

Trabalho laboratorial 2

REGIME: Diurno

TURMA: PL2-Grupo1

Autores:

David Drumond, N^a2181042

Edgar Paulo, N^o2181665

Docente:

Professor Eliseu Ribeiro

28 de Maio de 2020

Conteúdo

1	Introdução	1
2	Grafcet	2
3	Inicialização	3
4	Funcionamento em regime permanente	4
5	Manutenção	5
6	Buzina	5
7	Emergência	6
8	Aplicação	7
9	Conclusão	8

Lista de Figuras

1.1	Ponte móvel	1
2.1	Tabela de variáveis	2
3.1	Inicialização do programa	3
4.1	Main Logical POU	4
6.1	Funcion Block Clock	5
7.1	Emergência	6
8.1	Plataforma EasySim 1	7

1 Introdução

Tendo por objetivo o desenvolvimento de competências no âmbito da automatização de processos, foi proposto a elaboração de um sistema de comando de uma ponte móvel.

A estruturação gráfica das ações e recetividades foi efetuada através método Grafcet, e a sua posterior execução na aplicação PC WORX, através de linguagem de blocos e de texto estruturado.

Durante a implementação das especificações solicitadas no caderno de encargos, foram adicionadas funcionalidades suplementares como, por exemplo, o botão Start e o botão de Emergência.



Figura 1.1: Ponte móvel

2 Grafcet

Nos Grafcet's nível 1 e 2 encontra-se esquematizado o fluxo de ações e transições necessárias a implementação do sistema de controlo da ponte. Os grafkets de nível 1 e 2 encontram-se em anexo.

Na sua conceção procurou-se minimizar o número de etapas, garantindo, contudo, a separação e coordenação dos elementos esquerdo e direito da ponte, uma vez que estes utilizam sensores próprios para detetar a sua localização.

Por forma a minimizar o desgaste, a descida da ponte é feita de forma desfasada, ou seja, primeiro desce um lado, e 3 segundos depois desce o outro.

Na tabela 2.1 é possível verificar as variáveis utilizadas.

Grafcet		PC WORX
Nível 1	Nível 2	Tag
Mover	Mover	Mover
Ponte esquerda em baixo	BaixoEsq	BaixoEsq
Ponte direita em baixo	BaixoDir	BaixoDir
Ponte esquerda em cima	CimaEsq	CimaEsq
Ponte direita em cima	CimaDir	CimaDir
Botão Start	Start	Start
Botão Emergencia	*****	Emerg
Desce ponte esquerda	DesceEsq	DesceEsq
Desce ponte direita	DesceDir	DesceDir
Sobe ponte esquerda	Sobe Esq	Sobe Esq
Sobe ponte direita	Sobe Dir	Sobe Dir
Conta manobras	manobra	manobras_Fb
Buzina	Buzina	Buzina
Sinal Verde	Verde	Verde
Sinal Vermelho	Vermelho	Vermelho
Sinal Amarelo	Amarelo	Amarelo

Figura 2.1: Tabela de variáveis

3 Inicialização

Quando o sistema inicia é efetuado um reset às variáveis de saída, ficando em modo standby até que o utilizador pressione o botão Start. Uma vez que na ativação deste botão não é possível prever qual a posição prévia das pontes, pelo que é ativado por defeito o semáforo vermelho e a buzina, que toca de forma intermitente com frequência de 0,5 Hz.

De seguida, caso os sensores BaixoEsq ou BaixoDir não se encontrem ativos, as pontes descem com um desfasamento de três segundos uma da outra, mantendo o sinal luminoso e sonoro. Posteriormente é contabilizada mais uma manobra.

Quando a ponte estiver em baixo, o sinal verde é ativado, e a buzina e o sinal vermelho desligados. Para a inicialização recorreu-se a lógica condicional para colocar o programa na etapa E0, e blocos SR (Set Dominant) nas restantes etapas.

A figura 3.1 ilustra o processo de inicialização.

```
(*Network etapas*)

(*Inicializacao da etapa 0*)
if (E16 and T5) or (E7 and T6) then
E0:=1;
end_if;

(*Desativa a etapa 0*)
if E1 then
E0:=0;
end_if;

SR_1 (SET1:= E0 and start ,RESET:=(E2 and E4) or E8);
E1:=SR_1.Q1;
```

Figura 3.1: Inicialização do programa

4 Funcionamento em regime permanente

As movimentações da ponte dependem sempre da ação do utilizador sobre o botão mover.

A subida do mecanismo é precedida por uma temporização que implica a mudança do semáforo para amarelo durante 5 segundos e posteriormente para vermelho durante 10 segundos. Durante o processo de abertura e enquanto a ponte permanece aberta, a iluminação vermelha deve estar ativa e a buzina ligada à frequência de 0.5Hz.

Após cada manobra o sistema valida a necessidade de manutenção da ponte¹.

A ativação e desativação das etapas é efetuada com recurso a blocos SR², cumprido o fluxo do Grafcet. Adicionalmente encontram-se implementados blocos TON³ e CTU⁴ para controlo das temporizações e contagem de manobras.

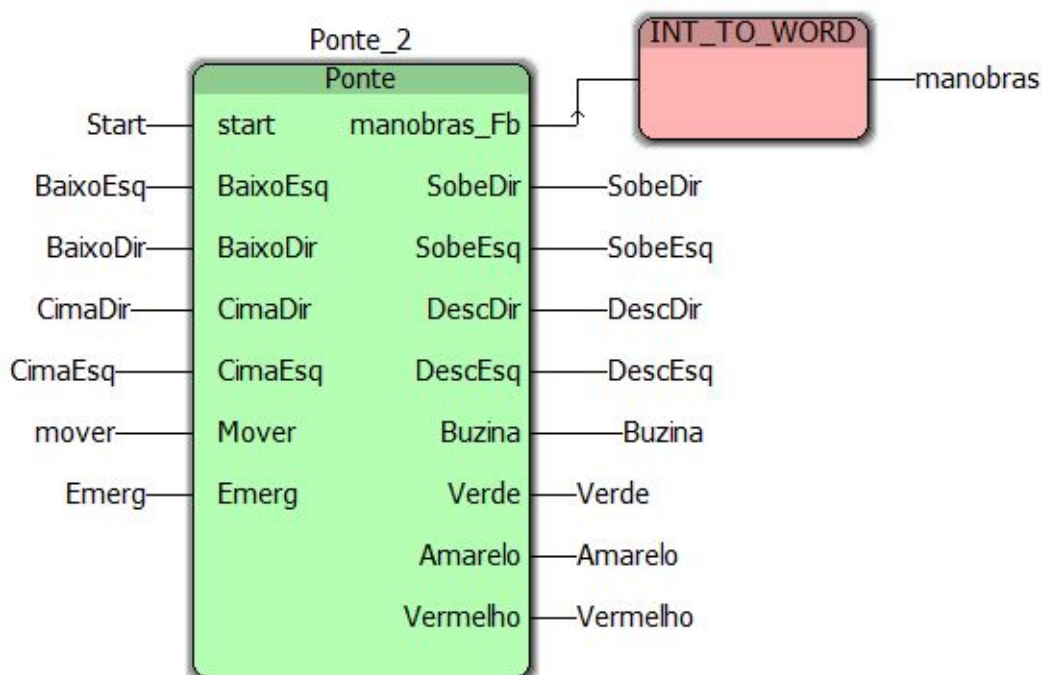


Figura 4.1: Main Logical POU

¹para mais detalhes consultar a pagina 5

²Função Set/Reset, em caso de estarem ativos o Set e o Reset, o Set é Dominante

³Temporizador ao tralho (Timer On-Delay)

⁴Contador incremental (Counter Up)

5 Manutenção

Os ciclos de subida ou descida das pontes são contabilizados recorrendo às movimentações da ponte esquerda, evitando assim a duplicação de contadores.

Após 5000 ciclos deve ser efetuada a manutenção pelo que a buzina atua durante 20 segundos. Por segurança do sistema, e para garantir a manutenção, o próprio entra em paragem, migrando para a etapa 0.

6 Buzina

A intermitência da buzina é garantida através de um temporizador que se encontra em loop de 2 segundos, ou seja, ao atingir o tempo definido efetua reset a si próprio, reiniciando o ciclo. A gestão deste temporizador é efetuada através de uma função específica de nome Clock, que utiliza lógica combinacional permitindo a comutação da saída. A desativação do bloco implica que a saída seja colocada a "0". A figura ilustra a implementação desta função.

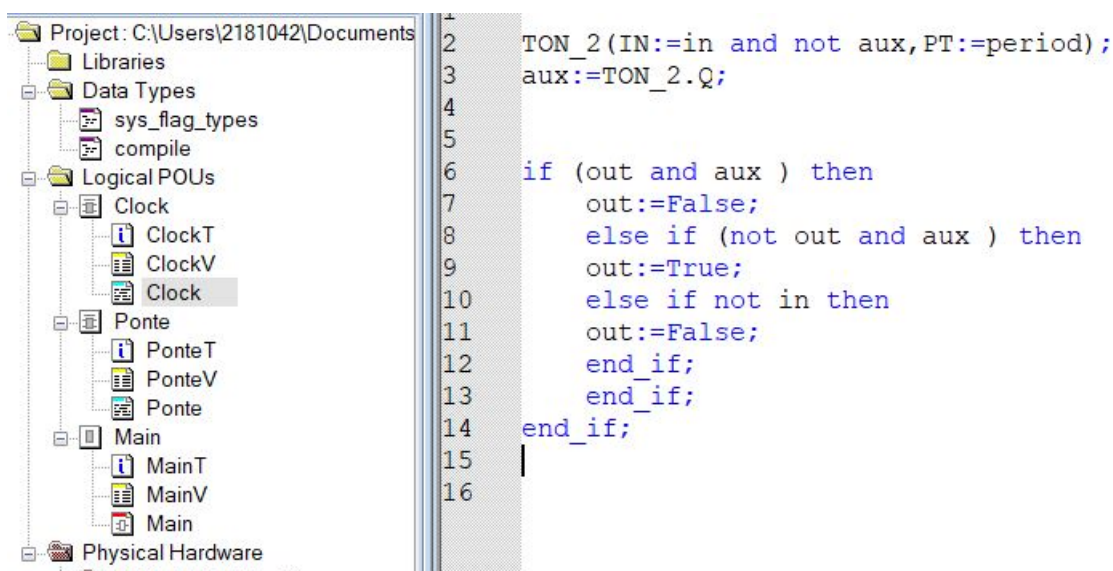


Figura 6.1: Funccion Block Clock

7 Emergência

A emergência está implementada utilizando lógica inversa, ou seja, quando este sinal se encontra a "0" significa que a mesma está ativada.

Quando é detetado este estado, o sistema mantém-se na etapa atual, acionando o sinal vermelho e a buzina de forma intermitente com uma frequência de 0.5Hz. Nesta situação todas as saídas são desativadas, pelo que funcionamento normal retoma-se após a desativação do sinal de emergência.

A figura 7.1 ilustra a implementação do estado de emergência.

```

102 (*Verifica situacao de emergencia(Logica inversa)*)
103 if Emerg then
104     SR_14 (SET1:=E1 or E10, RESET:=E8 or E0);
105     Vermelho:=SR_14.Q1;
106
107     Amarelo:=E9;
108     Verde:=E8;
109     (*Afetação da buzina*)
110     if E7 or E16 then
111         Buzina:=true;
112     else
113         SR_15 (SET1:=E1 or E11, RESET:=E8 or E0);
114         (*Buzina_aux:=SR_15.Q1;*)
115         Clock_3 (in:=SR_15.Q1, period:=T#1s);
116         Buzina:=Clock_3.out;
117     end_if;
118     DescEsq:=E2;
119     DescDir:=E5;
120     SobeEsq:=E12;
121     SobeDir:=E14;
122     (*Conta o numero de operações realizadas*)
123     CTU_1 (CU:=E3 or E13 , RESET:= E7 or E16 , PV:=5000);
124     manutencao:=CTU_1.Q;
125     manobras_Fb:=CTU_1.CV;
126     else
127     (*caso a emergencia seja ativada desliga
128     todas as saidas coloca o botao vermelho e Buzina a frequencia de 0.5Hz*)
129     Clock_4 (in:=not Emerg, period:=T#1s);
130     blink:=Clock_4.out;
131     Vermelho:=blink;
132     Buzina:=blink;
133
134     DescEsq:=False;
135     DescDir:=False;
136     SobeEsq:=False;
137     SobeDir:=False;
138     Amarelo:=False;
139     Verde:=False;
140
141 end_if;
142

```

Figura 7.1: Emergência

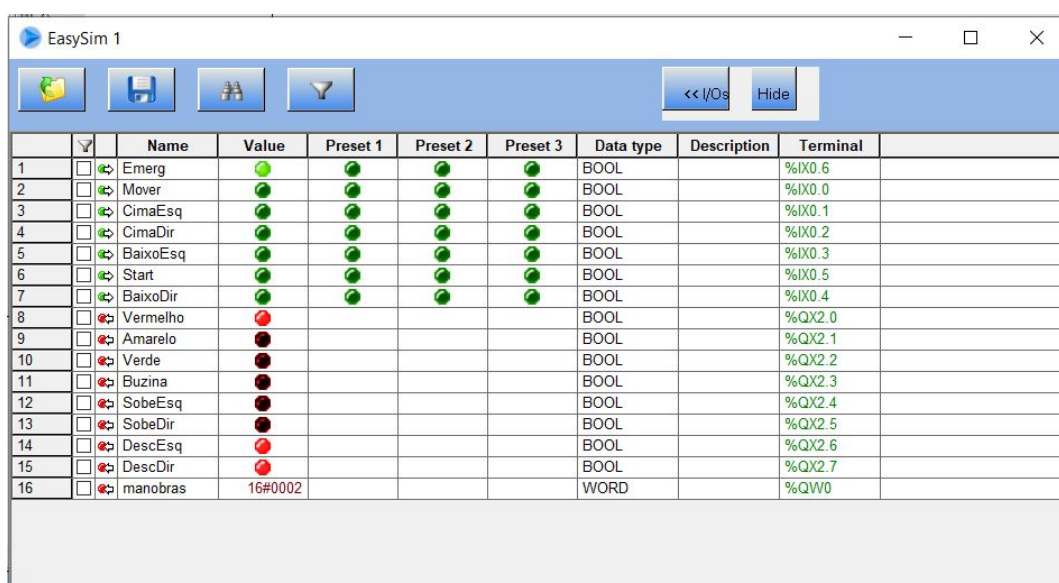
8 Aplicação

O desenvolvimento do Grafcet foi efetuado na aplicação CADe SIMU⁵, e posteriormente em ambiente gráfico.

O projeto foi implementado na aplicação PC WORX 6.30.2902 da Phoenix Contact utilizando linguagem de blocos e de texto estruturado. Esta aplicação permite manter o nome das variáveis utilizadas no Grafcet, otimizando a implementação do sistema.

Para simular o projeto recorreu-se à aplicação EasySim 1, que permite manipular as variáveis de entrada, podendo assim visualizar o comportamento do sistema implementado.

A figura 8.1 ilustra o ambiente de simulação.



The screenshot shows the EasySim 1 application window. It features a toolbar with icons for file operations and a table of variables. The table has columns for Name, Value, Preset 1, Preset 2, Preset 3, Data type, Description, and Terminal. The variables are listed in rows 1 through 16. The 'Value' column shows green circles for boolean variables and red circles for word variables. The 'Terminal' column shows hexadecimal addresses for boolean variables and a word address for the last variable.

	Name	Value	Preset 1	Preset 2	Preset 3	Data type	Description	Terminal
1	Emerg	●	●	●	●	BOOL		%IX0.6
2	Mover	●	●	●	●	BOOL		%IX0.0
3	CimaEsq	●	●	●	●	BOOL		%IX0.1
4	CimaDir	●	●	●	●	BOOL		%IX0.2
5	BaixoEsq	●	●	●	●	BOOL		%IX0.3
6	Start	●	●	●	●	BOOL		%IX0.5
7	BaixoDir	●	●	●	●	BOOL		%IX0.4
8	Vermelho	●				BOOL		%QX2.0
9	Amarelo	●				BOOL		%QX2.1
10	Verde	●				BOOL		%QX2.2
11	Buzina	●				BOOL		%QX2.3
12	SobeEsq	●				BOOL		%QX2.4
13	SobeDir	●				BOOL		%QX2.5
14	DescEsq	●				BOOL		%QX2.6
15	DescDir	●				BOOL		%QX2.7
16	manobras	16#0002				WORD		%QW0

Figura 8.1: Plataforma EasySim 1

⁵Consular em www.cadesimu.net

9 Conclusão

O desenvolvimento deste projeto possibilitou a introdução a aplicação PC WORX da Phoenix Contact, e a utilização da linguagem de texto estruturado no âmbito da programação de autómatos.

A programação em texto estruturado apresenta semelhanças com outras linguagens de programação, e é desta forma mais simples, nomeadamente pela possibilidade de uso de lógica combinacional. Por outro lado, a interação com a linguagem de blocos facilita a implementação de comandos.

A aplicação da Phoenix Contact, pelas suas características ao nível da gestão de variáveis, apresentou-se intuitiva, com a possibilidade de utilizar os nomes presentes no Grafcet nível 2.

A adição de funcionalidades suplementares permitiu explorar a organização das variáveis globais e a sua interligação às cartas digitais do autómato.

Os testes efetuados ao sistema de controlo desenvolvido através do EasySim e do PC WORX, revelam que o mesmo tem um comportamento estável e cumpre com o caderno de encargos solicitado, considerando-se cumpridos os objetivos propostos.

Anexos

-
1. Gracet de Nível 1
 2. Gracet de Nível 2
-

Graficet de nivel 1

