



**INSTITUTO SUPERIOR DE ENGENHARIA
DE LISBOA**

Área Departamental de Engenharia Mecânica



Projeto de Programação de Autómatos Enchimento e Embalamento de Garrafas

Relatório dos trabalhos da unidade curricular de Programação de Autómatos

Grupo 2:

Paula Costa N.º 45685

Tiago Rodrigues N.º41575

Valter Francisco N.º39383

Orientador: Mário J. G. C. Mendes

Junho de 2019

1. Índice

2. Introdução.....	3
3. Descrição do Problema.....	4
3.1 Definições Técnicas.....	5
4. Descrição do Equipamento.....	6
4.1 Definição das entradas e saídas.....	7
5. Descrição do Programa	8
5.1 Configuração do Autômato	8
5.2 Programação do autômato	13
5.2.1 OB1 (LAD).....	13
5.2.2 FC1 – Refill (LAD)	17
6.1 FC3 – Valve_Control (SCL)	18
7.1 FB1 – Auto_Mode (GRAPH).....	19
6. Descrição do HMI	24
7. Conclusões e Trabalho Futuro.....	29
8. Bibliografia.....	30
9. Anexos	31
9.1 Anexo 1 – Main [OB1].....	31
9.2. Anexo 2 – Auto_mode [FB1].....	33
9.3. Anexo 4 – Tag Tables	34

Figura 1 - Ilustração do sistema (adaptado)	4
Figura 2 - PLC no Tia Portal	8
Figura 3 - OB1 - Network 1: Auto_Mode	14
Figura 4 - OB1- Network 2: Level_Conversion	15
Figura 5 - Network 3: Counter_Full_Bottles	16
Figura 6 - Network 4: Counter_Bottles_inside_system	16
Figura 7 - Network 1: tank_refill	17
Figura 8 - FC3 - Valve_Control	18
Figura 9 - Step 1 e Transition 1	19
Figura 10 - Step 2 e Transition 2	20
Figura 11 - Step 3 e Transition 3	20
Figura 12 - Step 4 e Transition 4	21
Figura 13 - Step 5 e Transition 5	21
Figura 14 - Step 6 e Transitions 6 e 7	22
Figura 15 - Step 7 e Transition 8	23
Figura 16 - System Information screen	24
Figura 17 - System diagnostics screen	25
Figura 18 - System settings screen	25
Figura 19 - User administration screen	26
Figura 20 – Root screen	27

2. Introdução

O trabalho proposto apresenta uma estação de enchimento de garrafas, que terá dois modos de funcionamento, um modo automático para funcionamento autónomo do processo e um modo manual utilizado para efetuar diversos testes e/ou calibrações.

As garrafas são transportadas por um tapete e o enchimento é feito por um funil acoplado a um tanque.

No modo automático o movimento deste tapete será controlado pelos sensores dispostos ao longo da linha de enchimento.

O nível do contentor será controlado através de um sensor de nível que ativará ou desativará uma bomba de carregamento quando este atingir os limites desejados.

O sensor presente no final do tapete fará a contagem de garrafas cheias e a cada 12 garrafas ativará os cilindros empacotadores.

No modo manual o controlo dos vários pontos e a verificação dos vários sensores será feita através de um HMI.

De forma a aplicar os conhecimentos adquiridos na aula, usaram-se três linguagens de programação: *Ladder Diagram* (LAD), *Sequential Function Charts* (SFC, Grafcet ou GRAPH) e *Structured Control Language* (SCL ou texto estruturado). O código criado consiste num programa estruturado com diferentes soluções de forma a permitir explorar tantas ferramentas quanto possível.

Também foi possível a aplicação de boas práticas de programação como manter as variáveis organizadas e uso de *tags* de fácil identificação.

3. Descrição do Problema

O problema proposto em aula consiste numa estação de enchimento de garrafas automatizada, composta por um tapete rolante que inclui três sensores que detetam a entrada e a saída das garrafas, tal como a posição de enchimento. A Figura 1 ilustra o posicionamento de cada componente do sistema.

Quando uma garrafa chega à posição de enchimento o tapete para e um funil de enchimento despeja o conteúdo na garrafa num período pré-estabelecido de 5.5 segundos que findos cessa automaticamente o enchimento. Ao atingir as 12 garrafas cheias o sistema deve proceder ao embalamento por via do acionamento automático de 2 cilindros.

A alimentação do funil de enchimento é garantida por um tanque de 4000 litros mantido em controlo de nível, para tal, uma bomba é acionada aos 2% e desativada aos 98%.

A movimentação do tapete é garantida por um motor cujo sentido de rotação é controlado.

Existe um conjunto de botoeiras, que permitem ligar e parar o sistema, colocar em modo automático ou manual e deslocar o tapete para a frente ou para trás.

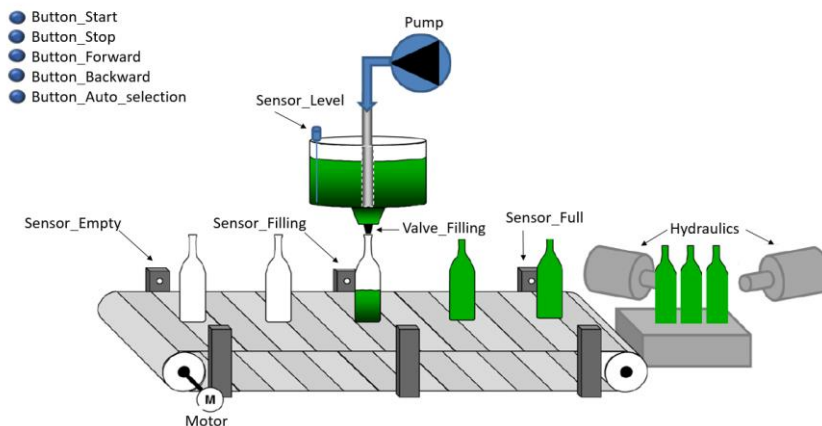


Figura 1 - Ilustração do sistema (adaptado)

3.1 Definições Técnicas

De forma a iniciar o projeto foi necessário ter em consideração algumas questões técnicas para a correta escolha e configuração do autómato correto para o trabalho pretendido.

O botão *Start* consiste num atuador de contacto momentâneo, normalmente aberto, o que significa que estará normalmente sem sinal e ao ser pressionado emite um pulso voltando à posição zero quando solto.

O botão *Stop* é normalmente fechado, isto é, existe sempre uma corrente a passar no mesmo e quando é pressionado, corta o sinal. Em termos de programação a variável “botão pressionado” é representado pelo booleano 1, então, quando o botão é pressionado o circuito elétrico do mesmo é interrompido e interpretado no PLC como a passar da variável de 0 para 1.

O sensor de nível do tanque emite um sinal analógico, e a válvula do funil de enchimento recebe um sinal também analógico. No caso a transmissão entre o autómato e o instrumento será elétrica, por exemplo, 4 mA equivale a zero no CPU e 20 mA equivale a 27648. O inverso é válido para as saídas.

A bomba de carregamento que faz o enchimento do tanque recebe um sinal binário, que com auxílio de um relé procede à sua (des)ativação.

Os sensores dispostos ao longo do tapete emitem sinais digitais sempre que uma garrafa os cruza.

O movimento deste tapete é controlado por um motor que pode girar nos dois sentidos dependendo da saída, também digital, que no motor acionará o relé correspondente ao sentido de rotação pretendido.

A última parte do processo é constituída por dois cilindros de embalamento que recebem sinais digitais. Neste caso consideramos que o sinal aciona solenoides que por sua vez induzem a atuação de pistões hidráulicos.

4. Descrição do Equipamento

O PLC escolhido foi um CPU 1511-1 PN, a razão para esta escolha foi para garantir que existia memória suficiente para qualquer extensão que o projeto tomasse, bem como o acesso a todas as linguagens de programação para maior flexibilidade. Outro ponto importante era a utilização de contadores, funcionalidade disponível no PLC escolhido. Considerando que era um projeto em que o *budget* não era um ponto a tomar em consideração, escolher o modelo topo de gama trouxe diversas garantias e facilidades, como a utilização de um simulador mais robusto, que facilitou a verificação de erros no código.

As cartas utilizadas foram DI 16/DQ 16x24VDC/0.5A BA_1, utilizada para as entradas a partir de 0.0 até 1.7 bits, DI 16x24VDC BA_1, utilizada para os *inputs* dos sensores, desde 16.0 a 17.7 bits, por último uma carta AI/AQ 4xU/I/RTD/TC/2xU/I ST_1, para os *inputs* e *outputs* analógicos em *words*, nomeadamente o sensor de nível do tanque e a válvula de enchimento, respetivamente.

O HMI utilizado foi TP1500 Basic color PN, com as seguintes especificações:

- 15.0" TFT display;
- 1024 x 768 pixéis;
- 256 cores;
- Touch screen;
- 1 x PROFINET.

Esta escolha foi baseada maioritariamente em dois pontos, o primeiro foi a compatibilidade com o autómato escolhido, tendo em conta que a presença de uma ligação PROFINET facilita a conexão entre os dois. O segundo ponto foi o tamanho e qualidade do ecrã de forma a facilitar a **visualização**.

Commented [PC1]: Se calhar metia as características q o TIA da, profinet e blablá

Commented [TR2R1]: Podes escrever se quiseres 😊
@Paula Costa

Commented [PC3R1]: ja tá parece-me

4.1 Definição das entradas e saídas

Em seguida será apresentada a tabela relativa aos sinais utilizados no processo, incluindo uma breve descrição, a sua *tag*, e o seu endereço no autômato.

Tabela 1 - Lista de sinais utilizados no processo

Descrição	Tag	Endereço no Autômato
Botão Start	Button_Start	I 0.0
Botão Stop	Button_Stop	I 0.1
Troca entre modo manual e automático	Button_Auto_selection	I 0.4
Faz o motor do tapete andar para a frente	Button_Forward	I 0.2
Faz o motor do tapete andar para trás	Button_Backward	I 0.3
Sensor de entrada das garrafas	Sensor_Empty	I 16.5
Sensor para enchimento das garrafas	Sensor_Filling	I 16.6
Sensor de saída das garrafas	Sensor_Full	I 16.7
Sensor de nível do tanque	Sensor_Level	IW 120
Bomba de enchimento do tanque	Pump	Q 9.0
Cilindro A de embalamento	Hydraulic_A	Q 9.1
Cilindro B de embalamento	Hydraulic_B	Q 9.2
Motor do tapete anda para a frente	Motor_Forward	Q 9.3
Motor do tapete anda para trás	Motor_Backward	Q 9.4
Funil de enchimento de garrafas	Valve_Filling	QW 164

5. Descrição do Programa

5.1 Configuração do Autômato

Para configuração do autômato referido anteriormente utilizamos o software TIA Portal (Totally Integrated Automation Portal) da Siemens na versão V14.

Após adicionado o CPU e as cartas referidas surge a sua representação, como na Figura 2.

O *slot* zero fica vazio reservado à aplicação de uma fonte de alimentação, que no caso não foi necessário utilizar por se tratar de exemplo académico e o simulador não requerer a unidade.

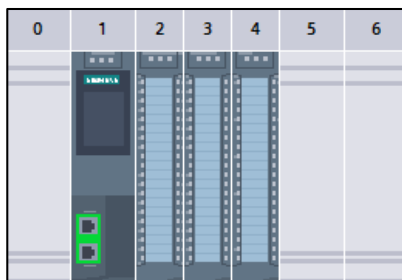


Figura 2 - PLC no Tia Portal

O *slots* 1 contem o módulo do CPU que contem as comunicações, por o TIA Portal, fazer automaticamente a configuração básica do autômato não foram necessárias mais alterações.

O *slots* 2, 3 e 4 correspondem às cartas de entradas e saídas de sinais, nestas cartas e através do *device configuration* alteramos os endereços de entrada para 0, 16 e 120 para cada uma respetivamente. Nas saídas alterámos para 9 no *slot* 2 e 164 na carta analógica.

O slot 4, por conter a carta analógica, tem de existir a conversão dos sinais, então nas propriedades da carta, no separador input/output definimos que o sinal seria do tipo de variação de corrente entre 4 e 20 mA, tanto para os inputs como para os outputs.

Dentro do PLC, na opção *show all tags* definimos as *tags* referidas na Tabela 1, e as memórias referidas na Tabela 2. Desta forma ficaram definidas as variáveis simbólicas para facilitar a construção do código.

A tabela seguinte inclui uma breve descrição, *tag*, tipo de informação e endereço no autómato das memórias utilizadas.

Tabela 2 - Lista de Memórias utilizadas

Descrição	Tag	Tipo	Endereço no Autómato
Memória utilizada para o flanco positivo do botão <i>Start</i>	Memory_SR_Start	Booleano	M 0.0
Memória utilizada no <i>Set/Reset</i> da bomba de enchimento do tanque	Memory_SR_refill	Booleano	M 0.1
Variável do grafcet para contagem de garrafas	Bottles	Inteiro	MW 2
Memória do total de garrafas cheias	Memory_Full_Bottles	Inteiro	MW 6
Memória do número de garrafas no sistema	Memory_bottles_system	Inteiro	MW 8
Memória do nível no tanque de enchimento	Memory_level_word	Inteiro	MW 10
Memória do nível do tanque convertido em percentagem	Memory_level_percent	Inteiro	MW 12
Memória do número de garrafas para empacotamento	Memory_bottle_to_pack	Inteiro	MW 14

Tendo em conta que é necessário efetuar diversas ações, então também foi sentida a necessidade de criar blocos de funções com diferentes finalidades.

Na tabela abaixo estão apresentados os blocos, incluindo breves descrições de cada um, a sua *tag* e endereço, bem como a sua base de dados, caso exista, e a *tag* desta.

Tabela 3 - Lista de Blocos de Funções

Descrição	Bloco	Tag bloco	Base Dados	Tag Base Dados
Bloco de início do programa. Faz <i>call</i> às funções necessárias.	OB1	Main	Memória de Carregamento do Autómato	
Bloco do programa para funcionamento automático	FB1	Auto_Mode	DB1	Auto_Mode_DB
Bloco controla o nível do tanque	FC1	Refill	(não utiliza)	
Bloco controla da válvula de enchimento de garrafas	FC3	Valve_control	(não utiliza)	

Foram, também, utilizados dois contadores de garrafas, um para contar as garrafas no tapete e outro para contar as garrafas cheias.

Estes contadores têm duas finalidades, a primeira passa por apresentar a informação de quantas garrafas existem no sistema de enchimento, isto é entre o sensor de entrada no tapete (Sensor_Empty) e o sensor de saída do tapete (Sensor_Full).

O segundo contador faculta a informação de quantas garrafas foram cheias desde que o sistema está em funcionamento.

Na tabela abaixo será apresentada a sua descrição, tag, base de dados utilizada, tag da base de dados, memória utilizada e tag da memória.

Tabela 4 - Tabela de Contadores

Descrição	Base de Dados	Tag Base de Dados	Tag da Memória	Memória utilizada
Contador de garrafas no sistema	DB3	IEC_Counter_Bottles_system	Memory_bottles_system	MW 8
Contador de garrafas cheias	DB2	IEC_Counter_full	Memory_Full_Bottles	MW 6

Em seguida será apresentada uma lista de constantes utilizadas no processo, incluindo a sua descrição, *tag*, tipo e valor utilizado.

Tabela 5 - Constantes utilizadas

Descrição	Tag	Tipo	Valor
Nível tanque alto	Tank_level_high	Inteiro	27095
Nível tanque baixo	Tank_level_low	Inteiro	553
Número de garrafas para ativar os cilindros embaladores	Bottles_per_package	Inteiro	12
Abertura máxima da válvula de enchimento	Valve_open	Word	27648
Válvula de enchimento fechada	Valve_closed	Word	0
Fator multiplicativo para converter o nível do tanque para percentagem	Correction_to_percent	LReal	0.00361689814814815

Os valores utilizados nos níveis do tanque são obtidos a partir das restrições do sistema (níveis baixo = 2% e nível alto = 98%) e das características dos autómatos, no caso os equipamentos da Siemens utilizam uma gama de valores entre -27648 e +27648. Pela configuração que usamos os 27648 correspondem aos 100 % do nível e os 0 % correspondem a um valor de zero. Não sendo usado o domínio negativo. Desta forma, $27648 \times 0.98 = 27095$. Para o nível baixo o método é idêntico, mas com 0.02.

O *Correction_to_percent* consiste num fator a usar na conversão do nível do tanque de word para percentagem e este valor é obtido a partir do arredondamento de $\frac{100}{27648}$, com o maior número de casas decimais permitidas pelo sistema.

5.1 Programação do autómato

5.2.2 OB1 (LAD)

Neste ponto do trabalho iremos apresentar o código utilizado para controlar e comandar o processo. Em seguida será explicada a estrutura dos blocos para controlo do processo, começando pelo bloco OB1.

Antes de iniciar o código colocamos todas as constantes (referidas na Tabela 5) nas constantes do OB1, de forma mantê-las todas juntas, facilitando a alteração em caso de necessidade.

O OB1 contém o código base para o funcionamento, chamando as funções necessárias quando as suas condições forem satisfeitas. Este apresenta diversas funcionalidades, começando pela *Network 1*, a que damos o nome de *Auto_Mode* está apresentada na Figura 3, da página seguinte.

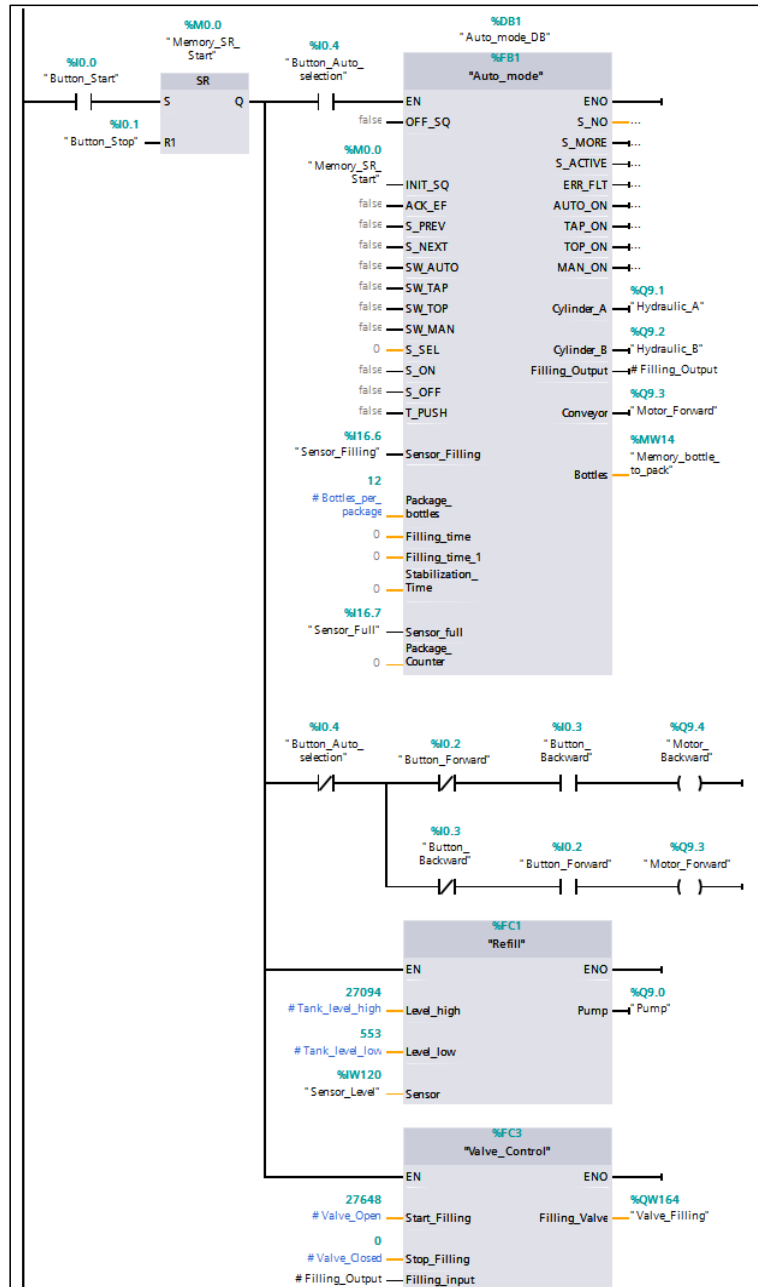


Figura 3 - OB1 - Network 1: Auto_Mode

Esta primeira linha de código começa por um *set-reset* que liga o programa quando o “Button_Start” é premido e só desliga o programa quando o “Button_Stop” é premido.

Em seguida ramifica-se em quatro, a primeira ramificação, quando o modo automático está seleccionado, o botão “Button_Auto_selection” está activo, faz call à função FB1 que controla todo o sequenciamento do processo de enchimento autónomo.

O segundo ramo funciona como modo manual de controlo do motor do tapete, quando o “Button_Auto_selection” e “Button_Forward” não estão activos, e “Button_Backward” está premido aciona o “Motor_Backward” que faz o tapete andar para trás, o oposto aciona o “Motor_Forward” fazendo o tapete andar para a frente.

A partir do momento que o “Button_Start” liga o programa é feito o *call* de duas funções, a função FC1 que é dedicada a controlar o nível presente no tanque e a função FC3 que controla a abertura da válvula de enchimento. Estas funções serão explicadas à frente.

A *Network 2*, a que chamámos “Level_Conversion”, vai fazer a conversão do nível do tanque, guardando o “Sensor_Level” numa memória através de um bloco “MOVE” para converter para percentagem utilizando um bloco de multiplicação com o fator de conversão referido na tabela de constantes. Esta *Network* tem como fim representar o nível em percentagem num HMI.

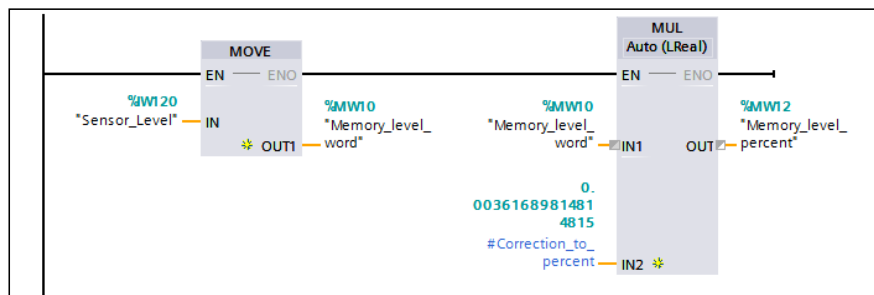


Figura 4 - OB1- *Network 2*: Level_Conversion

A *Network 3*: Counter_Full_Bottles é constituída por um bloco “CTU”, que incrementa uma unidade sempre que o uma garrafa activa o “Sensor_Full”, o seu reset é feito quando o processo é parado e o número é gravado numa memória de forma a ser apresentado no HMI.

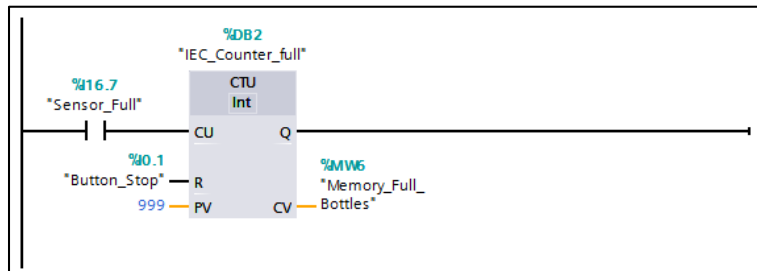


Figura 5 - *Network 3*: Counter_Full_Bottles

A *Network 4*: Counter_Bottles_inside_system, é constituída por um bloco “CTUD”, um contador que pode incluir incrementos e decrementos para fazer contagem do número de garrafas no sistema, o seu reset é feito quando o processo é parado, através do "Button-Stop".

A variável incrementa quando uma garrafa passa no sensor de garrafa vazia e decrementa quando passa no sensor de garrafa cheia. O valor final é guardado numa memória, este serve para determinar a diferença entre o número de garrafas que entrou no sistema *versus* o número de garrafas que saiu do sistema, servindo para determinar se o sistema está a funcionar como pretendido.

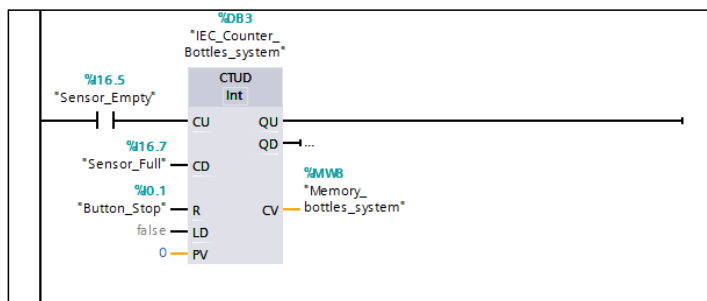


Figura 6 - *Network 4*: Counter_Bottles_inside_system

6.2.2 FC1 – Refill (LAD)

Começando por explicar os blocos mais simples, abaixo está apresentada a *Network 1: tank_refill*, que pertence ao bloco de programação FC1 denominado “Refill”.

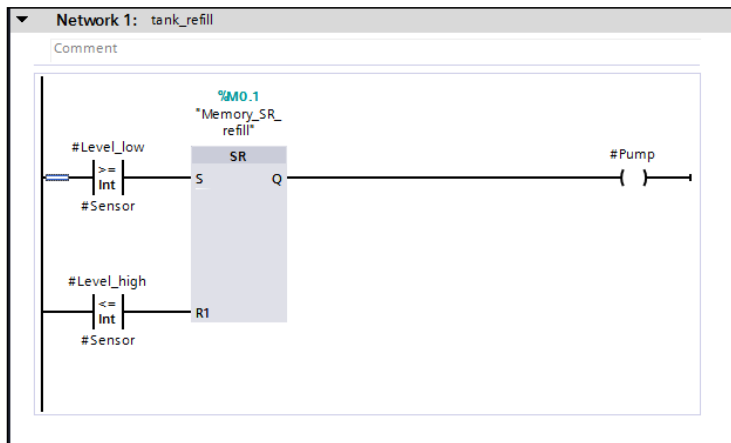


Figura 7 - *Network 1: tank_refill*

Como referido anteriormente, este bloco comanda o controlo da bomba de enchimento do tanque, utilizando um bloco de *set-reset* em que liga ou desliga a bomba segundo o valor que o sensor de nível está a medir, se este for maior ou igual à constante “Level_low” a bomba é acionada, se for maior ou igual à constante “Level_high” esta é desligada, constantes estas apresentadas na tabela de constantes.

7.2.2 FC3 – Valve_Control (SCL)

Para o controlo de válvula de enchimento a linguagem utilizada foi texto estruturado, no bloco de função FC3 a que foi dado o nome de “Valve_Control”, apresentado abaixo.

```
1 IF #Filling_input = 1 THEN
2
3     #Filling_Valve := #Start_Filling;
4
5 ELSE
6     #Filling_Valve := #Stop_Filling;
7
8 END_IF;
```

Figura 8 - FC3 - Valve_Control

Esta função recebe três inputs, *Filling_Output* (enviado da FB1), *Start_Filling* e *Stop_Filling* que são constantes definidas no OB1. Existe apenas um output, designado de *Filling_Valve*, e consiste numa *word* que segue direta para a válvula de controlo do enchimento.

Quando uma garrafa chega ao sensor de enchimento a FB1 envia sinal para abrir a válvula num sinal booleano, a FC1 ao receber o sinal verifica se o seu valor é 1, sendo verdade o output é definido com o valor que abre a válvula (*Start_Filling*), no caso de a condição anterior não se verificar então a válvula é fechada (*Stop_Filling*).

8.2.2 FB1 – Auto_Mode (GRAPH)

O bloco de função FB1 contém a função de controlo do processo quando este se encontra em modo automático, programado em *grafcet*, a importância de referir isto deve-se à linguagem ser preferencial para programar uma sequência de eventos/processos, dividindo-se em dois aspetos importantes: *steps* e *transitions*. Nos *steps* definem-se as ações que vão ser executadas e as *transitions* garantem que só é executado o *step* seguinte se um determinado requisito acontecer.

No *Step 1* foi introduzida uma variável utilizada para contagem de garrafas, chamada "Bottles" e, igualada a 0 pois é o início do processo, é feito, também, um *reset* ao contador de tempo de estabilização (TR na coluna de *Qualifier*) aquando a ativação deste *step*, isto é imposto pela condição S1 presente na coluna Event. O que garante a transição para o *step* seguinte (*Step 2*) é a variável "Bottles" ser efetivamente 0.

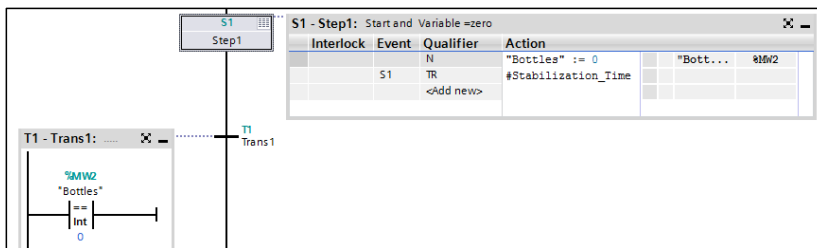


Figura 9 - Step 1 e Transition 1

O *Step 2*, que faz *reset* aos cilindros de embalagem e à válvula de enchimento e ativa o movimento do tapete para a frente. A *Transition 2* garante que só se passa para o *Step 3* quando o sensor de enchimento for ativado pela chegada de uma garrafa.

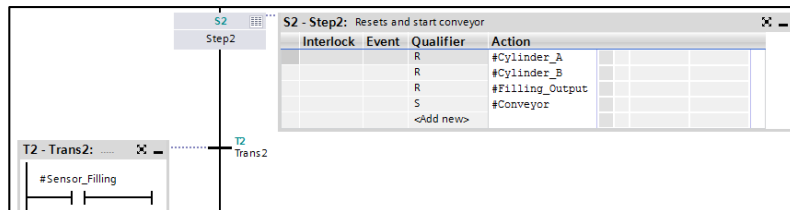


Figura 10 - Step 2 e Transition 2

No *Step 3* o movimento do tapete é cessado, a válvula de enchimento é aberta e inicia-se a contagem de 5,5 segundos, que é o tempo predefinido para encher uma garrafa, o processo só passa ao *step* seguinte quando este tempo é dado por terminado, garantido pela *Transition 3*.

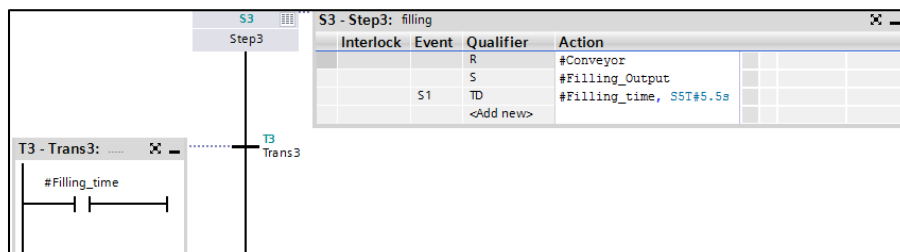


Figura 11 - Step 3 e Transition 3

No *step* seguinte (*Step 4*) a válvula de enchimento é fechada, é feito o *reset* ao contador do tempo de enchimento e inicia-se um contador de estabilização, para que o tapete não inicie o movimento antes que todo o líquido seja despejado devidamente, evitando derrames.

Semelhante ao que se passou anteriormente, a Transition 4 só se torna verdadeira quando o este tempo de estabilização, 1 segundo, termina.

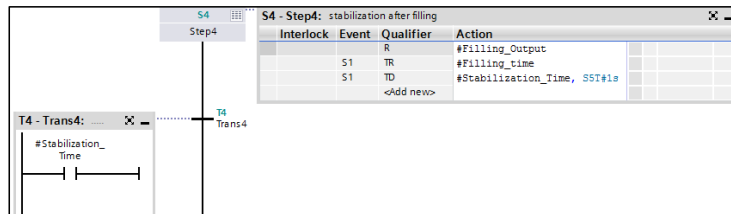


Figura 12 - Step 4 e Transition 4

Estando esta parte concluída, o tapete volta a ser ativado e é feito o *reset* ao contador do tempo de estabilização, tudo no Step 5. Para passar ao step seguinte, o sensor de garrafa cheia tem de ser ativado, na Transition 5.

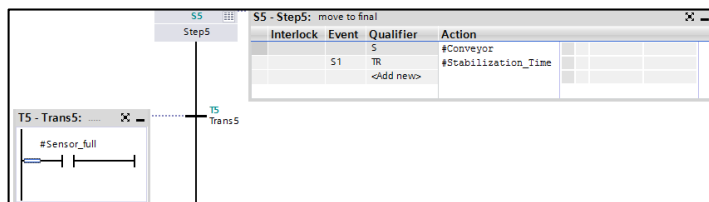


Figura 13 - Step 5 e Transition 5

No *Step 6* é incrementa-se uma unidade à variável “Bottles” e, dependendo do número desta contagem, são apresentadas duas hipóteses de transição:

T6 - A variável apresenta um número inferior à constante “Package_Bottles”, que corresponde ao número de garrafas necessário para fazer um *pack* (*Transition 6*) e o programa segue para o *Step 2*, apresentado anteriormente. Voltando a fazer outro ciclo do processo.

T7 - A variável é igual ao número necessário para fazer um pack, 12 garrafas, (*Transition 7*) e seguirá para o *Step 7*.

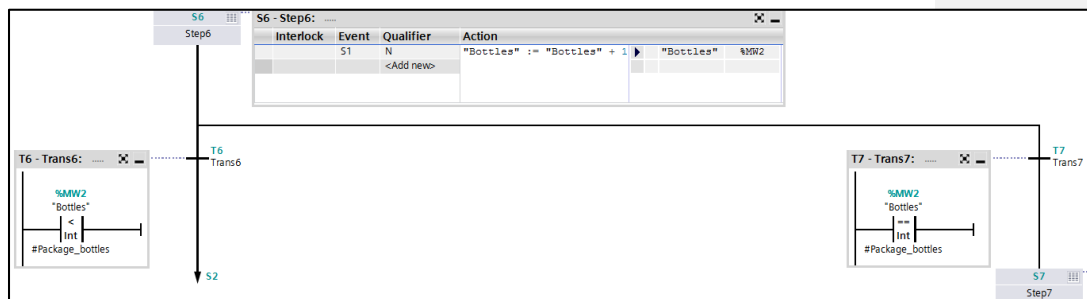


Figura 14 - Step 6 e Transitions 6 e 7

Após a *Transition 7* ambos os cilindros de empacotamento são ativados e é iniciada a contagem de um tempo de estabilização de 2 segundos, para que um *pack* não seja enviado para o processo seguinte sem estar devidamente acabado (*Step 7*). Quando o *timer* concluir a contagem (*Transition 8*), volta ao *Step 1* e reinicia todo o processo de contagem, enchimento e embalamento de garrafas.

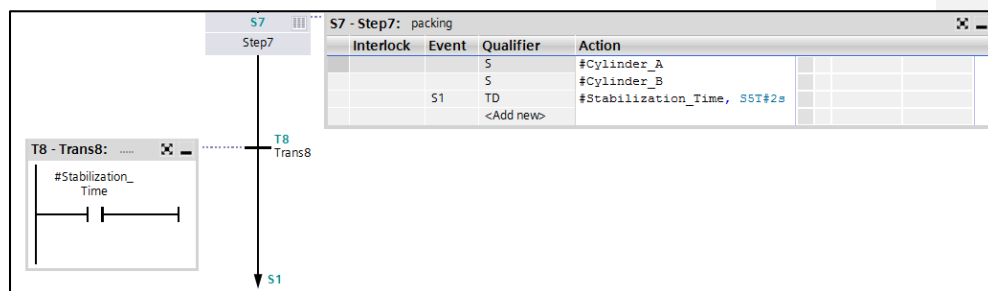


Figura 15 - Step 7 e Transition 8

6. Descrição do HMI

Quanto ao HMI, os diversos *screens* serão apresentados nas imagens abaixo e após a sua apresentação será feita uma breve descrição sobre o que foi desenvolvido.

Na Figura 16 encontra-se o *screen* de informações do sistema com os componentes e autores.

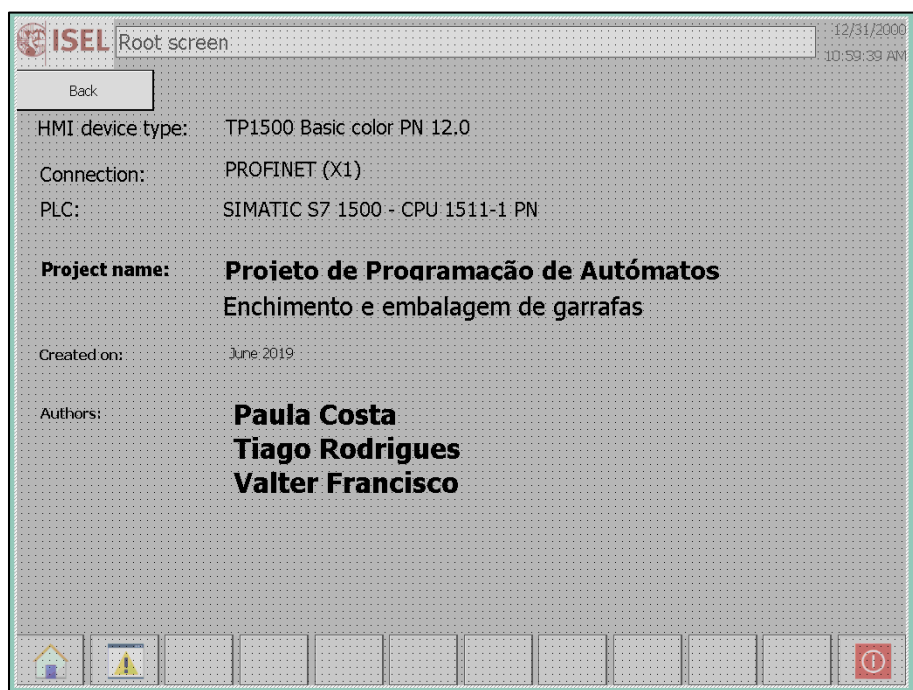


Figura 16 - System Information screen

No ecrã de diagnostico podemos ver como será a disposição das mensagens de erro do sistema, apresentando data e hora, bem como uma pequena mensagem de texto com o erro ocorrido.

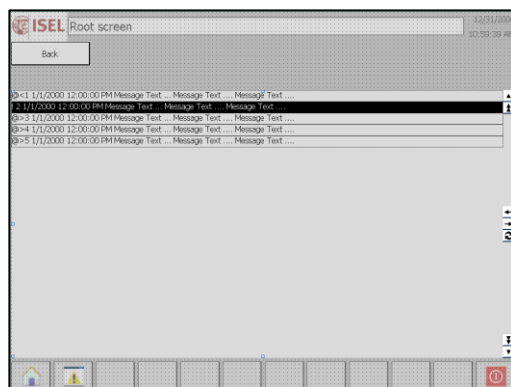


Figura 17 - System diagnostics screen

O ecrã de *settings* é um *template* que vem por defeito com intuito de calibrar o *touch* do ecrã, ou de fazer limpeza.

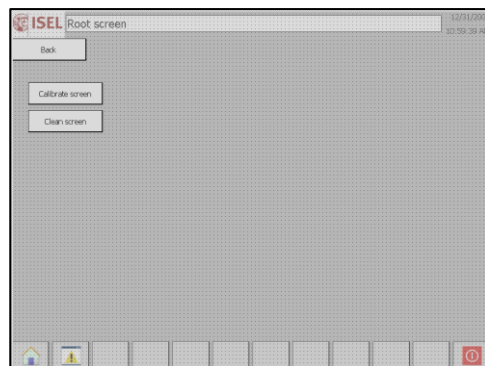


Figura 18 - System settings screen

Neste ecrã está apresentado como seria o *display* da lista de utilizadores.

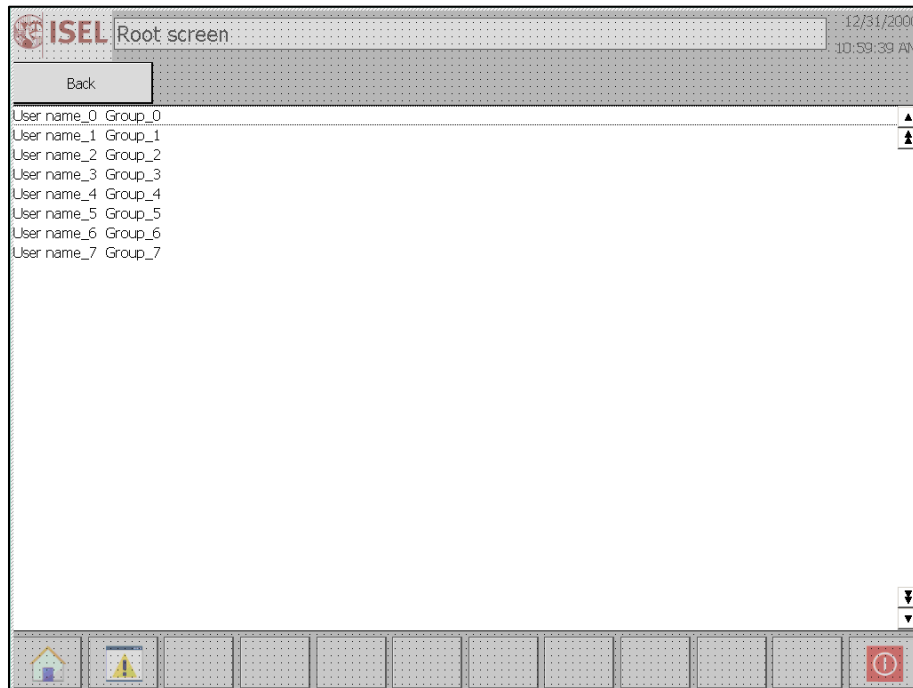


Figura 19 - User administration screen

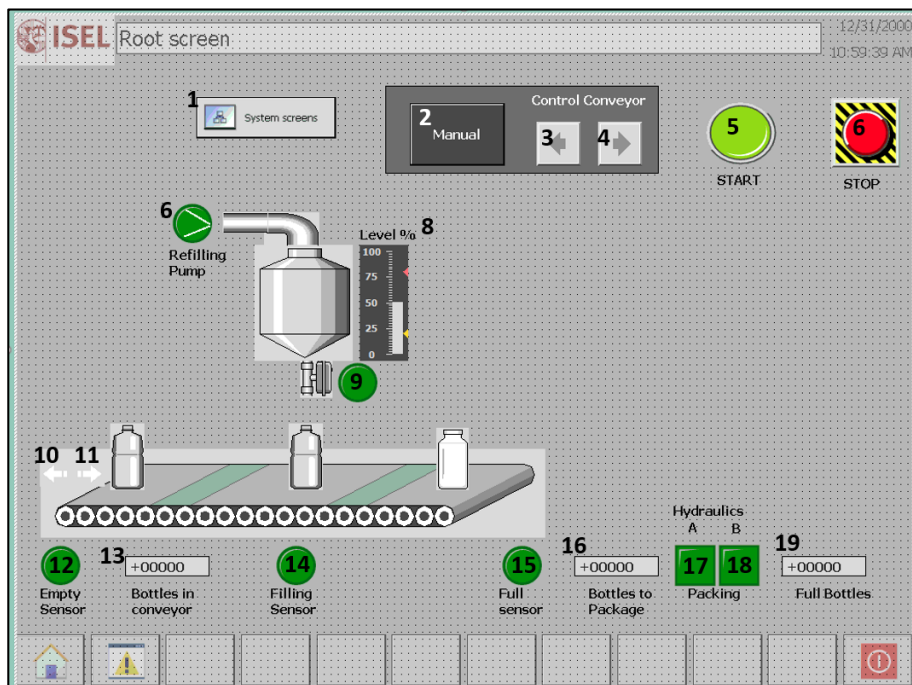


Figura 20 – Root screen

Por último está apresentado o ecrã raiz que ilustra o processo. Os números não existem no que seria o *display* real, são apenas para a listagem das imagens apresentadas, juntamente como uma breve explicação e a *tag* utilizada.

Há que notar que para que os botões apresentados possam interagir com o sistema físico teriam que ser usadas memórias no lugar de *inputs*, por isso algumas das entradas apresentadas na tabela 1, foram alteradas para memórias e o programa foi alterado de acordo.

As alterações estarão apresentadas em anexo.

Tabela 6 - Tabela explicativa do display do HMI

Número	Tag	Descrição	Endereço
1	System screens	Abre o painel de system screen	
2	Buton_Auto_selectionI	Activa/desactiva o modo manual	M 16.4
3	Button_Backward	Faz o motor do tapete andar para trás	M16.3
4	Button_Forward	Faz o motor do tapete andar para a frente	M 16.2
5	Button_Start	Inicia o processo	M 16.0
6	Button_Stop	Pára o processo	M 16.1
7	Pump	Mostra quando a bomba de carregamento está ativa	Q 9.0
8	Memory_Level_percent	Apresenta graficamente o nível medido pelo sensor de nível em percentagem	MW 12
9	Auto_mode_DB_Filling_Output	Válvula de enchimento fechada	
10	Motor_Backward	Motor do tapete andar para trás	Q 9.4
11	Motor_Forward	Motor do tapete andar para a frente	Q 9.3
12	Sensor_Empty	Acende quando o sensor está ativo	I 16.5
13	Memory_Bottles_system	Mostra o número de garrafas no sistema	MW 8
14	Sensor_Filling	Acende quando o sensor está ativo	I 16.6
15	Sensor_Full	Acende quando o sensor está ativo	I 16.7
16	Bottles	Mostra o número de garrafas a empacotar	MW 2
17 e 18	Hydraulic_A; Hydraulic_B	Quando o cilindro de empacotamento A fica ativo acende o quadrado da esquerda, quando o B fica ativo acende o da direita	Q 9.1; Q 9.2
19	Memory_Full_Bottles	Mostra o número de garrafas cheias	MW 6

7. Conclusões e Trabalho Futuro

O objetivo fulcral deste trabalho prendia-se com o enchimento de garrafas e o seu embalamento em packs de 12 unidades.

Através da linguagem LAD definiu-se a função FC1, “Refill”, bem como o bloco principal OB1, utilizou-se SFC para programar a FB1, “Auto_mode” e ainda o texto estruturado para a função FC3, “Valve_Control”.

Conseguiu-se cumprir o objetivo, existe controlo sob nível de líquido para encher as garrafas, bem como a abertura da válvula do tanque, quer se esteja no modo manual ou automático. É possível parar o processo a qualquer altura através do “Button_Stop”. Consegue-se saber qual a quantidade de garrafas cheias e no sistema, até serem embaladas, as garrafas são embaladas em packs de 12 unidades.

Em suma, houve uma preocupação em aplicar diferentes linguagens no trabalho, para o tornar mais rico do ponto de vista académico, e fomentar o conhecimento que é adquirido ao aplicar diferentes ferramentas. A linguagem SFC para sequências, como é o caso do enchimento das garrafas e posterior embalamento, torna-se bastante prática as etapas de cada processo ficam mais claras.

Para um trabalho futuro seria uma mais valia considerar interrupções no programa, para o caso de a máquina avariar, por exemplo e, seria interessante ter um outro sensor para verificar se as embalagens estão dentro dos conformes estabelecidos e a respetiva contagem de embalagens rejeitadas e boas, para avaliar a capacidade produtiva da secção.

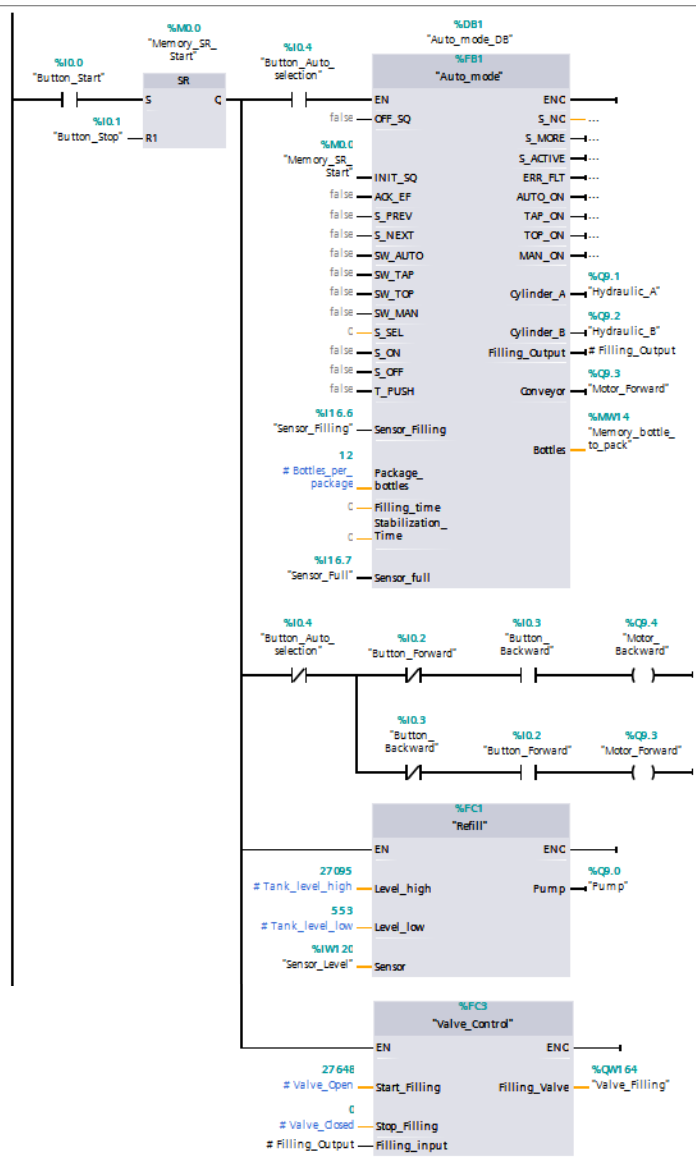
8. Bibliografia

Slides da Unidade Curricular, Programação de Autómatos, Mário Mendes

Commented [TR4]: Tão e isto?

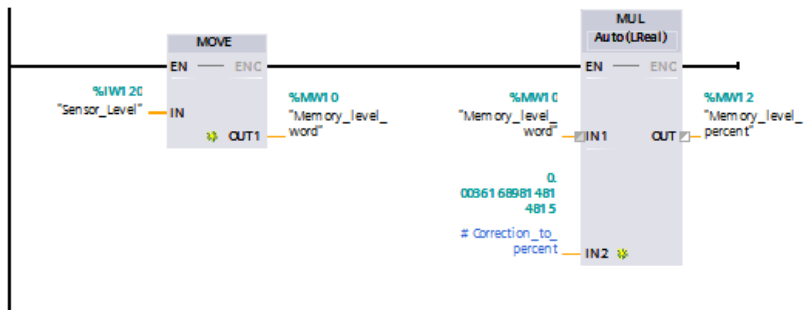
9. Anexos

9.1 Anexo 1 – Main [OB1]



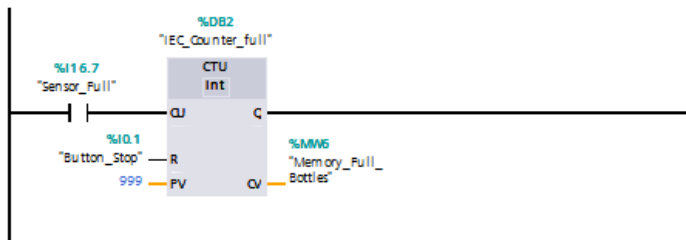
Network 2: Level_Conversion

Comment



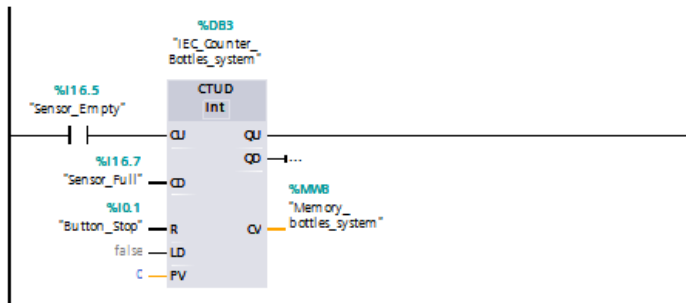
Network 3: Conter_Full_Bottles

Comment

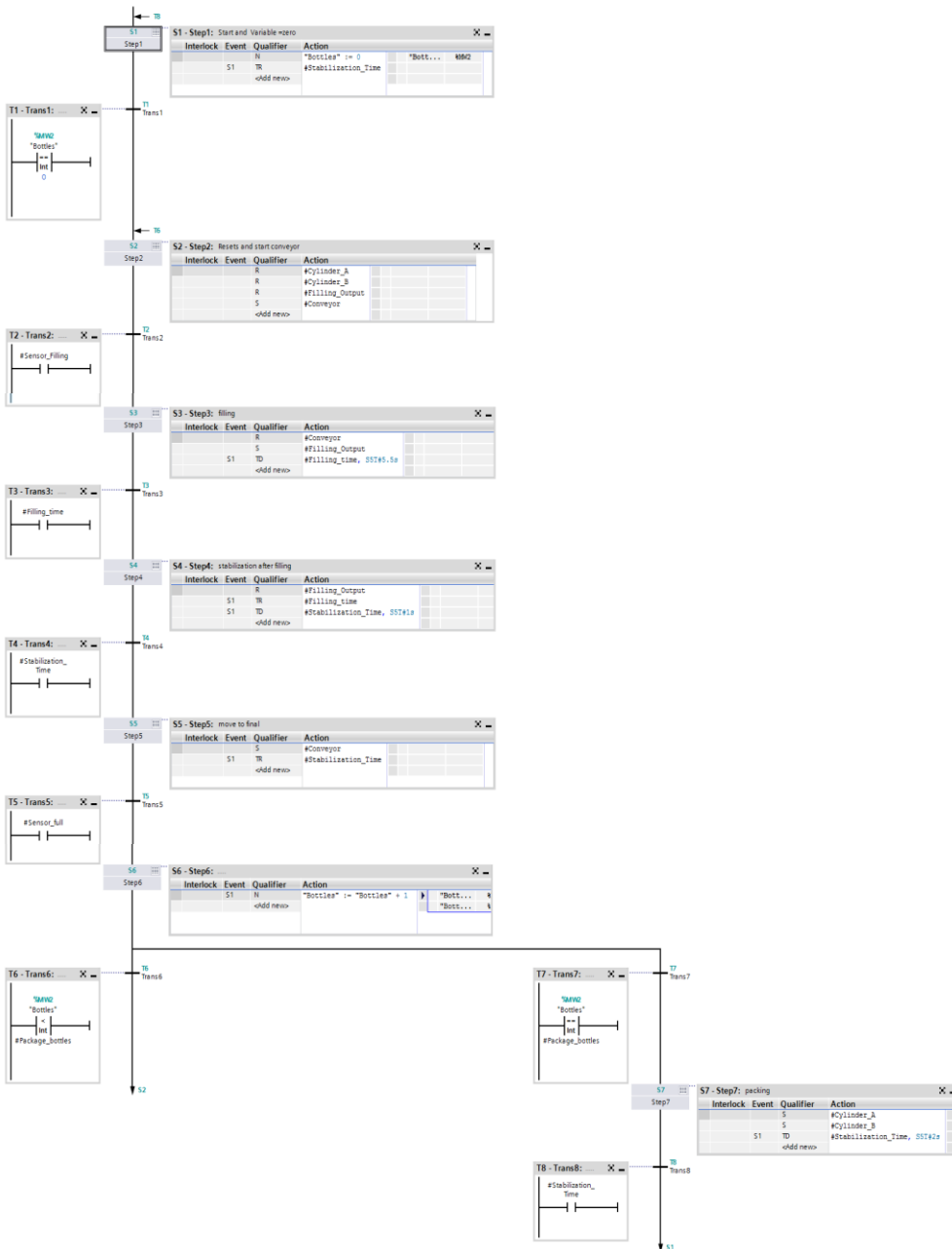


Network 4: Counter_Bottles_inside_system

Comment



9.2. Anexo 2 – Auto_mode [FB1]



9.3. Anexo 4 – Tag Tables

PLC tags										
	Name	Tag table	Data type	Address	Retain	Access...	Write...	Visibl...	Supervision	Comment
1	Button_Start	Default tag table	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		NO instant contact
2	Button_Stop	Default tag table	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		NC Contact
3	Button_Forward	Default tag table	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		on/off button
4	Button_Backward	Default tag table	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		on/off button
5	Button_Auto_selection	Default tag table	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		on= auto off=Manual
6	Sensor_Empty	Default tag table	Bool	%I16.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		1st sensor
7	Sensor_Filling	Default tag table	Bool	%I16.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Middle sensor
8	Sensor_Full	Default tag table	Bool	%I16.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Last Sensor
9	Sensor_Level	Default tag table	Word	%IW120	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Tank level sensor
10	Valve_Filling	Default tag table	Word	%QW164	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Filling valve output
11	Pump	Default tag table	Bool	%Q9.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		tank refill tank
12	Hydraulic_A	Default tag table	Bool	%Q9.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13	Hydraulic_B	Default tag table	Bool	%Q9.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
14	Motor_Forward	Default tag table	Bool	%Q9.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	Motor_Backward	Default tag table	Bool	%Q9.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	Memory_SR_Start	Default tag table	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		memory of positive pulse from start button
17	Bottles	Default tag table	Int	%MW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		variable from grafket counter
18	Memory_SR_refill	Default tag table	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		memory of tank refill system
19	Valve_Open	Default tag table	Word	%MW4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		output to valve control (from FC3)
20	Memory_Full_Bottles	Default tag table	Int	%MW6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Memory off ALL full bottles
21	Memory_bottles_system	Default tag table	Int	%MW8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Number of bottles in system
22	Memory_level_word	Default tag table	Int	%MW10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Memory from level
23	Memory_level_percent	Default tag table	Int	%MW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Memory from level converted to %
24	Memory_bottle_to_pack	Default tag table	Int	%MW14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
25	<Add new>				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

HM tags						
	Name	Data type	PLC tag	Coding	Acquisition cycle	Acquisition mo...
	Auto_mode_DB_Filling_Output	Bool	Auto_mode_DB.Fillin...	Binary	1 s	Cyclic in op...
	Pump	Bool	Pump	Binary	1 s	Cyclic in operat...
	Sensor_Empty	Bool	Sensor_Empty	Binary	1 s	Cyclic in operat...
	Sensor_Filling	Bool	Sensor_Filling	Binary	1 s	Cyclic in operat...
	Sensor_Full	Bool	Sensor_Full	Binary	1 s	Cyclic in operat...
	Hydraulic_A	Bool	Hydraulic_A	Binary	1 s	Cyclic in operat...
	Button_Auto_selection	Bool	Button_Auto_selection	Binary	1 s	Cyclic in operat...
	Button_Backward	Bool	Button_Backward	Binary	1 s	Cyclic in operat...
	Hydraulic_B	Bool	Hydraulic_B	Binary	1 s	Cyclic in operat...
	Button_Forward	Bool	Button_Forward	Binary	1 s	Cyclic in operat...
	Button_Start	Bool	Button_Start	Binary	1 s	Cyclic in operat...
	Button_Stop	Bool	Button_Stop	Binary	1 s	Cyclic in operat...
	Auto_mode_DB_Package_bottles	Int	Auto_mode_DB.Package...	Binary	1 s	Cyclic in operat...
	Bottles	Int	Bottles	Binary	1 s	Cyclic in operat...
	Memory_Full_Bottles	Int	Memory_Full_Bottles	Binary	1 s	Cyclic in operat...
	Memory_bottles_system	Int	Memory_bottles_system	Binary	1 s	Cyclic in operat...
	Memory_level_percent	Int	Memory_level_percent	Binary	1 s	Cyclic in operat...
	Tag_ScreenNumber	UInt	<Undefined>	Binary	1 s	Cyclic in operat...
	<Add new>					