



**FEUP** Universidade do Porto  
Faculdade de Engenharia

Programação 1

## Simulador ambiente fábrica

Guião

Versão 0.1

Valter Joaquim Ramos Costa (ee09115@fe.up.pt)

Revisão:

Luís Teixeira (luisft@fe.up.pt)

12 de Novembro de 2014

# CONTEÚDO

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Simulador</b>	<b>4</b>
2.1	Elementos da fábrica . . . . .	4
2.1.1	Tapetes Lineares . . . . .	4
2.1.2	Tapetes Rotativos . . . . .	4
2.1.3	Mesa de trabalho . . . . .	5
2.1.4	Máquina Ferramenta . . . . .	5
2.1.5	Pusher . . . . .	6
2.1.6	Comandos dos equipamentos . . . . .	6
<b>3</b>	<b>Biblioteca de funções</b>	<b>7</b>
3.1	Introdução . . . . .	7
3.2	Descrição das funções . . . . .	7
<b>4</b>	<b>Configuração da biblioteca</b>	<b>11</b>
<b>5</b>	<b>Trabalho prático</b>	<b>12</b>
5.1	Introdução . . . . .	12
5.2	Instruções de utilização . . . . .	12
5.3	Exercício Prático . . . . .	13
5.3.1	Exercício 1 . . . . .	13
5.3.2	Exercício 2 . . . . .	13
5.3.3	Exercício 3 . . . . .	13
	<b>Referências</b>	<b>15</b>

# 1 INTRODUÇÃO

Este guião apresenta um trabalho produzido para Programação 1 no sentido de apresentar um exemplo de aplicação num ambiente real.

O presente trabalho pode ser dividido em duas partes: simulador Java e uma biblioteca escrita em C para a comunicação entre o simulador e o código produzido pelo estudante.

O simulador consiste num programa em java, o qual pode ser executado em qualquer plataforma que disponha de uma máquina virtual Java (ex. Windows, Linux, OSX). Cada equipamento (tapete, máquina ferramenta, pusher, etc) simulado por este simulador será controlado através de entradas e saídas na sua maioria binárias. O protocolo de comunicação usado para a ligação é o Modbus.

A biblioteca de funções consiste num conjunto de rotinas preparadas para a comunicação e controlo dos elementos constituintes da fábrica.

## 2 SIMULADOR

### 2.1 Elementos da fábrica

Neste simulador, cada equipamento (tapete, máquina) aparece sob a forma de pontos coloridos no canto superior esquerdo da sua representação. Os pontos com fundo cinzento indicam o estado dos comandos que o equipamento está a receber, e os pontos com fundo preto o estado dos respectivos sensores. O vermelho indica desativado (FALSE), e o verde ativo (TRUE).

#### 2.1.1 Tapetes Lineares

Os tapetes lineares com um sensor no meio têm três quadrados (Figura 2.1) que indicam respectivamente:

- quadrado da esquerda (fundo cinzento):  
ordem de deslocamento para esquerda/cima;
- quadrado do meio (fundo preto):  
sensor de presença de peça;
- quadrado da direita (fundo cinzento):  
ordem de deslocamento para direita/baixo;



Figura 2.1: Imagem tapete linear no simulador

#### 2.1.2 Tapetes Rotativos

Os tapetes rotativos, para além de uma linha superior com o mesmo significado dos tapetes lineares, dispõem de uma segunda linha com o seguinte significado (Figura 2.2):

- quadrado da esquerda (fundo cinzento):  
ordem de rotação no sentido direto;
- quadrado da esquerda (fundo preto):  
tapete na posição extrema de rotação no sentido direto;

- quadrado da direita (fundo preto):  
tapete na posição extrema de rotação no sentido inverso;
- quadrado da direita (fundo cinzento):  
ordem de rotação no sentido inverso;



Figura 2.2: Imagem tapete rotativo no simulador

### 2.1.3 Mesa de trabalho

A mesa de trabalho dispõe de um único sensor (quadrado de fundo preto) que representa o sensor de presença de peça (Figura 2.3).

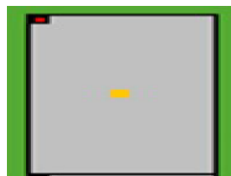


Figura 2.3: Imagem da mesa de trabalho no simulador

### 2.1.4 Máquina Ferramenta

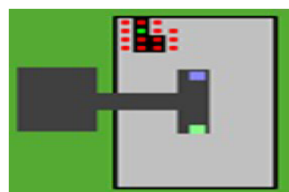


Figura 2.4: Imagem da máquina ferramenta no simulador

Os tapetes associados às máquinas ferramenta, para além de uma linha superior com o mesmo significado dos tapetes lineares, dispõem de uma segunda linha com o estado da máquina associada (Figura 2.4). Nesta segunda linha:

- quadrado da esquerda (fundo cinzento):  
ordem de acionamento da ferramenta da máquina;
- quadrado do meio (fundo preto):  
ferramenta alinhada e pronta para ser acionada;

- quadrado do meio (fundo cinzento):  
ordem de troca de ferramenta;
- quadrado da direita (fundo cinzento):  
ordem de troca de ferramenta (sentido inverso);

As duas linhas seguintes dizem respeito à deslocação da máquina nos eixos YY e ZZ respectivamente (no simulador a deslocação no eixo dos ZZ é representada por uma deslocação no eixo XX). Cada uma das linhas dispõe de:

- quadrado da esquerda (fundo cinzento):  
ordem de translação para a esquerda/cima;
- quadrado do meio (fundo preto):  
tapete na posição extrema de translação para a esquerda/cima;
- quadrado do meio (fundo preto):  
tapete na posição extrema de translação para a direita/baixo;
- quadrado da direita (fundo cinzento):  
ordem de translação para a direita/baixo.

### 2.1.5 Pusher



Figura 2.5: Imagem do pusher no simulador

Os tapetes associados aos pushers, para além de uma linha superior com o mesmo significado dos tapetes lineares, dispõem de uma segunda linha com o estado do pusher associado (Figura 2.5). Nesta segunda linha:

- quadrado da esquerda (fundo cinzento):  
ordem de recolher;
- quadrado do meio (fundo preto):  
pusher na posição extrema de recolhido;
- quadrado do meio (fundo preto):  
pusher na posição extrema de esticado;
- quadrado da direita (fundo cinzento):  
ordem de empurrar.

### 2.1.6 Comandos dos equipamentos

Se clicar com o botão direito do rato em cima de um dos recursos do simulador (por exemplo, num tapete linear), poderá alterar o estado dos actuadores desse recurso, bem como inserir ou remover uma peça. Repare no entanto, que a mudança de estado do atuador do recurso é feita de forma imediata, pelo que permanecerá no novo estado até haver novo comando.

## 3 BIBLIOTECA DE FUNÇÕES

### 3.1 Introdução

A biblioteca de funções desenvolvida em linguagem C consiste num conjunto de rotinas que implementam funcionalidades básicas do simulador.

### 3.2 Descrição das funções

**int moveTapete(int tapete, int direcao);**

Descrição: Permite pôr em andamento ou parar um dado tapete.

Entradas:

- tapete -> inteiro que representa o tapete a colocar em andamento/parar (ver tabela 3.1).
- direcao -> inteiro que representa qual a direção que o tapete deve tomar (ver tabela 3.2).

Saídas:

- Esta função retorna 1 se sucesso e 0 se o tapete ou a direção passados forem errados.

Tapete	Tipo
AT1	Linear
SAT1	Linear
SAT2	Rotativo
SAT3	Linear
SAT4	Linear
SAT5	Linear
SAT6	Rotativo
CT1	Linear

Tabela 3.1: Designação dos tapetes no simulador

Direção	Explicação
POS	Coloca o tapete a andar/rodar no sentido positivo
NEG	Coloca o tapete a andar/rodar no sentido negativo
STOP	Pára o movimento do tapete

Tabela 3.2: Direção dos tapetes no simulador

**int rodaTapete(int tapete, int direcao);**

Descrição: Permite pôr a rodar ou parar a rotação de um dado tapete.

Entradas:

- tapete -> inteiro que representa o tapete a colocar a rodar/parar (ver tabela 3.1).
- direcao -> inteiro que representa qual a direção que o tapete deve tomar (ver tabela 3.2).

Saídas:

- Esta função retorna 1 se sucesso e 0 se o tapete ou a direção passados forem errados.

**int lerSensorTapete(int tapete, int parametro);**

Descrição: Permite ler o estado dos sensores dos tapetes (lineares ou rotativos)

Entradas:

- tapete -> inteiro que representa o tapete para o qual se pretende saber o estado do sensor (ver tabela 3.1).
- parametro -> inteiro que representa que sensor deve ser lido(ver tabela 3.3).

Saídas:

- Esta função retorna o estado do sensor lido (1 se o sensor estiver ativo, 0 se estiver desativo).

Parametro	Explicação
POS	Lê o sensor no tapete rotativo no sentido positivo
NEG	Lê o sensor no tapete rotativo no sentido negativo
PSENSOR	Lê o sensor de presença de peça

Tabela 3.3: Parametro dos tapetes no simulador

**int tapeteLivre(int tapete);**

Descrição: Testa o estado de um tapete para verificar se este está livre para receber/enviar uma peça.

Exemplo: Se uma transferência de uma peça entre 2 tapetes estiver em andamento o estado desses dois tapetes é ocupado e portanto esta função retorna 0.

Entradas:

- tapete -> inteiro que representa o tapete para a qual se pretende saber se está livre ou não (ver tabela 3.1).

Saídas:

- Esta função retorna 0 se o tapete estiver ocupado e 1 se estiver livre.

**void inicializaMaquina(int maquina);**

Descrição: Permite inicializar a máquina para a qual se pretende atuar.

Nota: Esta função deve ser chamada pelo menos uma vez antes de ligar uma dada máquina.

Entradas:

- maquina -> inteiro que representa a máquina que se pretende inicializar (ver tabela 3.4).



Saídas:

- Não se aplica.

Maquina	Tipo
MAQ1	Máquina 1
MAQ2	Máquina 2

Tabela 3.4: Designação das máquinas no simulador

**void ligaMaquina(int maquina);**

Descrição: Permite ligar a máquina para trabalhar a peça.

Entradas:

- maquina -> inteiro que representa a máquina que se pretende ligar (ver tabela 3.4).

Saídas:

- Não se aplica.

**void desligaMaquina(int maquina);**

Descrição: Permite desligar a máquina.

Entradas:

- maquina -> inteiro que representa a máquina que se pretende ligar (ver tabela 3.4).

Saídas:

- Não se aplica.

**void empurraPeca(int pusher,int direcao);**

Descrição: Permite empurrar ou recolher o "braço"do pusher.

Entradas:

- pusher -> inteiro que representa o pusher para o qual se pretende atuar (ver tabela 3.5).
- direcao -> inteiro que representa o sentido do movimento do "braço"(ver tabela 3.6)

Saídas:

- Não se aplica.

Pusher	Tipo
PUSHER1	Pusher 1

Tabela 3.5: Designação do pusher no simulador

Direção	Explicação
POS	Coloca o tapete a andar/rodar no sentido positivo
NEG	Coloca o tapete a andar/rodar no sentido negativo
STOP	Pára o movimento do tapete

Tabela 3.6: Direção do pusher no simulador

**void lerSensorPusher(int pusher,int parametro);**

Descrição: Permite ler o estado dos sensores do pusher.

Entradas:

- pusher -> inteiro que representa o pusher para o qual se pretende saber o estado (ver tabela 3.5).
- parametro -> inteiro que representa que sensor deve ser lido do pusher (ver tabela 3.7)

Saídas:

- Esta função retorna o estado do sensor lido do pusher(1 se pusher na posição extrema, 0 se pusher numa posição intermédia).

Parametro	Explicação
POS	Lê o sensor no pusher no sentido positivo
NEG	Lê o sensor no pusher no sentido negativo
PSENSOR	Lê o sensor de presença de peça (na mesa de trabalho)

Tabela 3.7: Parametro do pusher no simulador

**void temporizador(int segundos);**

Descrição: Implementa um temporizador de espera.

Entradas:

- segundos -> inteiro que representa o tempo(em segundos) a esperar.

Saídas:

- Não se aplica.

## 4 CONFIGURAÇÃO DA BIBLIOTECA

Para poder utilizar este simulador é necessária a instalação da biblioteca funções de mod-bus para estabelecer comunicação. Esta biblioteca deve ser instalada seguindo os próximos passos:

1. Abra um terminal e navegue até ao diretório onde se encontra a pasta sfs-Prog1 e entre na mesma via terminal.
2. Execute o seguinte comando para instalar a biblioteca: `./install.sh`  
Este comando só necessita de ser executado uma vez.
3. No mesmo diretório execute: `java -jar sfs.jar`  
Este comando coloca em funcionamento o simulador da fábrica.
4. Repita o procedimento do ponto 1.
5. No terminal aberto em 4, execute o comando: `./sfs.sh`  
Este comando compila e executa o código. Se a ligação foi efetuada deverá aparecer a seguinte mensagem no terminal:  
Conexão estabelecida com sucesso no ip 127.0.0.1 porta 5502.

Nota importante: É importante que siga estas instruções pela ordem que está apresentada. O simulador gráfico deve ser sempre iniciado em primeiro lugar e, só depois executar o comando do ponto 5. Para parar qualquer um destes dois processos basta selecionar o respetivo terminal e efetuar Ctrl+C. Sempre que alterar código e quiser colocar novamente em execução deverá repetir o ponto 5.

## 5 TRABALHO PRÁTICO

### 5.1 Introdução

O objetivo deste trabalho é a automatização de uma linha de produção de uma fábrica. Este é um caso de aplicação que é normalmente resolvido com recurso a autómatos. Contudo, neste contexto o propósito é a aplicação de máquinas de estados para o controlo desta fábrica. A figura 5.1 mostra a disposição dos elementos da fábrica bem como o nome dos mesmos.

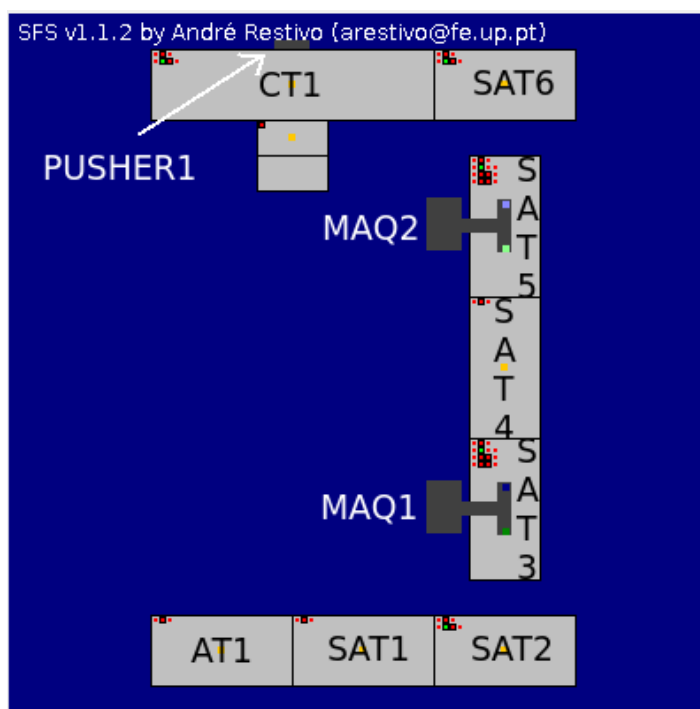


Figura 5.1: Disposição dos elementos da fábrica no simulador

### 5.2 Instruções de utilização

As peças são colocadas no simulador manualmente sobre um tapete clicando com a tecla direita do rato e selecionando Blocks->Add P1. Deverão ser **sempre** colocadas no tapete de entrada AT1 (Figura 5.1) e a peça seguinte **só deverá ser colocada quando a primeira chegar ao tapete SAT1**. De forma análoga, para retirar uma peça basta clicar com a tecla direita por cima da peça e selecionar Blocks->Remove.

## 5.3 Exercício Prático

### 5.3.1 Exercício 1

Este exercício tem como objetivo fazer uma caixa avançar do tapete AT1 para o tapete SAT1. Comece por copiar o código exemplo e colar na função `maquinaEstado()` disponível no ficheiro `main.c` na pasta `sfs-Prog1`.

Código exemplo:

```
switch(state)
{
    case 1:
        if (tapeteLivre(SAT1) && !lerSensorTapete(SAT1,PSENSOR))
        {
            moveTapete(AT1,NEG);
            moveTapete(SAT1,NEG);
            state=2;
        }
        break;
    case 2:
        if (lerSensorTapete(SAT1,PSENSOR))
        {
            moveTapete(AT1,STOP);
            moveTapete(SAT1,STOP);
            state=3;
        }
        break;
}
```

Em seguida, garanta que o simulador da fábrica está a correr (ponto 3 da configuração da biblioteca).

O próximo passo será compilar e executar o código gerdo (ponto 5 da configuração da biblioteca).

Por fim, coloque uma peça do tipo P1 no tapete AT1. Neste momento, a peça deverá deslocar-se do tapete AT1 para o tapete SAT1 e parar. Quando a peça chegar ao SAT1 remova-a.

### 5.3.2 Exercício 2

O objetivo deste exercício é colocar uma peça do tipo P1 no tapete AT1 e transportá-la pelo SAT1->SAT2->SAT3. Quando chegar a SAT3 pára onde será trabalhada pela máquina 1 para uma peça do tipo P2(ver tabela 5.1). Nota 1: o processo de transformação P1->P2 é correto se a peça ficar de cor vermelha. Nota 2: o tapete SAT2 é um tapete rotativo.

Matéria Prima	Peça Final	Tipo de Máquina	Tempo de Processamento
P1	P2	MAQ1	15 s
P2	P5	MAQ2	10 s

Tabela 5.1: Designação dos tapetes no simulador

### 5.3.3 Exercício 3

Neste exercício o objetivo é colocar uma peça do tipo P1 (castanha) no tapete AT1 e fazê-la avançar até à máquina 1 onde será trabalhada para uma peça do tipo P2 (vermelha).

Após o processo de transformação a peça deverá ser transportada até à máquina 2 onde será trabalhada para uma peça do tipo P5 (verde). Finalmente, deverá ser transportada até ao tapete CT1 onde deverá ser empurrada pelo pusher 1.

## BIBLIOGRAFIA

- [Automação, 2011] Automação (2011). Linha de produção flexível versão 1.6. URL: [https://sigarra.up.pt/feup/pt/conteudos\\_geral.ver?pct\\_pag\\_id=249640&pct\\_parametros=pv\\_ocorrencia\\_id=283643&pct\\_grupo=31120#31120](https://sigarra.up.pt/feup/pt/conteudos_geral.ver?pct_pag_id=249640&pct_parametros=pv_ocorrencia_id=283643&pct_grupo=31120#31120).
- [Restivo, 2014] Restivo, A. (2014). Shop floor simulator. URL: <https://github.com/arestivo/sfs>.