

DESENVOLVIMENTO DE APLICAÇÃO PARA A OTIMIZAÇÃO DE PROCESSO PRODUTIVO

Bruno Filipe Roca Mendes



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2015

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Bruno Filipe Roca Mendes, N° 1130027, 1130027@isep.ipp.pt

Orientação científica: Prof. Dr. Ramiro de Sousa Barbosa, rsb@isep.ipp.pt

Co-Orientação científica: Prof. Dr. Manuel Fernando dos Santos Silva, mss@isep.ipp.pt

Empresa: Grohe Portugal – Componentes Sanitários, Lda.

Supervisão: Eng.º Arlindo Francisco, afrancisco@grohe.pt



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

3 de agosto de 2015

Dedico este documento aos meus pais, Hilário e Teresa, fonte de inspiração e conforto, exemplos de dignidade e amor. À minha irmã, Cheila, fonte de juventude e carinho. E à Sara, como amor e gratidão pela sua compreensão, carinho e apoio para a realização deste trabalho.

Agradecimentos

Aos Engenheiros Manuel Silva e Ramiro Barbosa, pela confiança, pela orientação deste trabalho, pela amizade, paciência e conselhos que contribuíram para o meu crescimento científico e intelectual.

Ao Engenheiro Arlindo Francisco e ao Sr. Oliveira, por todo o apoio e conselhos que me foram dando ao longo de todo o meu percurso na Grohe-Portugal. As suas orientações e passagens de experiências daquilo que é o dia-a-dia de uma indústria e de como ela deve ser vista e estudada, foram dos pontos mais enriquecedores em todo este percurso. Um obrigado também pela forma acolhedora e descontraída com que me receberam e integraram.

A todo o departamento da Manutenção pela confiança depositada na minha pessoa e pela disponibilidade demonstrada para me ajudar sempre que necessitei.

Desejo também agradecer à minha namorada pela sua companhia e incentivo, determinante para concluir esta etapa.

Dedico um agradecimento especial a toda a minha família e amigos por todo o apoio e força depositada em mim ao longo de todo o percurso académico.

Por último, a todas as pessoas que de alguma forma contribuíram direta ou indiretamente para a realização deste trabalho.

Resumo

Empreendimentos industriais requerem o acompanhamento completo e preciso de dados de máquinas em tempo real, integrando as atividades do chão de fábrica com a engenharia, suporte e negócios. É reconhecido que sistemas de monitorização são fundamentais para melhorar a qualidade do produto, reduzir os tempos e para a adaptação rápida às mudanças de produção. Os sensores de processos possuem custos altos, não são compactos e são de difícil implementação. Portanto, seria estratégico desenvolver sistemas de monitorização que usufruem ao máximo dos recursos já existentes nas máquinas e ferramentas e que não representem custos adicionais elevados e de difícil implementação. O uso de PLC nas indústrias é crescente. Estes mostram-se como ferramentas potenciais para atender a estas exigências. Os PLC possuem os recursos necessários para a aquisição de informações e controlo das máquinas, facilitando a interação com um sistema SCADA.

O objetivo do presente trabalho é pesquisar soluções sobre sistemas de monitorização que possam substituir um sistema instalado antiquado e encontrar uma solução para o controlo de um processo de vazamento. Para tal, foram consideradas duas soluções: uma passaria por desenvolver um sistema de monitorização que controlasse o processo de vazamento, e outra que permitisse ser o PLC a fazer o controlo, adquirindo um HMI apenas para monitorização. Esta última solução revelou-se como a melhor, visto que um PLC tem mais durabilidade em relação a um PC, em termos de atualizações de *hardware* e *software*.

Assim, foram criadas funções no PLC para fazer o controlo de vazamento e foi adquirido uma consola Beijer, e respetivo *software* de programação SCADA, para a criação do sistema de monitorização. Os resultados mostram que o PLC é capaz de executar o controlo, não sendo necessário investir em PC que são mais caros, bastando para isso adquirir sistemas SCADA simples e capazes.

Palavras-Chave

SCADA, PLC, HMI, controlo, monitorização, TIA Portal, válvula, transdutor, processo de vazamento.

Abstract

Manufacturing companies require complete real time and accurate monitoring of machine and tools data for integrating the shop floor with other company activities, as engineering, maintenance and business. It is acknowledged that monitoring systems capabilities are unavoidable to improve product quality, to reduce production time, and to the rapid adaptation in the changes of production. The use of external sensors gives high costs and these do not have the advantages of compactness and ease of use. Therefore, it is extremely advantageous to develop monitoring systems, maximizing the use of the resources already existent in the machines tools and without high additional costs and hard implementations. The use of PLC on industries is growing. This kind of control has been a great tool to attend these present requirements. It has the necessary resources to get and manage all the information and control of the machines, and easily interacts with SCADA systems.

The objective of this study is to research solutions for monitoring systems that can replace an outdated installed system and find a solution to control a pouring process. To this purpose were considered two solutions: one is to develop a monitoring system that controls the casting process, and another one that lets the PLC to do the control, and acquiring an HMI only for monitoring. This last option has proved to be the best solution since a PLC has more durability than a PC, in terms of hardware and software upgrades.

Thus functions were created in the PLC to control the pouring process, and was acquired a Beijer console and its respective SCADA programming software for the creation of the monitoring system. The results show that the PLC is able to perform the process control without the need to invest in PC which are more expensive, being enough to purchase simple and capable SCADA systems.

Keywords

SCADA, PLC, HMI, control, monitoring, TIA Portal, valve, transducer, pouring process.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJETIVOS	2
1.3. CALENDARIZAÇÃO	3
1.4. ORGANIZAÇÃO DO RELATÓRIO	3
2. GROHE PORTUGAL	5
2.1. INTRODUÇÃO À EMPRESA	5
2.2. PROCESSO PRODUTIVO – ATIVIDADES PRINCIPAIS	7
2.3. PROCESSO PRODUTIVO – DEPARTAMENTOS DE SUPORTE	19
3. SISTEMAS SCADA	23
3.1. INTRODUÇÃO	23
3.2. ORIGEM	25
3.3. ARQUITETURA	26
3.4. COMPONENTES	28
3.5. FUNCIONALIDADES	36
3.6. INTEGRAÇÃO DE SISTEMAS	37
4. CONTROLADORES LÓGICOS PROGRAMÁVEIS	39
4.1. CONTROLADOR LÓGICO PROGRAMÁVEL	39
4.2. PRINCÍPIO DE FUNCIONAMENTO	41
4.3. ASPETOS DE <i>HARDWARE</i>	42
4.4. ASPETOS DE <i>SOFTWARE</i>	43
4.5. LINGUAGENS DE PROGRAMAÇÃO	44
5. ARQUITETURA DO SISTEMA	47
5.1. INTRODUÇÃO	47
5.2. BP260 X.....	48
5.3. ESTUDO DOS COMPONENTES QUE INTEGRAM O SISTEMA.....	56
5.4. TIA PORTAL	64
5.5. PROGRAMA DA MÁQUINA	68
5.6. IX DEVELOPER DA BEIJER ELECTRONICS.....	72
6. IMPLEMENTAÇÃO PRÁTICA	81
6.1. ARQUITETURA DO NOVO SISTEMA	81
6.2. PROGRAMAÇÃO PARA O CONTROLO DE VAZAMENTO	82
6.3. CONTROLO PID APLICADO	96
6.4. CRIAÇÃO DA INTERFACE-HOMEM-MÁQUINA	99
7. TESTES E RESULTADOS	119
7.1. PARAMETRIZAÇÃO PID BIDÃO.....	119

7.2.	PARAMETRIZAÇÃO DO PID PARA O FORNO – TESTE 1	129
7.3.	PARAMETRIZAÇÃO DO PID PARA O FORNO – TESTE 2	135
7.4.	RESULTADOS DOS TESTES NA MÁQUINA	139
8.	CONCLUSÕES	149
	REFERÊNCIAS DOCUMENTAIS.....	151
	ANEXO A. FLUXO PRODUTIVO DA GROHE-PORTUGAL.....	153
	ANEXO B. DATASHEET DO CABO DE COMUNICAÇÃO ENTRE A CONSOLA HMI (BEIJER) E O PLC SIEMENS E CARACTERÍSTICAS DA CONSOLA IX T15B.....	155
	ANEXO C. ÁRVORE DE MENUS DA NOVA CONSOLA BEIJER.....	157
	ANEXO D. ENDEREÇAMENTO INDIRETO [22]	159
	ANEXO E. FUNÇÃO BLOCO CONT_C (PID) [22].....	165
	ANEXO F. COHEN-COON TUNING RULES [23]	173
	ANEXO G. DIFERENÇAS ENTRE HMI ANTIGO E NOVO HMI BEIJER.....	179
	ANEXO H. COMPONENTES INFORMÁTICOS	187

Índice de Figuras

Figura 1	Calendarização do projeto.....	3
Figura 2	Fotografia da fábrica em Albergaria-a-Velha.....	6
Figura 3	Transvase de liga de latão.	8
Figura 4	Machos de areia.....	9
Figura 5	Corte dos cachos.....	10
Figura 6	Granalhadoras.....	11
Figura 7	Ferramenta de maquinagem.	12
Figura 8	Robô de lixamento.	13
Figura 9	Lixamento manual.....	14
Figura 10	Unidade de polimento.	15
Figura 11	Linha de cromagem e niquelagem.....	16
Figura 12	Linha de descromagem e desniquelagem.	17
Figura 13	Exemplo de uma linha de montagem.	18
Figura 14	Esquema dos departamentos da Grohe.....	19
Figura 15	Exemplo de uma estação de controlo de um sistema SCADA [2].	24
Figura 16	Arquitetura de um sistema SCADA.	27
Figura 17	Estruturação do sistema SCADA.	28
Figura 18	Sistema SCADA a comunicar com equipamentos de campo [4].	29
Figura 19	Sistema de comunicação para interface SCADA.	31
Figura 20	Níveis da pirâmide de automação [10].	32
Figura 21	Exemplo de um programa de supervisão [15].	37
Figura 22	Configurações das máquinas de chão de fábrica.	38
Figura 23	Ciclo de funcionamento de um PLC [18].	41
Figura 24	Sistema PLC [19].	42
Figura 25	BP260 X.....	48
Figura 26	Diagrama do modo de funcionamento da cabeça X.....	55
Figura 27	Arquitetura do sistema antigo.....	56
Figura 28	<i>PC-Adapter</i> para comunicação MPI [20].	59
Figura 29	Válvula proporcional do forno.	60
Figura 30	Modelo da válvula E/P <i>pressure regulator</i> , Series ED12 da AVENTICS.....	61
Figura 31	Transdutor a ser usado no forno para leitura dos valores de pressão.	61
Figura 32	Estrutura do projeto da BP260 X em SIMATIC.	64
Figura 33	Visão Geral do TIA Portal [21].	65
Figura 34	TIA Portal <i>portal view</i>	66
Figura 35	TIA Portal <i>project view</i>	66
Figura 36	Conversão do projeto em SIMATIC para TIA Portal.	67
Figura 37	Interface TIA Portal do programa da máquina.....	68

Figura 38	Diagrama da hierarquia das funções do programa.	69
Figura 39	Janela inicial do iX Developer.	74
Figura 40	Janela de seleção do tipo de consola a utilizar.	75
Figura 41	Janela para a escolha da família do controlador e respetivo protocolo de comunicação. 76	76
Figura 42	Área de trabalho do iX Developer.....	76
Figura 43	Menu de declaração das <i>tags</i> da consola.	78
Figura 44	Menu de ações.	79
Figura 45	Servidor de alarmes do iX Developer.	80
Figura 46	<i>AlarmViewer</i> do iX Developer.	80
Figura 47	Arquitetura do novo sistema.....	82
Figura 48	Fluxograma da função de controlo de vazamento.	84
Figura 49	Grafcet nível 1 para a função de controlo de vazamento.	85
Figura 50	Inicialização de variáveis.	86
Figura 51	Configuração do tempo para o envio de valores.	86
Figura 52	Configuração para a leitura dos valores de <i>setpoint</i>	87
Figura 53	Escrita dos valores de pressão lidos pelo transdutor.	88
Figura 54	Incrementação da posição dos apontadores.....	88
Figura 55	Condições de varrimento do DB.	89
Figura 56	Código do FC4 “RESET_DB”.	90
Figura 57	Cálculos para a conversão do valor analógico (1).....	91
Figura 58	Cálculos para a conversão do valor analógico (2).....	92
Figura 59	Atribuição da saída da válvula.	93
Figura 60	Condições iniciais para ativar o vazamento.	93
Figura 61	Gestão do vazamento.....	95
Figura 62	Chamada das funções criadas dentro da Função FC1.	96
Figura 63	Bloco PID da Siemens.....	97
Figura 64	Ferramenta gráfica do bloco PID.	99
Figura 65	Ecrã principal do novo programa.	100
Figura 66	Controlo e visualização do processo de fabrico.	101
Figura 67	Janela <i>pop-up</i> com gráfico de controlo.	101
Figura 68	Visualizador de alarmes.	102
Figura 69	X HEAD MENU.	103
Figura 70	X WORKSTATION SETTINGS PAGE 1.....	104
Figura 71	X WORKSTATION SETTINGS PAGE 3.....	105
Figura 72	X WORKSTATION SETTINGS PAGE 4.....	106
Figura 73	X WORKSTATION SETTINGS PAGE 5.....	108
Figura 74	Linear Scale Check Page.....	109
Figura 75	Proximity Switch Check Page.....	110
Figura 76	Criação de uma nova curva de vazamento.	111
Figura 77	Carregar e guardar curva de vazamento.	111
Figura 78	Modificar curva de vazamento.	112

Figura 79	Atribuição de novos valores de pressão para a configuração da curva.	113
Figura 80	Apagar curvas de vazamento.....	113
Figura 81	Menu do administrador.	114
Figura 82	Menu de login.....	115
Figura 83	Configurações de rotação da máquina.....	115
Figura 84	Configurações do forno.....	116
Figura 85	Sistema criado para simular o comportamento do forno.....	120
Figura 86	Variação do valor analógico da válvula.	120
Figura 87	Gráfico da evolução dos valores de pressão determinada pelo transdutor ao longo do tempo (bidão).	122
Figura 88	Reta tangente à curva (bidão).	124
Figura 89	Gráfico com as condições necessárias para aplicar método Cohen-Coon (bidão)....	125
Figura 90	Fórmulas para calcular as constantes PID segundo o método de Cohen-Coon.....	125
Figura 91	Atribuição das constantes K_c e T_i ao bloco PID “CONT_C”.	127
Figura 92	Valores estabelecidos de pressão para a curva pré-definida.....	128
Figura 93	Resultados obtidos com o controlador PID aplicado ao bidão.....	128
Figura 94	Forno da máquina.....	130
Figura 95	Variação do valor analógico da válvula (teste 1).	130
Figura 96	Gráfico obtido através dos valores de pressão do transdutor ao longo do tempo (teste 1). 132	
Figura 97	Reta tangente à curva (teste 1).	133
Figura 98	Gráfico com as condições necessárias para aplicar método Cohen-Coon (teste 1)....	134
Figura 99	Variação do valor analógico da válvula (teste 2).	135
Figura 100	Gráfico obtido através dos valores de pressão do transdutor ao longo do tempo (teste 2). 136	
Figura 101	Reta tangente à curva (teste 2).	137
Figura 102	Gráfico com as condições necessárias para aplicar método Cohen-Coon (teste 2).138	
Figura 103	Definição de uma curva com valores aleatórios.....	140
Figura 104	Resposta do sistema de controlo com as constantes $K_c = 0,6816$ e $T_i = 0,208$	140
Figura 105	Resposta do sistema de controlo com as constantes $K_c = 0,978$ e $T_i = 0,233$	141
Figura 106	Criação de uma nova curva (teste).	142
Figura 107	Resposta do sistema de controlo (teste) com as constantes $K_c = 0,6816$ e $T_i = 0,198$. 142	
Figura 108	Definição de uma nova curva (curva real).	143
Figura 109	Resposta do sistema de controlo (curva real) com as constantes $K_c = 0,6816$ e $T_i =$ $0,198$. 144	
Figura 110	Inserção do macho de areia na coquilha para a realização de um teste real.....	145
Figura 111	Resposta do sistema de controlo (teste real) com as constantes $K_c = 0,6816$ e $T_i =$ $0,198$. 145	
Figura 112	Peça final obtida com o novo sistema de controlo de vazamento.	146
Figura 113	Resposta do sistema de controlo (teste real) com o sistema antigo.....	147
Figura 114	Ponteiro Any.	163
Figura 115	Diagrama de blocos do CONT_C.	168

Figura 116	Esquema do Controlo PID.....	174
Figura 117	Aquisição de valores através do gráfico.....	175
Figura 118	Regras Cohen-Coon.	176
Figura 119	Main menu.....	179
Figura 120	Menu do programa (controlo).	180
Figura 121	Menu do programa (HMI antigo).	180
Figura 122	Menu configuração das cabeças (HMI antigo).....	181
Figura 123	Menu configuração das cabeças (HMI novo).....	181
Figura 124	Criação de uma nova curva de vazamento.	182
Figura 125	Criação de uma nova curva de vazamento.	182
Figura 126	Menu para guardar, carregar e/ ou apagar as curvas.	183
Figura 127	Menu da cabeça X (HMI antigo).....	183
Figura 128	Carregar e guardar curva de vazamento.	184
Figura 129	Modificar curva de vazamento.	184
Figura 130	Atribuição de novos valores.	185
Figura 131	Apagar curvas de vazamento.....	185

Índice de Tabelas

Tabela 1	Principais características da CPU S7-315-2 DP.	58
Tabela 2	Comparação da CPU da máquina com a nova CPU a implementar.	63
Tabela 3	Requisitos recomendados para a instalação do iX Developer.	73
Tabela 4	Descrição da área de trabalho do iX Developer.	77
Tabela 5	Correspondência dos valores a inserir em graus.	107
Tabela 6	Relação tempo com pressão (bidão).	121
Tabela 7	Relação entre o tempo e pressão e respetivos declives (bidão).	122
Tabela 8	Expressão da reta tangente ao longo do tempo (bidão).	123
Tabela 9	Relação tempo com pressão (teste1).	131
Tabela 10	Relação tempo com pressão e respetivos declives (teste 1).	132
Tabela 11	Relação tempo com pressão (teste 2).	136
Tabela 12	Relação tempo com pressão e respetivos declives (teste 2).	137
Tabela 13	Formats for the declaration of an ANY pointer.	160
Tabela 14	Coding of data types for the ANY pointer.	161
Tabela 15	Coding of the memory areas for the ANY pointer.	162
Tabela 16	Input parameter CONT_C.	169
Tabela 17	Output parameters CONT_C.	171

Acrónimos

AC	– <i>Alternating Current</i>
API	– <i>Application Programming Interface</i>
C#	– <i>C sharp</i>
CEP	– <i>Controlo Estático de Processos</i>
COM	– <i>Component Object Model</i>
CP	– <i>Módulos de comunicação</i>
CPU	– <i>Central Processing Unit</i>
DB	– <i>Data Block</i>
DC	– <i>Direct Current</i>
DCOM	– <i>Distributed Component Object Model</i>
DCS	– <i>Distributed Control System</i>
DDE	– <i>Dynamic Data Exchange</i>
DEE	– <i>Departamento de Engenharia Eletrotécnica</i>
DLL	– <i>Dynamic Link Library</i>
DP	– <i>Double Port</i>
E/P	– <i>Electric/ Pressure</i>

E/S	– Entrada/ Saída
EPROM	– <i>Electrically Erasable Programmable Read-Only Memory</i>
FB	– <i>Function Block</i>
FBD	– <i>Function Block Diagram</i>
FC	– <i>FunCtion</i>
FDDI	– <i>Fiber Distributed Data Interface</i>
FLASH-EPROM	– FLASH – <i>Electrically Erasable Programmable Read-Only Memory</i>
FM	– Módulos de Função
GM	– General Motors
HMI	– <i>Human Machine Interface</i>
HTML	– <i>Hyper Text Markup Language</i>
I/O	– <i>Input/ Output</i>
IEC	– <i>International Electrotechnical Commission</i>
IL	– <i>Instruction List</i>
IP	– <i>Internet Protocol</i>
IPC	– <i>Industrial Portable Computer</i>
IPX	– <i>Internetwork Packet Exchange</i>
ISEP	– Instituto Superior de Engenharia do Porto

LAD	– <i>Ladder Diagram</i>
LAN	– <i>Local Area Network</i>
MMI	– <i>Man Machine Interface</i>
MPI	– <i>Multi Point Interface</i>
MW	– <i>Memory Word</i>
NEMA	– <i>Nation Electrical Manufactures Association</i>
NetDDE	– <i>Network Dynamic Data Exchange</i>
NVRAM	– <i>Non-Volatile Random Access Memory</i>
OB	– <i>Organization Block</i>
ODBC	– <i>Open DataBase Connectivity</i>
OLE	– <i>Object Linking and Embedding</i>
OPC	– <i>Open Platform Communication</i>
PC	– <i>Personal Computer</i>
PID	– <i>Proportional Integral Directive</i>
PLC	– <i>Programmable Logic Controller</i>
RAM	– <i>Random Access Memory</i>
RTDB	– <i>Real Time DataBase</i>
RTU	– <i>Remote Terminal Units</i>
SAP	– <i>Systems, Applications and Products in data processing</i>

SCADA	– <i>Supervisory Control And Data Acquisition</i>
SD Card	– <i>Secure Digital Card</i>
SFC	– <i>Diagrama Sequencial de Funções</i>
SLW	– <i>Shift Left Word</i>
SP	– <i>SetPoint</i>
SPX	– <i>Sequenced Packet Exchange</i>
SQL	– <i>Structured Query Language</i>
ST	– <i>Structured Text</i>
STL	– <i>Statement List</i>
TCP	– <i>Transmission Control Protocol</i>
TD	– <i>Transmission Data</i>
TI	– <i>Tempo integral</i>
USB	– <i>Universal Serial Bus</i>
WAN	– <i>Wide Area Network</i>

1. INTRODUÇÃO

Este documento descreve o projeto desenvolvido na Grohe-Portugal, como forma de realizar a unidade curricular Tese/Dissertação, do Mestrado em Engenharia Eletrotécnica e de Computadores, lecionado pelo Departamento de Engenharia Eletrotécnica (DEE), do Instituto Superior de Engenharia do Porto (ISEP). Este capítulo começa com uma breve apresentação da empresa assim como uma explicação do problema em causa. São apresentados os objetivos e as motivações do autor e uma calendarização onde estão indicadas as diferentes tarefas envolvidas, terminando com a organização do documento.

1.1. CONTEXTUALIZAÇÃO

Este projeto surgiu do desejo de realizar um trabalho no âmbito da automação industrial e da oportunidade apresentada pela Empresa Grohe-Portugal, situada em Albergaria-a-Velha, que mais uma vez colaborou com o ISEP. Este trabalho insere-se no âmbito do Curso de Mestrado em Engenharia Eletrotécnica e de Computadores, Ramo de Automação e Sistemas.

O grupo Alemão Grohe é líder mundial no fabrico de produtos sanitários e fornecedor de sistemas e exportador de natureza global. Fundado em 1936, por Friedrich Grohe, conta hoje com 140 escritórios de representação por todo o mundo, e seis unidades de produção, estando três localizadas na Alemanha e as restantes em Portugal, Tailândia e Canadá. Atualmente integra o Grupo Nipónico LIXIL. No que diz respeito às instalações portuguesas, estas encontram-se em funcionamento desde 1998 e localizam-se em Albergaria-a-Velha, distrito de Aveiro.

A Grohe-Portugal contribui com uma parte fundamental na produção total do grupo, pois não só produz peças de gama comum como também tem exclusividade em certos produtos, como é o caso das torneiras TERMOSTÁTICA GROTHER 1000 COSMOPOLITAN, da linha de torneiras MINTAS TOUCH e das torneiras GROHE BLUE.

Tendo como principais metas os elevados padrões de qualidade, tecnologia, *design* e sustentabilidade, a empresa procura destacar-se de todas as outras do grupo, buscando assim as melhores soluções para os problemas que surjam ou que poderão surgir. Desta forma, é proposto o projeto em causa, que visa solucionar um problema com que esta empresa se debate num dos seus processos de fabrico.

Como é sabido, no meio industrial não é viável a constante atualização de equipamentos, tanto a nível de *hardware* como a nível de *software*, o que pode levar a que surjam dificuldades para a manutenção e/ou produção, pois a partir de determinada altura muitos dos equipamentos são descontinuados pelos seus fabricantes e os que os substituem, por vezes, tornam-se incompatíveis com os mais antigos.

Este é um dos problemas que poderá vir a atingir um dos processos de fabrico que, apesar de funcionar corretamente, apresenta limitações caso surja a necessidade de substituir alguns dos componentes, mais concretamente no que toca ao seu sistema de monitorização.

Atualmente o sistema de monitorização funciona com um computador, cujo *hardware* e *software* já não são fabricados, e em caso de avaria não haverá solução possível senão conceber algo mais recente para o mesmo propósito, melhorando algumas falhas/aspectos que possam ser encontrados.

1.2. OBJETIVOS

O objetivo principal deste projeto de estágio é apresentar uma solução que permita a substituição do equipamento utilizado para o controlo e monitorização do processo de vazamento de peças por equipamento mais recente, tendo como base as mesmas funcionalidades e a melhoria de alguns aspetos que sejam encontrados. Desse modo, para a realização deste projeto deve-se:

- Realizar a Interface Homem Máquina (*Human Machine Interface* – HMI) que deverá interagir com um *Programmable Logic Controller* (PLC) Siemens S7-300 de forma a

No terceiro capítulo são apresentados os sistemas SCADA, sendo introduzidos vários conceitos, como a sua origem, e como estes funcionam.

No quarto capítulo é dado destaque aos Controladores Lógicos Programáveis, visto que são um dos componentes essenciais para o funcionamento do sistema a implementar. Serão apresentados aspetos de *hardware* e de *software*, assim como as várias linguagens de programação possíveis de utilizar para a sua programação.

No quinto capítulo é apresentada a arquitetura do sistema, ou seja, neste capítulo é apresentado o modo de funcionamento da máquina, os componentes essenciais para a resolução do problema, o estudo do programa a ser adaptado e também o estudo das ferramentas necessárias para a programação e configuração do *hardware*.

No capítulo seis é apresentado o trabalho desenvolvido ao longo deste projeto. É apresentada e explicada a programação da nova função de controlo de vazamento e da consola utilizada.

No capítulo sete são apresentados todos os resultados que, mediante as disponibilidades, foram obtidos, terminando-se depois com a conclusão no capítulo oito.

O documento contém oito anexos diferentes. No primeiro anexo (A) é apresentado o Fluxo produtivo da Grohe-Portugal. No segundo anexo (B) são apresentadas as ligações elétricas para dimensionar um cabo de comunicação entre uma consola Beijer e um PLC da Siemens (MPI). No Anexo C é apresentada a árvore de menus do novo HMI e no Anexo D é explicado o endereçamento indireto para a programação por instruções num PLC Siemens. No Anexo E é explicado o bloco PID do TIA Portal, enquanto no Anexo F é apresentado o procedimento para a sintonia de um bloco PID. No anexo G são feitas comparações entre a HMI antiga e a nova a ser implementada e, por último, o Anexo H apresenta o conteúdo do CD-ROM deste documento.

2. GROHE PORTUGAL

A Grohe AG é a maior empresa da Europa e líder mundial no que diz respeito ao fabrico e fornecimento de acessórios sanitários, e tem como base quatro princípios (qualidade, tecnologia, design e sustentabilidade) dos quais se orgulha e pelos quais é reconhecida. Atualmente integra o Grupo LIXIL, uma grande empresa Japonesa ligada à área da construção e afins, também reconhecida em todo o mundo. O presente capítulo aborda de forma mais detalhada a estrutura e funcionamento de um dos constituintes do grupo Grohe, a Grohe Portugal, situada em Albergaria-a-Velha, Portugal.

2.1. INTRODUÇÃO À EMPRESA

Sediada em Hemes, Alemanha, o grupo Grohe encontra-se espalhado por todo o mundo. No que diz respeito a centros de produção, a Grohe marca presença em Portugal, Tailândia, Canadá e Alemanha. Um dos principais centros de produção, e o que se tem vindo a destacar ao longo dos anos, é a Grohe Portugal em Albergaria-a-Velha.

A 28 de Maio de 1998 surge uma nova instalação fabril denominada Grohe Portugal Componentes Sanitários, Lda. Esta empresa começou com um volume de produção médio de um milhão de produtos acabados, e a partir do ano de 2004, as instalações aumentaram, assim como a produção anual que atinge valores superiores a cinco milhões de unidades. Em

2012 a Grohe Portugal (Figura 2) é reconhecida como sendo a melhor unidade do grupo desse ano, facto pelo qual se orgulha.

Com uma equipa de cerca 800 trabalhadores e a tecnologia mais avançada do grupo, esta unidade é responsável por cerca de 25% da produção total da marca. O processo produtivo da Grohe Portugal está organizado em seis fases que ditam todo o processo: a Fundição, Maquinagem, Lixamento, Polimento, Galvânica e, por último, a Montagem.

O regime de funcionamento da fábrica engloba 3 turnos rotativos de 8 horas, fazendo com que esta funcione 24 horas por dia. Normalmente opera 5 dias por semana, de segunda a sexta-feira; no entanto, caso seja necessário (por exemplo para fazer manutenção, ou para produções importantes) o regime pode ser estendido para 6 dias semanais de segunda a sábado.

Toda a informação do grupo Grohe, como modelos de produtos, encomendas e ordens de produção, encontra-se uniformizada e centralizada de modo a proporcionar uma melhor organização e gestão global, sendo para tal usado o *software Systems, Applications and Products in data processing* (SAP).



Figura 2 Fotografia da fábrica em Albergaria-a-Velha.

2.2. PROCESSO PRODUTIVO – ATIVIDADES PRINCIPAIS

Como referido na Secção 2.1, o processo produtivo está organizado em seis fases, a Fundição, Maquinagem, Lixamento, Polimento, Galvânica e Montagem. É através deste processo produtivo que a matéria-prima utilizada (liga metálica à base de latão) é trabalhada de várias maneiras dando origem aos produtos finais, mais propriamente às torneiras.

De forma mais clarificada, encontra-se no Anexo A um esquema do fluxo produtivo contendo as várias atividades do processo produtivo.

2.2.1. FUNDIÇÃO

A Fundição é o processo base de toda a atividade produtiva, pois é nesta fase que a matéria-prima é moldada dando forma ao aspeto quase final do produto.

Existem quatro tipos de liga de latão produzidas para serem respeitadas as diferentes legislações e exigências dos seus clientes; dentre os quatro tipos produzidos, a maior diferença destaca-se nas percentagens de chumbo utilizadas.

A obtenção da liga é feita através da fusão dos vários elementos constituintes que são submetidos a altas temperaturas (cerca de 1100 °C) dentro de um forno (fusão central). A fusão central é constituída por dois fornos, com cerca de 2500 kg de capacidade cada, que funcionam de forma alternada.

Após ser obtida a mistura de um dos fornos, esta é vertida para um recipiente de metais revestido internamente com material refratário, com uma capacidade de cerca de 1000 kg (Figura 3), que é carregado por uma empilhadora responsável pela distribuição da mistura líquida pelos vários fornos presentes nas máquinas dos processos de vazamento (baixa pressão e gravidade), responsáveis por manter a temperatura ideal da liga para que possa ocorrer o vazamento. Esses fornos também estão equipados com balanças que dão a indicação relativa à quantidade de material de que dispõem, informação essa que pode ser observada no posto de controlo.



Figura 3 Transvase de liga de latão.

A liga obtida é apenas um dos constituintes para que se possa produzir a peça; são também necessários os machos de areia e as coquilhas que irão moldar o produto final.

Os machos de areia, apresentados na Figura 4, destinam-se a fazer as cavidades interiores das torneiras, ou seja, este produto apresenta várias formas pelo que o seu aspeto dita o aspeto das cavidades do produto final e, como tal, tem que ser produzido de acordo com o tipo de torneira pretendido. Resultam da mistura de areia, resina, catalisador e silicone, mistura essa que é colocada na caixa de machos com a forma pretendida e colocados na “macharia”. Esta, através da ação do calor (a uma temperatura que ronda os 220 °C), faz com que o catalisador acelere o processo químico da resina permitindo que a mistura ganhe forma sólida.



Figura 4 Machos de areia.

Posteriormente o macho é rebarbado para retirar canais de areia ainda existentes e que não sejam necessários. De seguida são inseridos nas coquilhas, estando o molde formado e pronto para ir para o processo de vazamento.

No processo de vazamento a liga de latão no estado líquido é vertida para o interior do molde, preenchendo o espaço vazio entre a coquilha e o macho, dando origem à peça. O processo de vazamento é distinguido por se efetuar de duas maneiras distintas: baixa pressão ou por gravidade. No processo de baixa pressão a liga é, de certa forma, empurrada para o interior da coquilha por ação da pressão. Por outras palavras, a liga entra na coquilha devido à diferença de pressões existentes, pois no forno onde se encontra a liga líquida está uma determinada pressão, sendo submetida a uma outra pressão proveniente de uma válvula reguladora de pressão, fazendo com que a liga “suba”; o processo é feito de baixo para cima e é automático. No que diz respeito ao processo por gravidade, este é orientado por um operário, que tenha experiência, e que verte a liga para dentro do molde com uma colher revestida de material refratário, processo de cima para baixo. Apesar de ser um processo manual, para algumas peças este processo é o que garante melhores resultados em termos de

qualidade. Atualmente existem seis máquinas de vazamento automático (baixa pressão), denominadas BP (BP1 à BP6), e duas máquinas de vazamento gravítico.

Por fim é obtido o corpo da torneira no seu estado bruto e ainda com o sistema de gitagem. Assim sendo, será posto em prática o corte desses excessos de metal. No processo de corte são separadas as peças do sistema de gitagem. Obtém-se uma peça isolada, todavia ainda não terminada.

Efetuada o corte (Figura 5) ainda é necessário enviar as peças para dentro de uma das duas Granalhadoras existentes (Figura 6), pois no interior dos corpos poderá ainda existir resto de areia do macho. Na Granalhadora as peças são colocadas dentro de uma espécie de tambor que, por rotação, faz revolver as peças (estas batem umas contra as outras), retirando os restos de areia e excessos metálicos existentes.



Figura 5 Corte dos cachos.



Figura 6 Granalhadoras.

Desde do início do processo produtivo que são realizados testes de qualidade do produto. Ainda dentro do forno do centro de fusão são recolhidas três amostras, que são vertidas para um molde específico, dando origem a três copos de metal para posteriormente serem alvo de testes.

O primeiro teste serve para analisar o material através de uma espectroscopia. O espectro é analisado verificando se a composição química da liga respeita os parâmetros estipulados para aquele tipo de liga.

O segundo teste passa por ser um polimento à amostra para verificar se a liga se encontra uniforme. Caso sejam encontradas algumas imperfeições na superfície do material, a liga não se encontra apta para a próxima fase. Este teste é realizado na Fundição pois assim previne o despiste tardio deste tipo de problemas.

Por último, é analisada a forma como a liga flui no molde e cria o copo, retirando assim informações sobre aspectos químicos da peça que poderão traduzir-se em outro tipo de informações de extrema importância, como por exemplo a resistência do material.

2.2.2. MAQUINAGEM

Após concluídos todos os processos da Fundição, as peças são transportadas para a fase de maquinagem. Como o próprio nome indica, esta é a fase onde são maquinadas as peças (passos, furos, roscas, etc.), permitindo assim a montagem dos componentes quando chegar a fase da montagem. Neste departamento o trabalho é todo automatizado, sendo a função dos operários apenas carregar / descarregar as máquinas (Figura 7).



Figura 7 Ferramenta de maquinagem.

Findo este processo, é necessário proceder à lavagem das peças para remover restos de limalha acumulados nestas e efetuar a remoção da emulsão (uso de óleos sintéticos com o propósito de reduzir o atrito e aquecimento do metal), para garantir maior segurança.

De forma a evitar o acréscimo de valor a uma peça que poderá acabar na sucata, são realizados testes de estanquicidade ainda no processo de maquinagem. Estes testes voltam a ser efetuados posteriormente na montagem.

2.2.3. LIXAMENTO / POLIMENTO

No Lixamento/Polimento são realizados os tratamentos da peça ao nível da superfície, sendo um dos trabalhos mais cuidadosos e de extrema importância, pois não podem ser admitidas peças defeituosas. Este departamento enquadra processos automáticos e/ou manuais, quer para o lixamento, quer para o polimento (Figura 8).



Figura 8 Robô de lixamento.

Após as peças serem maquinadas no departamento anterior, estas são submetidas ao lixamento que, numa fase inicial, é executado de forma automática com recurso a robôs, que tratam a maior parte da superfície da peça. No entanto, os braços robóticos não conseguem tratar algumas áreas de difícil acesso (ou por vezes também devido ao desgaste da ferramenta), de tal forma que é necessário enquadrar um processo de lixamento manual (Figura 9) realizado pelos operadores, que ficam responsáveis por aperfeiçoar o lixamento

das peças. De salientar que nem sempre é necessário realizar a atividade manual, pois em peças mais simples os robôs dão conta de todo o lixamento.



Figura 9 Lixamento manual.

Findo o processo de lixamento, todas as peças são submetidas a uma inspeção realizada por um dos postos de inspeção que se encontram espalhados por todo o departamento. Aqui os operários analisam pormenorizadamente todas as peças para que nenhuma com defeito siga para o processo de polimento. Em caso de defeito as peças voltam a ser submetidas ao lixamento manual.

O processo de polimento (Figura 10) tem como objetivo polir as peças para que estas fiquem brilhantes e livres de imperfeições. O sistema de polimento é organizado numa célula de produção, constituída por uma ou duas máquinas e normalmente três operários responsáveis pelo funcionamento dessa célula. Um fica responsável por fazer o carregamento e/ou descarregamento de peças, um outro inspeciona-as constantemente para garantir que não haja qualquer tipo de imperfeição e um último, designado como recuperador, repara

pequenas falhas que possam ter ocorrido em processos anteriores que só ficam visíveis após o polimento.



Figura 10 Unidade de polimento.

A existência de peças mais complexas leva a que resíduos, como limalha, pó de latão e pasta de polir, se acumulem no seu interior, o que provoca a contaminação do tanque de cromagem da próxima fase. A contaminação de um desses tanques pode provocar defeitos na ordem dos 100%, não só na peça com resíduos, como em todas as outras que usem o mesmo tanque. Daí que, após o polimento, todas as peças passam por um processo de lavagem e desgorduramento.

2.2.4. GALVÂNICA

O corpo da torneira é terminado neste departamento, ou seja, a última etapa do processo de tratamento é realizado na Galvânica, dando origem ao aspeto físico final das peças. Findo o processo de lixamento/polimento, as peças chegam à Galvânica e são colocadas em suspensões metálicas, para que depois possam percorrer as diferentes fases do processo de eletrodeposição (Figura 11). Inicialmente as peças são limpas para que depois sejam

submetidas aos banhos de níquel (niquelagem) e de crómio (cromagem), obtendo-se então a cor característica e um aumento da durabilidade da peça, já que este tratamento permite preservar os corpos de oxidações.



Figura 11 Linha de cromagem e niquelagem.

Todo o processo é monitorado para garantir que tudo esteja dentro das especificações, visto que este é um tratamento delicado. Terminado o tratamento, algumas das peças, cujo histórico de rejeição seja alto, são submetidas a novas inspeções visuais. Em caso de conformidade são enviadas ao departamento de montagem para serem terminadas. Por vezes algumas peças apresentam defeitos, o que leva à existência de um processo “inverso” chamado descromagem e desniquelagem, apresentado na Figura 12, que é realizado caso as peças defeituosas necessitem de voltar a um dos departamentos anteriores ou mesmo para a sucata.



Figura 12 Linha de descromagem e desniquelagem.

2.2.5. MONTAGEM

O departamento de montagem é o maior departamento da fábrica, assim como o que contém mais funcionários. Este departamento é responsável por todas as operações restantes, nomeadamente a montagem dos restantes componentes, o seu embalamento, paletização e, por fim, a sua expedição.

A maior parte das linhas de montagem foram concebidas com um *layout* na forma de “U” (Figura 13), sendo constituídas por três bancadas, normalmente albergando dois ou três operários. Ao serem rececionadas nas bancadas, as peças passam por quatro fases distintas. No entanto, antes de se iniciarem essas fases, todas as peças são submetidas novamente a inspeções, algumas mais breves (caso já tenham sido inspecionadas na Galvânica), e outras mais rigorosas caso a peça seja proveniente do exterior (a fábrica não tem capacidade para produzir peças suficientes de acordo com a procura e necessidade exigidas). Aprovada a peça, dá-se então início à primeira fase, que consiste em montar todos os componentes que ainda estão em falta nesta.



Figura 13 Exemplo de uma linha de montagem.

Na segunda etapa são realizados um ou dois testes, de água e ar, para verificar se a peça não apresenta fugas. Caso a peça apresente algum constituinte termostático é realizado um teste que comprove o funcionamento do mesmo. Em caso de não conformidade de algum dos componentes é de imediato efetuada a troca do mesmo e só em casos em que a gravidade do defeito seja maior é que a peça volta para a Galvânica para ser desniquelada / descromada para que depois seja enviada para a sucata.

Se a peça passar nos testes anteriores, passa para a próxima fase onde são colocados os componentes finais, como os manípulos e alavancas, onde é realizada uma limpeza e uma inspeção visual. Se a peça for aprovada segue para a etapa de embalagem e posteriormente paletização.

Estando pronta para a expedição, a paleta é submetida a mais um controlo, com vista a preservar os padrões de qualidade. Esta nova inspeção é realizada por uma equipa denominada Firewall, que verifica se as peças estão dentro das especificações, quer a nível físico da peça, quer no conjunto de componentes e instruções agregadas à montagem final.

O processo de inspeção desta equipa passa basicamente por escolher três produtos terminados de uma paleta completa e verificar se os componentes usados são os corretos, se há defeitos na peça, a correta colocação na caixa, entre outras, dando origem a vários veredictos:

- **três peças conformes (OK):** a paleta está correta e pode ser selada, devidamente etiquetada e colocada no camião;
- **pelo menos uma peça não conforme (NOK):** são recolhidas dez peças para serem inspecionadas. Caso o número de peças não conformes seja inferior a quatro, essas peças são corrigidas e a paleta é validada. Se o número de peças não conformes é igual ou superior a quatro, a paleta é rejeitada e será verificada na sua totalidade corrigindo tudo o que estiver incorreto.

2.3. PROCESSO PRODUTIVO – DEPARTAMENTOS DE SUPORTE

No que diz respeito à restante estrutura organizacional da Grohe, existem outros departamentos que contribuem para as atividades principais do processo produtivo, dando apoio às outras divisões, disponibilizando-lhes não só recursos como também todas as condições para a obtenção do produto final. Na Figura 14 são apresentados os vários departamentos que constituem a organização.

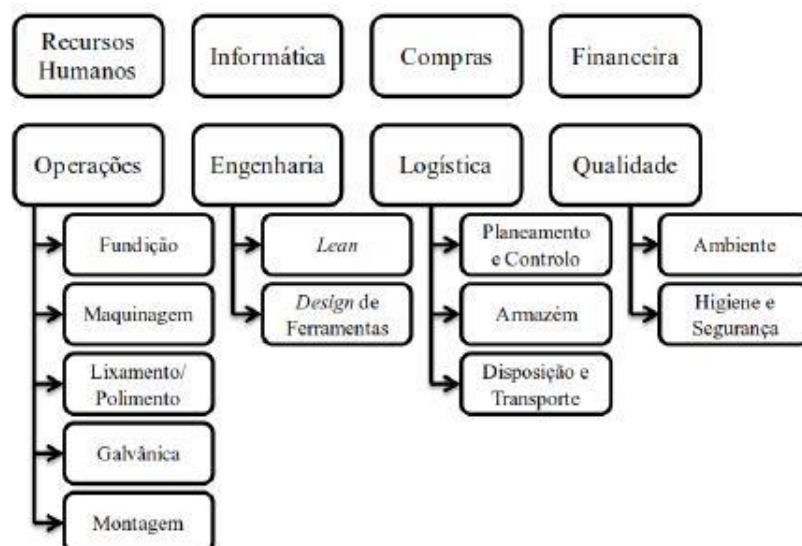


Figura 14 Esquema dos departamentos da Grohe.

De seguida é efetuada uma breve explicação de cada um dos constituintes da organização.

2.3.1. RECURSOS HUMANOS

Este departamento tem por finalidade selecionar, gerir e nortear os colaboradores na direção dos objetivos e metas da empresa. A área de recursos humanos também lida com estratégias de recursos humanos, relações sindicais, relações de trabalho e técnicas afetas à função tais como recrutamento, seleção, treino, planos de cargos e salários, avaliação de desempenho, incentivos e remuneração.

2.3.2. INFORMÁTICA

O departamento de informática é responsável pela manutenção de todo o material informático da organização, e tem um papel de grande importância pois está encarregue da gestão da plataforma SAP em que assenta a organização da empresa.

Por toda a zona fabril estão colocados vários “*PC Panels*” que integram o sistema *E-plant* que comunica com o SAP. Através deste sistema os operários registam-se nos diferentes postos de trabalho e registam também os produtos de entrada/saída do departamento, defeitos de peças e material rejeitado.

2.3.3. COMPRAS

Qualquer atividade industrial necessita de matérias-primas, componentes, equipamentos e serviços que possa operar. O departamento de compras é responsável pelo estabelecimento do fluxo dos materiais na empresa, pelo seguimento junto ao fornecedor, e pela agilização da entrega. Prazos de entrega não cumpridos podem criar sérias perturbações para os departamentos de produção.

2.3.4. FINANCEIRA

A seu cargo encontra-se o planeamento financeiro da fábrica. Está ainda incluído neste departamento a contabilidade associada à fábrica.

2.3.5. ENGENHARIA

É um dos grandes pilares de sustentação da Grohe Portugal. Das diversas funções que este departamento tem a seu encargo, podem-se destacar algumas com maior relevância:

- estudo de métodos e tempos;
- engenharia do produto;
- desenvolvimento de novas ferramentas;
- investimentos e controlo de custos.

Também dispõe de uma equipa de *Lean Manufacturing*, que constantemente se rege por uma política de melhoria contínua na fábrica.

2.3.6. LOGÍSTICA

O departamento de Logística é responsável por prover recursos, equipamentos e informações para a execução de todas as atividades de uma empresa, sendo as que merecem maior ênfase o planeamento e controlo da produção, o armazenamento e as questões relacionadas com o transporte de mercadorias.

2.3.7. QUALIDADE

Sendo este um dos valores pela qual a Grohe se rege e pretende que os seus produtos sejam reconhecidos, cabe ao departamento da qualidade a responsabilidade de garantir que todos os produtos e matérias são manuseados e montados com recurso às ferramentas e técnicas mais apropriadas.

Além da qualidade, estão ainda incluídos neste departamento as secções do ambiente e segurança no trabalho.

3. SISTEMAS SCADA

Tendo o projeto como objetivo a criação de um sistema de controlo e de uma Interface Homem Máquina, neste capítulo são apresentados os sistemas Supervisory Control And Data Acquisition (SCADA), que foram estudados de modo a ter uma perceção, tanto a nível do utilizador, como a nível do programador, de como um sistema baseado em SCADA funciona, fazendo uma abordagem a aspetos históricos, arquitetónicos e funcionais.

3.1. INTRODUÇÃO

Nos últimos anos tem-se verificado uma utilização cada vez mais frequente e generalizada de meios computacionais e informáticos nas empresas e indústrias dos mais diversos ramos de atividade. A utilização destes meios permite melhorar e tornar mais eficiente a execução de tarefas isoladas, quer ao nível dos processos produtivos, quer ao nível da gestão. Neste contexto, os sistemas SCADA desempenham um papel fundamental no que respeita à recolha de informação e ao controlo dos processos industriais. Assim, deve-se estruturar a instalação industrial e organizar os seus elementos constituintes (unidades de aquisição de dados, Controladores Lógico Programáveis (PLC), sistemas de supervisão existentes, entre outros), de modo a facilitar toda a monitorização e/ou controlo exigidos.

Os sistemas SCADA são aplicações de *software* que além de servirem como interface homem/máquina, como *softwares Human Machine Interface (HMI)* ou *Man Machine*

Interface (MMI), podem também efetuar controlo e distribuir informações entre estações através da rede, com bom desempenho e segurança. São *softwares* robustos e confiáveis para aplicações de grande porte e para aplicações distribuídas em várias estações de trabalho [1].

Estes sistemas abrangem aplicações cada vez mais diversificadas, podendo estar presentes em diversas áreas, revelando-se de crucial importância na estrutura de gestão das empresas, facto pelo qual deixaram de ser vistos como meras ferramentas operacionais, ou de engenharia, e passaram a ser considerados como uma importante fonte de informação (Figura 15).



Figura 15 Exemplo de uma estação de controlo de um sistema SCADA [2].

São implementados no topo de sistemas de controlo, formando centrais de controlo que comunicam com equipamentos de campo através de redes e unidades de terminais remotos, as *Remote Terminal Units* (RTU) e/ou PLC, propiciando uma visão geral de toda a área de produção [1].

3.2. ORIGEM

O controlo de supervisão surgiu pela primeira vez em sistemas de energia elétrica, devido à necessidade de operar equipamentos de subestação remota sem recorrer ao envio de pessoal ao local. Por volta de 1940 apenas era usado um par de fios para estabelecer a ligação entre equipamentos, daí o aproveitamento do potencial da multiplexação num par de linhas que passou a ser usado, baseado nas ideias de *Magnetic Stepping Switch* desenvolvido pelas companhias telefónicas em meados dos anos 30. Tendo como um grande problema a segurança, foi adotado o procedimento “*select-check-operate*”, onde o operador esperava pela confirmação do dispositivo antes de começar a utilizá-lo. Aproveitando mais “sugestões” dos sistemas de retransmissão dos telefones e os seus esquemas de codificação, a Westinghouse e a North Electric Company desenvolveram o controlo de supervisão VISICODE [3].

A General Electric e a Control Corporation também desenvolveram os seus próprios programas de controlo de supervisão independentes. Estes foram utilizados em oleodutos, companhias de gás e até em aeroportos, para as luzes da pista de aterragem. Estes sistemas tornaram-se populares entre 1950 e 1965 e por volta de 1960 surge a telemetria, desenvolvida para fins de monitorização. Com o advento da tecnologia computadorizada de baixo custo, *software* e computadores permitiram a realização das funções anteriormente realizadas através do painel de instrumentos e por telemetria [3].

O primeiro “empurrão” foi dado pelos computadores de 8 e 16 bit, também denominados minicomputadores. O segundo foram os microprocessadores, que apenas surgiram vários anos depois. Os computadores oferecem flexibilidade na programação e comunicação com a aquisição de dados de campo que eram antigamente adquiridos através de equipamentos *hard wired* [3].

Em suma, os sistemas SCADA permitiam informar periodicamente o estado do processo industrial, monitorizando sinais representativos de medidas e estados de dispositivos, através de um painel de lâmpadas e indicadores, sem qualquer interface com o operador. Com a ascensão tecnológica, os computadores passaram a ter um papel fulcral de gestão na recolha e tratamento de dados, tornando possível a sua visualização num ecrã e a geração de comandos de programação para execução de funções de controlo complexas.

Atualmente os sistemas SCADA utilizam tecnologias de comunicação e computação cada vez mais avançadas para automatizar a monitoração e controlo dos processos industriais, efetuando a recolha de dados em ambientes complexos, eventualmente dispersos geograficamente, e a respetiva apresentação de modo amigável para o utilizador, com recurso a HMI.

3.3. ARQUITETURA

3.3.1. ARQUITETURA DE *HARDWARE*

Uma arquitetura típica de um sistema de supervisão, ou sistema SCADA, é composta por estações do tipo cliente e estações do tipo servidor de dados, conforme ilustrado na Figura 16. Uma estação do tipo servidor de dados (*SCADA server*) comunica com equipamentos de campo através dos controladores, como por exemplo PLC, que podem ser conectados diretamente ou por redes de campo (*fieldbus*). Através destas estações pode-se armazenar e partilhar informações do chão de fábrica para as estações do tipo cliente através de redes *Local Area Network/Wide Area Network* (LAN/WAN). Assim, informações da estação do tipo servidor podem ser acedidas de qualquer ponto da rede através de aplicações de *software* especiais (*SCADA client*), ou através de navegadores de páginas *Hyper Text Markup Language* (HTML), tendo-se, neste último caso, algumas limitações de recursos. Adicionalmente, respeitando as normas de segurança da rede corporativa (*firewall*), pode-se até mesmo disponibilizar estas informações na Internet [1].

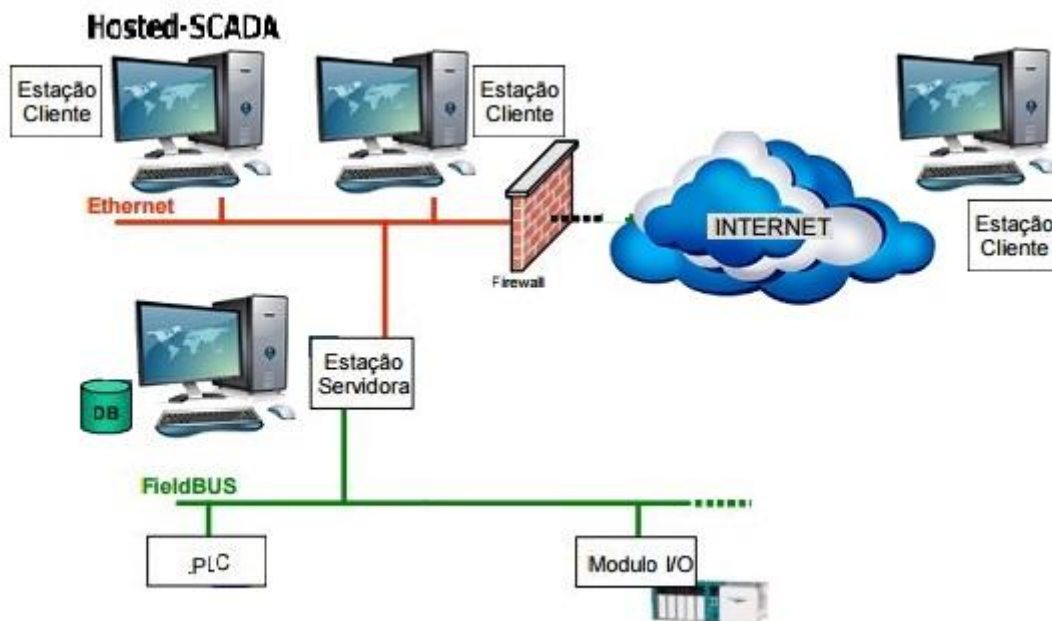


Figura 16 Arquitetura de um sistema SCADA.

3.3.2. ARQUITETURA DE SOFTWARE

As várias gamas de *software* existentes realizam muitas tarefas e são baseados em *Real Time DataBase* (RTDB) localizada num ou mais servidores. Os servidores são responsáveis pela aquisição de dados e manuseamento (por exemplo, verificação de alarmes, cálculos, arquivo, etc.) dum conjunto de parâmetros, normalmente aqueles a que estão ligados.

De acordo com Figura 17, a arquitetura de *software* de um sistema SCADA pode ser dividida da seguinte maneira:

- SCADA *Server*: responsável pela comunicação e troca de dados com equipamentos de campo, compartilhamento, processamento e armazenamento de informações, execução de tarefas em tempo real, gestão de eventos e de alarmes, comunicação com aplicações locais (por exemplo, EXCEL, *Structured Query Language* (SQL), ACCESS, etc.);
- SCADA *Client*: responsável pela interface homem-máquina, geração de gráficos de tendência, apresentação de alarmes e *logs*, e integração com aplicações locais;
- SCADA *Development Environment*: ambiente de desenvolvimento dos módulos de SCADA *Server* e *Client*.

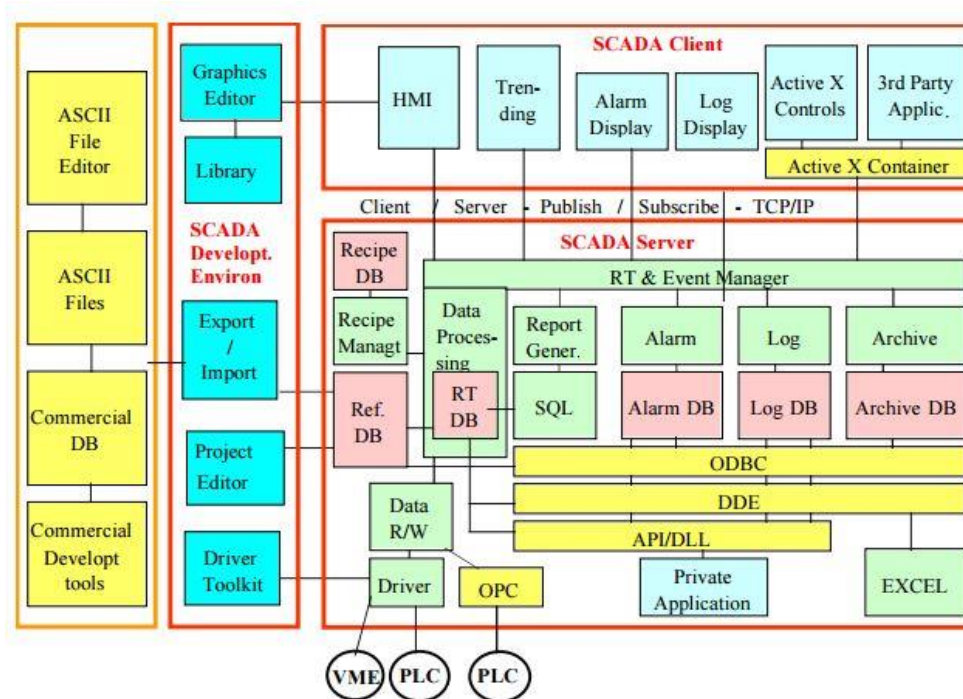


Figura 17 Estruturação do sistema SCADA.

3.4. COMPONENTES

Os componentes físicos de um sistema de supervisão podem ser resumidos, de forma simplificada, em:

- Equipamentos de campo:
 - Sensores;
 - Controladores;
 - Atuadores;
 - Módulos de aquisição de sinais (digitais e/ou analógicos);
- Sistema de comunicação;
- Estação SCADA do tipo Servidor (SCADA *Server*);
- Estação SCADA do tipo Cliente (SCADA *Client*).

3.4.1. EQUIPAMENTOS DE CAMPO

No que diz respeito aos equipamentos de campo, estes são responsáveis pela leitura de grandezas físicas, tomada de decisões e atuações no processo, ou seja são equipamentos de interface direta com a área de produção e das máquinas. Como referido anteriormente, dele fazem parte os sensores, os módulos de aquisição de sinais (RTU), controladores (como por exemplo os PLC) e os atuadores (equipamentos como motores, inversores, contactores, relés, entre outros), como é apresentado na Figura 18.

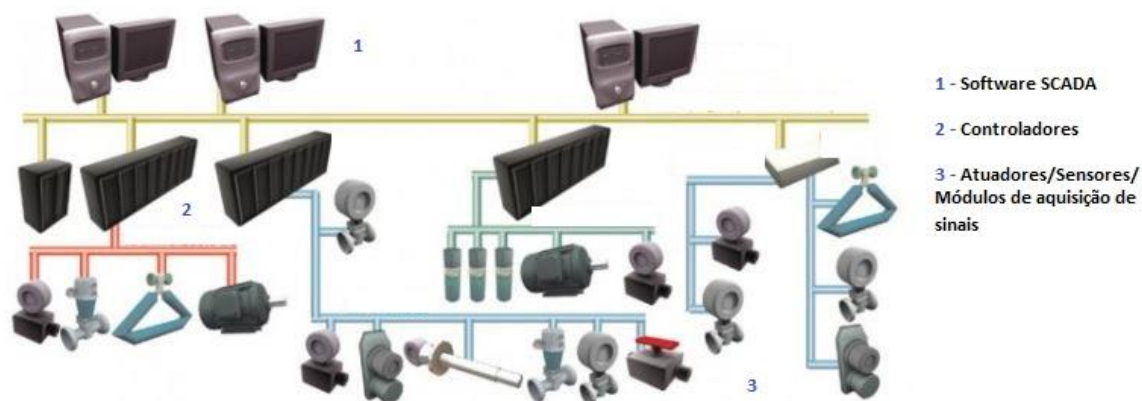


Figura 18 Sistema SCADA a comunicar com equipamentos de campo [4].

Os sensores são equipamentos que produzem sinais de tensão ou corrente, apresentando alguma propriedade física medida (temperatura, força, pressão, velocidade, etc.). Fornecem informações necessárias para a percepção do que ocorre nas áreas de produção e/ou máquinas. Fazendo uma analogia ao corpo humano, os sensores formam os “olhos e ouvidos” de um sistema SCADA. Geralmente, os sinais de sensores requerem algum tipo de processamento antes de serem utilizados. Alguns padrões industriais de sinal de saída analógico de sensores são 0-5 V ou 0-10 V para curtas distâncias, e 4-20 mA ou 0-20 mA para longas distâncias com controladores. Saídas digitais binárias são normalmente empregues por sensores para a indicação de estados discretos de equipamentos, tais como nível “1” para indicar ligado e nível “0” para indicar desligado [5].

Adicionalmente são utilizados os atuadores elétricos de válvulas, *drivers* e inversores de frequência de motores, atuadores pneumáticos e hidráulicos, entre muitos outros, que em analogia ao corpo humano representam as “mãos” de um sistema SCADA.

Contudo, antes que qualquer automação ou monitorização remota possa ser alcançada, a informação que é passada para, e de, um equipamento de campo deve ser convertida para um formato que seja perceptível para o sistema SCADA. Assim existem os módulos de

aquisição de sinais conhecidos como *Remote Terminal Units* ou *Remote Telemetry Units* (RTU). Estes são utilizados para converter sinais elétricos, recebidos pelos equipamentos de campo, em “linguagem” usada para transmitir dados através de um canal de transmissão.

Tanto as instruções para a automação dos equipamentos de campo, como a lógica para controlo, são geralmente armazenadas localmente. Tal facto deve-se à limitação típica da taxa de comunicação entre uma plataforma SCADA e os equipamentos de campo. Assim essas instruções encontram-se nos PLC, que antigamente eram fisicamente separados dos RTU. Os PLC são equipamentos de controlo que tomam decisões baseando-se em sinais de outro *hardware* e/ou transdutores (botões, chaves de fim de curso, sensores, outros PLC, computadores, etc.) e em lógicas previamente programadas. Hoje em dia, os PLC já apresentam RTU incorporadas. São disponibilizados em módulos, apresentando como principais constituintes:

- Módulo CPU;
- Módulo de entradas e saídas digitais e/ou analógicas;
- Módulo de interface de comunicação com redes de campo (RTU);
- Fonte de alimentação.

Os PLC surgiram pela primeira vez na automação industrial e podem ser utilizados em separado ou em conjunto com os sistemas SCADA. A necessidade de ter PLC conectados a uma rede de comunicação não era um requisito para estas aplicações, pois apenas eram usados para substituírem sistemas tradicionais de lógica a relés ou controlo pneumático [6].

Os sistemas SCADA, por outro lado, advêm das aplicações remotas, cujo objetivo era obter informações básicas de fontes remotas. Assim, os RTU utilizados nestes sistemas não tinham a necessidade de efetuarem controlo pois o algoritmo de controlo local era mantido em lógicas de chaveamento a relés, ou por PLC. No entanto, com a necessidade de alterações do algoritmo e/ou parâmetros de controlo através de sinais remotos, surgiu a necessidade da utilização de equipamentos capazes de executarem tanto tarefas de controlo, como de comunicações remotas. Por esta razão é que atualmente se verifica uma fusão das funcionalidades dos PLC e RTU. Tornou-se comum a existência de PLC com a capacidade de realizar comunicações remotas, bem como RTU capazes de executarem funções de

controle. Através desses desenvolvimentos, a diferença entre um PLC e um RTU tornou-se mínima.

3.4.2. SISTEMAS DE COMUNICAÇÃO

Na Figura 19 apresenta-se em destaque (a cores) o sistema de comunicação, dentre todos os componentes típicos de um sistema SCADA.

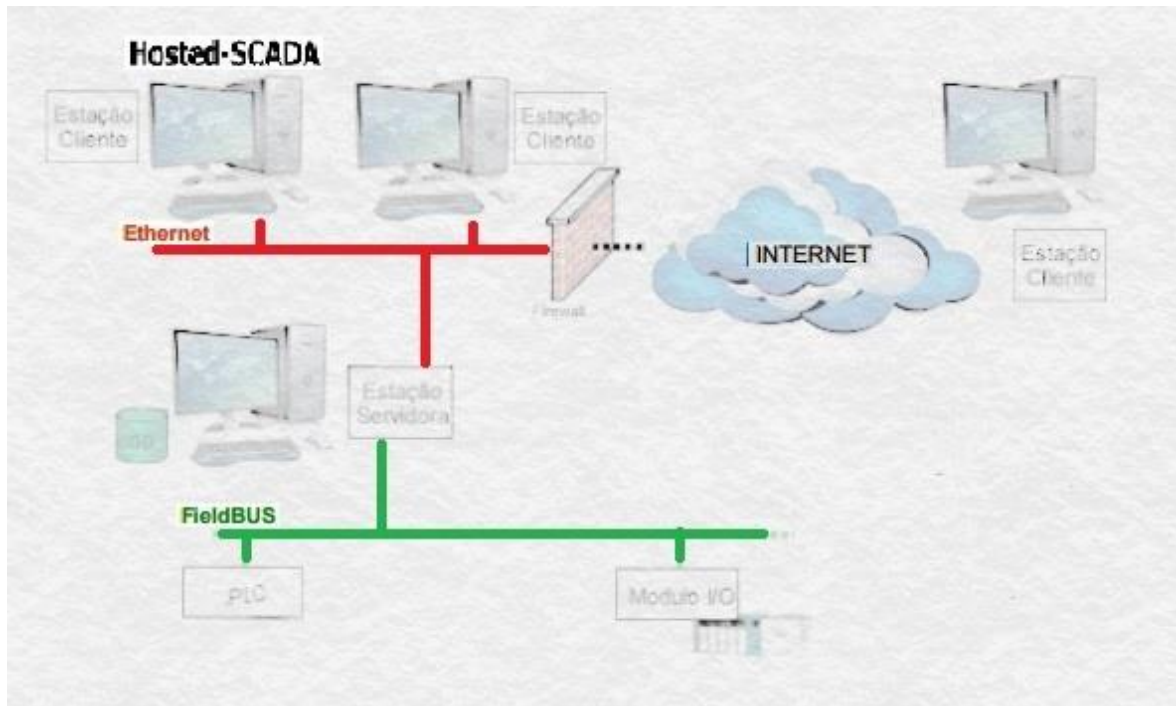


Figura 19 Sistema de comunicação para interface SCADA.

A troca de dados entre os equipamentos de campo e o *software* são da responsabilidade do sistema de comunicação, que apresenta como principais funções [7][8][9]:

- Comunicar: permite troca de dados em tempo real com os atuadores, através de uma variedade de métodos, tais como: taxas de *refresh* em *Inputs/Outputs* selecionados, compartilhamento de *inputs*, *interlocking* entre controladores, etc.;
- Configurar: permite configurar todos os equipamentos que se encontram na rede, modificando parâmetros, somente com um clique do rato ou através de um controlador lógico;

- Recolha de dados: possibilita a recolha de dados de equipamentos de campo, permitindo a visualização em *displays* de HMI, cálculos de tendências e análises, administração de manutenção e solução de problemas.

Atualmente existe uma grande quantidade de tecnologias de conceção aberta ou proprietária disponível no mercado em situação consolidada, sendo especificamente desenvolvidas para atender aos requisitos de cada um dos cinco níveis de dispositivos e equipamentos característicos de um sistema de automação, apresentados na Figura 20.

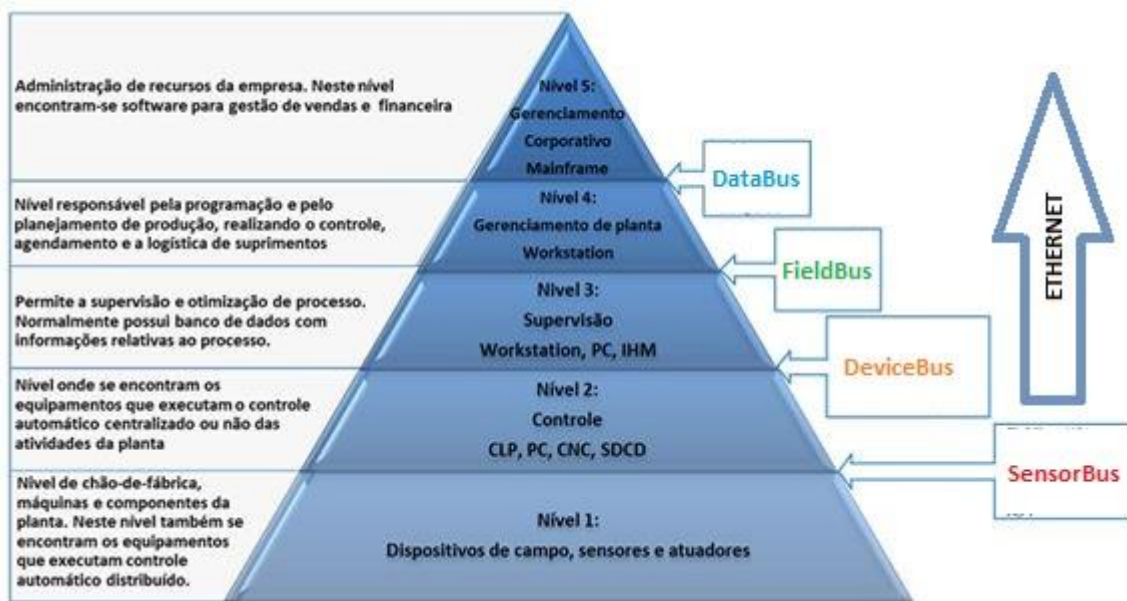


Figura 20 Níveis da pirâmide de automação [10].

Assim são destacadas quatro classes de redes de comunicação, apresentadas na Figura 20 [11]:

- **DataBus**: responsável pela comunicação entre computadores, com capacidade de manipular grandes quantidades de informação em tempo não crítico (ex.: Ethernet *Transmission Control Protocol/Internet Protocol* (TCP/IP), *Fiber Distributed Data Interface* (FDDI), etc.);
- **Fieldbus**: aplicado para comunicações entre unidades inteligentes, por ser determinístico e com alto desempenho (ex.: Fieldbus Foundation, Fieldbus WorldFIP, Modbus, Profibus-FMS, Profibus-PA, etc.);

- **DeviceBus:** apresenta tempos de resposta curtos (alguns milissegundos) e é determinístico, específico para dispositivos de controlo e seus periféricos. (ex.: DeviceNet, Interbus-S, Profibus-DP, etc.);
- **SensorBus:** apresenta tempos de resposta muitíssimo curtos (entre 1 e 10 ms) e determinísticos, sendo destinado à comunicação ao nível dos sensores e atuadores (ex.: AS-i, Seriplex, etc.).

Normalmente utilizam-se redes do tipo FieldBus para aplicações de supervisão, que realizam a recolha de dados e controlo de processos.

Com o aumento de máquinas e sistemas, aumentou a necessidade de redes de cabos de par trançado devido ao grande número de canais de *Inputs/Outputs* (I/O). Como consequência dessa crescente utilização, ocorreu o aumento de esforços para a sua configuração, instalação, inicialização e manutenção. Além disso, cabos de par trançado possuem custos elevados, principalmente os utilizados para a transmissão de dados analógicos [12].

Como solução para este problema, surge em meados do final da década de 70 a primeira rede industrial de comunicações, denominada Fieldbus. Esta rede, que com apenas um único cabo de comunicação se conecta a vários I/O, pode ainda comunicar com vários outros níveis de automação (desde o nível 1 até ao nível 5, dependendo do tipo de rede). Assim, comparativamente às redes de cabos de par trançado, o Fieldbus apresenta vantagens no que diz respeito à redução de cabos, o que reduz custos e tempo de planeamento, instalação, configuração, manutenção, redução de terminais e gabinetes de controlo, melhor confiabilidade, com maiores proteções contra falhas em sinais analógicos e possibilidade de comunicação com vários níveis de automação.

Normalmente são usadas redes de comunicação Fieldbus em ambiente industrial, com meio físico baseado em cabos. No entanto, tal prática torna-se cara para sistemas que cobrem vastas áreas geográficas uma vez que o custo dos cabos é elevado, assim como o suporte e as instalações. O uso de linhas telefónicas é uma solução económica para sistemas com larga cobertura. A linha telefónica pode ser alugada ou discada (*dialup*). As alugadas são utilizadas para sistemas que requerem conexões *online* com as estações remotas; no entanto, este método é de custo elevado uma vez que é alocada indefinidamente uma linha por estação. As linhas discadas podem ser utilizadas em sistemas que requerem atualizações de informação em intervalos regulares. Assim uma estação pode ligar para um número

particular de uma estação remota, receber informações e enviar comandos. Quando a alternativa da linha telefónica não se encontra disponível, uma solução será fazer uso da comunicação via sistemas de rádio [6].

As comunicações entre equipamentos estão sujeitas a protocolos que devem ser respeitados e tecnologias a serem utilizadas. Assim os principais protocolos e tecnologias para troca de dados entre *hardware* de chão de fábrica e *software* são [13]:

- **Component Object Model (COM):** plataforma da Microsoft para componentes de *software* que é utilizada para permitir a comunicação entre processos e a criação dinâmica de objetos em qualquer linguagem de programação que suporte a tecnologia;
- **Distributed Component Object Model (DCOM):** é uma extensão do COM para a comunicação entre objetos em sistemas distribuídos. Disponibiliza objetos remotos e locais. É o COM para aplicações em rede;
- **Dynamic Data Exchange (DDE):** nasceu com o Windows 3.0 e é um protocolo padrão de troca de dados entre aplicações Windows. Utilizando o *driver* DDE, é possível recolher dados do chão de fábrica em Excel, por exemplo. É um protocolo lento, não confiável e foi desenvolvido para trabalhar numa máquina e modificado para pequenas aplicações em rede (NetDDE). Possui variações como FastDDE e AdvancedDDE que possuem os mesmos problemas do DDE original – não foi desenvolvido para aplicações que necessitem de integridade de dados e alta velocidade. Deverá ser totalmente substituído pelo *Open Protocol Communication* (OPC);
- **Network Dynamic Data Exchange (NetDDE):** protocolo para troca dinâmica de dados entre quaisquer redes suportadas por Windows, como TCP/IP, NET-BEUI, *Internetwork Packet Exchange/Sequenced Packet Exchange* (IPX/SPX9) e entre aplicações Windows, como Excel, Word, etc.;
- **Object Linking and Embedding (OLE):** é um sistema de objetos distribuídos e um protocolo desenvolvido pela Microsoft que permite a um editor disponibilizar parte de um documento para outro editor, e então reimportá-lo. Apesar de ser mais flexível, eficiente, e robusto do que o *Dynamic Data Exchange* (DDE), não possui o desempenho necessário para automação;

- **OLE for Process Control (OPC):** protocolo padrão de interfaces, propriedades e métodos para o uso de aplicações em controlo de processos e automação. Baseado nas tecnologias OLE e DCOM, o OPC permite uma interface comum de comunicação com diversos dispositivos de controlo de processos, indiferentemente do *software* de controlo ou dispositivos no processo;
- **Dynamic Linking Libraries (DLL):** é a implementação feita pela Microsoft para o conceito de bibliotecas compartilhadas nos sistemas operativos Windows e OS/2. Possui velocidade e integridade conforme programação das rotinas de acesso, sendo normalmente utilizada por aplicações na mesma máquina.

3.4.3. ESTAÇÃO SCADA DO TIPO SERVIDOR

As estações SCADA do tipo servidor fornecem as interfaces homem-máquina para os sistemas SCADA. Os computadores processam as informações enviadas e recebidas do chão de fábrica, apresentando-as de forma adequada para que os utilizadores possam trabalhar. Podem ser exibidos diagramas de todo o processo para facilitar a identificação com o sistema real. Cada ponto de I/O de uma unidade remota pode ser mostrado com uma correspondente representação gráfica.

Antigamente os sistemas SCADA eram fornecidos com *hardware*, sistemas operativos e *software* incompatíveis com sistemas de outros fornecedores. A expansão de um sistema só era possível através da contratação do fornecedor original. As plataformas de computadores eram caracteristicamente baseadas em arquitetura UNIX, e eram fisicamente removidas das redes de computadores dos outros domínios da companhia. Por oposição, atualmente muitos sistemas SCADA estão disponíveis para toda a rede de comunicação das empresas, e são implementados em servidores comumente empregues em aplicações de escritório, abrangendo as possibilidades de ampliação e integração com outros sistemas [6].

3.4.4. ESTAÇÕES SCADA DO TIPO CLIENTE

As estações do tipo cliente estão ligadas em rede com a estação do tipo servidor, para que ocorra o envio e acesso de informações. A troca de informações entre as estações pode ser realizada através de *software* especial (SCADA *Client*), ou através de navegadores de páginas HTML (*browsers*).

3.5. FUNCIONALIDADES

A funcionalidade de um sistema SCADA é similar para sistemas *Distributed Control System* (DCS) encontrados em soluções de automação, com exceção de que os sistemas SCADA contêm geralmente aquisição de dados e controlo de equipamentos geograficamente distribuídos, diferenciando-se dos sistemas DCS [14].

Resumidamente, as suas principais funcionalidades são:

- Aquisição de dados das estações remotas de I/O e controladores via *drivers* de comunicação e/ou interfaces OPC e NetDDE;
- Controlo automático em malha fechada ou controlo via atuação do operador;
- *Displays* de processo: representações gráficas dos equipamentos de campo atualizadas automaticamente para retratar uma situação atual, e gráficos em geral, por exemplo de tendências, para acompanhamento de variáveis do processo;
- *Standby*: sistemas duplicados em espera com a capacidade de tomar o controlo no caso de falhas no sistema principal;
- Integração: permite a comunicação com sistemas corporativos, pois suporta *Structured Query Language/Open Database Connectivity* (SQL/ODBC). Integração com aplicações locais via interfaces OPC, DDE, ActiveX, etc.;
- Segurança: controlo de acesso por utilizador aos vários componentes do sistema;
- Gestão da rede: monitoração das condições de comunicação da rede;
- Automação: permite automatizar tarefas via *scripts*, iniciando-as através de eventos;
- Outros: possui base de dados proprietária, ferramentas de *log* de dados, relatórios formatados, históricos, receitas, alarmes e Controlo Estático de Processos (CEP).

Na Figura 21 é apresentado um exemplo de uma aplicação que corre um sistema SCADA, programado de acordo com as necessidades do sistema a controlar e/ou monitorizar.

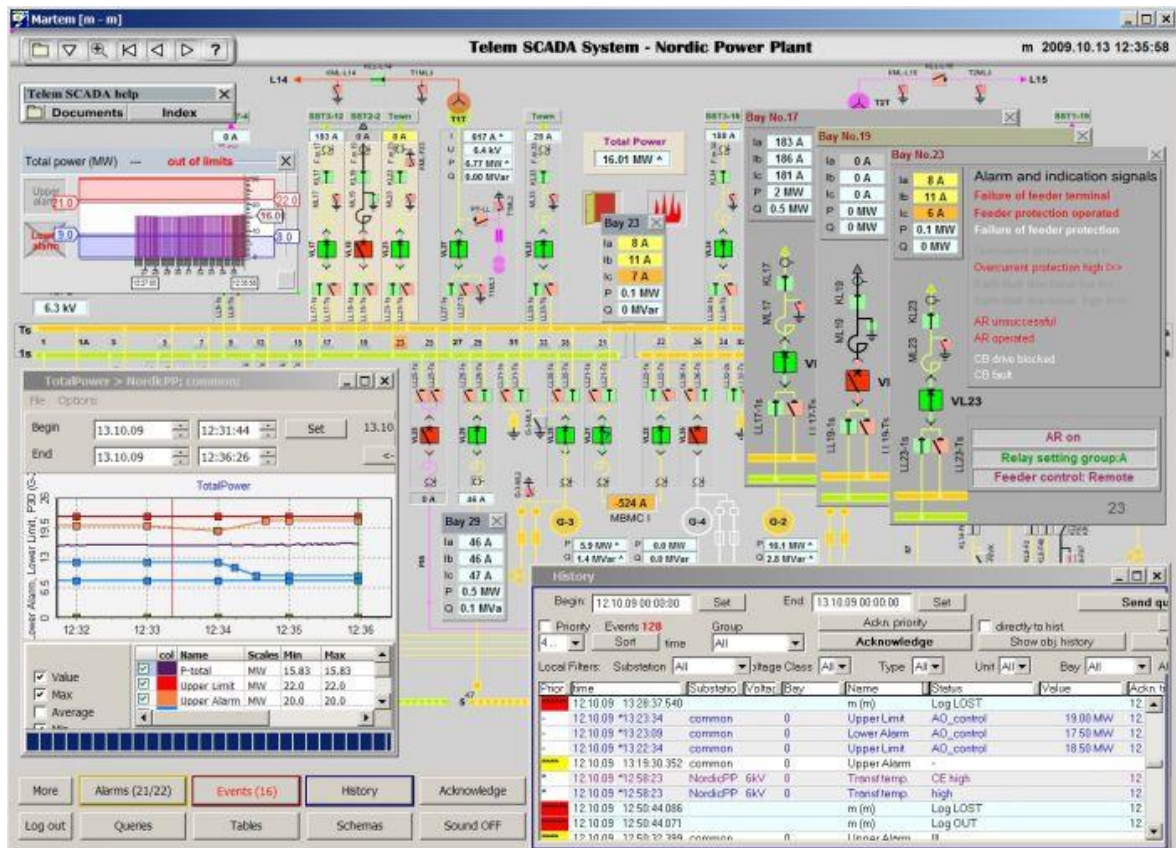


Figura 21 Exemplo de um programa de supervisão [15].

3.6. INTEGRAÇÃO DE SISTEMAS

O ambiente fabril pode ser composto por máquinas de diferentes configurações, com maior ou menor dificuldade de integração com o sistema SCADA e, como consequência, de integração com os sistemas corporativos e de gestão da produção. Máquinas com ambiente *Personal Computer* (PC) facilitam esta integração por permitirem a captura de dados utilizando protocolos padrão (por exemplo, OPC, DDE e NetDEE) já disponíveis nos sistemas SCADA, ou utilizando protocolos proprietários, sendo neste caso necessária a compra ou o desenvolvimento de *drivers* para a comunicação. Quanto às máquinas que não apresentam ambiente PC (ambientes heterogêneos), quando o fabricante não disponibiliza recursos e *drivers* de comunicação, torna-se difícil a integração das mesmas com os sistemas SCADA, exigindo-se em muitos casos a utilização de módulos digitais e/ou analógicos para a aquisição de sinais (*bits*, *words*, sinais analógicos de sensores, etc.) que podem ser convertidos em informações, como por exemplo uma combinação de *bits* que indicam uma determinada situação da máquina (informação). Na Figura 22 tem-se um esquema das possíveis configurações de máquinas de chão de fábrica: máquinas com ambiente PC e de

arquitetura aberta (ou parcialmente aberta), máquinas com ambiente PC e de arquitetura totalmente fechada e máquinas que não possuem ambiente PC.

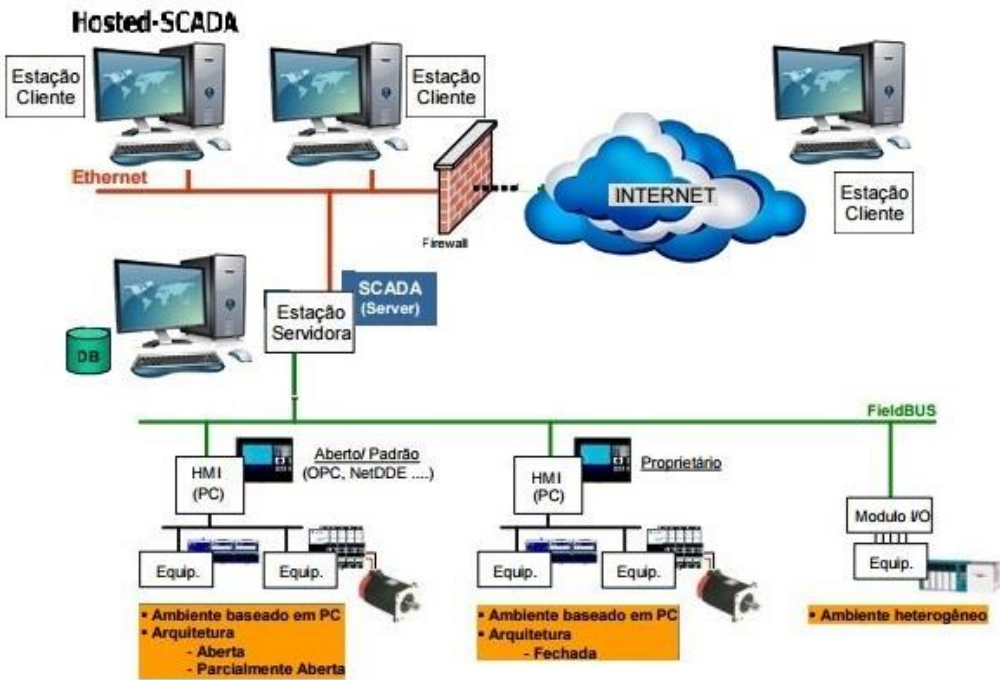


Figura 22 Configurações das máquinas de chão de fábrica.

4. CONTROLADORES LÓGICOS PROGRAMÁVEIS

Pretende-se neste capítulo fazer uma breve introdução aos PLC, de uma forma mais detalhada. Numa fase inicial do estágio revelou-se necessário o estudo do funcionamento e da programação de um controlador lógico programável, visto que o projeto em causa é controlado por um PLC e para a sua conclusão é necessário compreender e adaptar o código já existente, daí a importância da compreensão desta temática.

4.1. CONTROLADOR LÓGICO PROGRAMÁVEL

Os Controladores Lógicos Programáveis, do original inglês *Programmable Logic Controller* (PLC), são dos controladores mais utilizados na indústria, graças à sua grande flexibilidade de programação e de *hardware*.

Segundo a *National Electrical Manufacturers Association* (NEMA), um controlador lógico programável é um aparelho eletrónico digital que utiliza uma memória programável para o armazenamento interno de instruções para implementações específicas, tais como lógica, sequências, temporização, contagem e aritmética, para controlar, através de módulos de entradas e saídas, vários tipos de máquinas e/ou processos [16].

Os PLC foram desenvolvidos na década de 60, com a finalidade de substituir painéis de relés que eram utilizados nas indústrias para executar controlos, baseados em lógicas combinatórias e sequenciais. No passado os programas eram muito limitados e apenas

dispunham de uma função: selecionar as operações das máquinas e/ou processos. Além de apresentar uma operacionalidade muito reduzida, também eram constatados outros tipos de problemas como o alto consumo de energia, difícil manutenção e modificações de comandos/instruções, entre outros.

Com a industrialização da eletrônica ocorreu uma grande redução nos custos, aumentando a flexibilidade e, ao mesmo tempo, permitindo a utilização de comandos eletrônicos em grande escala.

No entanto, mediante as dificuldades encontradas na época para se implementarem controladores lógicos industriais, David Emmett e William Stone (da General Motors Corporation) prepararam as especificações detalhadas das necessidades da indústria e solicitaram aos fabricantes de instrumentos de controlo que desenvolvessem um novo tipo de controlador lógico que incorporasse determinadas características, características essas que tinham que permitir uma fácil programação e reprogramação para que a sequência de operação pudesse ser alterada, mesmo após a instalação, ser de fácil manutenção, garantir a fiabilidade em ambiente industrial, ser fisicamente menor que os sistemas de relés anteriormente usados, permitir a conexão a sistemas de recolha de dados e ter uma gama de custos inferior ou semelhante ao sistema de relés [16].

Em 1969 foi criado o primeiro protótipo pela General Motors (GM), no entanto, este não foi comercializado apesar de apresentar resultados satisfatórios. Entre 1970 e 1974 foram adicionadas algumas capacidades aos controladores, como as funções de temporização e contagem, funções aritméticas e de manipulação de dados, tornando-os mais poderosos. De 1975 a 1979 mais recursos foram adicionados que propiciaram expansões na capacidade de memória, controlos analógicos de malha fechada com algoritmos *Proportional-Integral-Derivative* (PID), utilização de estações remotas de interfaces de Entradas e Saídas (E/S) e comunicação com outros equipamentos ditos “inteligentes” [17].

Atualmente, através dos avanços tecnológicos, tanto a nível de *hardware* como de *software*, pode-se dizer que os PLC evoluíram para o conceito de controlador universal de processos, pois podem-se configurar para praticamente todas as necessidades de controlo de processos e com custos moderados.

4.2. PRINCÍPIO DE FUNCIONAMENTO

O PLC tem evoluído o seu princípio de funcionamento ao longo do tempo para incluir controlo sequencial de relés, controlo de movimento, controlo de processos, sistema de controlo distribuído e ligação através da rede. Usando capacidades como manipulação de dados, armazenamento, poder de processamento e comunicação, os PLC assemelham-se cada vez mais aos convencionais computadores de secretária.

A sua função passa por executar um programa previamente armazenado em memória, que por sua vez executa o processamento de sinais de entrada provenientes de diferentes tipos de sensores e outros dispositivos de campo, e fornece sinais de saída compatíveis com o acionamento de diversos componentes como atuadores, motores, válvulas, etc.

A Figura 23 ilustra um esquema do ciclo de funcionamento do PLC, resumindo a sua função em quatro etapas:



Figura 23 Ciclo de funcionamento de um PLC [18].

1. A primeira etapa diz respeito à inicialização do controlador;
2. Os sinais existentes na interface de entrada são transferidos para a memória de dados *Random Access Memory* (RAM);
3. É iniciado o varrimento de todos os dados armazenando na memória de programa (*scan*), para posteriormente serem usados para o correto funcionamento do programa. Dentro deste ciclo, todas as operações que estavam programadas no *software* são

executadas, é feita a habilitação de temporizadores/contadores, o armazenamento de dados processados na memória de dados, etc.;

4. Concluído o varrimento do *software*, o PLC transfere os dados processados (resultados das operações lógicas) para a interface de saída. Paralelamente, novos dados provenientes da interface de entrada irão atualizar a memória de dados;

4.3. ASPETOS DE *HARDWARE*

Normalmente, um PLC é constituído pela unidade de processamento, fonte de alimentação, dispositivo de programação, memória, secção de interface entradas/saídas e interface de comunicação. Um exemplo da constituição de um PLC é apresentado na Figura 24 [19].

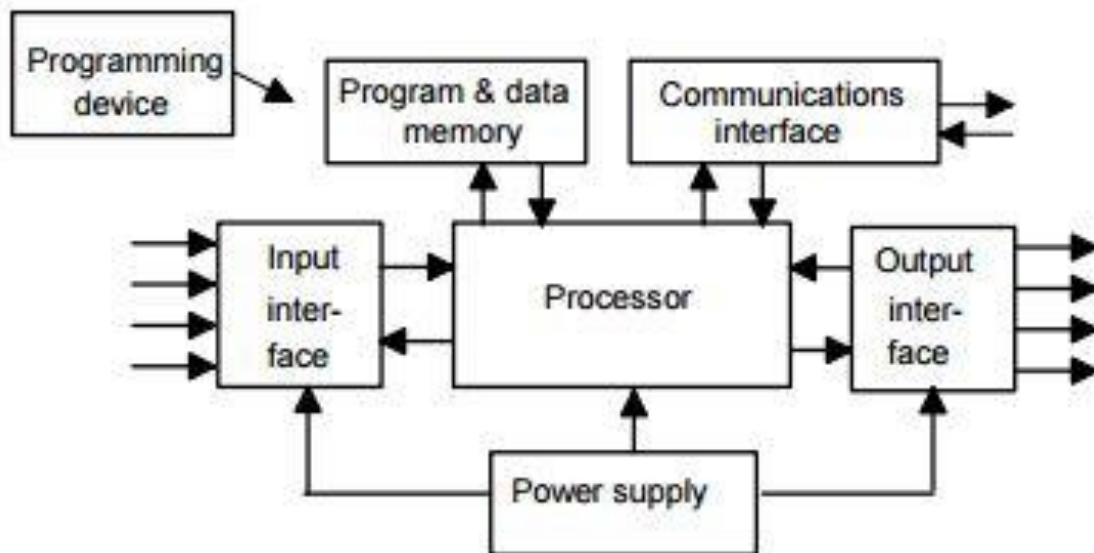


Figura 24 Sistema PLC [19].

- A unidade de processamento, ou unidade central de processamento (CPU), é a unidade que contém o microprocessador responsável por interpretar os sinais de entrada e dar início às ações de controlo, de acordo com o programa armazenado em memória, comunicando os sinais de ação para as saídas.
- A fonte de alimentação é necessária para converter a tensão alternada de entrada em tensão contínua necessária para o processador e para os circuitos da interface de entrada e saída dos módulos.

- O dispositivo de programação é usado para inserir o programa na memória do processador. O programa é desenvolvido no dispositivo e depois transferido para a unidade de memória do PLC.
- A memória de dados, também conhecida como memória de rascunho, serve para armazenar temporariamente os estados de E/S, marcadores *presets* de temporizadores/contadores e valores digitais para que a CPU possa processá-los. A memória de dados (geralmente memória RAM) é atualizada a cada ciclo de varrimento.
- A memória do utilizador serve para armazenar as instruções do *software* (programas que controlam a máquina ou a operação do processo), que são continuamente executadas pela CPU. Pode ser memória RAM, *Electrically Erasable Programmable Read-Only Memory* (EPROM), *Non-Volatile Random Access Memory* (NVRAM) ou *FLASH Electrically Erasable Programmable Read-Only Memory* (FLASH-EPROM).
- As secções de entrada e saída são onde o processador recebe/escreve a informação dos dispositivos externos. As entradas podem ser interruptores e/ou sensores. As saídas podem ser bobines do motor de arranque, válvulas, ou até mesmos outros controladores lógicos programáveis. Os dispositivos de entrada e saída podem ser classificados como sinais que são digitais ou analógicos. Dispositivos que permitam sinais discretos ou digitais são aqueles cujo estado apenas alterna de ligado (*On*) para desligado (*Off*) ou vice-versa. Os dispositivos analógicos dão sinais cuja intensidade é proporcional ao valor da variável a ser monitorada, por exemplo um sensor de temperatura pode dar uma tensão proporcional à temperatura.
- A interface de comunicação é usada para receber e transmitir dados em redes de comunicação de, ou para, outros PLC remotos. Preocupa-se com ações como a verificação do dispositivo, aquisição de dados, sincronização entre aplicações do utilizador e gestão da conexão.

4.4. ASPETOS DE *SOFTWARE*

Além do número de entradas e saídas, o que determina a utilização de um PLC são os recursos de *software* disponíveis, isto é, as funções que ele pode executar. Assim, de maneira geral os PLC apresentam como funções básicas:

- Lógica AND, OR, XOR;
- SET e RESET;
- Temporizadores e contadores;
- Cálculos com aritmética básica (+,-,*,/);
- Comparação de valores;
- Registos;
- Saltos no programa.

À medida que os PLC têm a sua capacidade de processamento aumentada, surge a necessidade de funções de *software* mais avançadas, tais como:

- Cálculos com vírgula flutuante;
- Cálculos integrais e trigonométricos;
- Malhas de controlo PID;
- Contagem rápida;
- Leitura de sinais analógicos;
- Leitura de sinais de temperatura;
- Linearização de sinais analógicos;
- Lógica difusa (*fuzzy logic*).

4.5. LINGUAGENS DE PROGRAMAÇÃO

Reconhecendo a necessidade de um padrão para PLC, por parte da comunidade industrial internacional, em 1979 foi designado um grupo de trabalho da *International Electrotechnical Commission* (IEC) com este propósito. Este grupo tinha como objetivo analisar o projeto completo de PLC (inclusive *hardware*), instalação, testes, documentação, programação e comunicações. Este grupo designou oito frentes de trabalho para desenvolver diferentes

partes do padrão para PLC. A Norma IEC 1131-3 é o único padrão global para programação de controlo industrial. Uma interface de programação padrão permite que pessoas com diferentes capacidades e formações, criem elementos diferentes de um programa, em etapas diferentes do ciclo de vida de um *software*: especificação, projeto, implementação, teste, instalação e manutenção. O padrão inclui a definição da linguagem Grafo Sequencial de Funções (SFC), usada para estruturar a organização interna do programa, e de mais quatro linguagens: Texto Estruturado (ST), Diagrama de Blocos Funcionais (FBD), Lista de Instruções (IL) e Diagrama *Ladder* (LD). Uma linguagem pode mostrar-se mais simples e legível, quando comparada com as outras, dependendo do que se pretende desenvolver e da sua complexidade.

4.5.1. DIAGRAMA SEQUENCIAL DE FUNÇÕES (*SEQUENTIAL FUNCTION CHART*)

O Diagrama Sequencial de Funções (SFC) é uma linguagem que foi desenvolvida com base no funcionamento do *Grphe Fonctionnel de Commande* (Grafcet), sendo constituída essencialmente por estados e transições. Fornece um método flexível e hierárquico de representar graficamente sistemas de controlo complexos. Um qualquer processo industrial pode ser dividido em diversos estados ou etapas bem definidas, que seguem uma determinada sequência. Em cada um destes estados podem ser executadas uma ou mais ações. Estas ações podem ser desenvolvidas em qualquer uma das linguagens referidas seguidamente. Para uma determinada ação terminar é necessário que as condições de transição para o próximo estado sejam satisfeitas. Podem existir sequências alternativas num processo e estas podem ser representadas usando caminhos divergentes no diagrama.

4.5.2. TEXTO ESTRUTURADO (*STRUCTURED TEXT*)

O Texto Estruturado (ST) é uma linguagem textual de alto nível semelhante ao C, C++ e ao PASCAL. Permite a indentação do código e a inserção de comentários que facilitam a compreensão do texto. Esta linguagem é principalmente destinada a resolver problemas de controlo analógico em tempo real. A Norma IEC-1131-3 define o ST como uma linguagem que consiste em declarações que podem ser usadas para atribuir valores a variáveis utilizando vários tipos de operadores aritméticos, booleanos, de comparação, execução condicional e ciclos iterativos. Podem ainda ser escritos blocos de funções utilizando o ST, que podem depois ser chamados em diversos pontos do programa.

4.5.3. DIAGRAMA DE BLOCOS FUNCIONAIS (*FUNCTION BLOCK DIAGRAM*)

O Diagrama de Blocos Funcionais (FBD) é uma linguagem gráfica muito utilizada na representação de sistemas de controlo industriais, adotando um conjunto de símbolos e convenções definidos na Norma IEC-1131-3 (1993). Representa um sistema de controlo em termos de fluxo de sinal entre os elementos de processamento, semelhante à metodologia adotada para o fluxo de sinal de circuitos eletrónicos. A linguagem FBD é mais adequada para expressar o comportamento contínuo de um sistema e deve ser considerada quando há uma necessidade de sequenciamento/priorização de funções. Nesta linguagem é possível a construção de novos blocos funcionais a partir de blocos mais simples (como blocos AND, OR ou NOT), reduzindo o tamanho do programa e melhorando a sua legibilidade. Esta linguagem permite uma ação retroativa do sinal, isto é, permite que um bloco de funções utilize os valores da sua saída em blocos ligados à sua entrada ou diretamente a uma das suas entradas.

4.5.4. LISTA DE INSTRUÇÕES (*INSTRUCTION LIST*)

A Lista de Instruções (IL) é uma linguagem de programação baseada em texto e assemelha-se à linguagem de baixo nível utilizada na programação de microprocessadores. É simples de implementar e, portanto, adotada por diversos fabricantes de PLC. Esta linguagem é geralmente utilizada para problemas simples que requerem código altamente otimizado. Apresenta regras de semântica muito bem definidas, que são utilizadas nas instruções. A conversão de IL para outras linguagens é difícil e só pode ser feita em certas circunstâncias. Por outro lado, a conversão para IL é relativamente mais fácil, mas não necessariamente mais simples.

4.5.5. DIAGRAMA LADDER

A linguagem LADDER é baseada no princípio de contactos elétricos. Cada um dos componentes pode possuir um número infinito de contactos que são limitados pela capacidade de memória do PLC. *Ladder* é uma palavra inglesa que quer dizer escada e este nome está associado à representação gráfica dos contactos: esta é feita horizontalmente, em linhas paralelas, fazendo lembrar os degraus de uma escada.

5. ARQUITETURA DO SISTEMA

Nos capítulos anteriores foram apresentados, de um modo global, os componentes/tecnologias que irão integrar este projeto. Como este passa por ser uma melhoria e não a criação de um novo sistema, é necessário referir e explicar o funcionamento do sistema existente, fazer um levantamento dos materiais necessários e verificar que melhorias poderão ser realizadas. Este capítulo tem então o objetivo de explicar toda a análise realizada em volta do sistema a otimizar, bem como o estudo do novo sistema a ser implementado.

5.1. INTRODUÇÃO

Para a realização deste projeto é necessário realizar um estudo prévio para conhecer o sistema implementado. É preciso saber que limitações poderão ser encontradas face à substituição dos componentes, uma vez que um dos objetivos do projeto passa por ser a atualização de *hardware* e *software* usados na máquina.

Assim sendo, para começar, é necessário analisar a máquina em questão, pois um dos principais passos passa por saber como a própria é constituída e como funciona, conhecendo assim exatamente o processo de fabrico, tornando-se, deste modo, mais simples a organização de ideias.

Posteriormente ao estudo do processo de fabrico existente, é necessário delinear quais os componentes que necessitam de ser substituídos e encontrar as respetivas soluções alternativas, tendo em atenção uma série de características e compatibilidades que não poderão ser descartadas e que serão discutidas nos capítulos seguintes.

Por fim, após estar decidido qual o *hardware* a ser utilizado, é necessário decidir qual o *software* que irá permitir configurar e interligar todos os componentes (os que se mantêm e os que serão substituídos).

5.2. BP260 X

5.2.1. CARATERÍSTICAS E DESCRIÇÃO DA MÁQUINA

Denominada por BP260 X ou Baixa Pressão (Figura 25), esta máquina integra uma das principais fases do processo produtivo, a fundição, e é responsável por moldar a matéria-prima, dando forma ao produto quase final.



Figura 25 BP260 X.

Esta máquina é utilizada para a fundição de latão e outras ligas de cobre, de modo a obter peças de dimensões pequenas ou médias/grandes, usando moldes de múltiplas cavidades. A matriz pode ter uma dimensão até 600 milímetros de diâmetro. Um sistema de proteção de barreira de luz permite que o operador trabalhe com condições de segurança, sendo que, a

máquina não tem sequências particularmente críticas. A BP260 X é projetada para funcionar com apenas um operador, que deve ficar dentro da área de trabalho da máquina.

As principais características desta máquina são listadas de seguida:

- os movimentos são acionados hidráulicamente por uma bomba de deslocamento variável (fluido hidráulico não-inflamável);
- contém uma válvula de segurança durante os movimentos da coquilha sobre o banho de grafite, encerramento da coquilha e inclinação do banco;
- apresenta redução do consumo de energia elétrica, funcionamento silencioso, temperatura do fluido hidráulico constante e, conseqüentemente, uma longa duração de equipamentos hidráulicos que são concedidos pela bomba de deslocamento variável;
- encerramento da coquilha com travagem hidráulica progressiva e ajustável para evitar danos no núcleo;
- interruptor de frenagem (ajustável) limite para todos os cilindros;
- ajuste rápido e funcional dos pontos para começar o vazamento;
- ciclo da grafite automático com ajuste independente para cada coquilha;
- rotação da coquilha durante o processo de grafite;
- pré-arranjo do ângulo de grafite;
- tempo de abertura da coquilha controlado;
- sistema de substituição da coquilha rápido e preciso;
- tanques de grafite separados, com agitadores de aço inoxidável com uma hélice do lado direito e do lado esquerdo;
- filtro de fluido hidráulico equipado com indicador de obstrução.

5.2.2. DESCRIÇÃO DA INSTALAÇÃO

De seguida é feita uma descrição das principais partes da célula de fabrico:

- manipulador com cabeças sobredimensionadas, composto por duas estações de trabalho, desfasadas por 180°, com a capacidade de rodar;
- proteção em todas as partes que possam estar em contato com os vapores provenientes da água de grafite;
- cabeça rotativa, por meio de um motor hidráulico, com bloqueio mecânico do manipulador;
- banco com inclinação ajustável (entre 0° e 90°);
- coquilhas rotativas entre 0° e os 90°.
- cilindros ejetores;
- forno elétrico indutivo B3R de 160 kW;
- sistema de pressurização com válvulas de segurança definidas a 1 bar;
- unidade hidráulica para o movimento do manipulador, a trabalhar com óleo hidráulico CASTROL 46 AWG, com controlo elétrico, e a visualização do nível externo;
- sistema de lubrificação com cartuchos automáticos;
- pressurização manual e indicadores paramétricos;
- painel de controlo do manipulador completo com PLC;
- painel de controlo do forno;
- plataforma metálica robusta, que delimita a zona de trabalho da máquina, e também a área de trabalho do operador;
- barreira de segurança classe 4;

- válvulas de segurança para evitar movimentos perigosos devido a travagem de tubos hidráulicos;
- resistências nos tanques de grafite;
- exaustores de aspiração de fumos em ambas as cabeças;
- PLC Siemens S7-300, com PC 840 supervisor e controlo das válvulas proporcionais;
- controlo do nível máximo de metal líquido no forno por meio de sinalização luminosa;
- painel perto da estação de trabalho do operador;
- elemento FESTO pneumático;
- fonte de alimentação de emergência para o forno.

5.2.3. ESTRUTURA E DESCRIÇÃO

O sistema é essencialmente constituído pelos seguintes elementos:

- uma estrutura de suporte em metal;
- uma máquina com comandos electro-hidráulicos chamados a partir da estação de trabalho do manipulador;
- uma unidade hidráulica que alimenta a máquina;
- um painel de comandos eléctrico para o manipulador;
- um controlo de pressurização de ar do forno, colocado dentro de um painel dedicado;
- tanques de imersão de grafite;
- um forno de indução eléctrico para o bronze, de baixa frequência, com um sistema de pressurização pneumático e sistema de elevação integrado;
- uma plataforma de suporte, no interior da qual são posicionados os componentes acima listados.

5.2.3.1. Estrutura de suporte metálica

A estrutura de metal é constituída por uma coluna cilíndrica adequadamente dimensionada, apresentando na extremidade superior uma roda, que permite a rotação do manipulador. Na base da coluna existe um alargamento, a fim de torná-la mais estável. A estrutura é presa ao solo com parafusos de expansão.

5.2.3.2. Manipulador

O manipulador torna possível a circulação do molde para a grafite, abertura e fecho, inclinação do banco para posicionar o macho de areia, inclinação lateral para facilitar as operações de colocação do macho de areia, rotação do molde, extração das peças de fundição e abertura frontal para a limpeza da coquilha.

5.2.3.3. Forno

A fase de fundição da peça ocorre com baixa pressão, quando o manipulador é posicionado verticalmente no forno. A aproximação entre o bico de vazamento e a boca da coquilha ocorre por meio do levantamento do forno utilizando um único motor hidráulico. O forno, alimentado por um sistema hidráulico que injeta ar sob pressão, executa a “baixa pressão”, dirigindo o líquido para dentro da coquilha, de baixo para cima, através do bico de vazamento. O forno B3R/160 é um forno de indução trifásico, de baixa frequência, em que a peça principal é constituída por dois indutores monofásicos (bobinas) que incorporam ligações de SCOTT (tipo de circuito usado para derivar energia elétrica), a fim de limitar o desequilíbrio das três fases.

5.2.3.4. Banho de grafite

O processo de grafite ocorre na estação onde são colocados os machos de areia. Este processo ocorre por meio de imersão da coquilha na água e grafite, mantido em suspensão por meio de agitadores elétricos. Cada tanque é fornecido com um sistema de refrigeração para a suspensão água-grafite, por meio do sistema de permutadores de calor de aço inoxidável contidos no fundo do próprio tanque.

5.2.3.5. Sistema hidráulico

Um único dispositivo equipado com uma unidade de bomba monobloco comanda o sistema. O sistema hidráulico foi concebido e dimensionado de forma a garantir os fluxos e pressões requeridas pelas diferentes fontes do sistema. O uso de bombas de fluxo variável faz com

que seja possível adaptar melhor o fluxo que sai da bomba, reduzindo a potência dissipada pelo calor e ruído.

5.2.3.6. Sistema de pressurização pneumática

O objetivo deste sistema é de gerar e controlar a pressão de pressurização do forno e, por conseguinte, da modelação. O controlo da fundição é exercida através de um computador que, por sua vez, controla os sistemas pneumáticos, constituídos por válvulas proporcionais e transdutores de pressão. No entanto, o transdutor de pressão é meramente informativo, ou seja, no painel é possível verificar os valores lidos por ele, mas este não influencia de modo algum o processo de vazamento. O *software* é instalado no computador que controla o vazamento, tendo em conta os parâmetros de base (pressão teórica e tempo) programados pelo utilizador, bem como um número de variáveis de processo tais como a pressão efetiva (*feedback*) e o peso do forno (células de carga).

5.2.3.7. Sistema elétrico

O sistema elétrico foi projetado e produzido de acordo com os regulamentos atualmente em vigor, a fim de satisfazer plenamente as necessidades atuais e futuras de automação e energia. A automação do sistema está sujeito a um PLC. O controlo de pressurização é exercido por um computador pessoal que está em constante comunicação com o PLC. O operador pode inserir e gravar curvas de pressurização, no PC, relativos às peças que estão a ser produzidas.

5.2.4. DESCRIÇÃO DO CICLO DE OPERAÇÃO

O *layout* da BP260 X é projetada para funcionar com apenas um operador, cuja função é posicionar os machos de areia, extrair a peça da coquilha e realizar a supervisão de qualidade. A máquina é constituída por uma coluna em que são montadas duas cabeças com capacidade de rotação. O operador encontra-se sobre uma plataforma, em frente à máquina rotativa.

O ciclo de trabalho começa com uma das duas cabeças colocadas em frente do operador. A cabeça é inclinada para permitir que o operador possa colocar o macho de areia na coquilha. O operador tem de limpar a metade da coquilha com um pulverizador de pressão de ar. De seguida, coloca o macho de areia no interior do invólucro, com o máximo de cuidado para não o partir, visto que o macho de areia é bastante frágil.

O operador volta a pulverizar metade da coquilha para eliminar partículas de areia que possam ter ficado dispersas após a colocação do macho de areia. Depois pode ser iniciado o ciclo automático pressionando o respetivo botão. A partir deste momento o ciclo é

completamente automático, e o operador só tem que esperar que a outra cabeça se posicione à sua frente e abra a coquilha.

Quando isto ocorre, ele extrai a peça fundida com a ajuda de um alicate. De seguida a máquina mergulha as duas metades da coquilha nos tanques para o processo de grafite. Em seguida, o operador tem de repetir a sequência de operações anteriormente descritas: pulverizar as duas metades da coquilha, colocar o macho de areia, limpá-los, etc.

A máquina posiciona os moldes no forno (a baixa pressão) de forma automática, dirigindo o líquido para dentro da coquilha, de baixo para cima, através do bico de vazamento, espera até que o metal derretido fique sólido e abre metade da coquilha para permitir que o operador possa extrair o produto.

A vantagem deste *layout* reside no facto de enquanto uma cabeça está posicionada sobre o forno para a fundição de baixa pressão, a cabeça oposta estar em frente do operador, que extrai a peça e coloca o macho de areia para fazer um novo produto. Há uma grande otimização dos tempos, através da redução dos tempos mortos entre uma peça moldada e a outra, em comparação com uma máquina de cabeça única.

O diagrama apresentado na Figura 26 descreve o funcionamento de uma das cabeças da máquina (sendo que o modo de funcionamento da outra é igual).

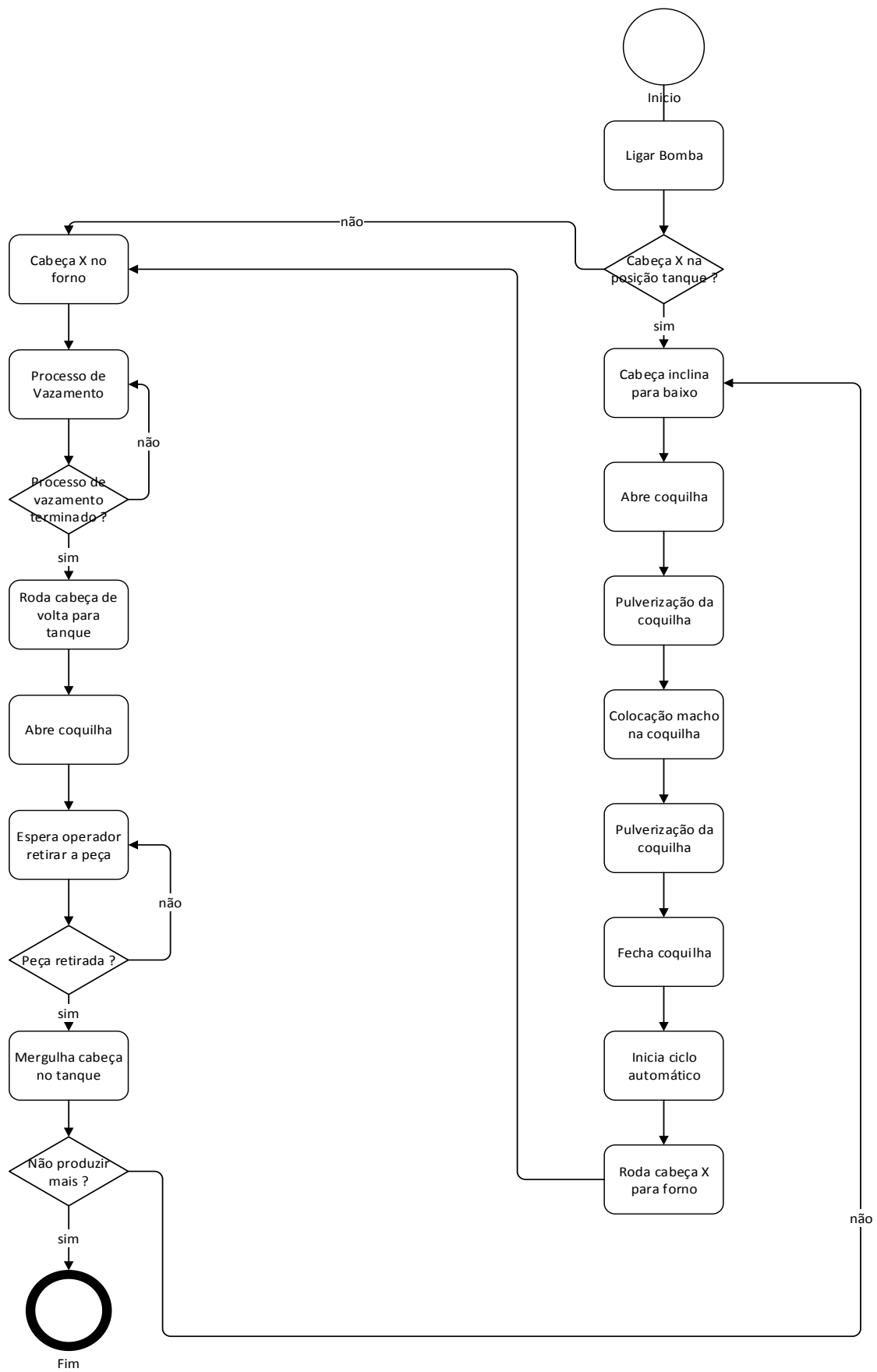


Figura 26 Diagrama do modo de funcionamento da cabeça X.

5.3. ESTUDO DOS COMPONENTES QUE INTEGRAM O SISTEMA

Findo o estudo genérico do funcionamento da máquina, o próximo passo é analisar cuidadosamente os componentes a serem substituídos nesta. Numa primeira abordagem do problema, a conclusão tirada foi que o mais viável seria encontrar componentes no mercado que substituíssem aqueles que são mais antigos, nomeadamente a válvula reguladora de pressão, e uma solução que substituísse todo o sistema de supervisão, a consola HMI e o PC.

Foi também definido como objetivo a conversão do programa do PLC, que se encontra em versão SIMATIC, para a nova versão disponibilizada pela Siemens, o TIA Portal, contornando assim futuros problemas de assistência ou manutenção que possam surgir.

Na Figura 27, é apresentada a arquitetura do sistema antigo. Os movimentos da máquina são comandados por um controlador (PLC), que por meio de atuadores e sensores, faz com que a máquina realize corretamente todos os seus movimentos. Ligado também ao PLC encontra-se um PC, responsável pelo controlo do processo de vazamento, que envia informação proveniente da válvula e do PLC para o HMI incorporado. Esta arquitetura incorpora também um transdutor de pressão.

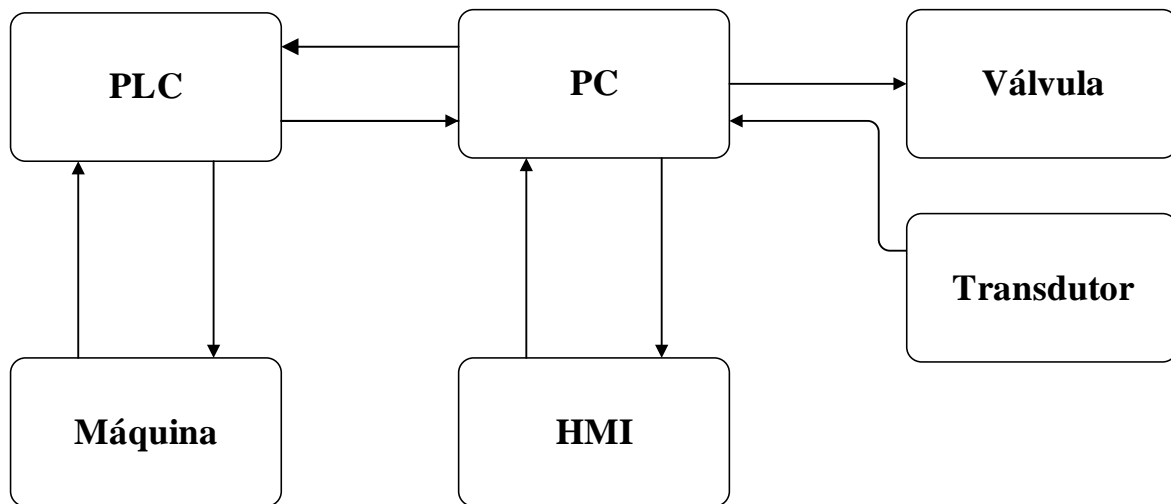


Figura 27 Arquitetura do sistema antigo.

Na análise ao sistema de controlo e monitorização da máquina, verificou-se que o controlo do processo de vazamento apenas utiliza a válvula e não usa o transdutor, ou seja o controlo não é realizado mediante os valores obtidos pelo transdutor, simplesmente é fornecido um valor à válvula e esta tenta cumprir essa especificação.

5.3.1. CONTROLADOR DA BP260 X

O sistema de aquisição de dados implementado na máquina é um autómato SIMATIC S7-300 com CPU 315-2 DP (6ES7 315-2AH14-0AB0). Sendo uma das soluções para sistemas integrados de produção, o S7-300 é um sistema modular utilizado em aplicações centralizadas ou distribuídas de pequeno a médio porte. As características desta CPU são apresentadas na Tabela 1.

O S7-300 não necessita de *racks* com números predefinidos de *slots* para ser montado (o que fornece uma economia de espaço e flexibilidade de configuração). Os módulos são interligados uns aos outros através de um barramento modular que fica encaixado numa calha DIN padrão.

Em termos de características funcionais, o S7-300 dispõe de diversas CPU (com diferentes capacidades), sendo utilizado em aplicações simples ou aplicações que requeiram grande desempenho.

A utilização de módulos de expansão permite uma diversidade de configurações para qualquer tipo de aplicação. Para tal, estão disponíveis:

- módulos de I/O (SM) – Digitais (24 V DC, 120/230 V AC) e analógicos (± 5 V, 0-10 V, 0/4 – 20 mA);
- módulos de comunicação (CP) – MPI, Profibus DP e Ethernet;
- módulos de função (FM) - Contadores, posicionamento, controlo de motor passo a passo, controlo em malha fechada (PID), entre outros.

Tabela 1 Principais características da CPU S7-315-2 DP.

CPU 315 - 2 DP	
Memória Central	64 kB/ 21 k RAM de instruções
Memória de carga	
Integrada	96 kB RAM
Memória volátil (SD Card)	4 Mbytes Flash-EPROM
Temporizadores	
Retentiva sem captura	0 a 127, ajustável
Retentiva com captura	0 a 127, ajustável
Alcance	10 ms a 9990 s
Interface Multiponto MPI	
Número de estações, máx.	32 estações no barramento MPI; PG/PC, OP, outros S7-300/400, C7, máx. 4 ligações estáticas e 4 ligações dinâmicas
Funções de Comunicação	Comunicação PG/OP; Comunicação por dados globais; Comunicação básica S7; Comunicação S7 (servidor).
Distância entre 2 estações	Sem repetidor: 50 m; Com 2 repetidores: 1100 m; Com 10 repetidores: 9100 m; Com fibra ótica: 23,8 km.
Velocidade de transmissão	187,5 kbits/s
PG/PC com STEP 7	Conexão via interface MPI
Espaço de endereçamento E/S	512/512 bytes

Numa configuração centralizada podem ser utilizados até cerca de 32 módulos de expansão. Toda a gama de CPU do S7-300 incorpora uma porta de comunicação *Multi Point Interface* (MPI) integrada e é a partir desta que a CPU é programada e parametrizada. A programação e a parametrização do PLC S7-300 podem ser realizadas através do *software* SIMATIC STEP 7 Professional ou TIA Portal.

5.3.2. COMUNICAÇÃO MPI

O MPI é uma interface proprietária do controlador lógico programável SIMATIC S7 da empresa Siemens. Este é usado para conectar as estações de programação (PC), consolas de operação e outros dispositivos na família SIMATIC. Esta tecnologia foi inspirada e desenvolvida segundo o protocolo Profibus.

O MPI baseia-se na norma EIA-485 (RS-485) e funciona normalmente com um *baud rate* de 187,5 kB/s. O MPI deve ter resistência no fim da linha, que é geralmente incluído no conector e ativado por um simples interruptor.

Assim, as principais características da interface MPI são:

- meio físico RS 485;
- taxas de transmissão de 19,2 kbps, 187,5 kbps ou 1,5 Mbps;
- distâncias de até 50 m (entre dois nós vizinhos) com 2 repetidores.

Cada nó MPI necessita de ter o seu próprio endereço (entre 0 e 126; os valores padrões são PG = 0, OP/TD = 1, e CPU = 2).

Para a comunicação realizada entre a estação de programação e o PC utilizou-se o *PC-Adapter*, mostrado na Figura 28. Esta interface de ligação permite estabelecer a comunicação entre uma porta USB (computador) e uma porta RS-485 (PLC).



Figura 28 *PC-Adapter* para comunicação MPI [20].

5.3.3. VÁLVULA E TRANSDUTOR DE PRESSÃO

A válvula de pressão é responsável por controlar o envio de ar para o forno, de acordo com o valor de pressão (mais corretamente a curva de pressão ao longo do tempo) que é introduzido no HMI. Esse valor atribuído é variável ao longo do tempo, ou seja, os valores de pressão que a válvula deve fornecer mudam ao longo do tempo, de modo a que as especificações da curva de vazamento sejam cumpridas. Na Figura 29, é apresentada o modelo da válvula instalada na máquina, que controla o envio de ar para o forno.



Figura 29 Válvula proporcional do forno.

A válvula é alimentada a 24 V DC, com tensão de saída de 0 a 10 V, 0 a 1 bar e com uma pressão máxima de 2 bar.

Apesar de a Grohe-Portugal ter este tipo de válvulas em *stock*, este modelo já foi descontinuado, tendo sido substituído por modelos mais recentes.

Uma solução encontrada para a substituição desta foi a válvula E/P *pressure regulator*, Series ED12 da AVENTICS, apresentada na Figura 30, também alimentada a 24 V DC, com tensão de saída de 0 a 10 V, de 0 a 10 bar e com pressão máxima de 11 bar.



Figura 30 Modelo da válvula E/P *pressure regulator*, Series ED12 da AVENTICS.

Relativamente ao transdutor de pressão (Figura 31), este serve para ler os valores de pressão do forno e vai ser usado de modo a que, com o controlo PID, o forno mantenha os níveis de pressão necessários para se conseguir fazer o vazamento das peças. O valor recebido pelo transdutor irá entrar como variável de realimentação do PID, que juntamente com as constantes P, I, D, e os valores de *setpoint* atribuídos, irão gerar o valor a ser usado pela válvula, para esta funcionar corretamente.



Figura 31 Transdutor a ser usado no forno para leitura dos valores de pressão.

Este transdutor é alimentado a 24 V DC, com uma tensão de saída de 0 a 10 V a que corresponde uma pressão de 0 a 2 bar.

5.3.4. SISTEMA DE SUPERVISÃO

O PC integrado neste sistema tem um papel fulcral pois é responsável não só pelo controlo da válvula proporcional, como também é o elo de ligação entre a própria válvula e o PLC que comanda o funcionamento da máquina, ou seja o processo de fabrico não se realiza caso o PC ou o PLC falhem, o que faz com o sistema tenha uma grande limitação neste ponto.

Outra grande limitação em torno do PC usado é o facto de parte do sistema ser controlado pelo próprio computador, uma vez que um computador pode trazer grandes limitações quer a nível de *hardware*, como a nível de *software*, como é o caso. O sistema operativo é bastante antigo (Windows 98), tendo já sido descontinuado, o *software* usado também se tornou obsoleto, o que traz grandes problemas de compatibilidade com os novos sistemas operativos hoje em dia utilizados. Em resumo, com o crescente desenvolvimento computacional e os requisitos atuais do próprio computador, não seria possível fazer o *update* para um sistema operativo mais recente. Mesmo que o PC tivesse essas capacidades seria completamente desnecessário, visto que o *hardware* presente é de tal forma antigo que já não existem *drivers* que pudessem ser instalados nos novos sistemas operativos.

Tendo em conta todos estes detalhes, foram consideradas duas possíveis soluções para o problema. Uma delas passaria por usar um *software* de programação, como o Visual Studio, e através de uma linguagem de programação orientada a objetos (como Visual basic ou C Sharp (C#)), criar-se um sistema SCADA capaz de monitorizar o sistema, controlar a válvula proporcional e servir de elo de ligação entre o PLC e o novo *hardware* de suporte para a aplicação. Uma outra solução passaria por recorrer ao PLC, para que este para além de controlar a máquina, passasse a controlar também o processo de vazamento, adquirindo depois somente um sistema SCADA para monitorização.

5.3.5. NOVO CONTROLADOR

Para a fase de testes e implementação prática, foi adquirida uma nova CPU 314. A razão pela qual não foi utilizada a CPU da máquina, diz respeito ao facto de a máquina continuar em produção. Daí a necessidade de adquirir uma nova CPU, não precisando assim de ter que a parar.

Comparativamente à CPU 315-2 DP, a CPU 314 apresenta apenas algumas diferenças a nível de memória, como é apresentado na Tabela 2, sendo que as outras características são bastante

semelhantes. Como pode ser verificado, as principais diferenças são que a CPU 314 tem uma memória central de 24 kB/8 k RAM de instruções, em vez de 64 kB/21 k RAM, e uma memória integrada de 40 kB RAM e 512/512 Bytes para endereçamento relativo a entradas e saídas.

Tabela 2 Comparação da CPU da máquina com a nova CPU a implementar.

CPU 315 - 2 DP		CPU 314	
Memória Central	64 kB/ 21k RAM de instruções	Memória Central	24 kB/ 8 k RAM de instruções
Memória de carga		Memória de carga	
integrada	96 kB RAM	integrada	40 kB RAM
Memória volátil (SD Card)	4 Mbytes Flash-EPROM	Memória volátil (SD Card)	4 Mbytes Flash-EPROM
Temporizadores	128	Temporizadores	128
Retentiva sem captura	0 a 127, ajustável	Retentiva sem captura	0 a 127, ajustável
Retentiva com captura	0 a 127, ajustável	Retentiva com captura	0 a 127, ajustável
Alcance	10 ms a 9990 s	Alcance	10 ms a 9990 s
Interface Multiponto MPI		Interface Multiponto MPI	
Número de estações, máx.	32 estações no barramento MPI; PG/PC, OP, outros S7-300/400, C7, máx. 4 ligações estáticas e 4 ligações dinâmicas	Número de estações, máx.	32 estações no barramento MPI; PG/PC, OP, outros S7-300/400, C7, máx. 4 ligações estáticas e 4 ligações dinâmicas
Funções de Comunicação	Comunicação PG/OP; Comunicação por dados globais; Comunicação básica S7; Comunicação S7 (servidor);	Funções de Comunicação	Comunicação PG/OP; Comunicação por dados globais; Comunicação básica S7; Comunicação S7 (servidor);
Distância entre 2 estações	Sem repetidor: 50 m; com 2 repetidores: 1100 m; Com 10 repetidores: 9100 m; Com fibra óptica: 23.8 km;	Distância entre 2 estações	Sem repetidor: 50 m; com 2 repetidores: 1100 m; Com 10 repetidores: 9100 m; Com fibra óptica: 23.8 km;
Velocidade de transmissão	187.5 kbits/s	Velocidade de transmissão	187.5 kbits/s
PG/PC com STEP 7	Conexão via interface MPI	PG/PC com STEP 7	Conexão via interface MPI
Espaço de endereçamento E/S	512/512 bytes	Espaço de endereçamento E/S	1/1 kB

No entanto, apesar das diferenças encontradas, a CPU 314 é suficiente para ser implementada na máquina e conseguir realizar todas as tarefas anteriormente realizadas, e as novas a serem incorporadas. Esta CPU tem a vantagem de ser mais barata do que a CPU 315-2 DP.

5.3.6. SIMATIC

O projeto em causa foi desenvolvido utilizando o SIMATIC da Siemens, que permite uma interação entre as várias aplicações que foram usadas para implementar o sistema da máquina, como o SIMATIC Manager (gestão do projeto), SIMATIC WinCC Flexible (desenvolvimento do HMI), SIMATIC WinCC Flexible Runtime (faz correr a aplicação criada no PC) e, caso necessário, o SIMATIC S7 PLCSIM (ferramenta de simulação do CPU que corre o programa criado).

A Figura 32 (lado esquerdo) mostra a estrutura do projeto sempre que se abre o SIMATIC Manager. Navegando pelas várias opções de seleção, é possível configurar o projeto de modo a atribuir os parâmetros de *hardware*, rede de comunicação, programação dos blocos e simulação do programa. Do lado direito da mesma figura é apresentada a janela de configuração do *hardware* do projeto, com a CPU selecionada e os módulos de expansão (contorno a laranja). Fazendo duplo clique na CPU, ou num dos módulos, é apresentada outra janela (contorno a vermelho) com as várias opções de configuração.

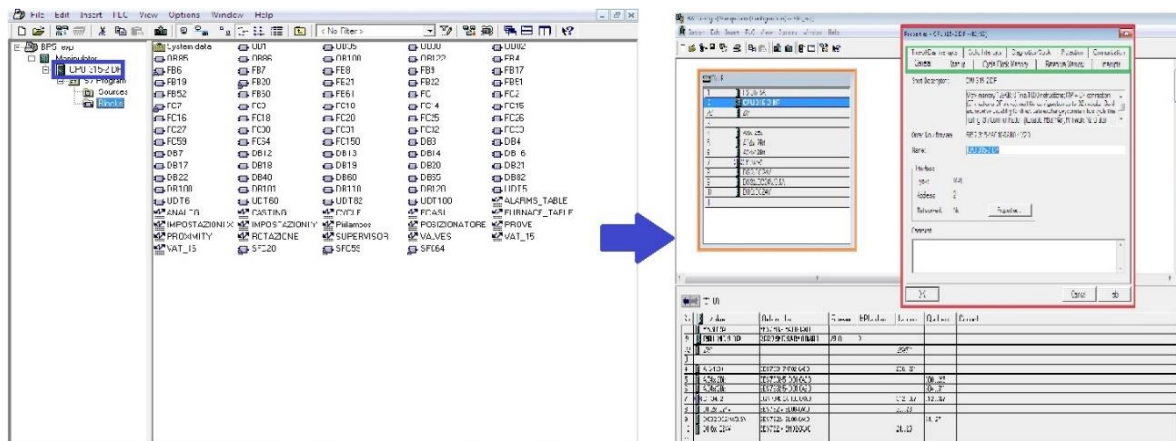


Figura 32 Estrutura do projeto da BP260 X em SIMATIC.

Expandindo a pasta *S7 program* é possível aceder a todos os componentes do programa, destacando-se a pasta *blocks*, que contém o *Organization Block 1* (OB1) e outras funções / funções de blocos criados na programação do projeto.

Tanto em SIMATIC como em TIA Portal, o OB1 é processado ciclicamente pela CPU, que faz uma leitura linha a linha e executa os comandos do programa. Quando a CPU volta à primeira linha do programa significa que completou exatamente um ciclo.

As funções de blocos (FB) e funções (FC) estão abaixo da OB1 em termos de hierarquia do programa; estas contêm uma parte do programa que pode ser chamado muitas vezes na OB1. Todos os parâmetros e dados estáticos na função de blocos são guardados em separado num bloco de dados (DB) atribuído. Já nas funções não é necessário um bloco de dados.

5.4. TIA PORTAL

Trabalhar em TIA Portal torna-se mais fácil e intuitivo do que em SIMATIC, visto que o próprio *software* proporciona um ambiente de desenvolvimento integrado para diversas

ferramentas de engenharia como, por exemplo, SIMATIC STEP 7 e SIMATIC WinCC, ou seja, é uma plataforma centralizada de engenharia que contém uma única interface para todas as tarefas de automação (configuração, comunicação, diagnóstico), várias bibliotecas com a informação necessária e uma consistência automática de dados em todo o projeto, ao contrário do que sucede em STEP 7, que abre janelas de visualização diferentes (fora do programa inicialmente aberto) para configurar hardware, e para programar blocos, sendo também o *software* WinCC à parte. Na Figura 33, é apresentada uma visão geral do *software* TIA Portal [21].



Figura 33 Visão Geral do TIA Portal [21].

O TIA Portal é um editor adaptado às necessidades do utilizador. Este contém dois tipos de visualização: a *Portal view* e a *Project view*. De acordo com o fabricante, este editor apresenta as seguintes características:

- *Portal view* – permite uma visualização orientada ao fluxo de desenvolvimento das tarefas e uma agilidade no desenvolvimento do projeto por meio de operações simples e intuitivas (Figura 34).

O *Portal view* é uma operação rápida e simples, para um rápido início orientado de *workflow*, o que o torna de fácil utilização. Permite um acesso rápido às tarefas através dos menus estruturados e facilidade nos primeiros passos rumo ao desenvolvimento das tarefas de automação.

No *Portal view* o utilizador é guiado às diferentes tarefas, através dos menus, como programação de blocos ou diagnóstico *online*.

- *Project view* – permite a organização hierárquica do projeto. Todos os editores, parâmetros e dados do projeto ficam disponíveis na mesma janela de visualização (Figura 35).

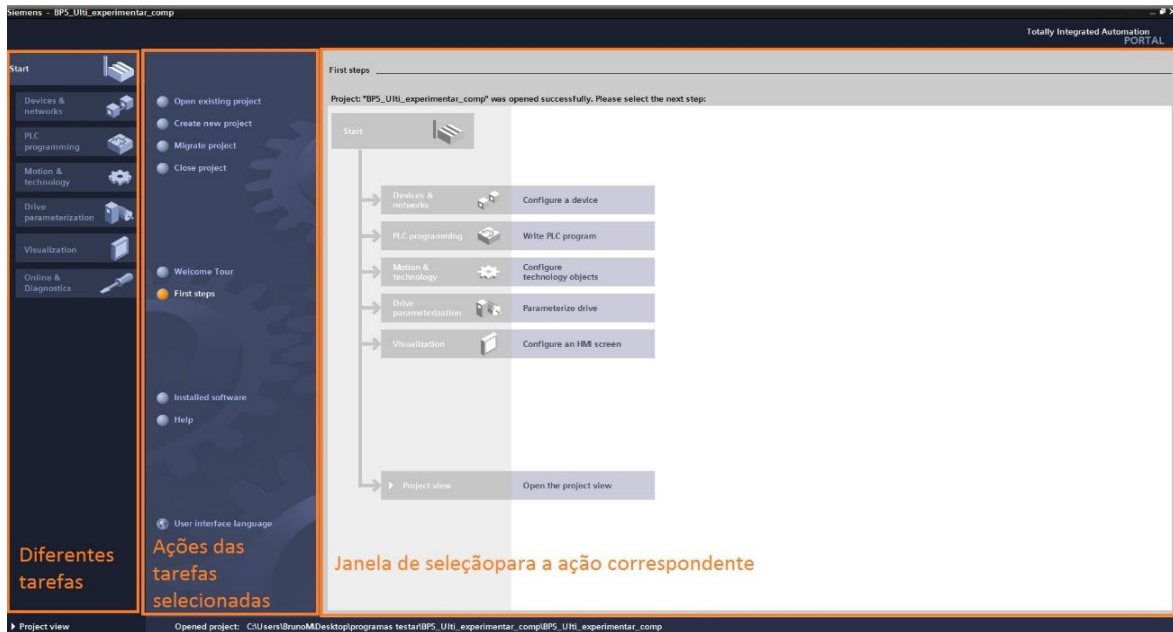


Figura 34 TIA Portal *portal view*.

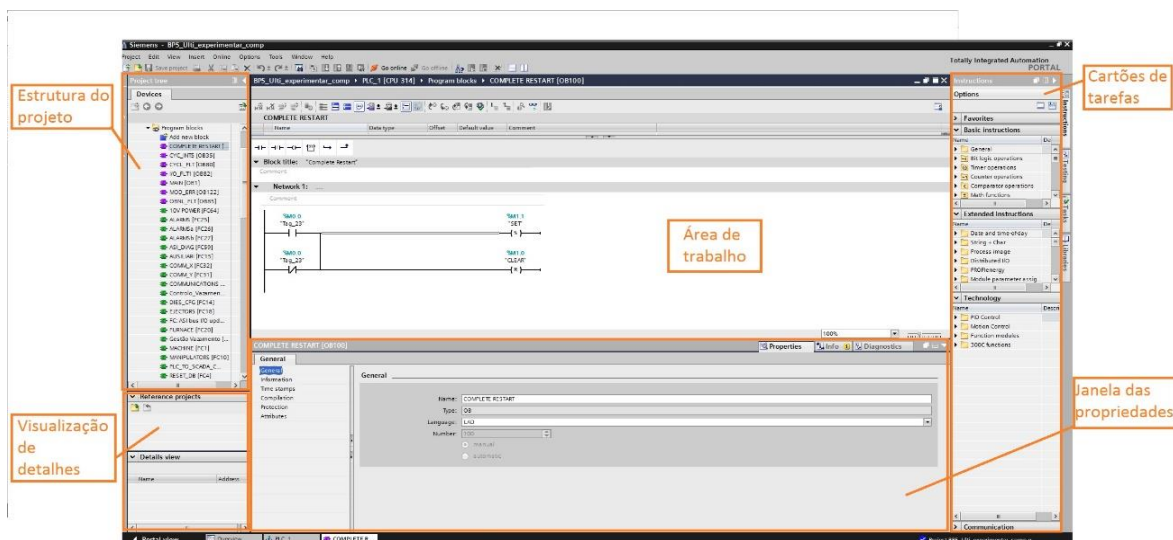


Figura 35 TIA Portal *project view*.

Para contornar futuros problemas de assistência ou manutenção, o programa da máquina foi convertido para o novo *software* da Siemens TIA Portal (Figura 36), visto que este também permite a configuração e programação de toda a gama de CPU S7-300.

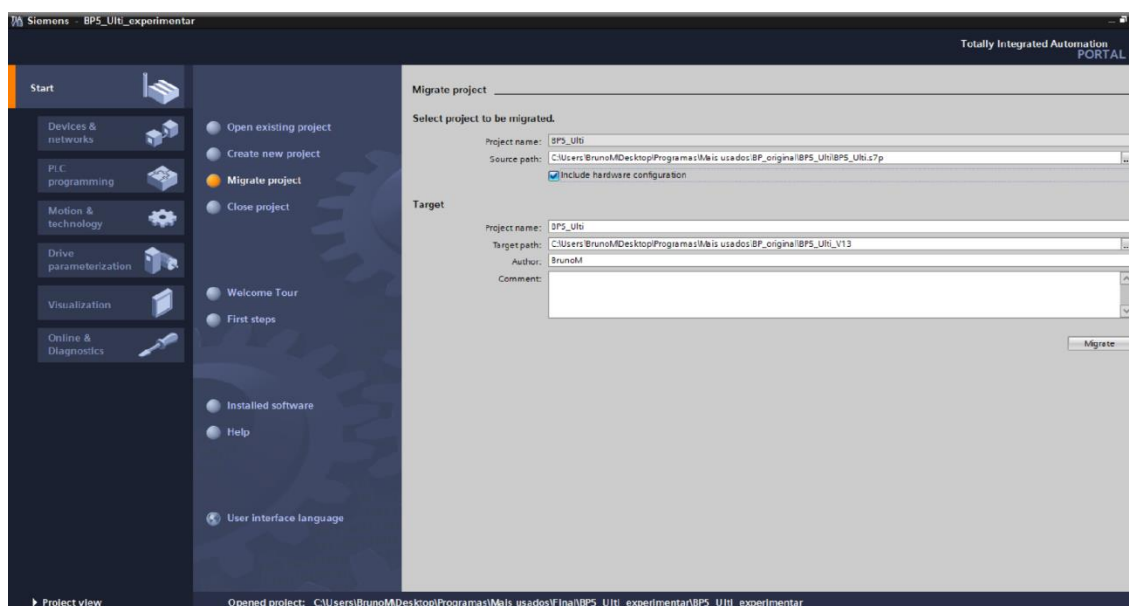


Figura 36 Conversão do projeto em SIMATIC para TIA Portal.

Após uma conversão bem-sucedida, todos os blocos como OB, FC, FB, DB, entre outros, bem como todas as configurações de *hardware*, foram gerados com compatibilidade com o *software* TIA Portal, mantendo assim o programa atualizado de acordo com as ferramentas de programação utilizadas atualmente. A Figura 37 mostra o resultado da migração do programa, onde se pode observar contornado a vermelho a CPU a ser utilizada e a opção de configuração de *hardware*, bem como os blocos que fazem parte do programa. Contornado a verde é apresentada uma imagem ilustrativa com a CPU utilizada e os módulos de expansão. Na parte inferior da imagem, contornado a azul é apresentada a configuração da CPU e seus módulos e, por fim, à direita e contornado a laranja, uma opção que permite selecionar outros *hardwares* disponíveis em catálogo.

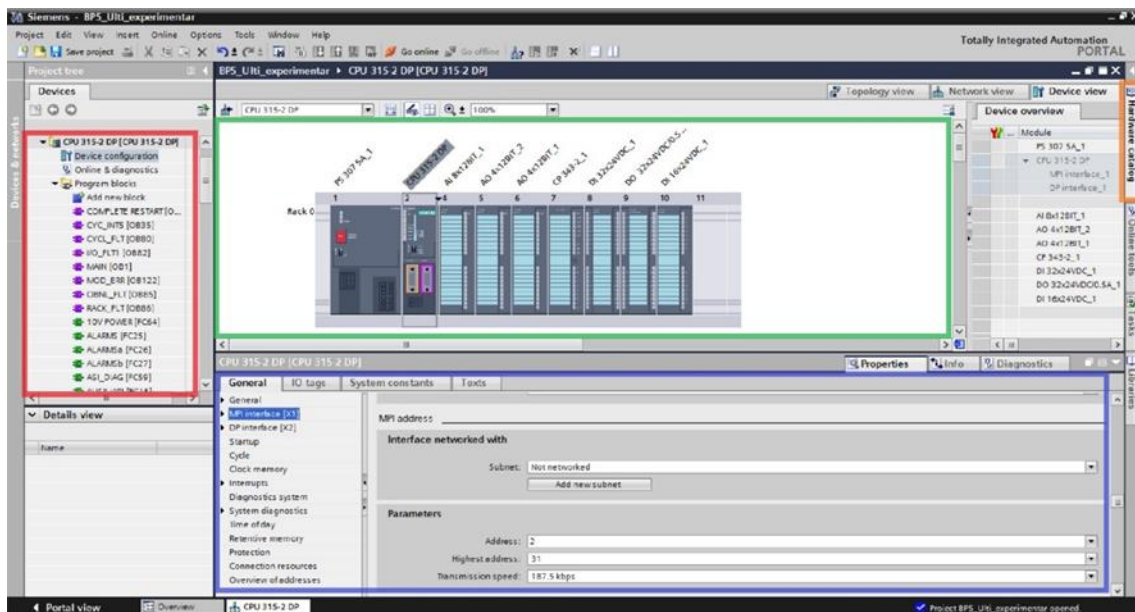


Figura 37 Interface TIA Portal do programa da máquina.

5.5. PROGRAMA DA MÁQUINA

Antes de criar qualquer tipo de função ou alteração ao programa é necessário perceber a forma como este está organizado e qual a sequência que segue, para desta forma se entender onde poderão ser encaixadas todas as alterações que forem consideradas. Na Figura 38 é apresentado um diagrama que mostra a hierarquia do programa da máquina.

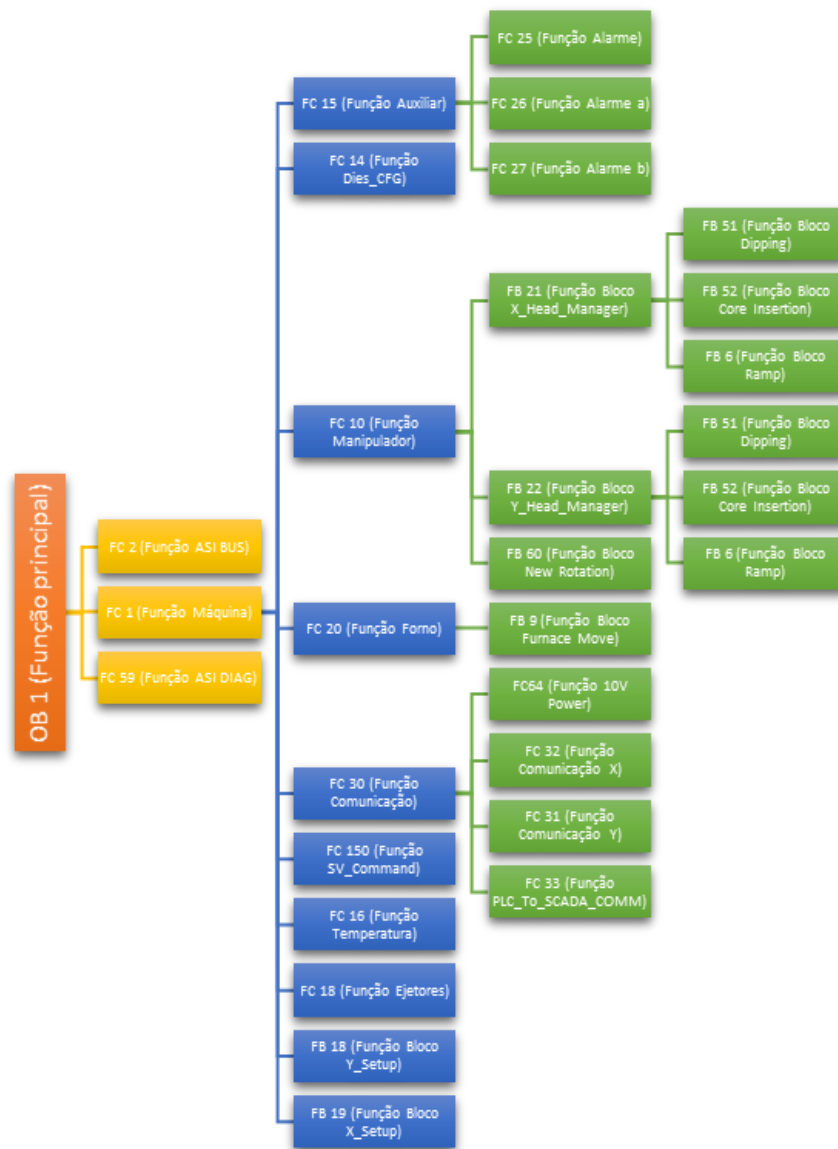


Figura 38 Diagrama da hierarquia das funções do programa.

OB 1 (Função Principal) – Os Blocos de Organização (OB) são responsáveis pelo controlo da execução do programa. Com os OB podem ser executadas respostas a eventos cíclicos, baseadas no tempo, ou interrupções durante a execução do programa. A OB1 é a função principal de um PLC SIEMENS, e sem ela não é possível programar o autómato. As restantes funções, sejam funções normais (FC) ou funções de blocos (FB) têm que estar ligadas à função principal.

FC 2 (Função ASI BUS) – Tem como principal função descodificar o barramento que interliga todos os componentes da máquina ao PLC.

FC 59 (Função ASI DIAG) – Usado para gravar os dados provenientes dos I/O, este bloco utiliza uma função chamada “RD_REC” para que esse propósito seja cumprido.

FC 1 (Função Máquina) – A função máquina funciona como uma função base pois chama todas as restantes funções. De um modo geral, esta função acaba por ser a função com “mais responsabilidade”, como pode ser visualizado no diagrama da Figura 38. As restantes funções (tanto FC como FB) estão todas ligadas à Função Máquina, que foi criada para uma melhor estruturação do programa.

FC 15 (Função Auxiliar) – Esta é a função que faz a gestão das funções Alarmes, como pode ser observado no diagrama da Figura 38.

FC 25, FC 26 e FC 27 (Funções de Alarme) – Estas funções são responsáveis por detetar falhas ao longo do funcionamento do sistema, estando programadas para esse fim, e interligadas a um *Data Block* (DB), onde são armazenadas informações em relação ao estado dos bits (bit a 0 tudo ok, bit a 1 sinal de alarme). Esta informação está ligada ao HMI que contém uma página dedicada ao estado dos alarmes, onde o operador pode visualizar se ocorreu algum tipo de erro ao longo do processo de operação.

FC 14 (Função *Dies_cfg*) – Guarda em memória quando é que a cabeça (x ou y) se encontra na posição de colocação ou de extração do macho de areia.

FC 10 (Função Manipulador) – Controla os movimentos da máquina, as rotações entre forno e tanque, e também os movimentos e rotações da coquilha. Sendo uma das funções principais, esta também é uma das mais complexas, visto que grande parte do controlo do processo é feito através desta função e das outras que lhe estão associadas.

FB 21 e FB 22 (Função Bloco X/Y *Head Manager*) – A programação desta FB foi realizada com o intuito de gerir/organizar os movimentos e rotações que dizem respeito à cabeça X/Y. Ao longo da função é evidenciada a estrutura dos movimentos e rotações, de acordo com as suas condicionantes e ordem pela qual se realizam. Ao mesmo tempo, esta função controla outras FB que simplificam a programação, dividindo determinadas tarefas em outras FB como, por exemplo, colocação/extração, banho de grafite e inclinação da bancada.

FB 51 (Função Bloco *Dipping*) – A FB 51 é programada de forma a realizar as tarefas às quais se encontra destinada. Este bloco está cheio de condições (neste bloco a programação é realizada através de *If/Else*, em linguagem PASCAL, deixando de ser LADDER como a

maior parte das anteriores) que ditam quando deve ser realizado o mergulho no banho de grafite.

FB 52 (Função Bloco *Core Insertion*) – Como a FB 51, este bloco também é programado à base de condições *If/Else* que ditam em que momento é que o macho de areia deve ser colocado ou extraído.

FB 6 (Função Bloco *Ramp*) – A função Bloco *Ramp* apenas dita quando é que a bancada deve ou não estar inclinada.

FB 60 (Função Bloco *New Rotation*) – Esta função gere as rotações da máquina, ou seja, este bloco é responsável por administrar as rotações realizadas entre forno e tanque ou vice-versa.

FC 20 (Função Forno) – Responsável por transmitir informações sobre o forno, tais como o peso e a quantidade de metal, e gere os movimentos realizados por este.

FB 9 (Função Bloco *Furnace Move*) – Esta função bloco controla os movimentos do forno, uma vez que este não é estático. O forno instalado na máquina tem a capacidade de subir, quando é necessário realizar o processo de vazamento, e uma vez este concluído desce. Esta função controla esses movimentos e dita se o forno se encontra em posição ou não.

FC 30 (Função Comunicação) – Estabelece a comunicação das cabeças X e Y com o HMI.

FC 31 e FC 32 (Função Comm_Y/X) – Acede a DB específicos para encontrar informações sobre a cabeça e transmite essa informação a outras variáveis que posteriormente poderão ser usadas como variáveis de acesso a um HMI.

FC 33 (Função PLC_To_Scada) – Função utilizada para saber quando está tudo pronto para efetuar o vazamento e quando este termina.

FC (Função 10V Power) – Ativa um alarme quando a tensão for igual ou inferior a 9 V.

FC 150 (Função SV_Command) – Envia informação relativa ao posicionamento dos equipamentos, nomeadamente se a cabeça X ou Y está no forno ou no tanque e se a bancada está inclinada ou na posição normal.

FC 16 (Função Temperatura) – Envia informações relativa à temperatura dos tanques e à temperatura do óleo da bomba.

FC 18 (Função Ejetores) – Função responsável por ativar/desativar os ejtores.

FB 18 e FB 19 (Função Bloco X/Y Machine_Setup) – Estas funções são responsáveis por estabelecer as configurações da máquina. São usadas duas funções distintas pois como a máquina opera com duas cabeças diferentes, que podem produzir peças diferentes, as configurações podem não ser as mesmas.

5.6. IX DEVELOPER DA BEIJER ELECTRONICS

De forma a ultrapassar os problemas de atualização, de *drivers* e sistemas operativos encontrados nos computadores, realizou-se um estudo em volta de um *software* SCADA já criado, o iX Developer da Beijer Electronics.

O *software* iX Developer é usado para configurar os iX Panels e as aplicações de controlo programados via PC, incluindo as aplicações para PC Industriais (IPC) da Beijer Electronics. O iX Developer contém todas as funções básicas necessárias para a criação de uma aplicação. As funções foram desenvolvidas tendo em mente as necessidades e as preferências dos utilizadores. Os objetos predefinidos no iX Developer podem ser usados para criar um processo completo de imagem, proporcionando uma visão geral de uma aplicação complexa. É possível personalizar os objetos predefinidos ou criar objetos próprios.

As consolas do operador podem ser conectadas a vários tipos de equipamentos de automação, tais como PLC, servos e unidades de sensorização, controlo e atuação. Para a troca de informação são usadas as *tags*, que podem pertencer ao sistema ou serem internas. As *tags* têm um nome simbólico e podem ser de diferentes tipos de dados. Objetos ligados às *tags* podem alterar valores no controlador, e os valores das *tags* podem ser refletidos ao alterar a aparência do objeto de várias maneiras.

No que diz respeito à comunicação entre o PLC e a consola da Beijer, teve que ser dimensionado um novo cabo, de acordo com o *datasheet* fornecido pela Beijer Electronics e que pode ser consultado no Anexo B. Ainda neste anexo são apresentadas as características da consola adquirida.

5.6.1. PROCESSO DE INSTALAÇÃO

O iX Developer é instalado num PC de desenvolvimento, em que podem ser criados, desenvolvidos e compilados projetos. O projeto é, de seguida, executado numa consola ou PC para observar e controlar um controlador (ou grupo de controladores).

Para tal, o PC utilizado deve obedecer a um conjunto de requisitos para que este possa correr o *software* de desenvolvimento. Na Tabela 3 são apresentados os requisitos recomendados.

Tabela 3 Requisitos recomendados para a instalação do iX Developer.

Parâmetros	Recomendação	
RAM	2 GB	
Processador	2 GHz ou superior	
Sistema Operativo	Microsoft Windows 7	
	Microsoft Windows Vista	
	Microsoft Windows XP SP3	
Placa de vídeo	Nível de Renderização 2	
	Versão do DirectX	Deve ser maior ou igual a 9.0
	RAM de Vídeo	Deve ser maior ou igual a 120 MB
	<i>Pixel shader</i>	Nível da versão deve ser maior ou igual a 2.0
	<i>Vertex shader</i>	Nível da versão deve ser maior ou igual a 2.0
	Unidades multitextura	O número de unidades deve ser maior ou igual a 4

São necessárias versões específicas de *software* para alguns objetos serem incluídos no projeto iX Developer. A simulação do projeto num PC de desenvolvimento pode também estar limitada. Por exemplo, é necessário o Windows Media Player para reproduzir áudio ou vídeo na consola, no entanto não é possível efetuar simulações no PC.

Após a instalação é criado um ícone para a ferramenta de configuração, no grupo chamado iX Developer, no menu iniciar do Windows.

5.6.2. CRIAÇÃO DE UM NOVO PROJETO

Ao iniciar a aplicação iX Developer aparecerá ao utilizador uma janela *wizard*, como pode ser visualizado na Figura 39, com várias opções. Do lado esquerdo estão disponíveis opções como a criação de um novo projeto, abrir projeto guardado, carregar projeto que esteja dentro de um *target* (consola) e uma opção que ajuda o utilizador a iniciar a ferramenta pela primeira vez (*User's guide*). Do lado direito é apresentada uma secção com o nome dos projetos usados recentemente, o que permite uma rápida inicialização e continuação do projeto.

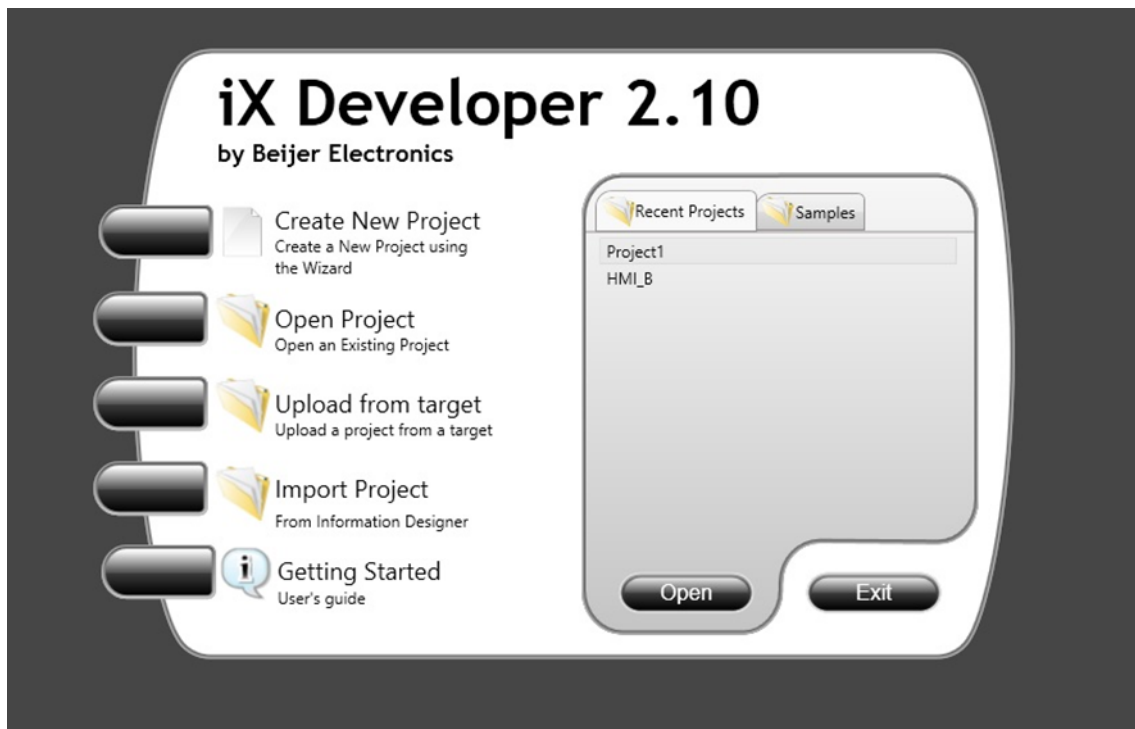


Figura 39 Janela inicial do iX Developer.

Ao seleccionar a opção “*Create New Project*”, surgirá ao utilizador uma nova janela, apresentada na Figura 40, com os vários modelos de consolas disponibilizadas pela Beijer Electronics. Para a realização do projeto da BP260 X, a consola usada foi a iX T15B.

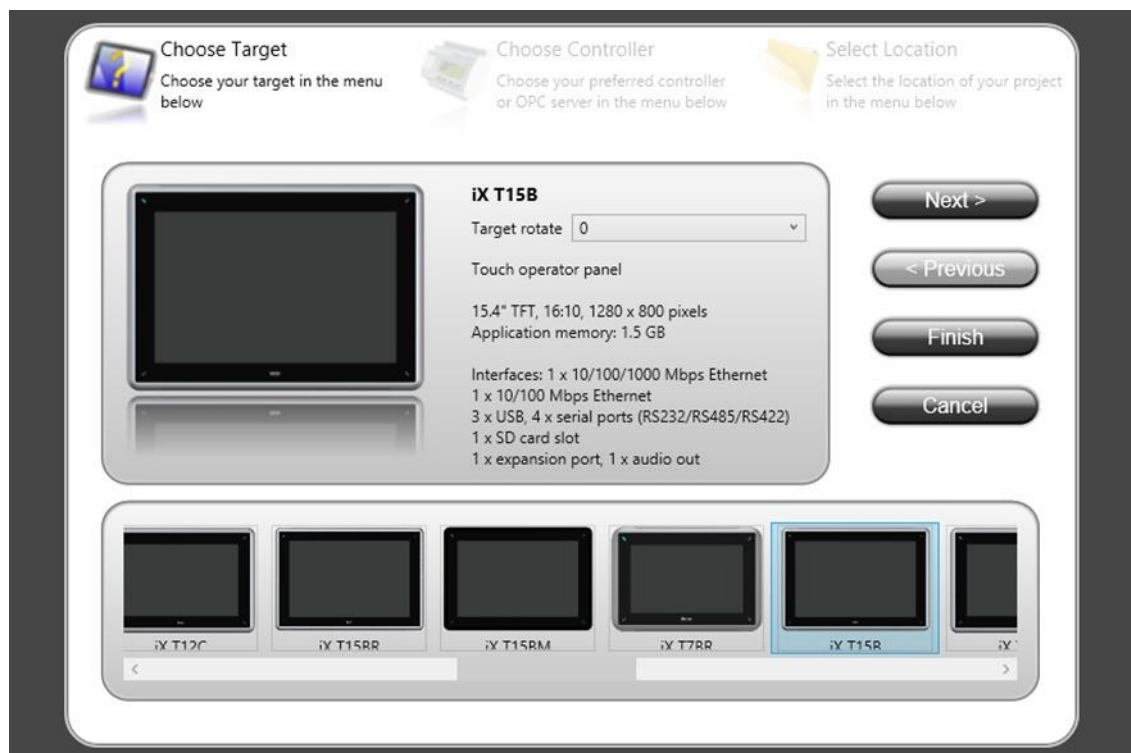


Figura 40 Janela de seleção do tipo de consola a utilizar.

Depois de selecionado o tipo de consola pretendida, é necessário selecionar o tipo de controlador com que esta irá interagir. Dessa forma, na Figura 41 mostra-se a janela para a seleção do controlador ou OPC Server. No caso da BP260 X, como esta é controlada por um S7-315-2 DP e o novo PLC a instalar também é um S7-300, o tipo de controlador será SIMATIC e será usado o S7 MPI Direct como protocolo. Por fim basta escolher o nome que será atribuído ao projeto, assim como a localização do ficheiro gerado.

Findas estas configurações iniciais, surge a área de trabalho mostrada na Figura 42 e descrita na Tabela 4. A área de trabalho mostra janelas e páginas de configuração para componentes do projeto, tais como controladores e funções, mas apenas mostra uma janela ou componente de cada vez. Quando várias janelas ou componentes são abertos, é exibida uma sequência de guias na parte superior da área de trabalho. Ao clicar numa dessas guias o respetivo conteúdo é posto em primeiro plano.

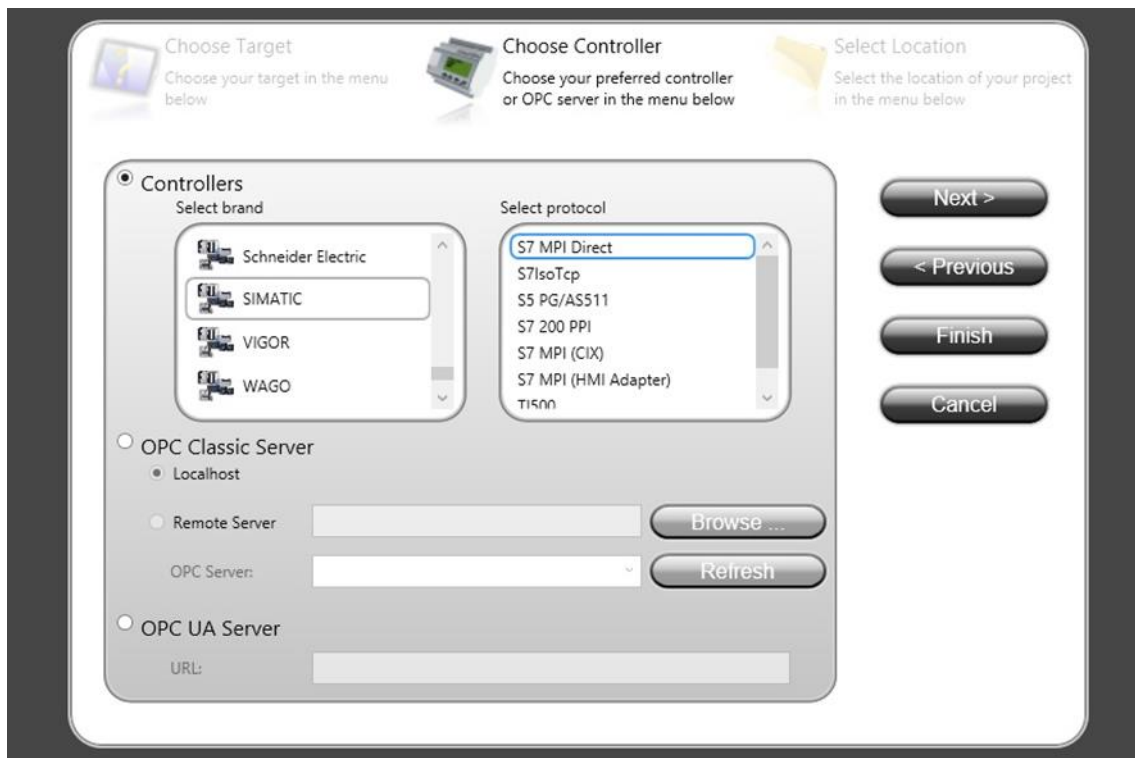


Figura 41 Janela para a escolha da família do controlador e respetivo protocolo de comunicação.

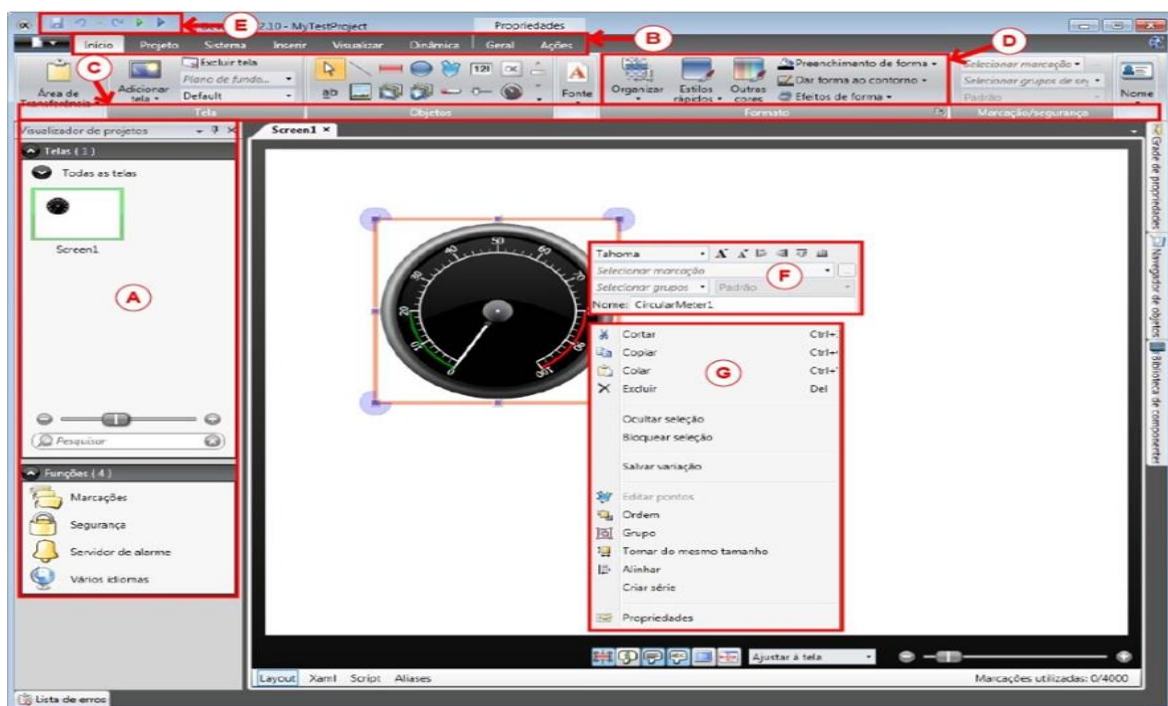


Figura 42 Área de trabalho do iX Developer.

Tabela 4 Descrição da área de trabalho do iX Developer.

Indicação em imagens	Componente da área de trabalho	Descrição na secção
A	Visualizador de projetos	Visualizador de projetos
B	Guias da faixa de opções	Grupos e controlos da faixa de opções
C	Grupos de controlo	
D	Controlos	
E	Barra de ferramentas de acesso rápido	Barra de ferramentas de acesso rápido
F	Minibarra de ferramentas	Minibarra de ferramentas e menu de contexto
G	Menu de contexto	

De acordo com a Figura 42, quando é aberto um novo projeto, fica ativa uma janela em branco na área de trabalho. O Visualizador de projetos está encaixado à esquerda.

As guias da faixa de opções estão localizadas na secção superior da janela de ferramentas. Cada guia da faixa de opções possui um ou vários grupos de controlo. Cada grupo possui um conjunto de controlos relacionados, que são usados para criar projetos e fazer configurações de objetos.

A barra de ferramentas de Acesso Rápido está sempre visível na parte superior da área de trabalho. Esta contém os comandos ‘Salvar’, ‘Desfazer’, ‘Refazer’, ‘Executar’ e ‘Simular’.

Ao clicar com o botão direito do rato no iX Developer são exibidos uma minibarra de ferramentas e um menu de contexto. A minibarra de ferramentas contém comandos específicos para o iX Developer como, por exemplo, conectar objetos às *tags* do controlador. O menu de contexto contém comandos normais, tais como copiar, colar, etc.

5.6.3. TAGS DO CONTROLADOR

Os objetos conectados às *tags* podem alterar valores no controlador, e os valores das *tags* podem ser refletidos ao alterar a aparência do objeto de várias maneiras. Os objetos permanecerão estáticos no ecrã até que sejam conectados a uma *tag*. A Figura 43 mostra a janela de criação das *tags* onde, ao ser inserida uma nova *tag*, é feita a atribuição do nome, é definido o tipo de dados, os direitos de acesso (*Read*, *Write*, *ReadWrite*) e a respetiva correspondência com a *tag* do controlador.

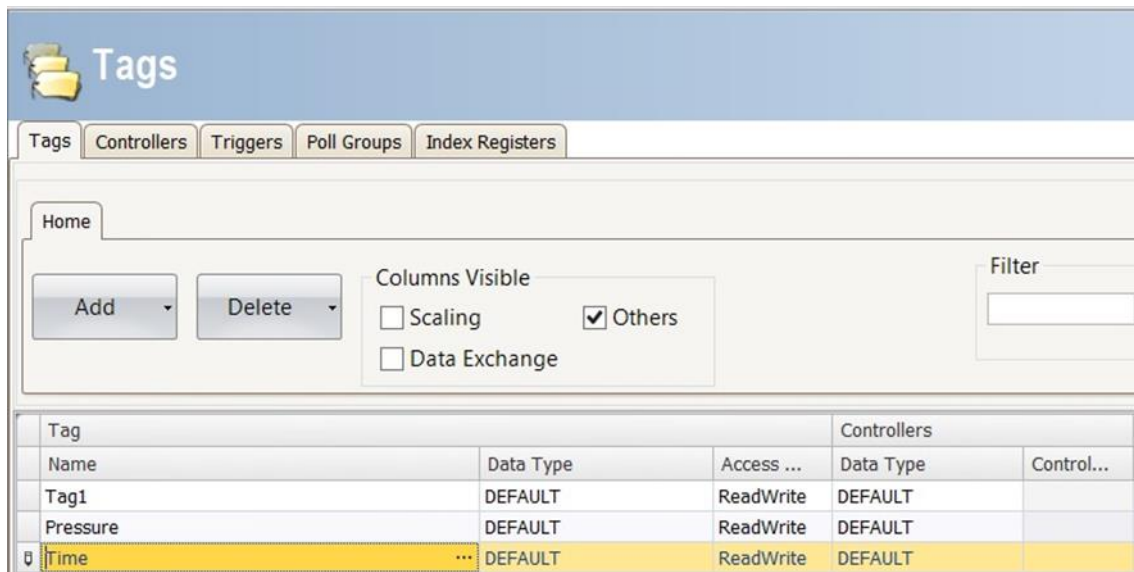


Figura 43 Menu de declaração das tags da consola.

5.6.4. NAVEGAÇÃO E SALTOS ENTRE ECRÃS

Um projeto do iX Developer é constituído por ecrãs com objetos, normalmente conectados por *tags* do controlador. Todos os ecrãs apresentam as mesmas funções básicas. É possível atribuir propriedades específicas a um ecrã para que este tenha um comportamento especializado dentro do projeto, como sejam:

- **Startup Screen** – O ecrã de inicialização é o primeiro a ser exibido durante o tempo de execução. Por padrão, a *Screen1* é utilizada como ecrã principal; no entanto, qualquer ecrã pode ser designado como ecrã de inicialização, bastando clicar com o botão direito do rato e selecionar “*Set as Startup Screen*”.
- **Background Screen** – Qualquer ecrã, exceto os ecrãs com *aliases*, podem ser usados como ecrã de fundo por outros ecrãs do projeto.
- **Screen template** – Um ecrã que é guardado como um *screen template* pode ser utilizado, não só pelo projeto atual, como também em futuros projetos.

Os saltos entre ecrãs são feitos com ações que podem ser atribuídas a, por exemplo, botões. Ao utilizar o “*Navigation Manager*” para adicionar ecrãs e criar *links* entre estes, são adicionados automaticamente botões no canto superior esquerdo do ecrã onde o *link* teve origem.

De acordo com a Figura 44, após ser adicionado um novo ecrã, para fazer o *link* de ligação entre os dois ecrãs através de um botão basta selecionar o botão e aceder à guia “*Actions*”

onde serão exibidas uma série de ações que poderão ser atribuídas a esse botão. De frisar que o iX Developer permite que os objetos tenham a capacidade de ações múltiplas, isto é, os objetos podem ter associadas várias ações, e não apenas uma.

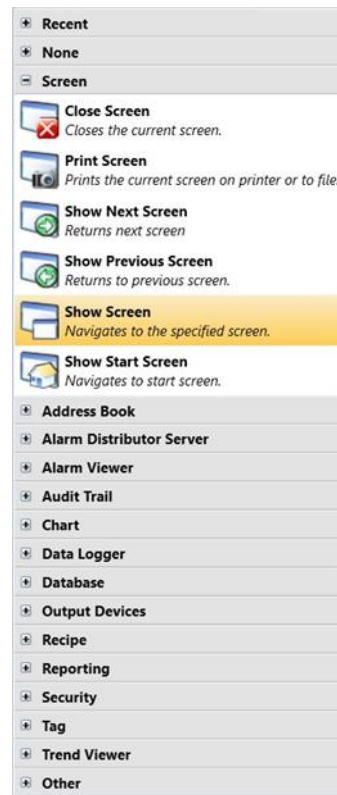


Figura 44 Menu de ações.

5.6.5. GESTÃO DE ALARMES

Os alarmes são utilizados para fazer com que o operador esteja ciente dos eventos que exigem ações imediatas. Um alarme é ativado quando determinadas condições são satisfeitas. Uma condição de alarme foi projetada como uma avaliação lógica de um valor da *tag*. Os alarmes podem ser divididos em grupos e criados em ordem de prioridade.

Quando um alarme é ativado, o *Alarm Indicator* (indicador de alarme) fica visível para alertar o operador, independentemente ao ecrã ativo. A aparência do alarme depende do seu estado atual. Selecionando a configuração “*General*”, na página de propriedades do servidor de alarme, é possível decidir quando é que o alarme deve ser ativado.

O indicador de alarme mostra o alarme mais grave na lista de alarmes com as seguintes indicações:

- Sinalizador vermelho, que pisca quando há qualquer alarme ativo do qual ainda não se tomou conhecimento.

- Sinalizador amarelo, que pisca quando não existe nenhum alarme ativo, mas existem alarmes inativos dos quais ainda não se tomou conhecimento.
- Sinalizador verde, que pisca quando há apenas alarmes ativos conhecidos.

O indicador de alarme desaparece quando todos os alarmes forem reconhecidos e estiverem de volta ao estado inativo.

Acedendo à guia *Alarm Items* no *Alarm Server*, como apresentado na Figura 45, é possível adicionar alarmes com base nas *tags* da lista de *tags* do controlador. As colunas “*Acknowledge Required*” e “*History*” devem estar marcadas, e devem ser introduzidas as condições necessárias para que os alarmes sejam ativados em caso de emergência.

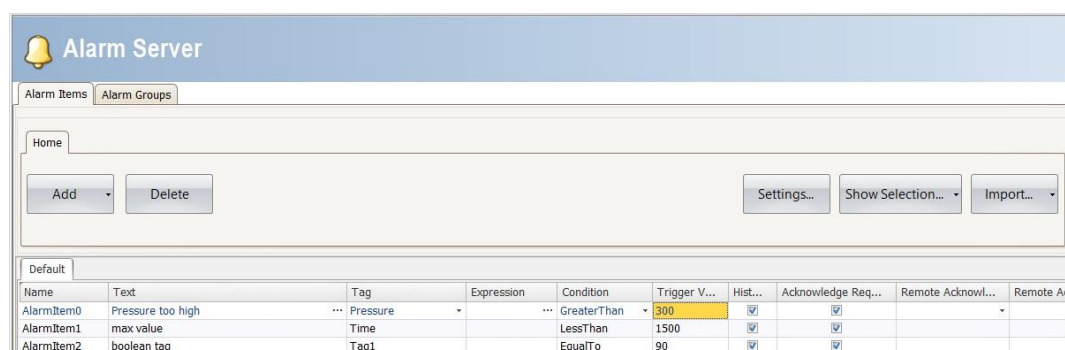


Figura 45 Servidor de alarmes do iX Developer.

O iX Developer dispõe de um ecrã, o “*Alarm Viewer*”, que permite visualizar todos os alarmes. Como pode ser verificado na Figura 46, o “*Alarm Viewer*” permite fazer o reconhecimento dos alarmes, individualmente ou todos de uma vez, apagar, filtrar e obter mais informações. Na janela de visualização é possível verificar o nome do alarme, o estado, uma breve descrição deste, o tempo em que se encontrou ativo, o tempo em que se encontrou inativo e também o instante de tempo em que foi reconhecido.

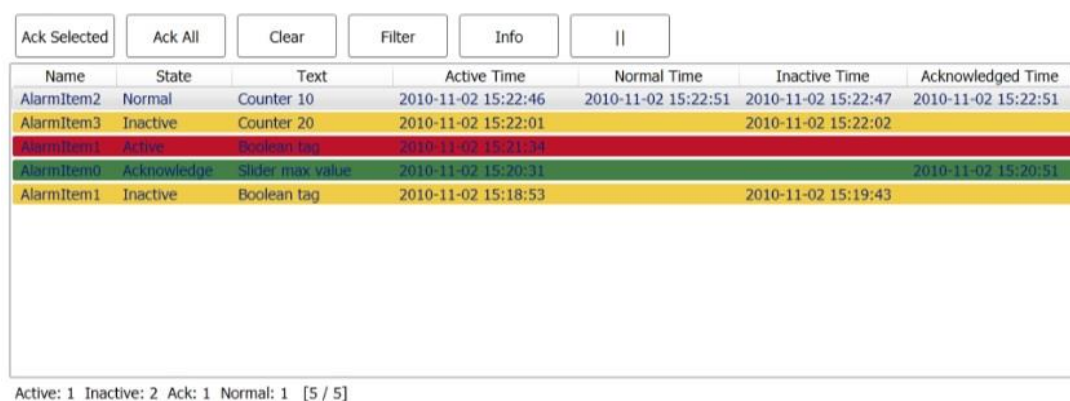


Figura 46 Alarm Viewer do iX Developer.

6. IMPLEMENTAÇÃO PRÁTICA

Na sequência do estudo realizado no capítulo anterior, e após o levantamento de todos os problemas, concluiu-se que para a implementação prática o mais viável seria realizar o controlo de vazamento através do PLC que controla a máquina, ligando o transdutor e a válvula ao controlador. Como o estudo da programação exigiu mais tempo do que inicialmente previsto, adquiriu-se uma consola da Beijer Electronics, juntamente com o software de programação para desenvolver o HMI, em vez de criar um sistema SCADA através de uma linguagem informática. Neste capítulo será explicada a programação do PLC e do HMI.

6.1. ARQUITETURA DO NOVO SISTEMA

O novo sistema a ser implementado (Figura 47) exclui o controlo do processo de vazamento realizado pelo PC, passando este a ser realizado pelo PLC. Neste novo sistema o PLC passa a ser a unidade de controlo central, visto que todos os equipamentos necessários estão ligados ao controlador, assim este passa a ser responsável por todo o controlo, quer da máquina quer do processo de vazamento.

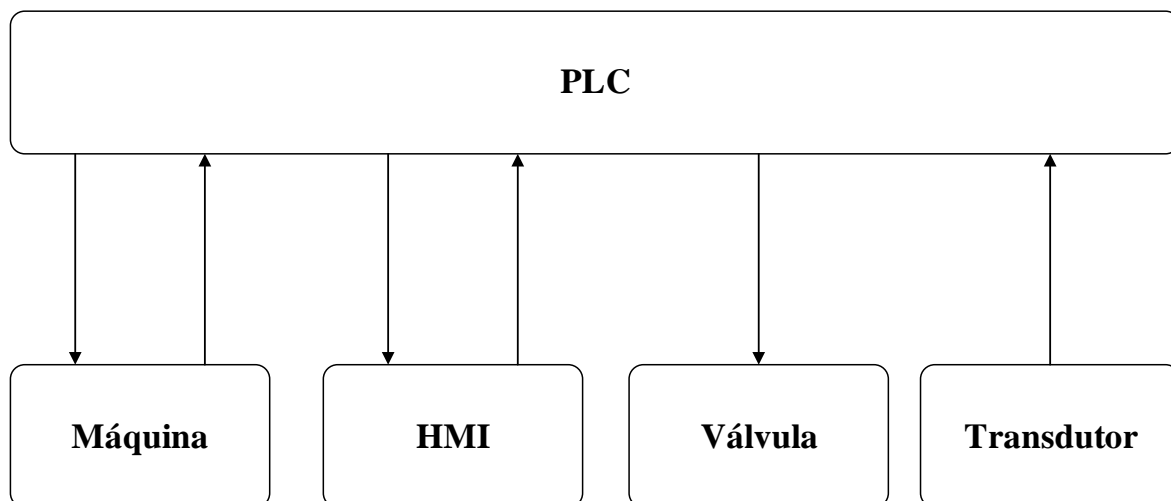


Figura 47 Arquitetura do novo sistema.

6.2. PROGRAMAÇÃO PARA O CONTROLO DE VAZAMENTO

Neste projeto, o controlo de vazamento é realizado quando uma das cabeças da máquina está na posição “forno”. Após o PC receber a indicação do PLC que todas as condições para iniciar o vazamento foram verificadas, a válvula começa a injetar ar para dentro do forno, de acordo com uma tabela que relaciona pressão com o tempo, para que dessa forma seja criada a curva de vazamento predefinida. O acesso à programação do controlo de vazamento do PC não foi possível, de modo que se tornou mais difícil a perceção de como o controlo era realizado.

Assim, o pensamento que levou ao desenvolvimento do novo controlo, estabeleceu que a válvula deveria receber valores de *setpoint* de pressão a intervalos pré-definidos de x segundos. De acordo com as tabelas das várias curvas de vazamento, todos os valores de pressão são estabelecidos a intervalos de 1 s, ou seja, para a nova função criada a válvula recebe, valores diferentes, de *setpoint* a cada 1 s.

Da mesma maneira, o transdutor lê os valores de pressão que se encontram no forno a cada 1 s. Para que este tipo de controlo seja possível, é utilizada uma função PID (função bloco já desenvolvida pela Siemens), em que se utilizam os valores do *setpoint* como parâmetro de entrada e do transdutor como valor de realimentação. Depois que seja aplicado o controlo PID, faz-se atuar a saída que está ligada ao endereço analógico da válvula. A sintonia do controlador PID é feita à parte, e será explicada detalhadamente mais à frente.

No PLC estão criados dois *Data Blocks* (DB), um referente aos valores de pressão para a cabeça X e outro para a cabeça Y da máquina.

De forma a clarificar a programação efetuada, na Figura 48 é apresentado um fluxograma de como a função criada funciona, de maneira a controlar o processo de vazamento.

Para ajudar na programação do PLC, a Figura 49 mostra o Grafcet nível 1 criado relativo ao funcionamento do controlo de vazamento.

De modo a facilitar a manipulação de dados, para a programação da nova função foi usada a linguagem por instruções (IL), mais conhecida por *Statement List* (STL) dentro dos PLC da SIEMENS.

De seguida é apresentado todo o código imposto na nova função de vazamento criada, a FC3 “Controlo_Vazamento”.

De acordo com a Figura 50, a função criada começa por inicializar os endereços dos ponteiros e do valor de *setpoint* com o valor zero, para garantir que não ocorra sobreposição de dados. No caso dos ponteiros (#Ponteiro_DB e #Ponteiro_DB2), se estes não fossem inicializados a zero, o valor podia ir acumulando o que poderia trazer erros no acesso às áreas de memória do DB, visto que acabaria por tentar aceder a endereços inexistentes.

No que diz respeito à variável “SP” (variável de *setpoint*), por precaução também foi inicializada a zero para garantir que o envio dos valores seja efetuado corretamente.

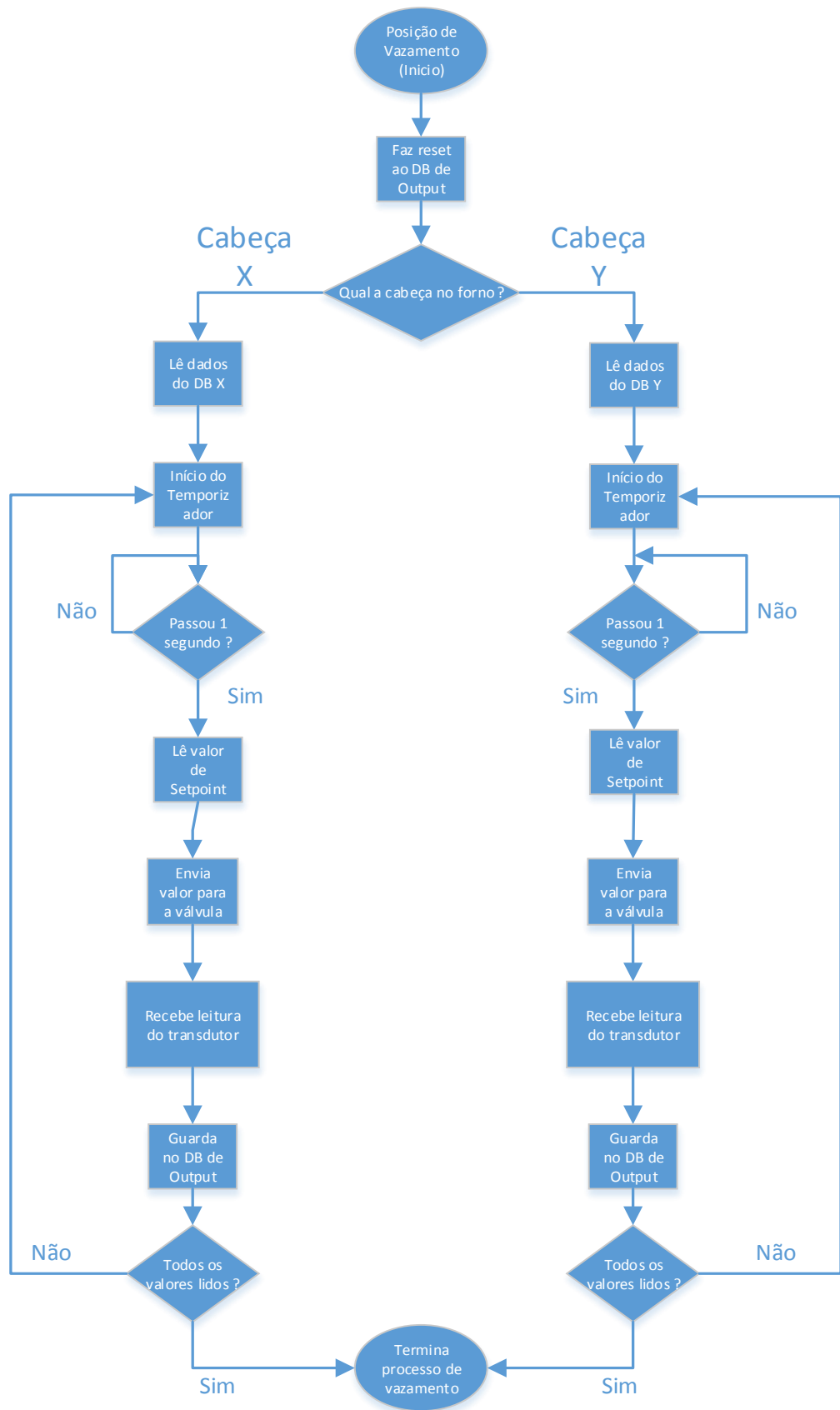


Figura 48 Fluxograma da função de controle de vazamento.

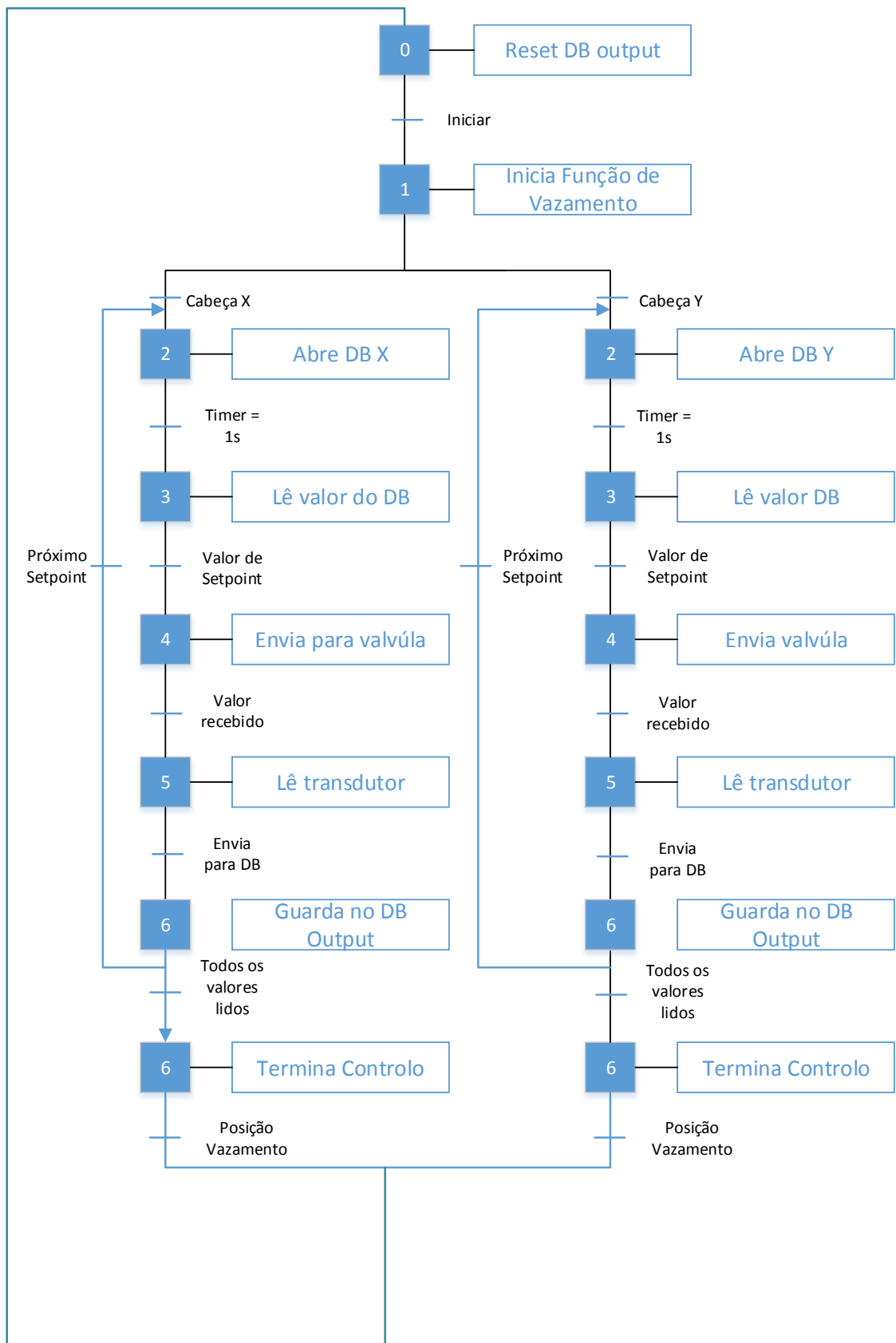


Figura 49 Graficet nível 1 para a função de controle de vazamento.

Network 1: Condições de utilização desta função pela primeira vez (valores dos ponteiros a "0")			
Comment			
1	A	"Bit_condições_iniciais"// bit necessario para para fazer avançar as condições iniciais	%M0.1
2	JC	Avança	
3	L	0 // inicia o endereço do ponteiro a zero	0
4	T	#Ponteiro_DB	
5	L	0 // inicia o endereço do ponteiro a zero	0
6	T	#Ponteiro_DB2	
7	L	0 // inicia o endereço de setpoint a zero	0
8	T	"SP"	%MD150
9	S	"BIT_Start_Timer" // bit para iniciar o timer	%M0.2

Figura 50 Inicialização de variáveis.

Estando as variáveis necessárias devidamente inicializadas, é preciso programar a função para aceder a *Data Blocks* diferentes, consoante a cabeça da máquina que esteja no forno, assim como configurar um temporizador de modo a que a leitura de cada valor seja processada em intervalos de 1 s. Na Figura 51 é apresentado o código da *Network 2* da função criada.

Network 2: Configura o Temporizador para enviar valores de 1 em 1 segundo			
Comment			
1	Avança : L	#Ponteiro_DB	
2	L	#Ponteiro_DB2	
3			
4	anf: A	"FCX_HEAD_ON_FURNACE"// Verifica se a cabeça X está no forno	%M142.1
5	JNB	nDB // se não estiver salta para o endereço nDB	
6	OPN	"Dados da Peça X" // abre o DB onde estão os valores da pressão da cabeça X	%DB1
7	L	"Offset_X"	%MW158
8	T	"entrada_analogica_PID"	%MW236
9	nDB: A	"FCX_HEAD_ON_TANK" // Verifica se a cabeça Y está no forno	%M142.0
10	JNB	nDB2	
11	OPN	"Dados da Peça Y" // abre o DB onde estão os valores da pressão da cabeça Y	%DB5
12	L	"Offset_Y"	%MW202
13	T	"entrada_analogica_PID"	%MW236
14	nDB2: NOP	0	
15	L	SST#1S // Carrega o Timer com o valor de 1 segundo	SST#1S
16	A	"BIT_Start_Timer" // este bit garante que o timer é renicializado	%M0.2
17	SE	"Temporizador Setpoint"// Ativa o Timer	%T0
18	L	"Temporizador Setpoint"// Carrega o valor do timer	%T0
19	L	SST#10ms // Carrega o valor de 10ms	SST#10ms
20	==I	// compara o valor do timer com o valor de 10ms	
21	R	"BIT_Start_Timer" // quando a RLO = 1 faz reset a este bit	%M0.2
22	L	"Temporizador Setpoint"	%T0
23	L	0	0
24	==I	// compara o valor do timer com o valor zero	
25	S	"BIT_Start_Timer" // se a RLO = 1 ativa este bit	%M0.2
26	JCN	Next // Salta para o fim da função	

Figura 51 Configuração do tempo para o envio de valores.

De modo a gerir qual o *Data Block* a ser lido, foram usados dois bits já existentes na programação da máquina. O bit "FCX_HEAD_ON_FURNACE" e o bit "FCX_HEAD_ON_TANK" são usados ao longo de toda a programação da máquina de forma a distinguir as posições da cabeça X ou Y. O "FCX" é referente à cabeça X e quando o bit "FCX_HEAD_ON_FURNACE" está a 1, significa que a cabeça X se encontra no forno; em contrapartida, quando o bit "FCX_HEAD_ON_TANK" está a 1 significa que a cabeça X está na posição de tanque e que a cabeça Y está no forno. Estes dois bits são os

mais indicados para esta gestão, pois nunca poderão estar ambos com o valor lógico 1 ao mesmo tempo.

De volta à programação da função, usando estes dois bits, foram criadas condições que permitissem gerir o comportamento da função de acordo com a cabeça que se encontre no forno, ou seja, quando a cabeça X está no forno é garantido o acesso a um DB, mas quando é a cabeça Y a estar no forno o acesso é feito a outro DB, com outros dados.

É necessário configurar o temporizador, criando também condições que facilitem a fluidez do programa como, por exemplo, a criação de uma sequência de código que garanta que o temporizador é reinicializado de cada vez que um novo valor seja lido. No excerto de código apresentado na Figura 51, a configuração do temporizador começa por carregar o valor 'L S5T#1S' em memória - esta instrução define o tempo a ser usado pelo temporizador (1 segundo). De forma a garantir que o temporizador já chegou a zero, é criada uma condição que vai comparando o valor do *timer* com o valor carregado em memória (zero) e quando estes forem iguais o controlo de vazamento continua. Caso não sejam, sai fora da função voltando a entrar no próximo ciclo de processamento, onde irá voltar a verificar esta condição e assim sucessivamente (esta condição teve que ser criada pois como o processamento de um PLC é cíclico, e o do S7-314 ronda os 11 ms, é necessário que a leitura cíclica continue mesmo que o temporizador ainda não tenha chegado a zero).

Para efetuar a leitura do DB é usado o endereçamento indireto (Figura 52), mediante o uso de apontadores que acedem a áreas de memória específicas e permitem fazer a leitura dos dados desejados.

```
Network 3: Endereçamento indirecto para ler dados de pressão da peça
Comment
1 L #Ponteiro_DB // carrega o apontador
2 SLW 3 // move o valor do apont. 3 bit para a esquerda (1Byte)
3 T #Posição_apontada // guarda esse valor nesta variavel
4 L DBW [ #Posição_apontada] // Lê o valor contido neste endereço
5 T "HMI" // %MD154
6 DTR // converte o valor do tipo Dint para Real
7 T "conv_real" // carrega valor convertido para esta variavel // %MD168
8 L 10.0 // 10.0
9 /R // divide o valor REal por 10.0
10 T "SP" // guarda na variavel "SP" // %MD150
```

Figura 52 Configuração para a leitura dos valores de *setpoint*.

Para o endereçamento indireto é utilizada uma instrução especial, denominada 'Shift left word by word' (SLW). A instrução 'Shift Left Word by Word' muda a *word* (bits 0 a 15) do acumulador 1, bit por bit para a esquerda. As posições de bits livres são preenchidas com

zeros. Os bits 16 a 31 do acumulador 1 permanecem inalterados. No Anexo D estão presentes informações sobre o endereçamento indireto e a instrução SLW.

O uso da instrução ‘SLW 3’ faz com que sejam movidos 3 bits para a esquerda. O resultado dessa operação é guardado numa nova variável, que será usada para identificar o *offset* do valor dentro do DB. A instrução ‘L DB [#Posição_apontada]’ carrega o valor do endereço do DB apontado pela variável #Posição_apontada.

À medida que é efetuada a leitura de cada valor de pressão, é feita a escrita para um novo DB, o DB de *Output*. A Figura 53 mostra a *Network 4* que apresenta o código utilizado para a escrita dos valores lidos pelo transdutor.

Network 4: Endereçamento indirecto para escrever valores do transdutor			
Comment			
1	OPN	"Output_PID" // abre o DB Output PID	%DB2
2	L	#Ponteiro_DB2 // carrega o apontador	
3	SLW	3 // move o valor do apont 3 bit para a esquerda	3
4	T	#Pos_apontada_escrever//guarda esse valor nesta variavel	
5	L	"transdutor_mBAr"	%MD194
6	T	DBW [#Pos_apontada_escrever] // escreve o valor no na posição apontada do DB Output PID	

Figura 53 Escrita dos valores de pressão lidos pelo transdutor.

De forma análoga ao que sucede na leitura dos valores, a escrita também é realizada através de um apontador, que mediante a posição apontada escreve o valor presente no transdutor de pressão para o endereço do DB de *Output* apontado.

A *Network 5*, apresentada na Figura 54, apresenta o código que é responsável por incrementar o valor dos apontadores (leitura e escrita) a cada segundo. O valor de incremento é 2 porque os dados dos DB são do tipo inteiro (*int*), ou seja de 16 bit ou 2 Byte, o que faz com que o incremento tenha o valor de 2. Por exemplo, se a primeira posição tem endereço ‘DB1.DBW0’, a segunda posição irá ser ‘DB1.DBW2’.

Network 5: Incremento da posição dos apontadores			
Comment			
1	L	2	2
2	L	#Ponteiro_DB	
3	+I	// incremento do valor apontado para o DB leitura de dados	
4	T	#Ponteiro_DB	
5	L	2	2
6	L	#Ponteiro_DB2	
7	+I	// incremento do valor apontado para o DB escrita de dados	
8	T	#Ponteiro_DB2	
9			

Figura 54 Incrementação da posição dos apontadores.

As duas últimas *Networks*, a 6 e 7, presentes na Figura 55, apresentam código que verifica se todas as posições do DB de leitura já foram percorridas e, consoante a condição, fazem a gestão da função, ou seja, se todas as posições já tiverem sido percorridas é efetuado *reset* ao bit que controla a função de vazamento (por outras palavras, termina a fase de vazamento). Caso ainda não tenham sido percorridas todas as posições do DB, foi adicionada uma instrução ('JCN anf') que faz um "salto" para o início da função fazendo a leitura e escrita do próximo valor.

Esta instrução de salto integra uma etiqueta 'anf' que é responsável por indicar onde irá continuar o processamento do programa. Esta etiqueta é visível na *Network 2*, ou seja, se todas as posições do DB ainda não tiverem sido percorridas o programa irá continuar a partir desta etiqueta e continuará a efetuar os saltos necessários até que a condição anterior (<=I) seja verdadeira.

Na *Network 7* é realizada mais uma comparação para garantir que todas as posições do DB foram realmente percorridas. Também é visível a etiqueta 'Next' que faz com que o ciclo de processamento continue enquanto o temporizador da *Network 2* não chegar ao fim.

```

Network 6: Varrimento do DB
Comment
1 L #Ponteiro_DB
2 L #Tamanho_DB
3 <=I // confirma se o apontador já percorreu todos os endereços do DB
4 JCN anf
5

Network 7: .....
Comment
1 L #Tamanho_DB
2 L #Ponteiro_DB
3 ==I // quando atingir o ultimo endereço faz reset à utilização desta função
4 R "BIT_Inicia_vazamento"
5 Next: S "Bit_condições_iniciais"
6
7

```

Figura 55 Condições de varrimento do DB.

Apesar do FC3 "Controlo_Vazamento" ser a principal função para o controlo do processo de vazamento, foi necessário criar outras funções que contribuíssem para o correto funcionamento desta operação. Assim, foram criadas mais quatro funções. Sempre que é iniciado um novo processo de vazamento, o FC4 "RESET_DB" certifica-se de colocar a zero todos os endereços do DB "Output". O FC6 "transdutor" e o FC7 "válvula", onde se encontram todos os cálculos realizados para a conversão do tipo de dado *word* (o valor recedido/enviado pelo transdutor/válvula é no formato *word*), para valores percentuais ou em milibar, para serem apresentados ao utilizador, ou para efetuar os cálculos do bloco PID.

Por último, o FC5 “Gestão Vazamento”, faz a gestão de quando e em que condições o processo de vazamento deve ser ativado. De seguida é apresentado o código e explicações detalhadas sobre cada uma das funções referidas.

A FC4 “RESET_DB” foi criada com o objetivo de colocar os vários endereços do DB de *Output* a zero, também recorrendo ao uso de apontadores para fazer a escrita do valor zero nos vários endereços. O código presente na Figura 56 ilustra a sua funcionalidade. No entanto, em vez de escrever o valor zero de segundo em segundo é carregado, no início da função, o número de endereços em memória, e o carregamento do valor é feito em apenas um ciclo de processamento da CPU (cerca de 11 ms). A substituição dos valores anteriores por zero tem que ser feita a esta cadência para garantir que não há nenhum valor diferente de zero no DB, antes que o transdutor comece a enviar os seus valores.

```
Network 1:
Comment
1 L 16 16
2 LOP5: T #Times
3
4 //Destino
5 L #Pos_Destino
6 SLW 3 3
7 L P#0.0
8 OW
9 T #Point2
10 L 0 0
11 OPN #N_DB_Destino
12 T DBW [#Point2]
13 L #Pos_Destino
14 + 2 2
15 T #Pos_Destino
16
17 L #Times
18 LOOP LOP5
```

Figura 56 Código do FC4 “RESET_DB”.

Para a chamada desta função tiveram que ser definidas duas variáveis do tipo entrada, nomeadamente a variável #Pos_Destino, que irá receber o valor introduzido para fazer o *reset* (valor ‘0’), e a variável #N_DB_Destino, que recebe o nome do DB a ser afetado.

A função FC6 “Transdutor” é responsável por receber a informação do transdutor de pressão e convertê-la para dados perceptíveis, ou seja, o valor obtido pelo transdutor não é claro, ou seja, o valor retornado vem no formato de *word*, daí que seja necessário converter esse valor para uma nova unidade, neste caso para milibar ou valores percentuais. Todos os cálculos efetuados estão presentes nas Figura 57 e Figura 58, apresentadas a seguir.

Na Figura 57, o valor analógico é recebido através do endereço %IW260 : P (PIW260). Este endereço faz parte do módulo analógico AI 8x12BIT ligado à CPU que, posteriormente, é

enviado para outro endereço de memória para facilitar todas as conversões e cálculos que se seguem.

Como o valor é recebido no formato inteiro (`int`) e para um correto controlo após todos os cálculos, é necessário converter esse valor para o formato real. As duas *Networks* seguintes (2 e 3) convertem primeiro o valor da variável para o formato de duplo inteiro (`Dint`) e depois para o formato Real, pois a conversão não pode ser direta (não é permitido converter de inteiro diretamente para real).

Depois do valor ser convertido para o formato Real, converteu-se o valor recebido pelo transdutor para um valor percentual e também para um valor em milibar (`mb`). A base para as conversões efetuadas foi uma regra dos três simples, pois testou-se qual o valor máximo recebido pela variável `PIW260` (27648,0), no entanto como o valor máximo deste transdutor corresponde a 2 bar, foi necessário multiplicar o valor recebido por dois, como é mostrado na *Network 1* da Figura 57, para assim ficar com uma escala igual à da válvula e mostrar os valores certos. O novo valor obtido após a multiplicação corresponde então a 100%. Assim o cálculo efetuado será a variável convertida para real a dividir por 27648,0 (valor máximo de uma `word`), sendo que o resultado obtido será multiplicado por 100,0 e daí será obtido o valor percentual do transdutor.

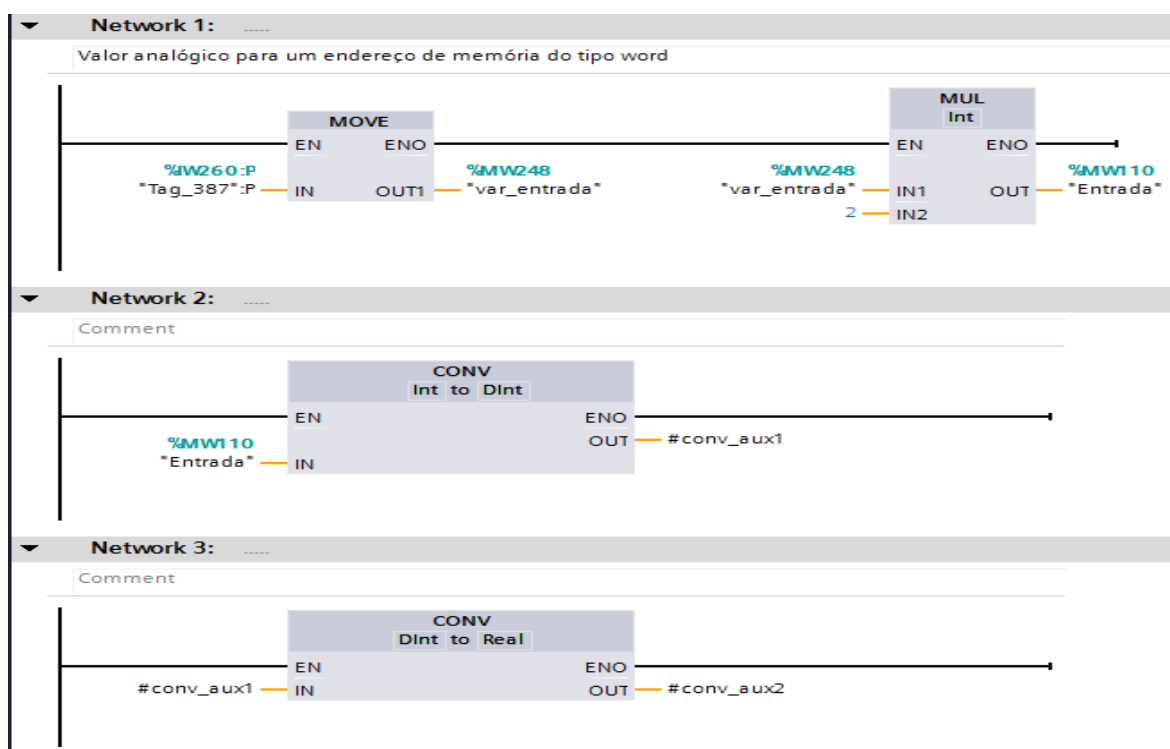


Figura 57 Cálculos para a conversão do valor analógico (1).

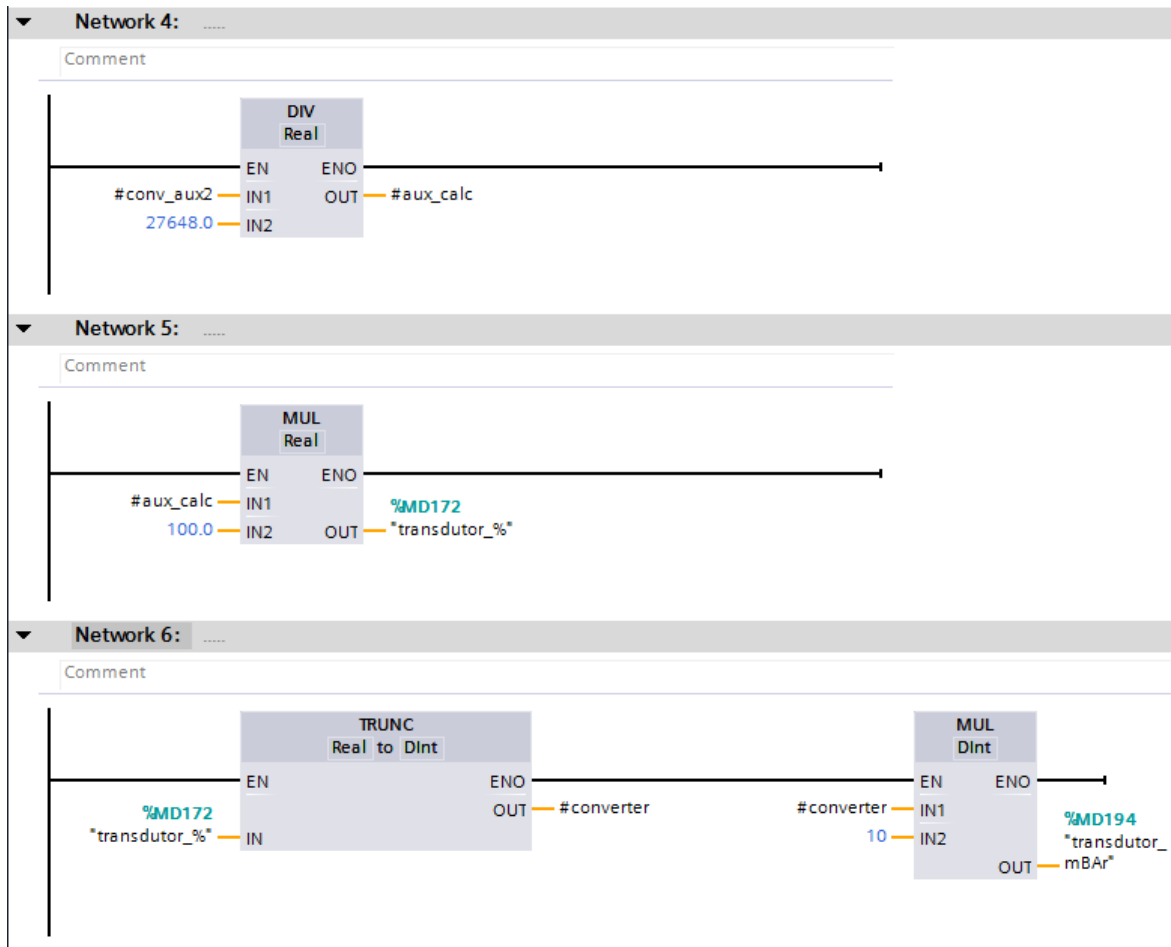


Figura 58 Cálculos para a conversão do valor analógico (2).

Com uma escala de relação percentagem-pressão 100% corresponde a 1000 mb, basta dividir o resultado percentual do transdutor por 10,0 e é obtido o valor real do transdutor em milibar, como é mostrado na *Network 6*. Antes deste cálculo foi aplicada a instrução ‘TRUNC’, que permite converter um dado do tipo real para o tipo Dint.

O valor a ser enviado para a válvula (QW288) terá que ser no mesmo formato que o valor recebido pelo endereço PIW260 do transdutor. No entanto a atribuição da saída pode ser direta visto que o PID faz os cálculos de acordo com o valor analógico da entrada PIW260 do transdutor e envia o resultado para o endereço QW288. De acordo com a Figura 59, o endereço de memória MW116 recebe o valor dos cálculos do PID e atribui esse valor diretamente ao endereço %QW288 : P (QW288) do módulo AO 4x12BIT_2 ligado à CPU.

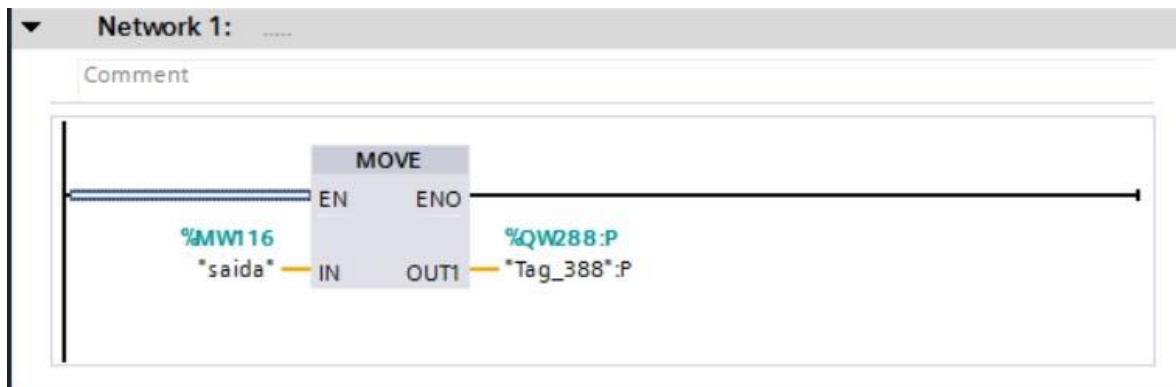


Figura 59 Atribuição da saída da válvula.

Também foi criada a função FC5 “Gestão_Vazamento”, responsável por gerir quando, e em que condições, é que o processo de vazamento deve ser ativado. Esta função foi programada recorrendo à linguagem *Ladder Diagram* (LAD).

Na Figura 60 é apresentada a *Network 1*. Nesta sequência, os vários *bits* apresentados, nomeadamente o “reset_db” (M0.4), o “stop_vazamento” (M0.5) e o “Bit_inicia_vazamento” (M0.3) apenas são ativados quando o estado lógico do *bit* do DB40.DBX2.3 (“SV_CMD_DB”.START_POURING) passa a ‘1’. Este *bit* foi selecionado, uma vez que ele só passa a ‘1’ quando todas as condições, antes de ser dada a permissão de vazamento, são garantidas. Esse conjunto de condições dizem respeito à cabeça X/Y estar no forno e o forno já ter subido até à altura estabelecida.

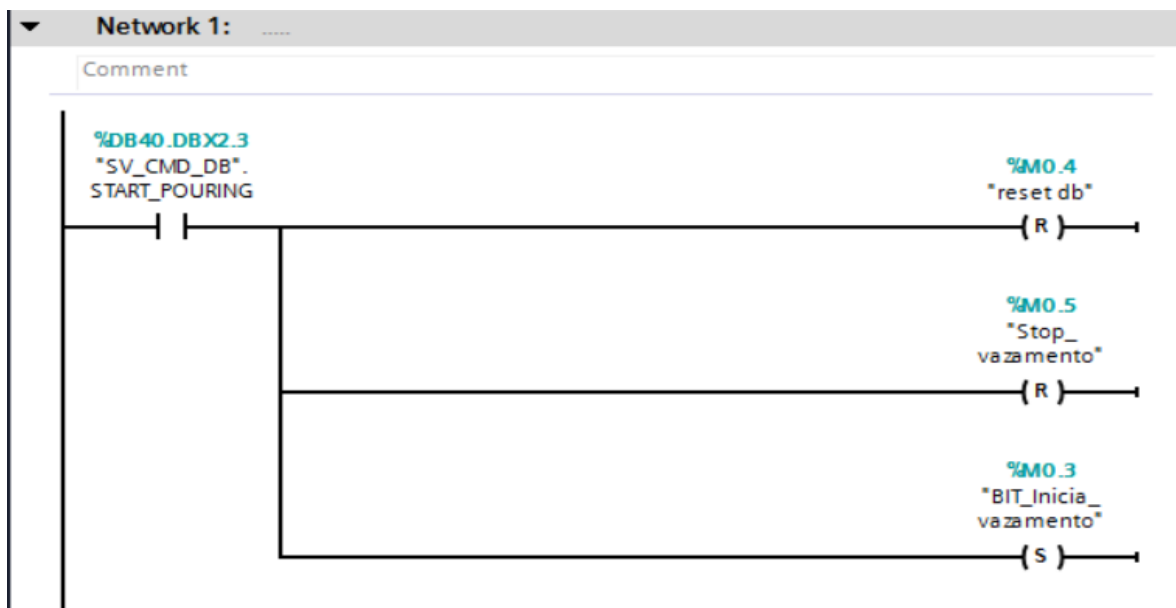


Figura 60 Condições iniciais para ativar o vazamento.

Quando todas as condições forem reunidas é realizado o *reset* aos *bits* M0.4 e ao M0.5, e é ativado (*set*) o bit M0.3 que inicia a função de vazamento como é mostrado na Figura 61, *Network 2*.

Por sua vez, esta função ativa o bit M0.1 “Bit_condições_iniciais” que é responsável por garantir que na primeira leitura da função “Controlo_Vazamento” certas variáveis sejam inicializadas com o valor ‘0’.

Quando o bit M0.3 “BIT_Inicia_vazamento” está no estado lógico ‘0’ este irá ativar o bit M0.4 “reset_db” (uma das condições para ativar a função “RESET_DB”) e o bit M0.5 “Stop_vazamento”, que dá a indicação que o processo de vazamento terminou. Também muda o estado lógico dos bits M0.1 “Bit_condições_iniciais” e M36.4 “SV_IS_POURING” para ‘0’.

Também na *Network 3* é onde é ativada a função FC4 “RESET-DB”, para garantir que sempre que é terminado o processo de vazamento os valores no DB de *output* são zerados, ficando assim prontos para receber nova informação quando for realizado outro vazamento. Ainda nesta *network* é gerido o *reset* de um dos parâmetros do bloco PID, que é necessário sempre que o bloco PID termine as suas funções. Este procedimento é explicado na secção 6.3.

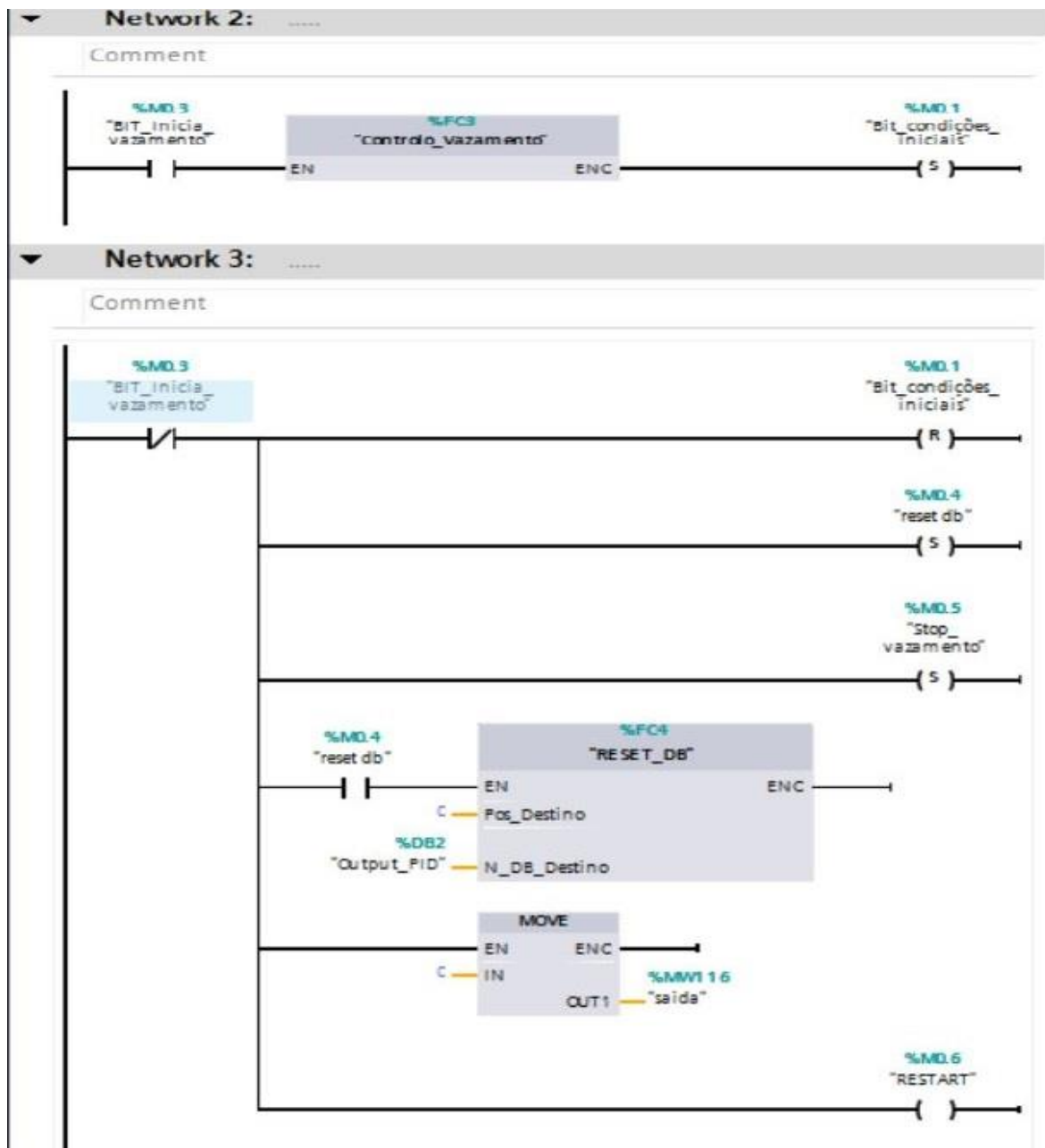


Figura 61 Gestão do vazamento.

De acordo com o estudo efetuado na Secção 5.3, as funções criadas são chamadas na função FC1 “MACHINE”, como é mostrado na Figura 62. Entre as várias funções chamadas dentro desta função, as funções FC6 “transdutor”, FC7 “válvula” e FC5 “Gestão_Vazamento”, foram inseridas logo a seguir à chamada das outras funções, para serem utilizadas sempre que necessário.

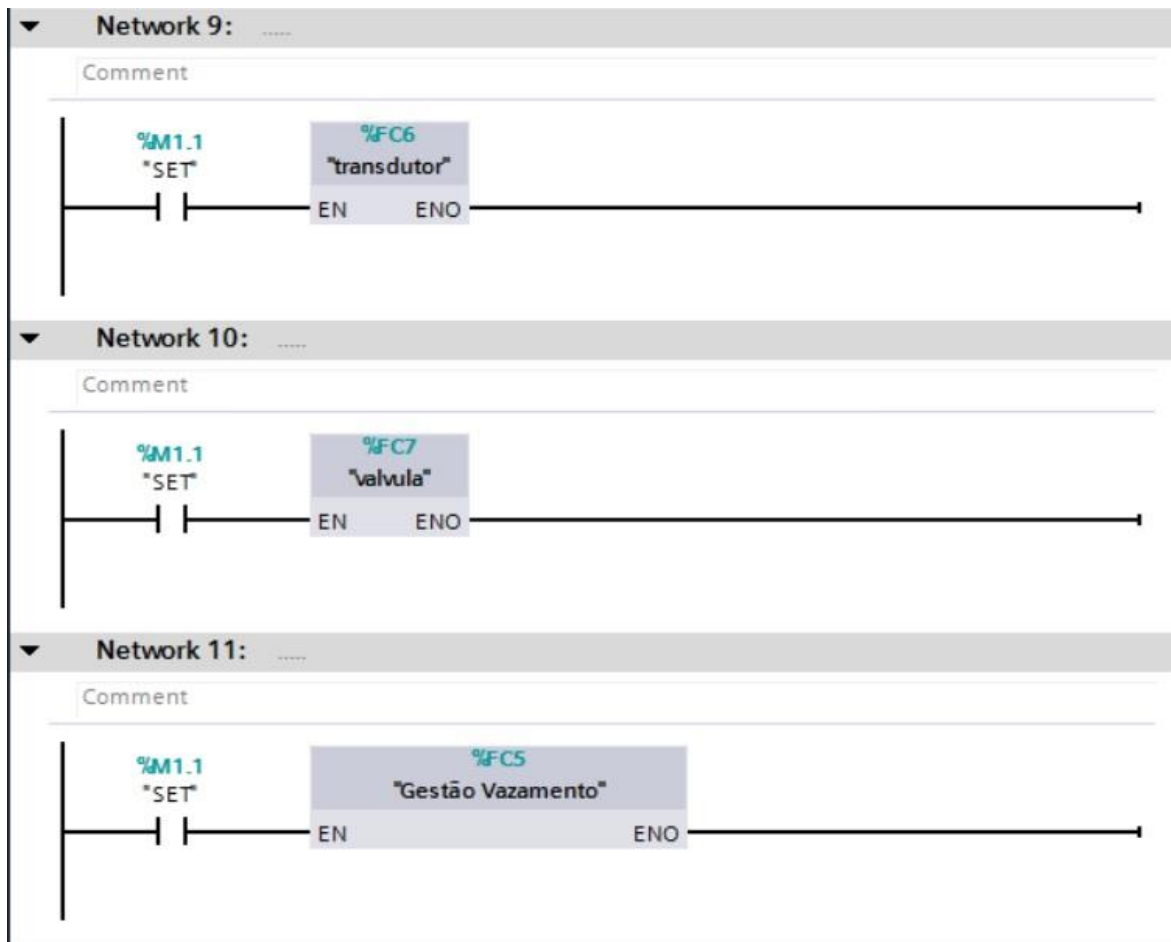


Figura 62 Chamada das funções criadas dentro da Função FC1.

6.3. CONTROLO PID APLICADO

Para a programação descrita na secção anterior funcionar de forma correta, foi necessário enquadrar um controlador PID para definir os valores a serem enviados para a válvula proporcional, em função dos valores de *setpoint* e lidos pelo transdutor de pressão.

Aproveitando as ferramentas fornecidas pelo TIA Portal, a aplicação deste controlador tornou-se mais fácil pois este *software* fornece um bloco, denominado por “CONT_C”, que tem um DB de instância associado que contém todas as variáveis necessárias para o seu correto funcionamento. Devido a ser um bloco desenvolvido pela Siemens, este encontra-se protegido, de forma que a programação envolvida não é disponibilizada aos utilizadores.

Na Figura 63 é apresentado o bloco PID com os seus vários parâmetros de entrada e de saída. No Anexo E são disponibilizadas informações mais detalhadas relativas a este bloco.

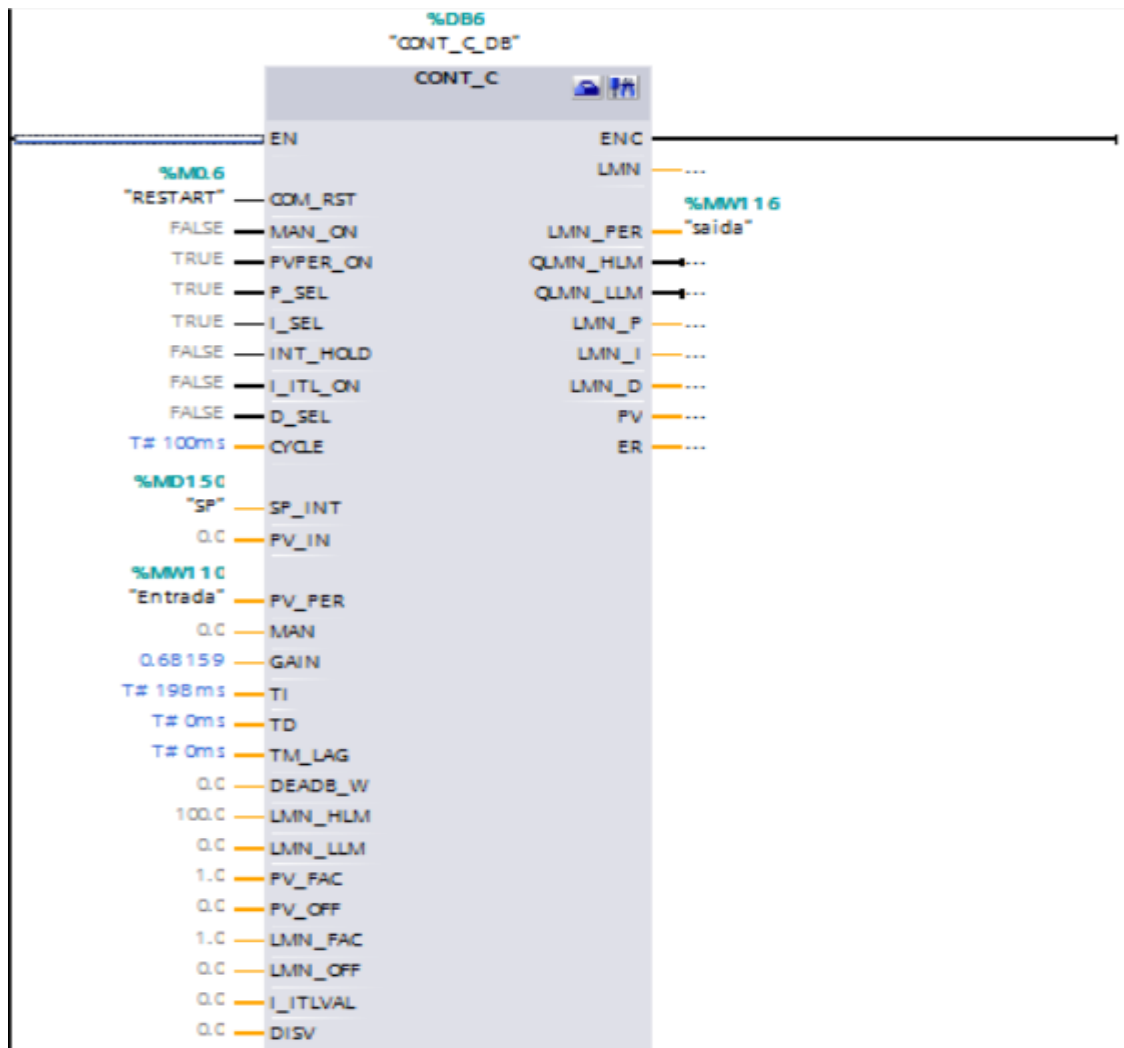


Figura 63 Bloco PID da Siemens.

Este bloco deve ser chamado dentro de um OB de interrupção cíclica (da OB 30 à 38), sendo normalmente usado o OB 35 (no projeto foi usado o OB 34, uma vez que o 35 já estava a ser usado). Deve ser estabelecido o tempo cíclico com que o bloco deve ser chamado para garantir que todos os cálculos são efetuados corretamente. Para tal, deve ser introduzido no parâmetro “CYCLE” o valor temporal correto.

Um dos parâmetros a ser utilizado é o “SP_INT”, onde é introduzido o valor de *setpoint*. Este valor é introduzido em percentagem, pelo que é necessário fazer a associação dos valores pretendidos, como por exemplo pressão, a uma escala percentual.

Da mesma forma que o “SP_INT”, é introduzido o “PV_IN” que diz respeito à entrada do valor lido pelo transdutor. Este pode ser introduzido em formato percentual, no entanto é

usado o endereço “PV_PER” onde os valores lidos pelo transdutor podem ser enviados diretamente para esta variável sem a necessidade de recorrer a fatores de escala.

Foi atribuído um endereço (M0.6, “RESTART”) ao parâmetro de entrada do bloco “COM_RST” para o bloco fazer *reset* aos cálculos efetuados anteriormente, antes de iniciar um novo controle. O endereço “RESTART” é controlado na função “Gestão_vazamento”, quando o processo de vazamento não está ativo este *bit* apresenta o estado lógico igual a ‘1’, ou seja, faz *reset*, quando a fase de vazamento é ativada então o estado lógico do *bit* passa a ‘0’.

Os restantes valores a serem definidos na entrada dizem respeito às constantes do controlador PID. À saída é apenas atribuído um endereço de memória (por exemplo o MW116), que é responsável por enviar os valores obtidos através dos cálculos efetuados no controlador para o endereço analógico atribuído à válvula.

Como foi usado o bloco “CONT_C” para efetuar o controle do processo, foi necessário apenas definir um método para sintonia dos parâmetros do PID, nomeadamente a constante proporcional (P), tempo integral (T_i) e tempo derivativo (T_d).

O método de sintonia usado foi o Método de Cohen-Coon, pois este é um método adequado para sistemas que requerem uma resposta rápida. Assim, seguindo as regras estabelecidas por este método (apresentadas no Anexo F), foram determinadas as várias constantes.

Como forma de obtenção de resultados, foi utilizada uma ferramenta disponibilizada pelo bloco PID (Figura 64) que permite a visualização gráfica dos valores de entrada e saída.

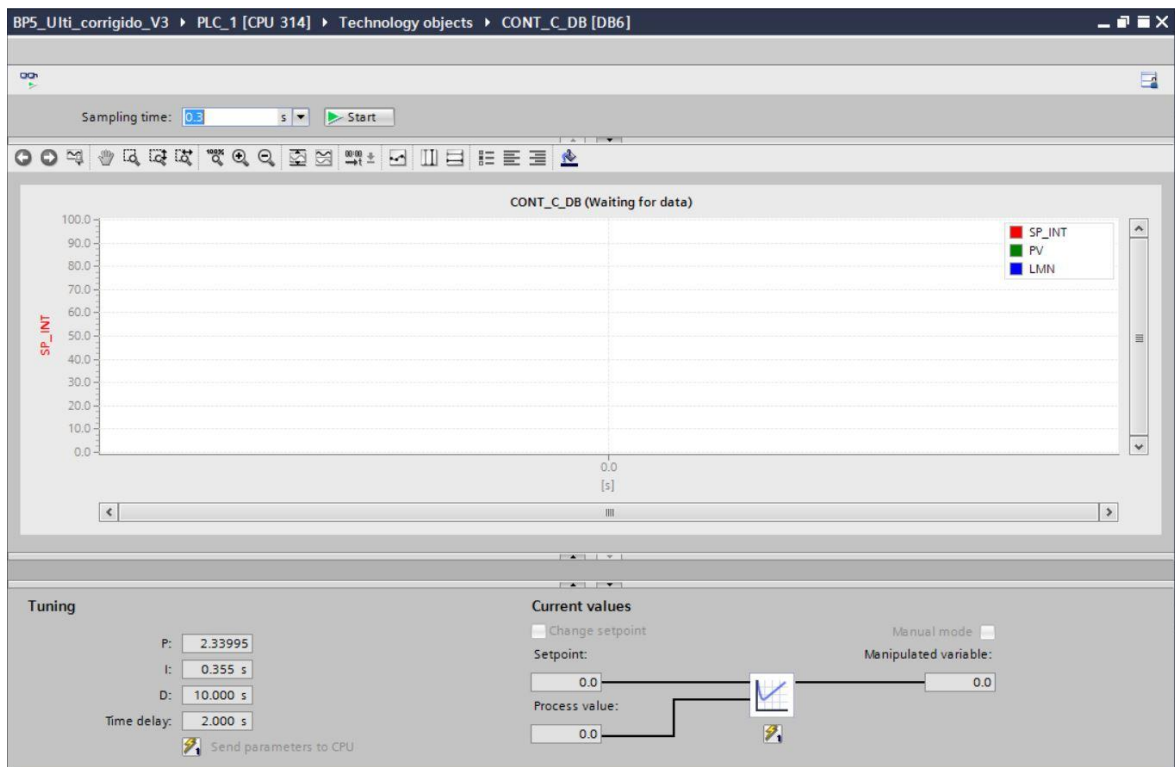


Figura 64 Ferramenta gráfica do bloco PID.

No capítulo 7 será mostrado como foi realizada a sintonia dos parâmetros do controlador PID.

6.4. CRIAÇÃO DA INTERFACE-HOMEM-MÁQUINA

Para completar um dos objetivos deste projeto, foi desenvolvida uma nova HMI para comunicar com o PLC, baseada na anterior, que permitisse o controlo e a visualização de um conjunto de variáveis, assim como o gráfico da curva de vazamento. Deste modo foi criada uma nova interface, que não “fugisse” muito à anteriormente desenvolvida, mas que colmatasse certas falhas que são apresentadas e discutidas no Anexo G.

Como referido no capítulo anterior, foi decidido usar um *software* SCADA já devidamente configurado, da Beijer Electronics, como ferramenta de desenvolvimento do HMI.

Com o estudo do programa foi possível criar a nova interface gráfica baseada na anterior. De seguida são apresentados os vários ecrãs criados, e as suas funcionalidades, e é feita uma comparação com a interface antiga. No Anexo C encontra-se a árvore de menus da nova consola.

Na Figura 65 é apresentado o ecrã principal da aplicação. Este é o primeiro a surgir quando a consola é ligada, e permite o acesso a outros ecrãs como o de Controlo e Visualização do Vazamento, definições de administrador e base de dados. Em todos os ecrãs criados foram inseridos objetos que mostram a data e hora e o logotipo da empresa.

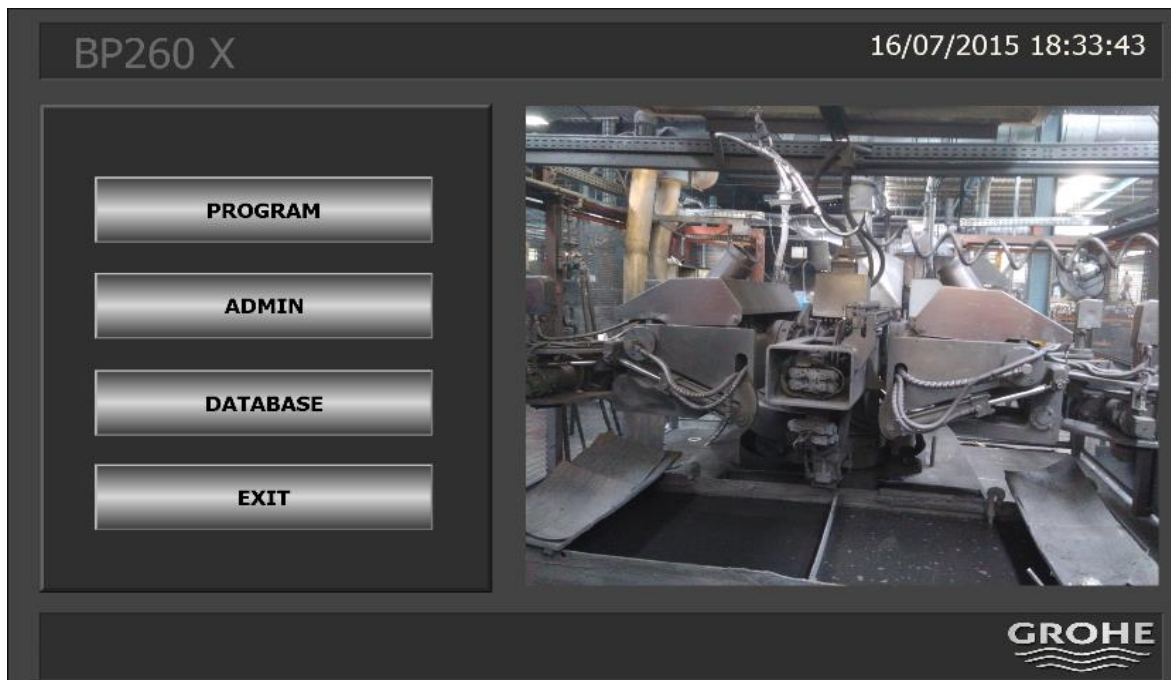


Figura 65 Ecrã principal do novo programa.

Ao pressionar o botão “PROGRAM” é apresentado um novo ecrã onde é possível efetuar o controlo e visualização de alguns parâmetros do processo de produção (Figura 66). O botão “ADMIN” reencaminha o utilizador para uma nova área onde poderão ser feitas as configurações mais críticas da máquina e onde também podem ser geridas contas de utilizador. O botão “DATABASE” serve simplesmente para carregar uma nova base de dados que possa estar armazenada na memória da consola ou através de uma PEN inserida via USB. Por último, o botão “EXIT” desliga a aplicação.

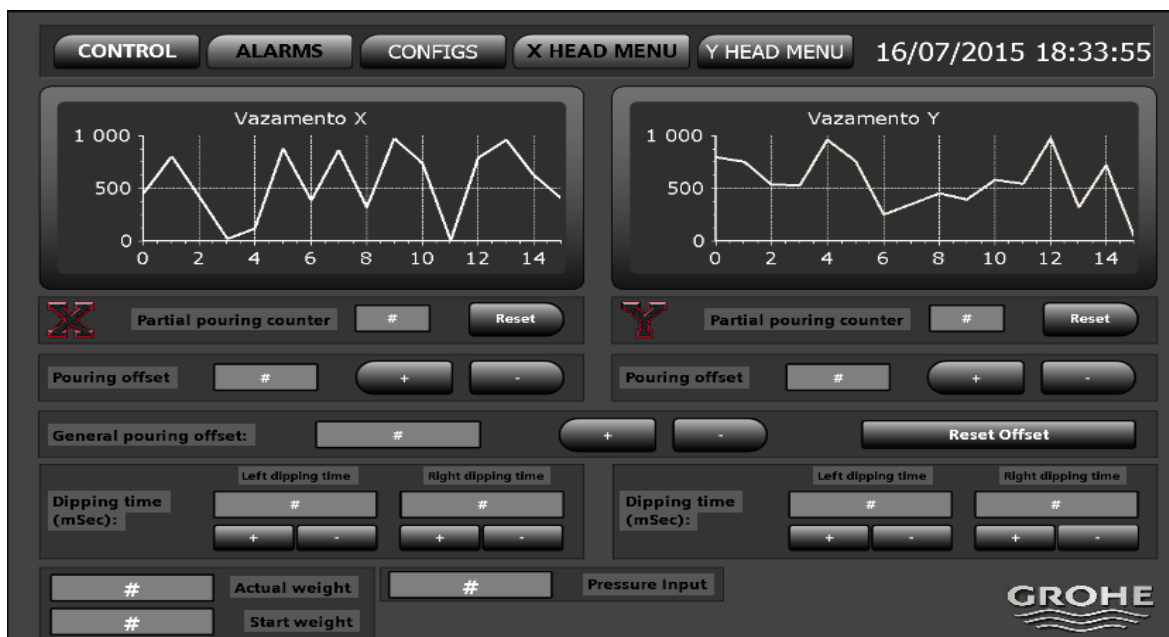


Figura 66 Controlo e visualização do processo de fabrico.

O controlo e visualização do processo de fabrico acaba por ser o ecrã mais usado. É com este ecrã que o operador mais interage, visto que é neste que é feita a visualização da evolução da curva de vazamento, que indica se a peça vazou corretamente ou não.

Quando a fase de vazamento é iniciada, surge em primeiro plano uma janela *pop-up* que amplia o gráfico (dependendo da cabeça que se encontra no forno) para uma melhor visualização do controlo (Figura 67).

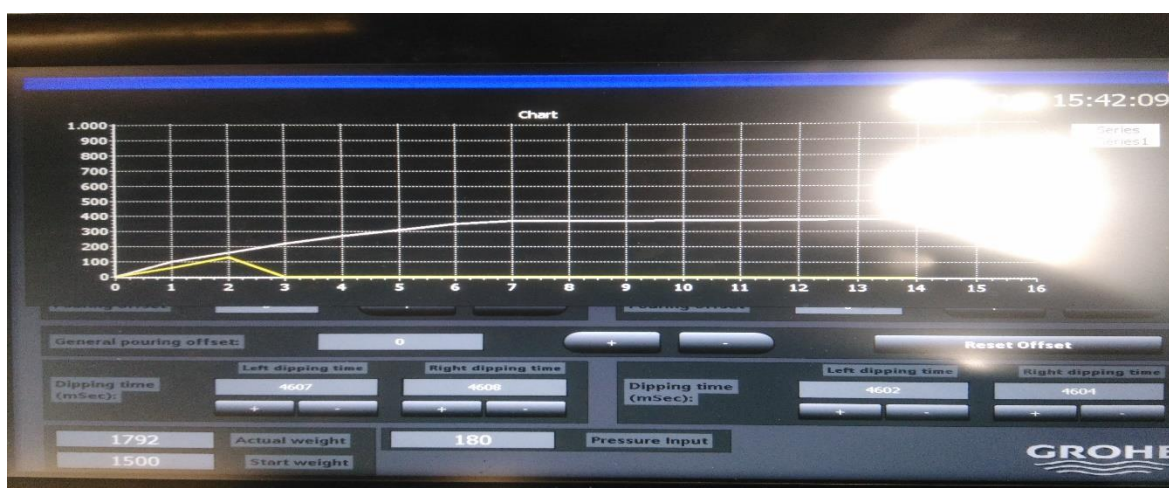


Figura 67 Janela *pop-up* com gráfico de controlo.

O ecrã de controlo contém dois gráficos: um para a cabeça X e outro para a cabeça Y. Por baixo de cada um deles existe uma variável (“*Partial pouring counter*”) que faz a contagem

do número de vazamentos em cada cabeça, com a possibilidade de fazer *reset* à contagem quando ocorre mudança de operador (no fundo acaba por ser um registo de produção).

Visto que o forno tem uma balança para medir o peso do metal fundido, também foram introduzidas variáveis que mostram o valor medido inicialmente (quando o forno está cheio) e o valor do peso à medida que são efetuados vazamentos. Existe outra variável que indica a pressão dentro do forno. Esta variável está a apresentar informação do transdutor de pressão instalado.

Na parte superior do ecrã existem vários botões que permitem navegar por outros ecrãs.

Carregando no botão “ALARMS” é apresentado o ecrã de visualização de alarmes (Figura 68) que mostra se há ou não alarmes ativos, assim como uma descrição indicativa do motivo do alarme.

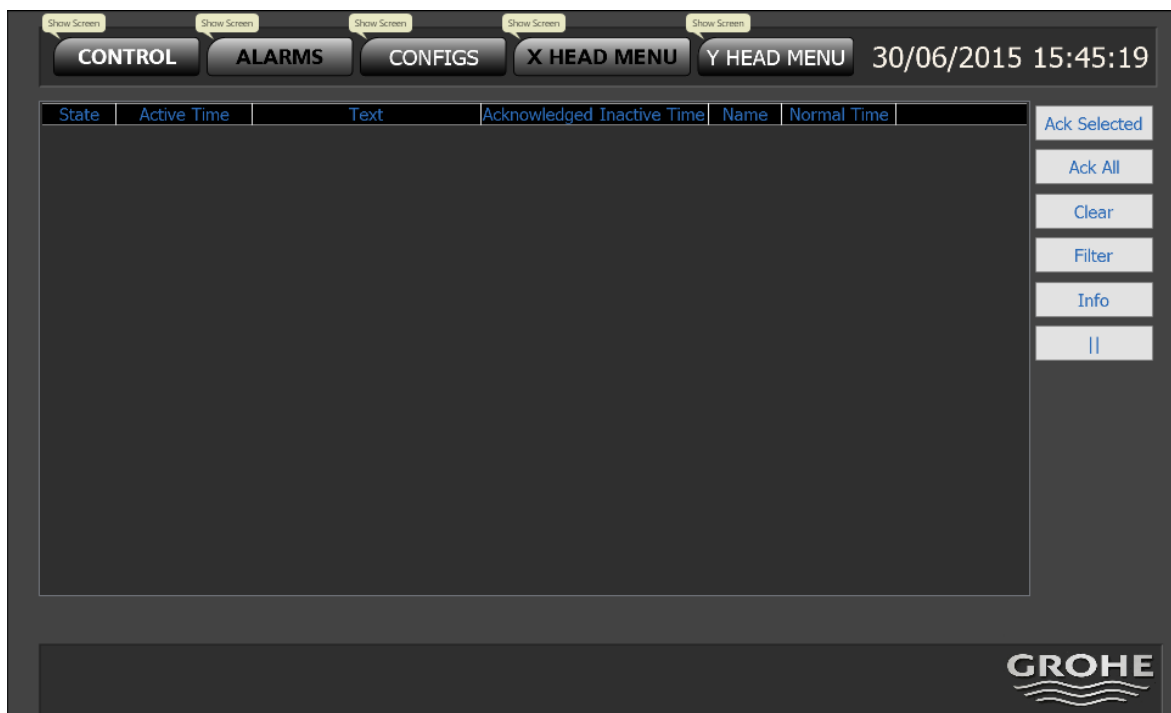


Figura 68 Visualizador de alarmes.

O “X HEAD MENU” e o “Y HEAD MENU” são ecrãs destinados a configurações das cabeças X e Y. Ambos são praticamente iguais a nível de parametrizações - os parâmetros introduzidos é que poderão ser diferentes. A Figura 69 mostra o conteúdo do “X HEAD MENU”, que permite o acesso a outros ecrãs de configuração mediante as necessidades do

operador. De referir que nestes menus são criadas, editadas, guardadas, ou carregadas, as curvas de vazamento.

Os acessos disponibilizados ao utilizador através deste ecrã são o “X WORKSTATION SETTINGS PAGE”, onde são efetuadas as configurações da cabeça X, o “LINEAR SCALE CHECK PAGE”, onde pode ser consultada informação sobre os parâmetros analógicos da cabeça X, o “PROXIMITY SWITCH CHECK PAGE”, um menu comum entre as duas cabeças, que dá informação sobre posições e, por último, os quatro botões que dizem respeito à configuração da curva de vazamento para a cabeça X: “NEW DIE CURVE DESIGN”, “SAVE ACTUAL CURVE AS”, “MODIFY ACTUAL CURVE” e “LOAD AND ERASE CURVES FROM ARCHIVE”.

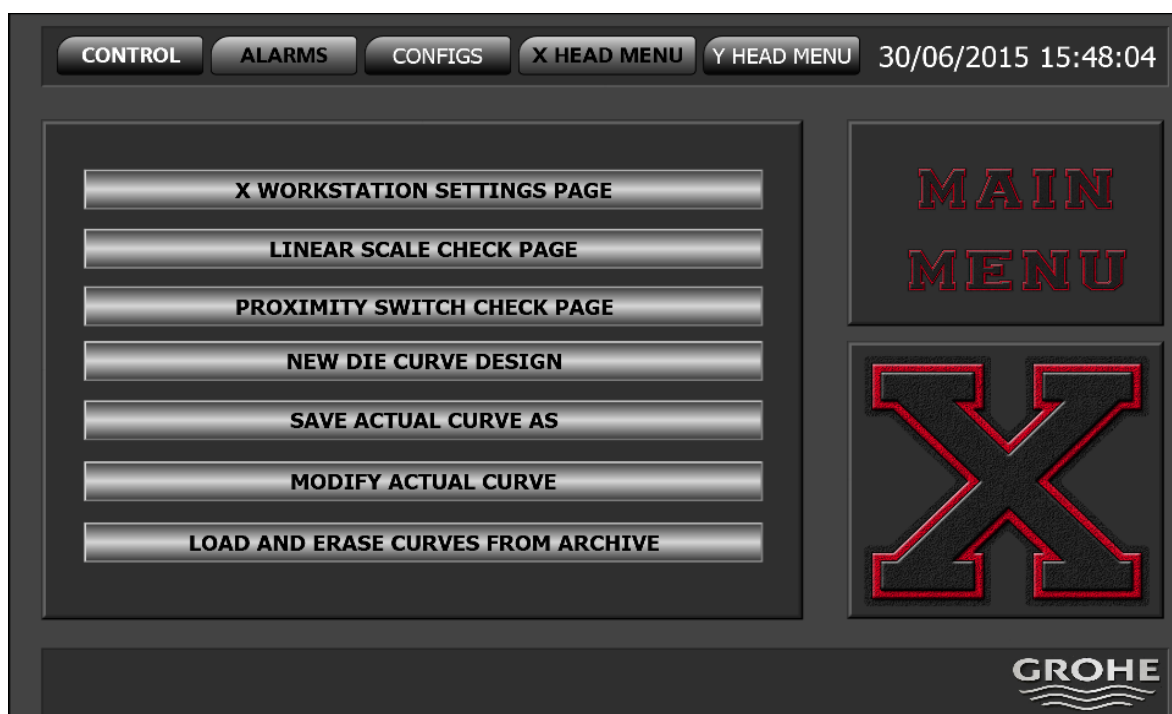


Figura 69 X HEAD MENU.

O “X WORKSTATION SETTINGS PAGE” permite o acesso a várias páginas de configuração da cabeça X. Nessas páginas constam todas as configurações que podem ser ativadas e modificadas. A Figura 70 mostra a primeira página de configuração, onde as indicações luminosas do lado direito mostram se o parâmetro está ativado ou desativado. As diversas configurações que podem ser efetuadas são as seguintes:

Workstation enabled – se esta função está ativada (para ambas as cabeças X e Y), a máquina trabalha com ambas as cabeças. Alternativamente também pode ser decidido trabalhar

apenas com uma cabeça. Para trabalhar apenas com uma cabeça, neste caso a cabeça Y, basta carregar no “*Workstation enabled*” (apresentado na Figura 70), sendo a confirmação dada quando o sinal luminoso apagar.

Dipping angle while in tank position enabled – o ângulo de rotação da coquilha para a grafite (o banco na posição de grafite e os braços alinhados com o eixo) é configurado na página 3 (“*X WORKSTATION SETTINGS PAGE 3*”, “*dipping angle duration time*”). O valor introduzido não corresponderá a um ângulo real, mas sim a um tempo que, por sua vez, corresponderá a um ângulo. A rotação começa logo que a coquilha deixe de estar alinhada com o eixo e vá para o banho de grafite.



Figura 70 X WORKSTATION SETTINGS PAGE 1.

Dipping angle before tank position enabled – o ângulo de rotação da coquilha antes do banho de grafite (banco na posição horizontal e braços alinhados com o eixo) é também configurado na página 3 (“*X WORKSTATION SETTINGS PAGE 3*”, “*dipping angle duration time*”). Aqui a coquilha roda por intermédio de um botão, seletor de ciclo, que deve ser carregado pelo operador, para esta ir para a posição de banho de grafite.

Left ejector enabled – comando para ativar ou desativar o ejetor esquerdo.

Right ejector enabled - comando para ativar ou desativar o ejetor direito.

Dies rotated towards cast position during cast insertion – roda a coquilha, quando o programa está na fase de colocação do macho de areia, com banco inclinado, e coquilha aberta. Caso este seletor esteja desativado a coquilha está aberta mas virada de frente para o operador.

Dies rotated towards cast position during spray function – este controlo serve para escolher a posição da coquilha, caso a máquina esteja em modo manual.

Mixers enabled – ativa ou desativa os misturadores.

A página número 2 do “X WORKSTATION SETTINGS” apenas contém endereços livres, caso seja necessário acrescentar alguma variável da máquina ao HMI.

A partir da página 3, a configuração de todas as variáveis é feita através da introdução de valores no *display* do lado esquerdo, como é mostrado na Figura 71.

Ao pressionar esse display surgirá um mini teclado para a introdução de novos valores.



Figura 71 X WORKSTATION SETTINGS PAGE 3.

Todos os valores introduzidos na página 3 “X WORKSTATION SETTINGS”, são valores numéricos que podem ir de 10 a 9990. Estes valores são submetidos em milissegundos. Mais uma vez foram deixadas algumas variáveis livres para futuros acrescentos.

Na Figura 72 segue-se a continuação das configurações, mas agora para parâmetros não temporais.

Dies weight due nozzle touching during pouring – de modo a garantir baixa pressão na coquilha, as seguintes condições devem ser estabelecidas: as coquilhas devem estar instaladas na máquina, a coquilha deve estar sobre o forno, o forno deve atingir a altura certa e o bico do forno à devida pressão. Esta pressão é definida neste campo, e o valor de 150 é válido para todos os casos.

Furnace vertical position during pouring – indica a posição (em milímetros) que o forno deve atingir de modo a que o bico entre em contacto com a coquilha. Este parâmetro deve ser configurado de acordo com a coquilha instalada, pois o forno deve atingir uma altura suficiente para o bico chegar até à coquilha. Para tal, deve-se aceder ao menu “*LINEAR SCALE CHECK PAGE*” e verificar o valor no campo “*furnace position*”. O valor registado quando o forno atinge a altura necessária, deve ser multiplicado por 10 e inserido neste campo.

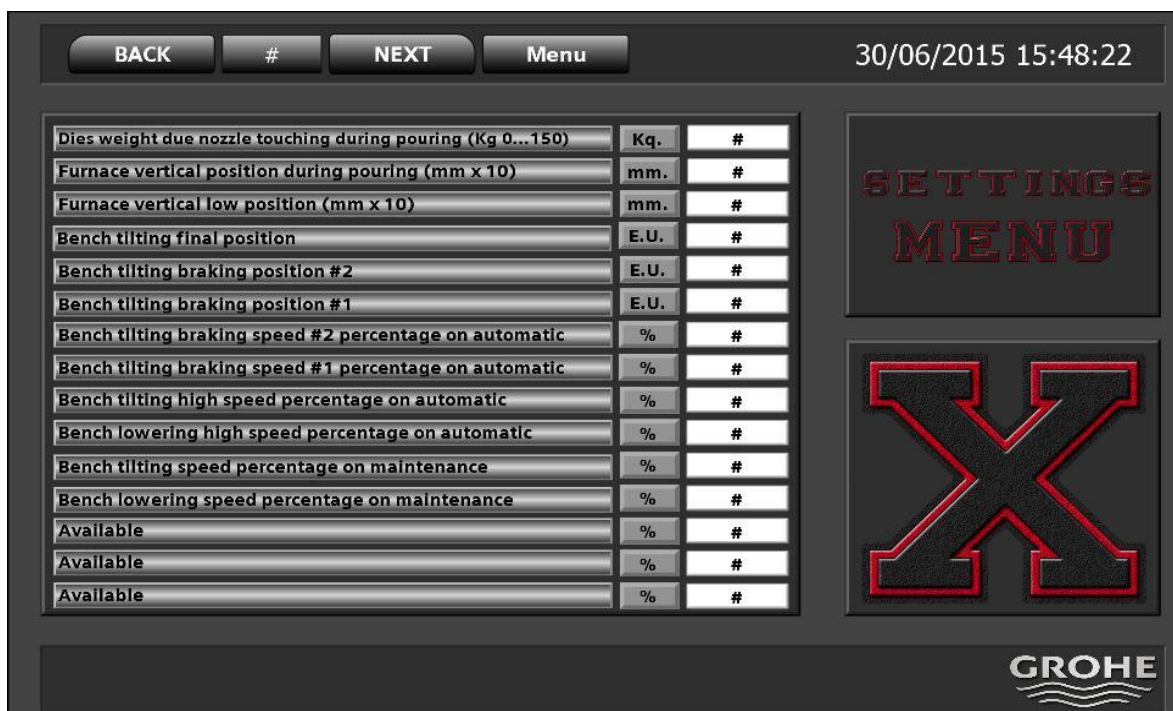


Figura 72 X WORKSTATION SETTINGS PAGE 4.

Furnace vertical low position – da mesma maneira que se procede para inserir o valor no campo anterior, este parâmetro utiliza o valor do “*furnace position*”, do ecrã “*LINEAR*

SCALE CHECK PAGE”, quando o forno esta na altura mínima. O valor lido é multiplicado por 10 e inserido neste campo.

Bench tilting final position – indica o ângulo de inclinação do banco. Os valores inseridos neste parâmetro estão de acordo com uma tabela pré-estabelecida (Tabela 5).

Bench tilting braking position #2 – indica o ângulo quando ocorre a segunda mudança de velocidade durante a inclinação do banco. Os valores inseridos neste parâmetro estão de acordo com uma tabela pré-estabelecida (Tabela 5).

Bench tilting braking position #1 - indica o ângulo quando ocorre a primeira mudança de velocidade durante a inclinação do banco. Os valores inseridos neste parâmetro estão de acordo com uma tabela pré-estabelecida (Tabela 5).

Tabela 5 Correspondência dos valores a inserir em graus.

DEGREES	VALUE
0°	0
15°	4608
30°	9216
45°	13824
60°	18432
75°	23040
90°	27648

Bench tilting braking speed #2 percentage – indica a percentagem da velocidade de travagem do banco (segunda velocidade). Quando maior a percentagem maior será a velocidade, sendo que o valor 100 corresponde à velocidade máxima.

Bench tilting braking speed #1 percentage - indica a percentagem da velocidade de travagem do banco (primeira velocidade). Quando maior a percentagem maior será a velocidade, sendo que o valor 100 corresponde à velocidade máxima.

Bench tilting high speed percentage – indica a percentagem da velocidade do banco durante a fase de levantamento.

Bench lowering high speed percentage – indica a percentagem da velocidade do banco, quando este baixa.

Bench tilting speed percentage on maintenance - indica a percentagem da velocidade do banco quando este levanta para o ciclo manual.

Bench lowering speed percentage on maintenance - indica a percentagem da velocidade do banco quando este baixa para o ciclo manual.

Por fim, é apresentada na Figura 73 a página 5 do “X WORKSTATION SETTINGS”, onde são realizadas as configurações finais da cabeça.

Cast weight – neste campo é inserido o peso real da peça depois de esta estar fundida. O valor do peso é inserido em miligramas.

Die closing speed percentage on maintenance – indica o valor percentual da velocidade de encerramento da coquilha, em modo de operação manual.

Die closing speed percentage on braking - indica o valor percentual da velocidade de encerramento da coquilha, quando esta está em modo de travagem.

Die closing speed percentage on automatic - indica o valor percentual da velocidade de encerramento da coquilha, em modo de operação automático.



Figura 73 X WORKSTATION SETTINGS PAGE 5.

O ecrã que se segue é o “LINEAR SCALE CHECK PAGE” (Figura 74). Esta página serve para verificar o estado dos potenciómetros em tempo real.

Back 30/06/2015 16:09:34

X WORKSTATION

Dies closing
Actual position #

Rotation
Actual position #

Bench inclination
Actual position #

Furnace position (mm.)
Actual position #

Vertical movement on furnace
Actual position #

LINEAR SCALE CHECK MENU GROHE

Figura 74 Linear Scale Check Page.

O ecrã “PROXIMITY SWITCH CHECK PAGE”, presente na Figura 75, mostra a aproximação da máquina face aos *inputs*. Quando o indicador luminoso está aceso é confirmada a posição, e quando apagado tal deixa de ser verdade.



Figura 75 Proximity Switch Check Page.

Ao aceder ao menu “NEW DIE CURVE DESIGN” é possível configurar uma nova curva de vazamento. Os dados são inseridos nas várias caixas de texto, apresentadas no lado direito da Figura 76. Por baixo da etiqueta “MBAR” são inseridos os vários valores das pressões desejadas e por baixo da etiqueta MSEC os valores temporais, em milissegundos, correspondentes.

Do lado esquerdo é apresentado um gráfico que representa a curva conforme os parâmetros desejados, sendo que o eixo x corresponde ao tempo (em segundos), e o eixo y corresponde aos valores de pressão (em milibar).

Ao carregar no botão “NEXT”, que se encontra na parte superior do ecrã, é permitido o acesso ao menu “SAVE ACTUAL CURVE AS” (presente na Figura 77), com botões que permitem guardar a curva anteriormente criada, ou simplesmente carregar uma curva guardada em memória. Também é garantido o acesso a este menu através dos ecrãs “X HEAD MENU”, ou “Y HEAD MENU”, anteriormente apresentados.

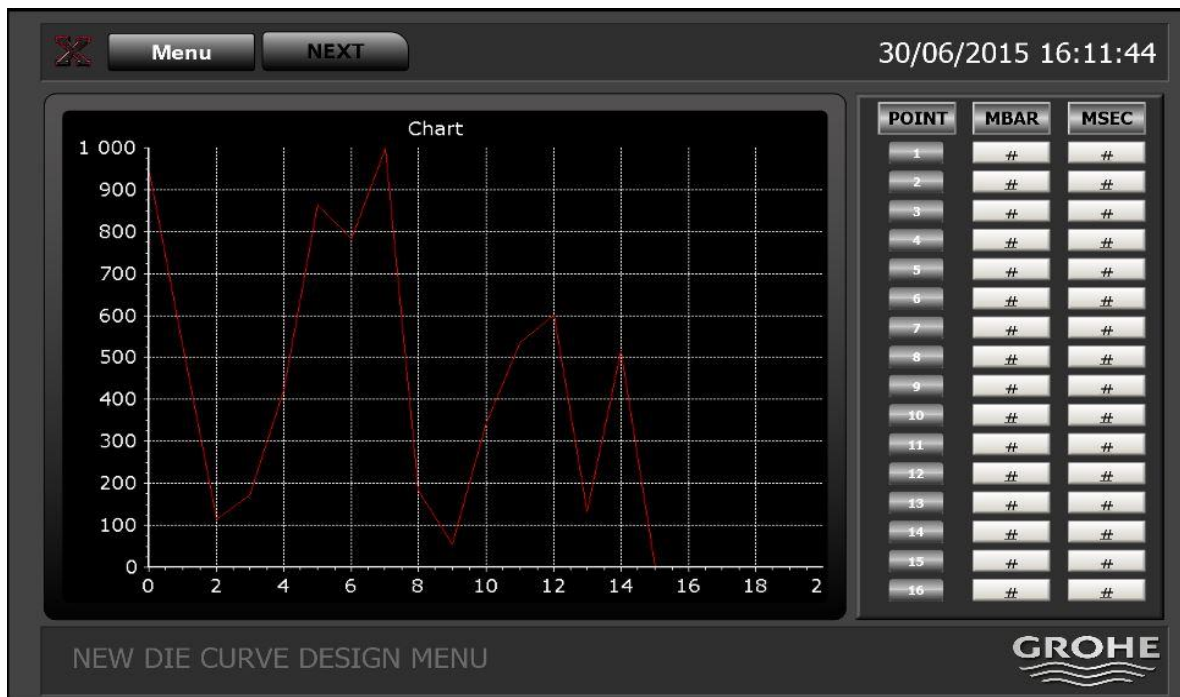


Figura 76 Criação de uma nova curva de vazamento.

Apesar dos ecrãs de configuração das curvas serem idênticos a nível estético, são diferentes na forma como foram configurados, pois as curvas são guardadas em *recipes* diferentes, mediante a cabeça utilizada.

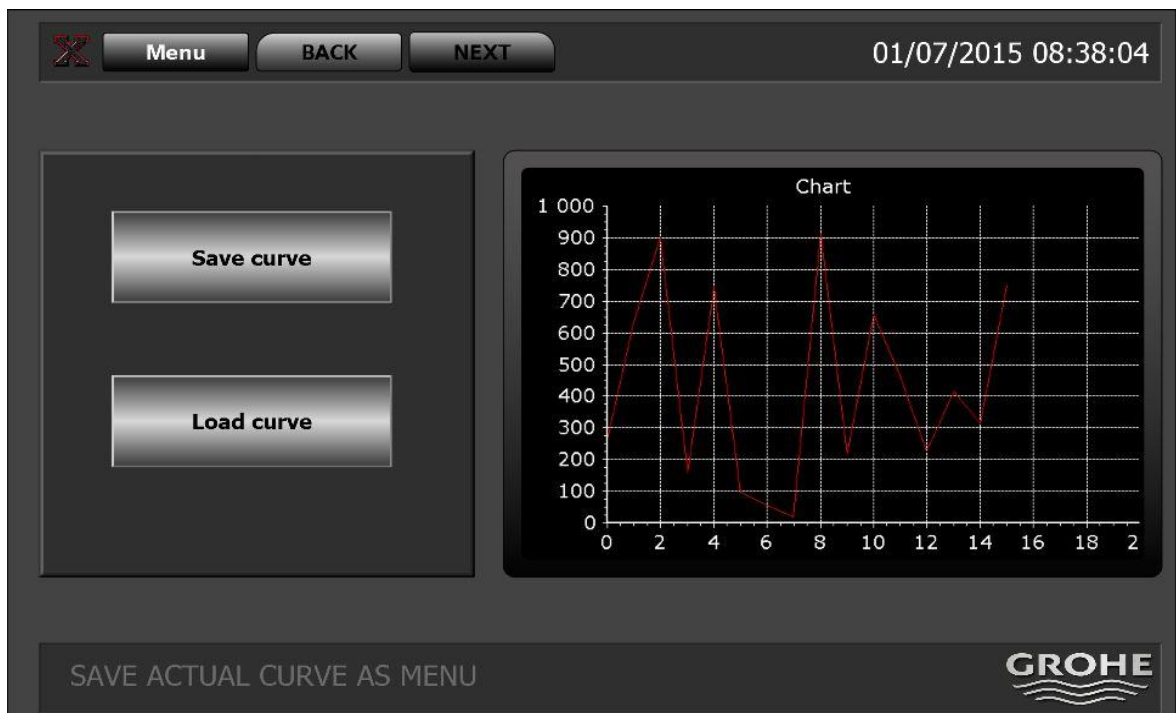


Figura 77 Carregar e guardar curva de vazamento.

Carregando novamente no botão “NEXT”, aparece ao utilizador o menu “MODIFY ACTUAL CURVE MENU” para modificar curvas, apresentado na Figura 78.

Este menu permite fazer alterações numa curva anteriormente criada. Para tal existem botões que permitem carregar curvas guardadas, como o “Load curve”, e curvas guardadas em outros dispositivos (como uma PEN), através do botão “Import curve”. Do mesmo modo, o botão “Export curve” permite enviar as curvas criadas para outro dispositivo de armazenamento.

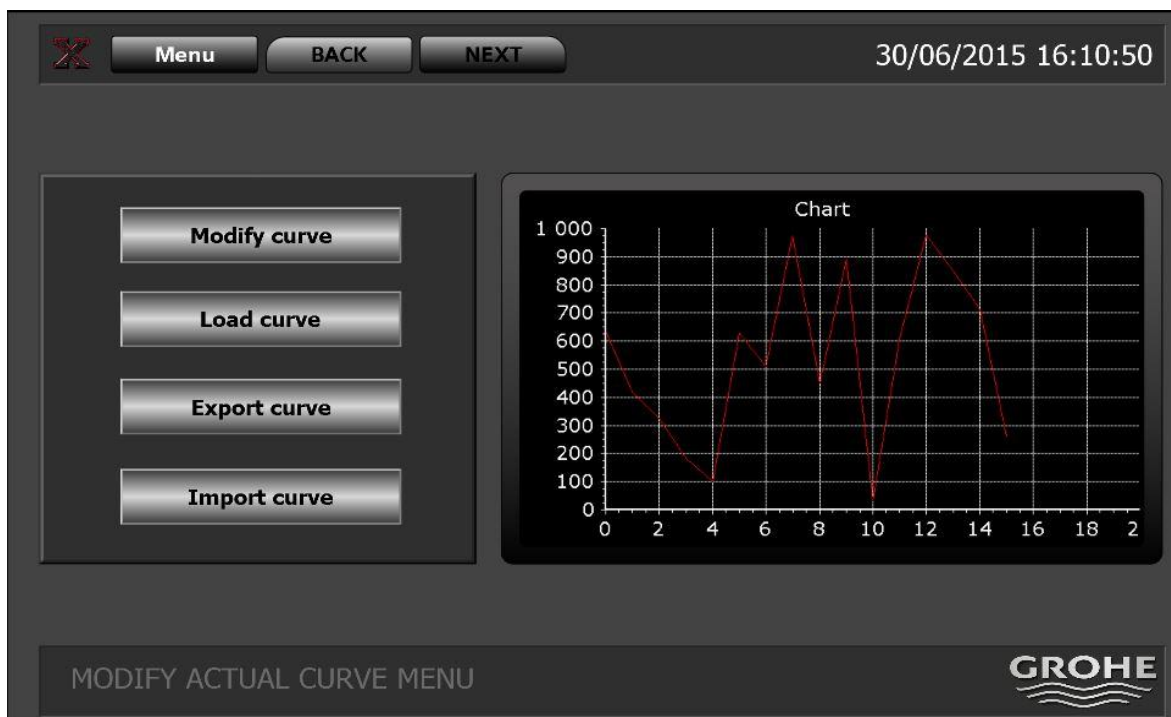


Figura 78 Modificar curva de vazamento.

Como referido anteriormente, este menu serve para modificar curvas já criadas. Carregando no botão “*Modify curve*” surgirá uma nova janela *popup*, apresentada na Figura 79, que disponibiliza a introdução de novos valores de pressão para a configuração da curva.

O último menu de configuração de curvas é o “LOAD AND ERASE CURVES FROM ARCHIVE MENU” (Figura 80). Este menu serve para apagar as curvas criadas de modo que contém os botões necessário para fazer o carregamento de curvas, quer da memória interna da consola, quer de um dispositivo de armazenamento externo.

POINT	MBAR	MSEC
1	#	#
2	#	#
3	#	#
4	#	#
5	#	#
6	#	#
7	#	#
8	#	#
9	#	#
10	#	#
11	#	#
12	#	#
13	#	#
14	#	#
15	#	#
16	#	#

Modify curve

Figura 79 Atribuição de novos valores de pressão para a configuração da curva.

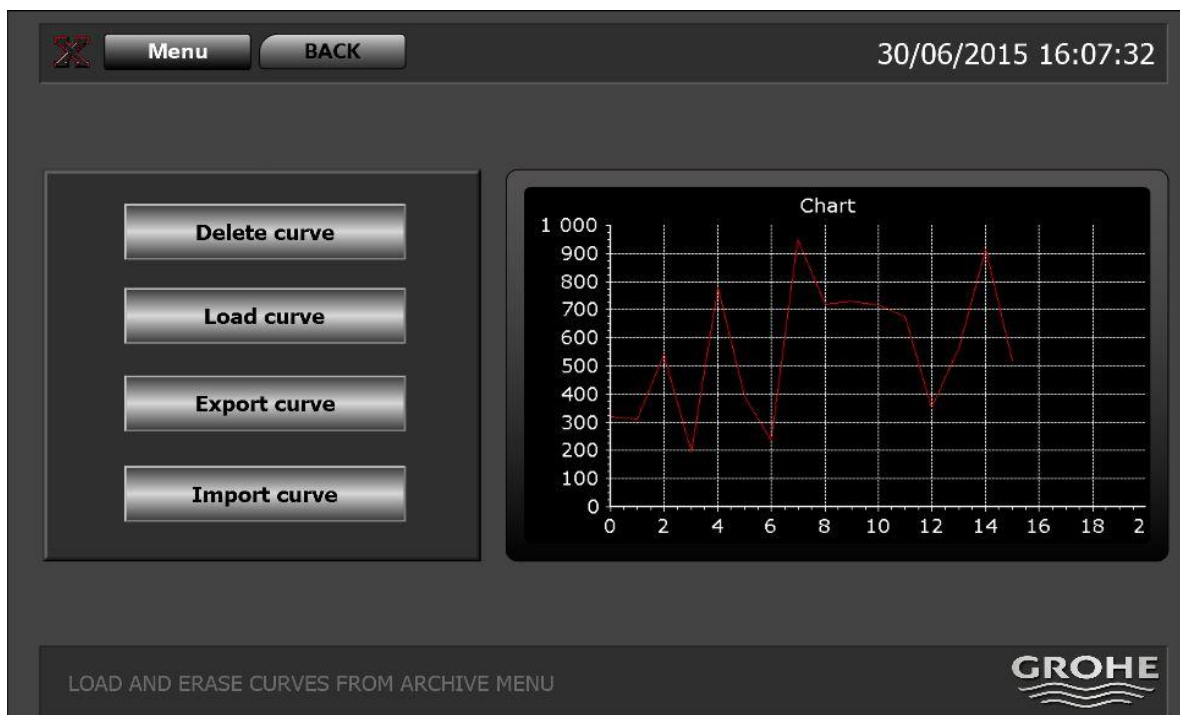


Figura 80 Apagar curvas de vazamento.

O acesso ao menu “ADMIN”, mostrado na Figura 81, permite aceder a outros tipos de configuração da máquina em geral e do forno associado a esta.

Estes menus apenas se encontram disponíveis quando os direitos de acessibilidade forem garantidos. Para tal é preciso ser efetuado o *login* com direitos de administrador. Estes menus encontram-se bloqueados pois as configurações destes parâmetros apenas deverão ser realizados por agentes mais especializados e que entendam o funcionamento da máquina a nível técnico.

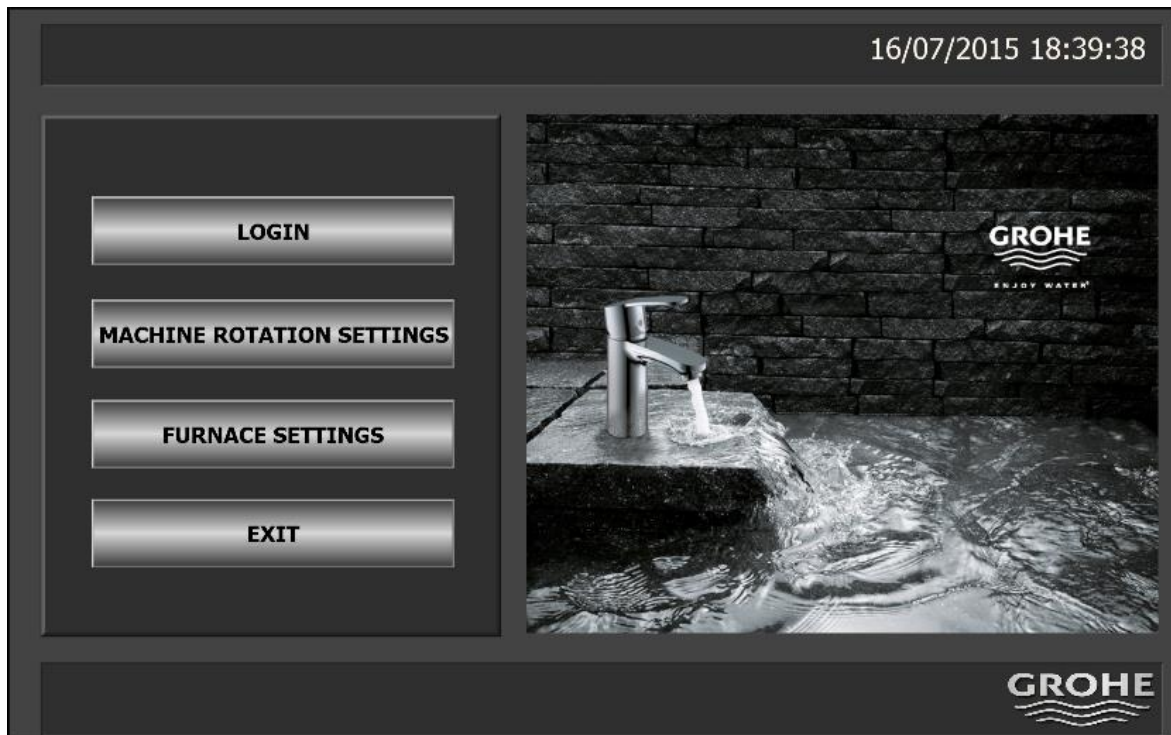


Figura 81 Menu do administrador.

Acedendo ao menu *login*, apresentado na Figura 82, é possível fazer o acesso com direitos de administrador, selecionando para isso o *user name* pretendido e a respetiva *password*.

Depois de garantidos esses direitos, ainda neste menu, é possível a criação de novos utilizadores (para operadores, ou pessoal de manutenção) ou importar os dados de todos os utilizadores para um dispositivo externo. Quando os direitos de administrador não forem mais necessários, o botão “LOGOUT” permite terminar sessão.

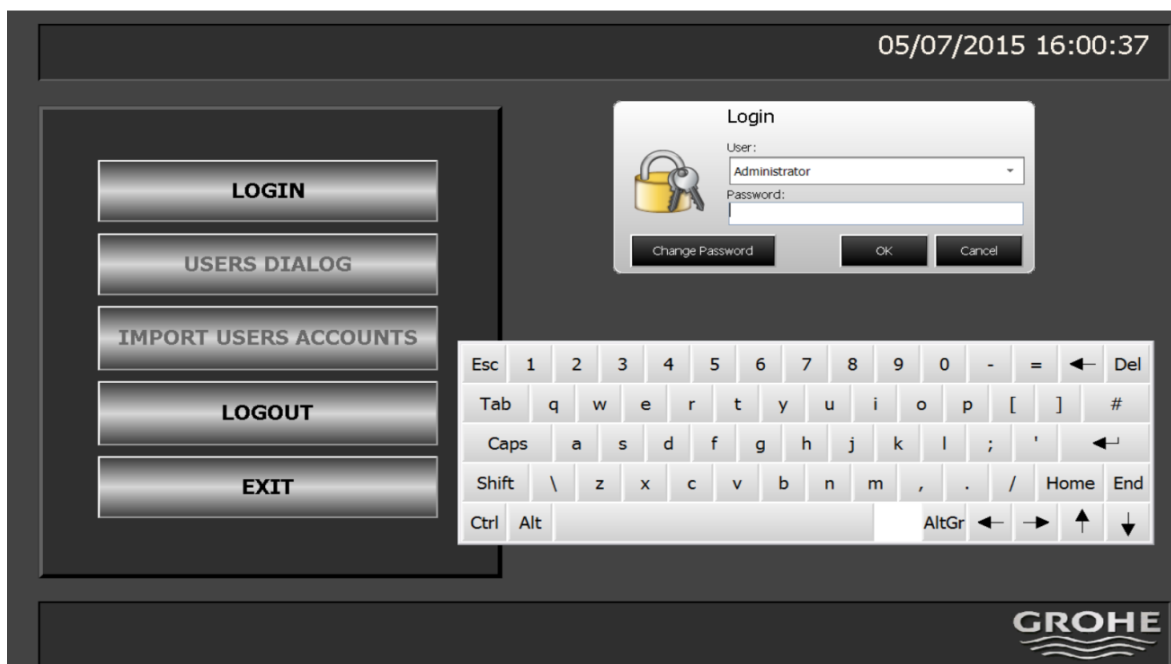


Figura 82 Menu de login.

Com o acesso ao menu “MACHINE ROTATION SETTINGS”, é possível a configuração dos parâmetros da máquina em geral, como mostrado na Figura 83. A entrada neste menu é feita através do menu “ADMIN”, depois de serem garantidos os direitos de administrador.

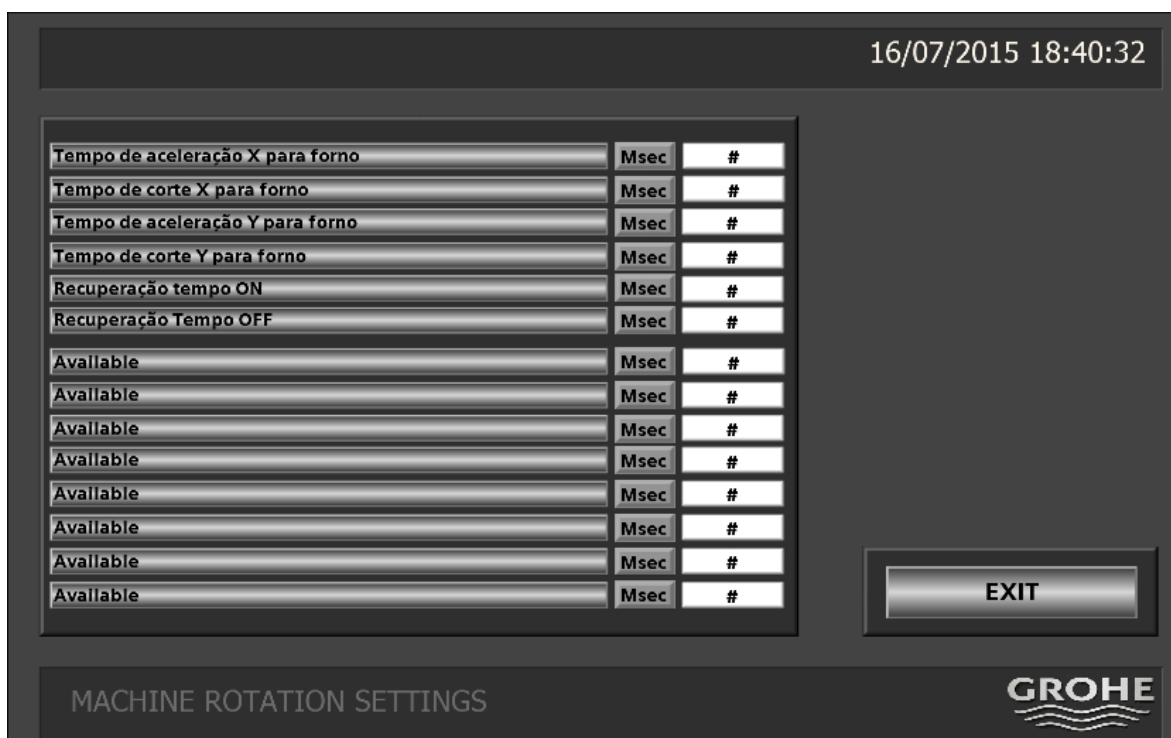


Figura 83 Configurações de rotação da máquina.

Os parâmetros que podem ser configurados neste ecrã, são os seguintes:

Tempo de aceleração X/Y para forno – apesar de estar apresentado como uma variável temporal, este parâmetro indica a velocidade de rotação da máquina, quando esta se encontra na zona 1. Os valores a serem inseridos vão de 0 a 13000, representando este último a velocidade máxima.

Tempo de corte X/Y para forno - este parâmetro indica a velocidade de rotação da máquina, quando esta se encontra na zona 2.

Recuperação tempo ON/OFF – nestes parâmetros é introduzido um valor temporal que indica a velocidade a atingir quando é necessário entrar em modo de recuperação (posição inicial da máquina).

Na Figura 84 é apresentado o ecrã com as configurações do forno. Estas configurações servem apenas para definir limites na quantidade de metal (em relação ao peso) e são úteis como indicações para se saber quando devem ser feitos novos reabastecimentos de metal no forno.

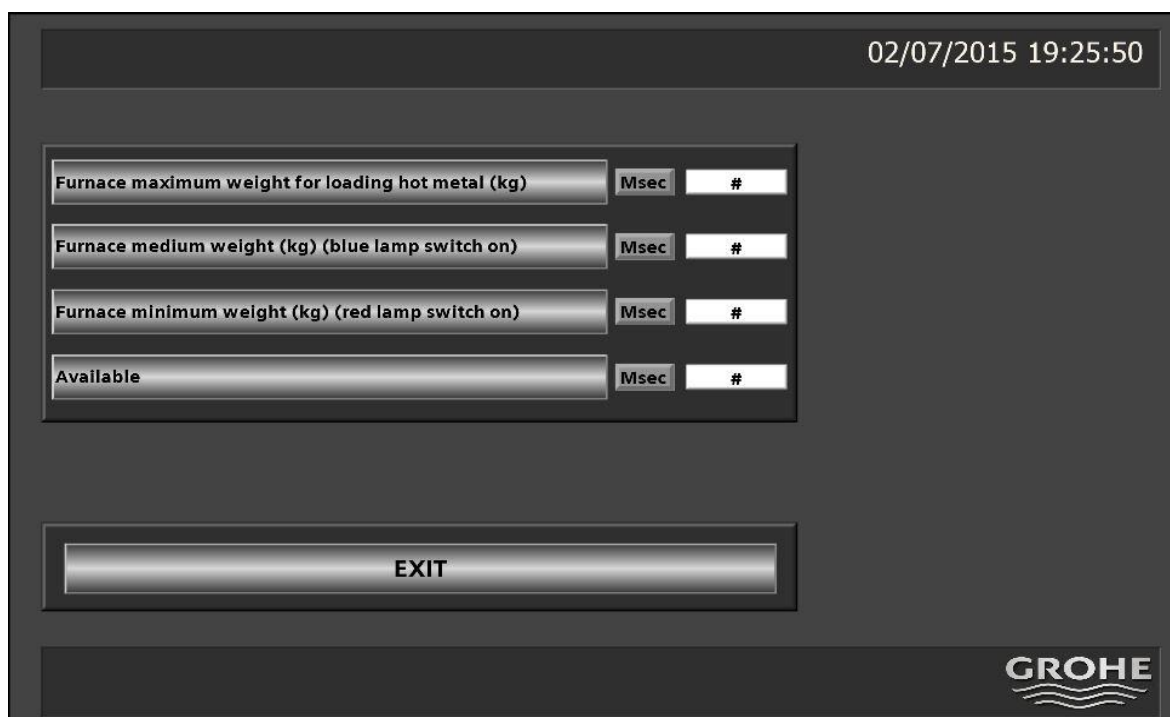


Figura 84 Configurações do forno.

Os parâmetros que podem ser configurados neste ecrã, são os seguintes:

Furnace maximum weight for loading hot metal – este valor indica a capacidade máxima do forno e é introduzido em quilogramas (kg). O peso medido pela balança do forno é apresentado no ecrã de controlo e visualização com a indicação de ‘*Actual weight*’.

Furnace médium weight (blue lamp on) – o valor a ser introduzido nesta variável serve como referência. Quando a quantidade de metal no forno estiver pela metade, acende-se uma lâmpada azul, para transmitir essa indicação.

Furnace minimum weight (red lamp on) – este valor corresponde ao valor mínimo de metal presente no forno. Quando este valor é atingido, é acesa a lâmpada vermelha. Esta é a indicação que o forno precisa de ser reabastecido.

7. TESTES E RESULTADOS

Com a programação do PLC e HMI já efetuadas, é necessário realizar testes que comprovem o correto funcionamento do sistema integrado. Para começar, foi criado um sistema que permitisse simular o forno da máquina para assim se puderem testar as funções criadas no PLC e verificar se o controlo é feito corretamente. Após a confirmação do funcionamento do novo sistema de controlo, procedeu-se à obtenção de resultados na máquina, testando também o HMI e a sua comunicação com o PLC.

7.1. PARAMETRIZAÇÃO DO CONTROLADOR PID PARA O BIDÃO

Para a parametrização do controlador PID, foi criado um sistema (apresentado na Figura 85) constituído pelo transdutor de pressão, a válvula proporcional, um controlador PLC Siemens com a CPU S7-314 e os módulos de entradas e saídas analógicas que ligam o transdutor de pressão e a válvula, respetivamente. Para simular o forno da máquina foi usado um bidão para testar o método de sintonia Cohen-Coon e a função de vazamento criada. Devido à demora na entrega da nova válvula E/P *pressure regulator*, Series ED12 da AVENTICS, foi usado o mesmo modelo da válvula já instalada no forno, acabando esta por ser usada na realização de todos os testes.



Figura 85 Sistema criado para simular o comportamento do forno.

Numa primeira fase, foi necessário determinar as constantes a serem enquadradas no bloco “CONT_C” do TIA Portal, seguindo as regras do método de sintonia de Cohen-Coon, descrito no Anexo F.

Recorrendo à ferramenta gráfica disponível através do bloco “CONT_C”, foi aplicado o valor de 10 % da capacidade máxima da válvula e esperou-se que o transdutor acompanhasse esse valor de pressão, permitindo que o sistema estabilizasse. De seguida impôs-se uma variação para os 30 % da capacidade total da válvula, para obter o gráfico desejado (apresentado na Figura 86).

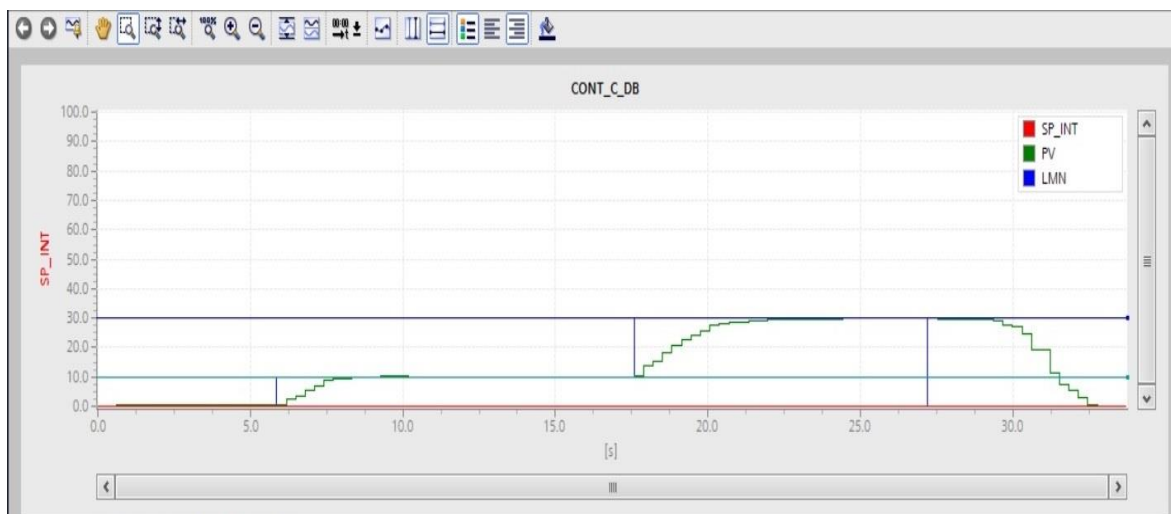


Figura 86 Variação do valor analógico da válvula.

Através da análise deste gráfico, foram obtidos vários valores que relacionam o tempo, em segundos (s), com a pressão, em percentagem (%), ou seja, os valores que descrevem a resposta do transdutor ao longo do tempo para atingir os 30 %, que correspondem a 300 milibar, da pressão pedida. A Tabela 6 mostra os valores obtidos.

Tabela 6 Relação tempo com pressão (bidão).

Tempo (s)	Pressão (%)
17,59	10,33
17,9	13,84
18,2	15,4
18,5	18,13
18,81	20,47
19,14	22,42
19,45	24,37
19,76	25,73
20,07	27,49
20,39	27,88
20,69	28,27
21	28,85
21,63	29,24
22,21	29,82
22,74	30,02

A Tabela 6 apresenta valores até ao instante de tempo 22,74 s, a que corresponde uma pressão de 30,02 %, porque a partir deste instante o sistema estabilizou e o valor de pressão manteve-se constante.

Para a obtenção do ganho do processo (gp), usou-se a expressão:

$$gp = \frac{\text{variação do transdutor}(\%)}{\text{variação da válvula}(\%)} \quad (1)$$

Através dos resultados apresentados na Tabela 6, o ganho do processo é:

$$gp = \frac{19,69}{20} = 0,9845 \quad (2)$$

Face aos resultados obtidos, e continuando a seguir as instruções do método Cohen-Coon, foi necessário calcular o declive máximo. Recorrendo ao Excel, foram introduzidos os valores da Tabela 6, e foi obtido o gráfico apresentado na Figura 87.

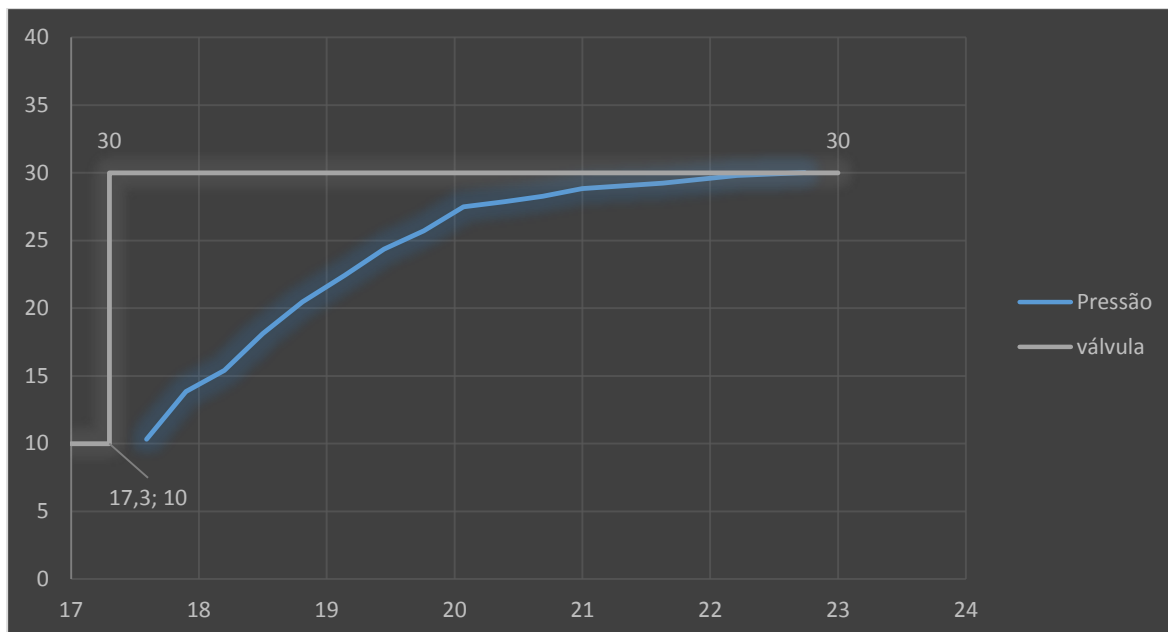


Figura 87 Gráfico da evolução dos valores de pressão determinada pelo transdutor ao longo do tempo (bidão).

Sendo a equação do declive dada por:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3)$$

Foram obtidos os valores apresentados na Tabela 7.

Tabela 7 Relação entre o tempo e pressão e respetivos declives (bidão).

Tempo	Pressão	$m = (y_2 - y_1) / (x_2 - x_1)$
17,59	10,33	11,32258065
17,9	13,84	5,2
18,2	15,4	9,1
18,5	18,13	7,548387097
18,81	20,47	5,909090909
19,14	22,42	6,290322581
19,45	24,37	4,387096774
19,76	25,73	5,677419355
20,07	27,49	1,21875
20,39	27,88	1,3
20,69	28,27	1,870967742
21	28,85	0,619047619
21,63	29,24	1
22,21	29,82	0,377358491
22,74	30,02	1,320140721

De acordo com estes valores, conclui-se que o declive é máximo no ponto (17,59; 10,33), apresentando o valor de $m = 11,323$.

Sabendo as coordenadas e o declive, determina-se a constante da equação da reta de acordo com:

$$y = mx + b \quad (4)$$

$$10,33 = 11,323 * 17,59 + b \quad (5)$$

$$b = -188,834 \quad (6)$$

Sabendo a constante b , a equação da reta tangente à curva é dada por:

$$y = 11,323x - 188,834 \quad (7)$$

Substituindo os valores referentes ao tempo na equação (7), são obtidos os valores apresentados na Tabela 8.

Tabela 8 Expressão da reta tangente ao longo do tempo (bidão).

Tempo	Pressão	$m = (y1-y0)/(x1-x0)$	$y = 11,32258065x - 188,834$
17,59	10,33	11,32258065	10,33
17,9	13,84	5,2	13,84
18,2	15,4	9,1	17,23677419
18,5	18,13	7,548387097	20,63354839
18,81	20,47	5,909090909	24,14354839
19,14	22,42	6,290322581	27,88
19,45	24,37	4,387096774	31,39
19,76	25,73	5,677419355	34,9
20,07	27,49	1,21875	38,41
20,39	27,88	1,3	42,03322581
20,69	28,27	1,870967742	45,43
21	28,85	0,619047619	48,94
21,63	29,24	1	56,07322581
22,21	29,82	0,377358491	62,64032258
22,74	30,02	1,320140721	68,64129032

Usando agora os valores de pressão da equação da reta, obtém-se o gráfico apresentado na Figura 88.

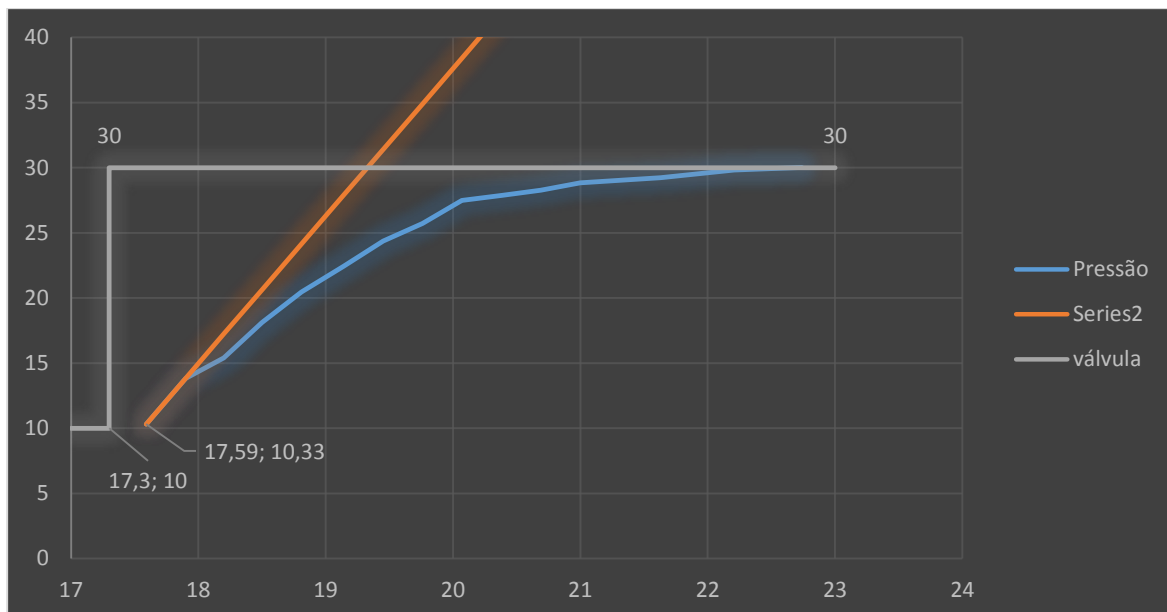


Figura 88 Reta tangente à curva (bidão).

De acordo com as regras do método de Cohen-Coon, é necessário calcular o ponto correspondente a 63 % da variação total do transdutor. Uma vez que a curva apresentada varia de 10,33 % até aos 30,02 %, então tem-se:

$$30,02 - 10,33 = 19,69 \quad (8)$$

$$19,69 * 0,63 = 12,4047 \quad (9)$$

$$12,4047 + 10,33 = 22,7347 \quad (10)$$

Fazendo a intersecção deste ponto com a curva inicial, é possível obter o valor temporal correspondente e assim conhecer as coordenadas do ponto, obtendo-se o gráfico da Figura 89.

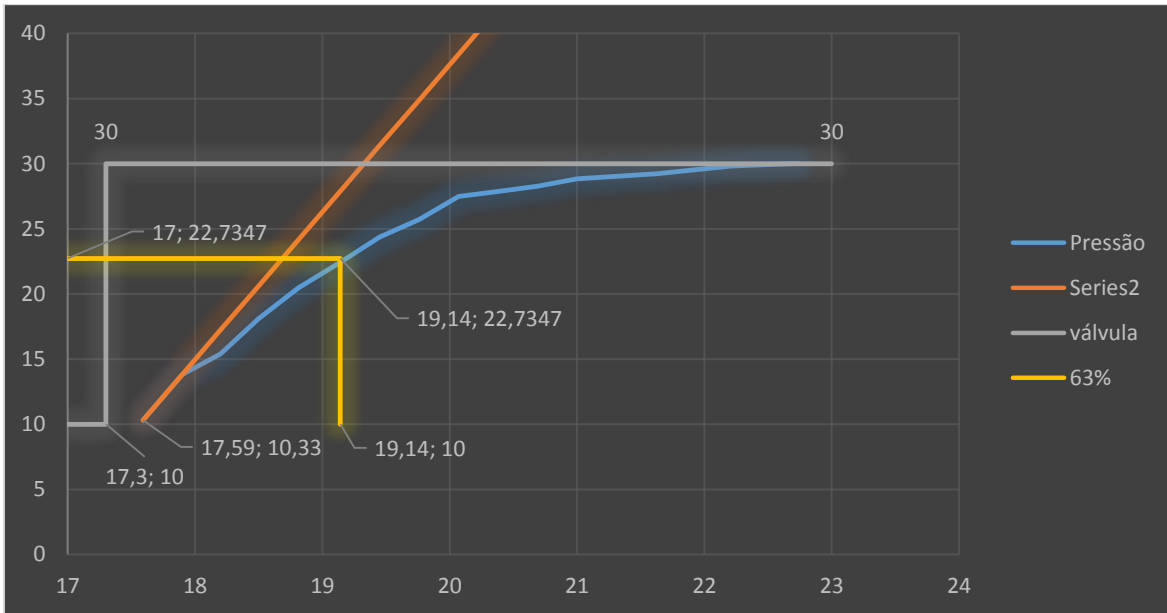


Figura 89 Gráfico com as condições necessárias para aplicar método Cohen-Coon (bidão).

Conhecendo os pontos essenciais, resta retirar o valor do *dead time* (t_d) e da constante de tempo τ . O cálculo destes valores é dado por:

$$t_d = 17,59 - 17,3 = 0,29 \quad (11)$$

$$\tau = 19,14 - 17,59 = 1,55 \quad (12)$$

Por fim, após estarem calculados todos os parâmetros necessários para a aplicação das fórmulas Cohen-Coon, resta aplicá-las. A Figura 90 mostra uma tabela com as diferentes expressões a aplicar, de acordo com o algoritmo de controlo desejado (P, PI ou PID).

www.opticontrols.com	Controller Gain	Integral Time	Derivative Time
P Controller:	$K_c = \frac{1.03}{g_p} \left(\frac{\tau}{t_d} + 0.34 \right)$		
PI Controller:	$K_c = \frac{0.9}{g_p} \left(\frac{\tau}{t_d} + 0.092 \right)$	$T_i = 3.33 t_d \frac{\tau + 0.092 t_d}{\tau + 2.22 t_d}$	
PD Controller:	$K_c = \frac{1.24}{g_p} \left(\frac{\tau}{t_d} + 0.129 \right)$		$T_D = 0.27 t_d \frac{\tau - 0.324 t_d}{\tau + 0.129 t_d}$
PID Controller: (Noninteracting)	$K_c = \frac{1.35}{g_p} \left(\frac{\tau}{t_d} + 0.185 \right)$	$T_i = 2.5 t_d \frac{\tau + 0.185 t_d}{\tau + 0.611 t_d}$	$T_D = 0.37 t_d \frac{\tau}{\tau + 0.185 t_d}$

Figura 90 Fórmulas para calcular as constantes PID segundo o método de Cohen-Coon.

Para este projeto, o algoritmo de controlo aplicado foi o PI. Aplicando as correspondentes fórmulas apresentadas na Figura 90, vem:

$$K_c = \frac{0,9}{gp} \left(\frac{\tau}{t_d} + 0,092 \right) \quad (13)$$

$$K_c = \frac{0,9}{0,9845} \left(\frac{1,55}{0,29} + 0,092 \right) \quad (14)$$

$$K_c = 4,970183 \quad (15)$$

$$T_i = 3,33t_d \frac{(\tau + 0,092t_d)}{\tau + 2,22t_d} \quad (16)$$

$$T_i = 3,33 * 0,29 * \frac{(1,55 + 0,092 * 0,29)}{1,55 + 2,22 * 0,29} \quad (17)$$

$$T_i = 0,670558 \text{ s} \quad (18)$$

Como é aconselhado pelo guia da sintonia dividir os valores das constantes por dois, vem:

$$\frac{K_c}{2} = 2,4851 \quad (19)$$

$$\frac{T_i}{2} = 0,3353 \text{ s} \quad (20)$$

Foram realizados vários testes para retirar os valores necessários para a sintonia do controlador PID. No entanto, como as constantes obtidas não variaram muito, apenas é apresentado um caso de estudo.

7.1.1. RESULTADOS DO SISTEMA DE CONTROLO APLICADO AO BIDÃO

Com as constantes K_c e T_i , é possível configurar o bloco “CONT_C” do TIA Portal. Assim foram introduzidas as novas constantes no bloco PID, como mostra a Figura 91.

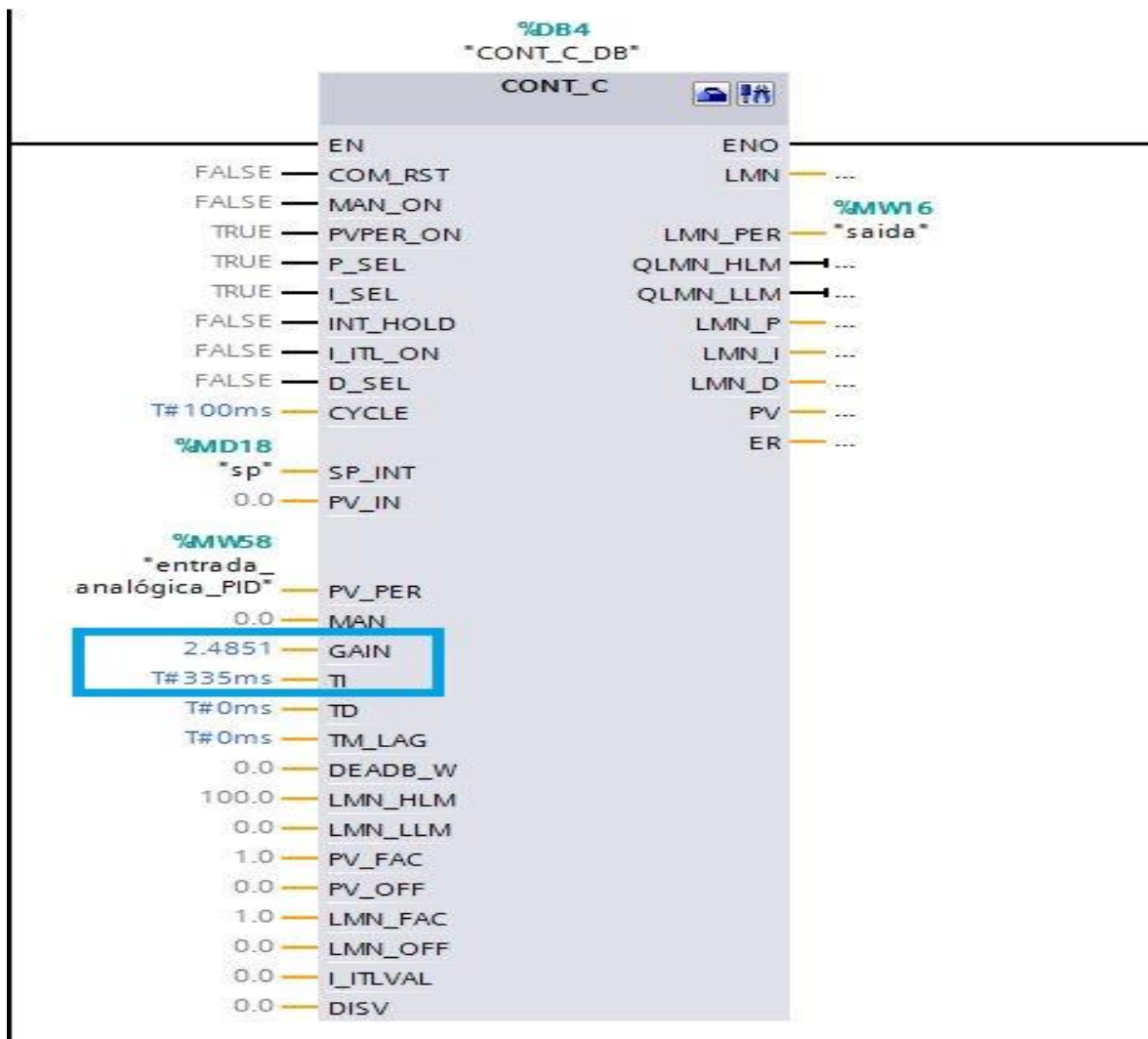


Figura 91 Atribuição das constantes K_c e T_i ao bloco PID “CONT_C”.

Posteriormente, recorrendo à função criada “controle_vazamento” foi possível testar o controlo da pressão dentro bidão. De salientar que para este teste apenas foram usadas as novas funções, simplesmente para verificar também o seu funcionamento.

De acordo com a Figura 92, foram definidos valores de pressão para simular uma curva pré-definida.

funciona_pls_3 > PLC_1 [CPU 314] > Program blocks > Dados da Peça X [DB1]

Dados da Peça X

	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint
1	Static						
2	a	Int	0.0	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	b	Int	2.0	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	c	Int	4.0	120	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	d_1	Int	6.0	135	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	e	Int	8.0	140	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	f	Int	10.0	147	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	g	Int	12.0	153	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	h	Int	14.0	159	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	i	Int	16.0	169	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	j	Int	18.0	180	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	g_1	Int	20.0	194	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	h_1	Int	22.0	200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	a_1	Int	24.0	210	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	a_2	Int	26.0	220	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	a_3	Int	28.0	220	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	w	Int	30.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 92 Valores estabelecidos de pressão para a curva pré-definida.

Após todas as configurações, o PLC foi colocado em modo *run* e iniciado o teste de pressão. Os resultados obtidos são apresentados na Figura 93.

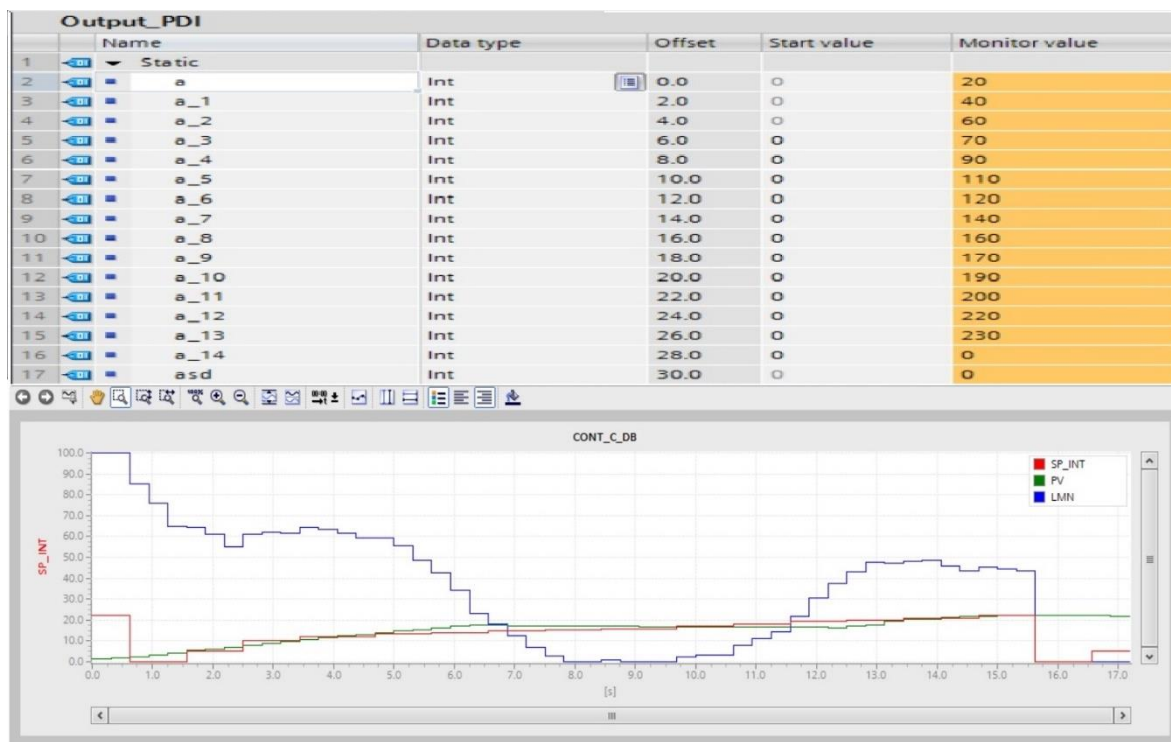


Figura 93 Resultados obtidos com o controlador PID aplicado ao bidão.

Como pode ser verificado através da Figura 93, a função criada acompanhou as especificações exigidas, no que toca a enviar os valores de segundo a segundo para a variável de *setpoint* do bloco “CONT_C”, visto que o bloco realizou os cálculos, e os valores foram enviados para a válvula e lidos pelo transdutor, assim como o *setpoint* foi variando ao longo do tempo.

Na imagem inferior da Figura 93, é apresentado um gráfico onde se pode verificar o acompanhamento dos valores lidos pelo transdutor (curva a verde) face aos valores de *setpoint* (curva a vermelho). A curva azul mostra o funcionamento da válvula ao longo do tempo. Esta é a que mais varia devido à relação com os valores do transdutor e cálculos PID, uma vez que a válvula tem que garantir os valores de *setpoint* consoante os valores lidos pelo transdutor.

Comparando os valores pré-definidos com os valores que foram sendo inseridos no DB de *output*, verifica-se uma diferença, principalmente nos valores iniciais lidos pelo transdutor. No entanto, tal facto pode estar relacionado com a pressão vinda da rede da fábrica, isto porque a pressão da rede é de 6 bar e a válvula apenas suporta 2 bar (no máximo). Assim, através de uma alavanca, foi aberta uma “quantidade pequena” (também como forma de segurança, pois a pressão excessiva podia rebentar com o bidão), e como não foi possível recorrer a um instrumento de medida para verificar a pressão de saída da rede, não foi possível garantir que a pressão de entrada na válvula era a adequada e/ou suficiente. Talvez por isto é que exista uma dificuldade para acompanhar os valores iniciais estabelecidos. No entanto, a determinada altura os resultados passam a ser semelhantes.

Este teste foi desenvolvido para garantir que a função criada no PLC funcionava corretamente, como pode ser evidenciado. De acordo com as condicionantes apresentadas, o objetivo foi cumprido.

Nas secções seguintes serão apresentados testes efetuados no forno da máquina, onde as condições de entrada de ar para a válvula são as apropriadas, e a partir dos quais poderão ser retiradas conclusões mais fiáveis.

7.2. PARAMETRIZAÇÃO DO CONTROLADOR PID PARA O FORNO – TESTE 1

Para a parametrização do controlador PID do forno integrado na máquina (apresentado na Figura 94), foram usados o mesmo transdutor e um modelo igual à válvula utilizados no

sistema apresentado na Subsecção 5.3.3, e apenas foi substituída a CPU da máquina (CPU 315-2 DP), pela nova CPU 314.



Figura 94 Forno da máquina.

Recorrendo novamente à ferramenta gráfica disponível através do bloco “CONT_C”, foi aplicado o valor de 10 % da capacidade máxima da válvula e esperou-se que o transdutor acompanhasse esse valor de pressão, permitindo que o sistema estabilizasse. De seguida provocou-se uma variação para os 50 % da capacidade total da válvula, para obter o gráfico desejado (Figura 95).

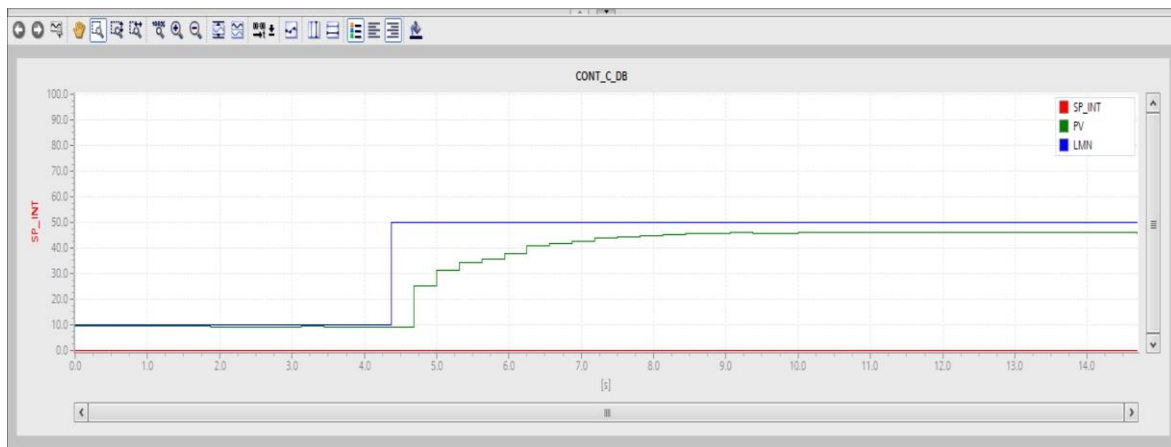


Figura 95 Variação do valor analógico da válvula (teste 1).

Através da análise deste gráfico foram obtidos vários valores que relacionam o tempo (em segundos) com a pressão (em percentagem), ou seja, os valores que descrevem o acompanhamento do transdutor ao longo do tempo para atingir os 50 %, que correspondem a 500 milibar, da pressão pedida. Devido a existirem fugas no forno, e como pode ser visto na Figura 95, a pressão no forno não chega a atingir os 500 milibar desejados. A Tabela 9 mostra os vários valores obtidos para este caso.

Tabela 9 Relação tempo com pressão (teste1).

Tempo (s)	Pressão (%)
4,688	9,13
4,688	25,22
5	31,3
5,315	34,35
5,628	35,65
5,942	37,83
6,245	40,87
6,559	41,74
6,873	42,61
7,187	43,91
7,501	44,35
7,815	44,78
8,129	45,22
8,443	45,65
9,06	46,09

Assim através dos resultados apresentados na Tabela 9, o cálculo do ganho do processo vem:

Face aos resultados obtidos, calculou-se o declive máximo. Recorrendo ao Excel, foram

$$gp = \frac{36,96}{40} = 0,924 \quad (21)$$

introduzidos os valores da Tabela 9, e foi obtido o gráfico apresentado na Figura 96.

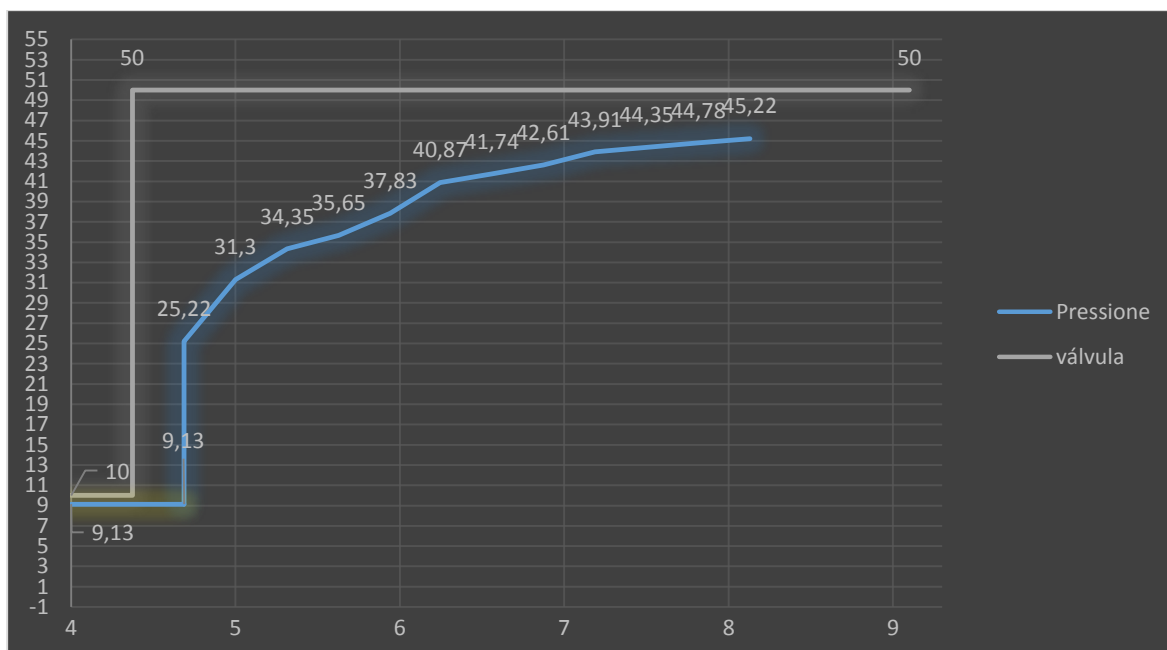


Figura 96 Gráfico obtido através dos valores de pressão do transdutor ao longo do tempo (teste 1).

A Tabela 10 mostra o resultado dos declives calculados.

Tabela 10 Relação tempo com pressão e respectivos declives (teste 1).

Tempo	Pressão	$m = (y_2 - y_1) / (x_2 - x_1)$
4,688	9,13	∞
4,688	25,22	19,48717949
5	31,3	9,682539683
5,315	34,35	4,153354633
5,628	35,65	6,942675159
5,942	37,83	10,0330033
6,245	40,87	2,770700637
6,559	41,74	2,770700637
6,873	42,61	4,140127389
7,187	43,91	1,401273885
7,501	44,35	1,369426752
7,815	44,78	1,401273885
8,129	45,22	1,369426752
8,443	45,65	0,713128039
9,06	46,09	5,087196468

De acordo com estes valores, conclui-se que o declive é máximo no ponto (4,688; 9,13), apresentando o valor de $m = \infty$.

Sendo o declive infinito, a reta obtida será como a mostrada na Figura 97, ou seja será uma reta vertical no ponto $x = 4,688$, tangente ao ponto onde o declive é infinito (4,688; 9,13). A reta tangente intersecta a curva no ponto 9,13 %.

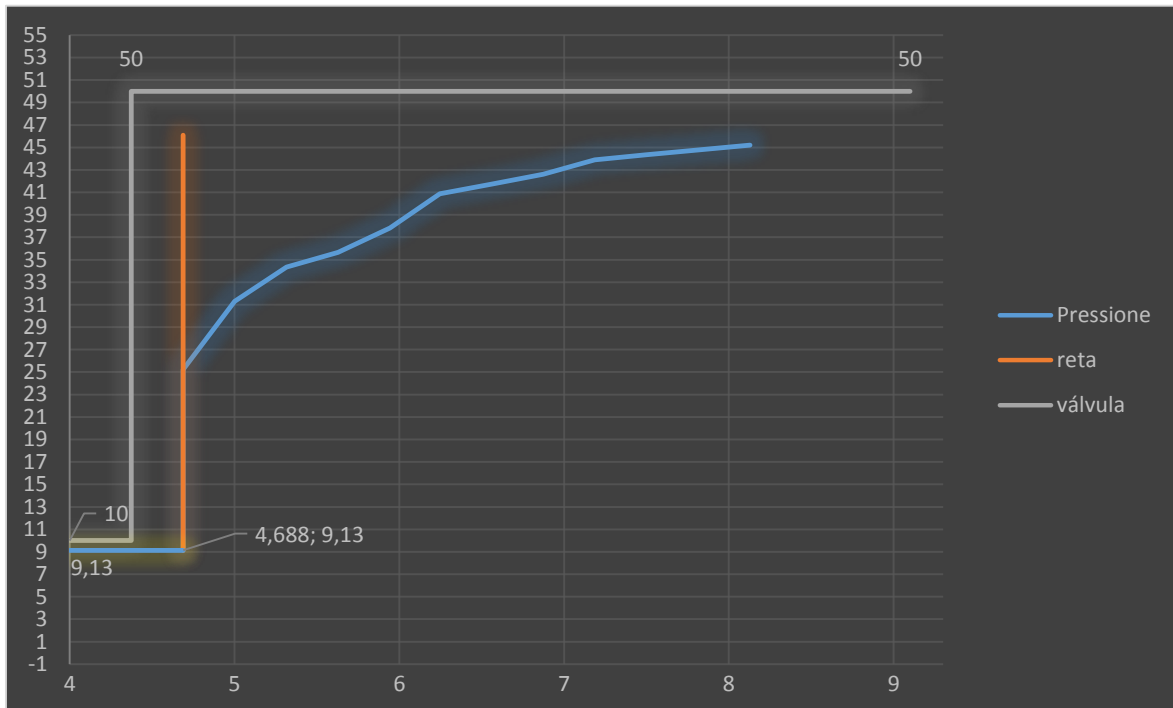


Figura 97 Reta tangente à curva (teste 1).

É necessário calcular o ponto para 63 % da variação total do transdutor, que varia de 9,13 % até aos 46,09 %, então:

$$46,09 - 9,13 = 36,96 \quad (22)$$

$$36,96 * 0,63 = 23,2848 \quad (23)$$

$$23,2848 + 9,13 = 32,4148 \quad (24)$$

Fazendo a intersecção deste ponto com a curva inicial, será possível obter o valor temporal e assim conhecer as coordenadas do ponto, obtendo-se o gráfico da Figura 98.

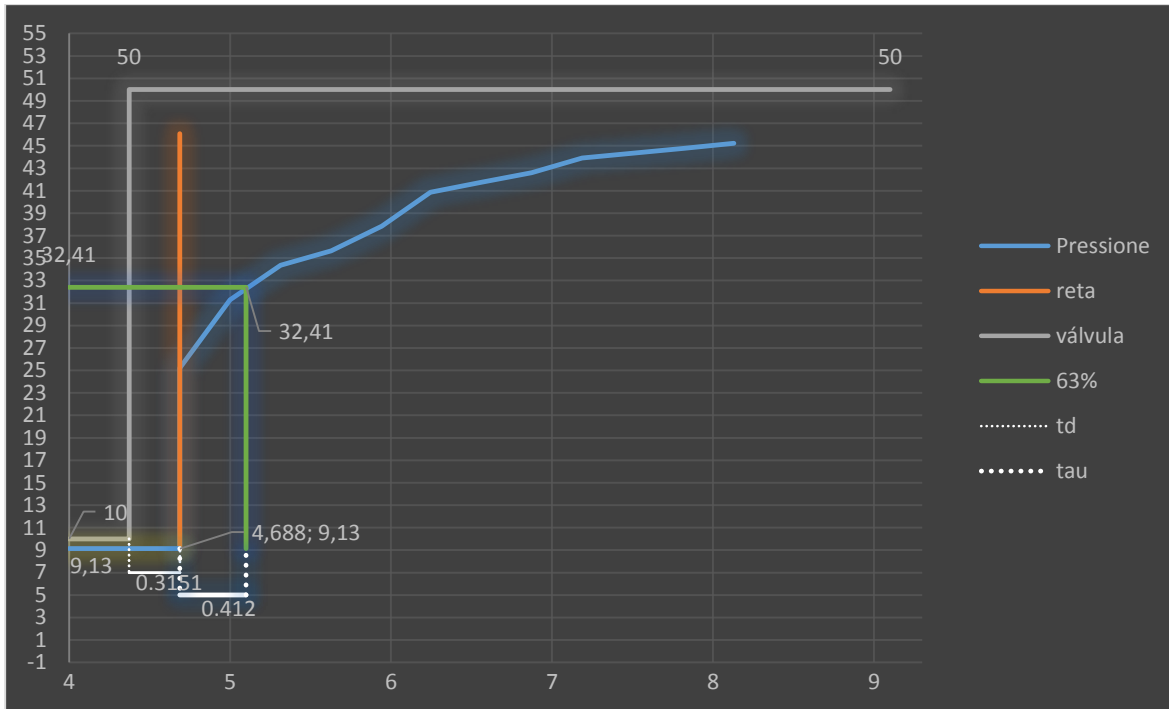


Figura 98 Gráfico com as condições necessárias para aplicar o método de Cohen-Coon (teste 1).

Conhecendo os pontos essenciais apenas resta retirar o valor do *dead time* (t_d) e da constante de tempo τ . O cálculo destes valores é dado por:

$$t_d = 4,688 - 4,3729 = 0,3151 \quad (25)$$

$$\tau = 5,1 - 4,688 = 0,412 \quad (26)$$

Após estarem calculados os parâmetros necessários para a aplicação das expressões relativas ao método de Cohen-Coon, estas foram aplicadas. Uma vez que o algoritmo de controlo aplicado foi o PI, vem:

$$K_c = \frac{0,9}{0,924} \left(\frac{0,412}{0,3151} + 0,092 \right) \quad (27)$$

$$K_c = 1,36317 \quad (28)$$

$$T_i = 3,33 * 0,3151 * \frac{(0,412 + 0,092 * 0,3151)}{0,412 + 2,22 * 0,3151} \quad (29)$$

$$T_i = 0,416296 \text{ s} \quad (30)$$

Como aconselhado pelo guia da sintonia, dividiram-se os valores das constantes por dois, obtendo-se:

$$\frac{K_c}{2} = 0,681585 \quad (31)$$

$$\frac{T_i}{2} = 0,20815 \text{ s} \quad (32)$$

7.3. PARAMETRIZAÇÃO DO CONTROLADOR PID PARA O FORNO – TESTE 2

De modo a verificar se as constantes obtidas na Secção 7.2 estão corretas, foram retirados novos valores, realizando para isso outro teste no forno, para no final as constantes serem comparadas.

Através da ferramenta gráfica do bloco “CONT_C”, foi aplicado o valor de 10 % da capacidade máxima da válvula e esperou-se que o transdutor acompanhasse esse valor de pressão, até estabilizar. De seguida provocou-se uma variação para os 40 % da capacidade total da válvula, para obter então o gráfico desejado (Figura 99).

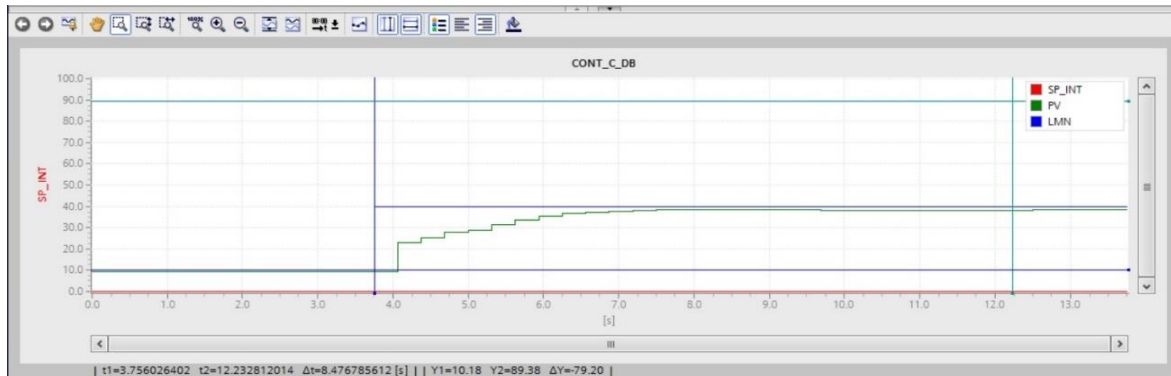


Figura 99 Variação do valor analógico da válvula (teste 2).

Através da análise deste gráfico foram obtidos vários valores que relacionam o tempo com a pressão, ou seja, os valores que descrevem o acompanhamento do transdutor ao longo do tempo para atingir os 40 %, que correspondem a 400 milibar, da pressão pedida. Devido a existirem fugas no forno, e como pode ser visto na Figura 99, a pressão no forno não chega a atingir os 400 milibar desejados. A Tabela 11 mostra os vários valores obtidos para este caso.

Tabela 11 Relação tempo com pressão (teste 2).

Tempo (s)	Pressão (%)
4,065	9,29
4,065	23,01
4,373	25,22
4,682	27,88
5,004	28,76
5,312	31,42
5,621	33,63
5,943	35,4
6,251	36,73
6,56	37,17
6,869	37,61
7,203	38,05
7,499	38,5

O ganho do processo (gp) é:

$$gp = \frac{29,21}{29,54} = 0,9855 \quad (33)$$

Face aos resultados obtidos, e continuando a seguir as instruções do método de Cohen-Coon, foi necessário calcular o declive máximo. Foram introduzidos os valores da Tabela 11 numa folha de cálculo do Excel, obtendo-se o gráfico apresentado na Figura 100.

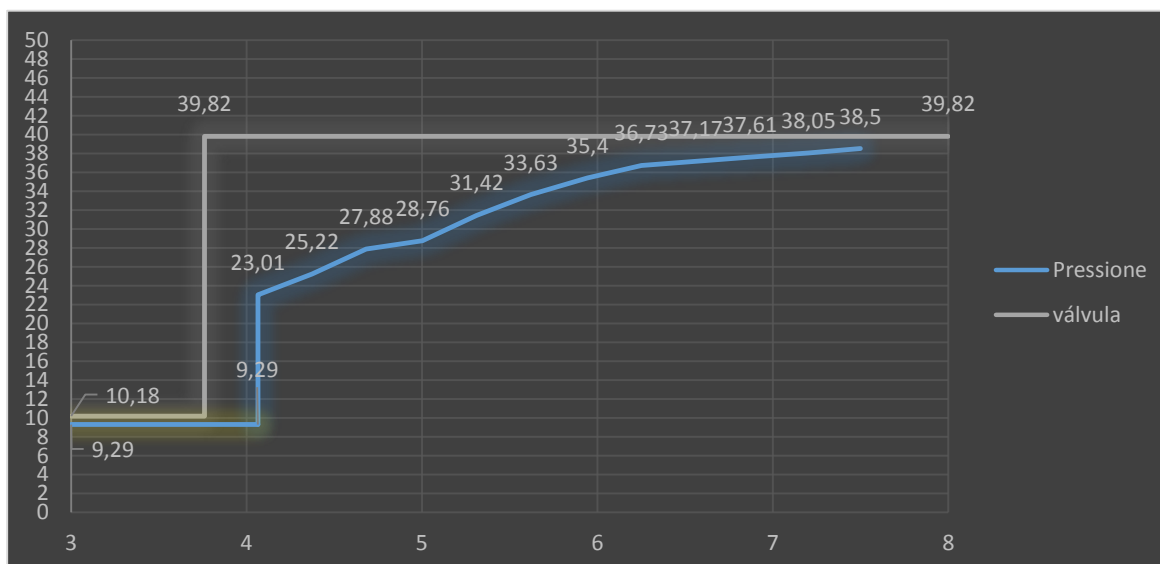


Figura 100 Gráfico obtido através dos valores de pressão do transdutor ao longo do tempo (teste 2).

Os valores para os declives dos segmentos de reta são apresentados na Tabela 12.

Tabela 12 Relação tempo com pressão e respectivos declives (teste 2).

Tempo	Pressão	$m = (y_2 - y_1) / (x_2 - x_1)$
4,065	9,29	∞
4,065	23,01	7,175324675
4,373	25,22	8,608414239
4,682	27,88	2,732919255
5,004	28,76	8,636363636
5,312	31,42	7,15210356
5,621	33,63	5,49689441
5,943	35,4	4,318181818
6,251	36,73	1,42394822
6,56	37,17	1,42394822
6,869	37,61	1,317365269
7,203	38,05	1,52027027
7,499	38,5	5,134017869

De acordo com estes valores, conclui-se que o declive é máximo no ponto (4,065; 9,29), apresentando o valor de $m = \infty$.

Sendo o declive infinito, a reta obtida será como a mostrada na Figura 101, ou seja será uma reta vertical no ponto $x = 4,065$, tangente ao ponto onde o declive é infinito (4,065; 9,29). A reta tangente intersecta a curva no ponto 9,29 %.

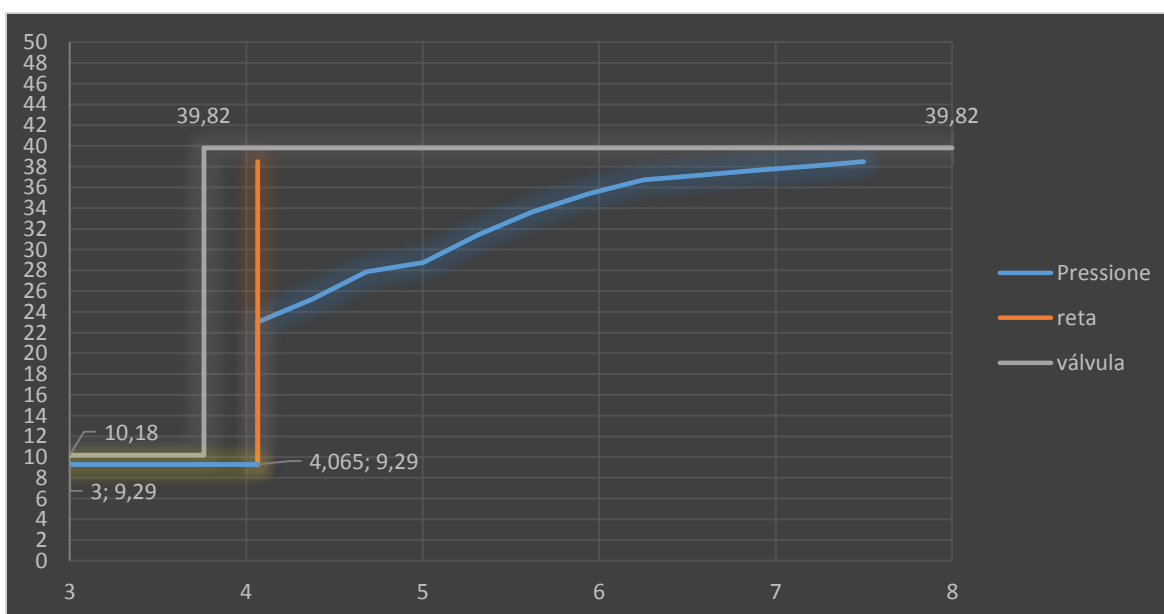


Figura 101 Reta tangente à curva (teste 2).

De acordo com as regras do método de sintonia de Cohen-Coon, é calculado o ponto para 63 % da variação total do transdutor. Uma vez que a curva apresentada varia de 9,29 % até aos 38,5 %, obtém-se:

$$38,5 - 9,29 = 29,21 \quad (34)$$

$$29,21 * 0,63 = 18,4023 \quad (35)$$

$$18,4023 + 9,29 = 27,6923 \quad (36)$$

Fazendo a intersecção deste ponto com a curva inicial, é possível obter o valor temporal e assim conhecer as coordenadas do ponto, obtendo-se o gráfico da Figura 102.

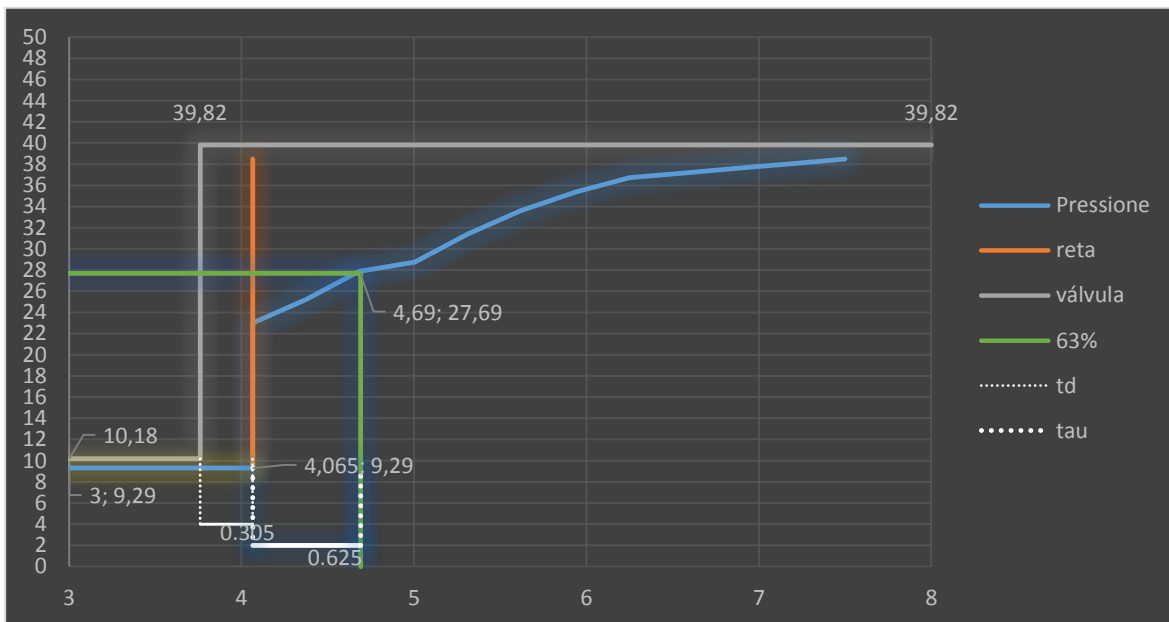


Figura 102 Gráfico com as condições necessárias para aplicar o método de Cohen-Coon (teste 2).

Conhecendo os pontos essenciais, apenas resta retirar o valor do *dead time* (t_d) e da constante de tempo τ , sendo estes valores:

$$K_c = \frac{0,9}{0,986} \left(\frac{0,625}{0,305} + 0,092 \right) \quad (37)$$

$$K_c = 1,955431 \quad (38)$$

$$t_d = 4,065 - 3,76 = 0,305 \quad (39)$$

$$\tau = 4,69 - 4,065 = 0,625 \quad (40)$$

Por fim, após os parâmetros necessários para a aplicação das fórmulas do método de Cohen-Coon estarem calculados, foram aplicados. Como o algoritmo de controlo aplicado foi o PI, tem-se:

$$T_i = 3,33 * 0,305 * \frac{(0,625 + 0,092 * 0,305)}{0,625 + 2,22 * 0,305} \quad (41)$$

$$T_i = 0,465619 \text{ s} \quad (42)$$

Dividindo os valores das constantes por dois, como aconselhado pelo guia da sintonia, vem:

$$\frac{K_c}{2} = 0,978 \quad (43)$$

$$\frac{T_i}{2} = 0,23281 \text{ s} \quad (44)$$

Perante os resultados dos dois testes, verifica-se uma ligeira diferença nos valores das constantes. Tal facto pode ter a ver com o *setpoint* definido, que foi diferente para os dois testes, sendo que no primeiro a pressão variou dos 10 % até aos 50 %, e no segundo dos 10 % aos 40 %. No entanto, a conclusão será tirada experimentando ambos os valores na prática e verificando se a diferença é significativa ou não.

7.4. RESULTADOS DOS TESTES NA MÁQUINA

Nesta secção são apresentados os resultados que foram possíveis obter, de acordo com as limitações de disponibilidade da máquina para testes.

São apresentados resultados para ambas as constantes calculadas e realizadas comparações. Para estes primeiros testes não foi usado metal fundido, o bico do forno foi tapado e apenas foi injetado ar para verificar se as especificações da curva criada eram cumpridas. Para tal, começou-se por definir uma curva com valores aleatórios (conforme é mostrado na Figura 103) para ambas as cabeças.

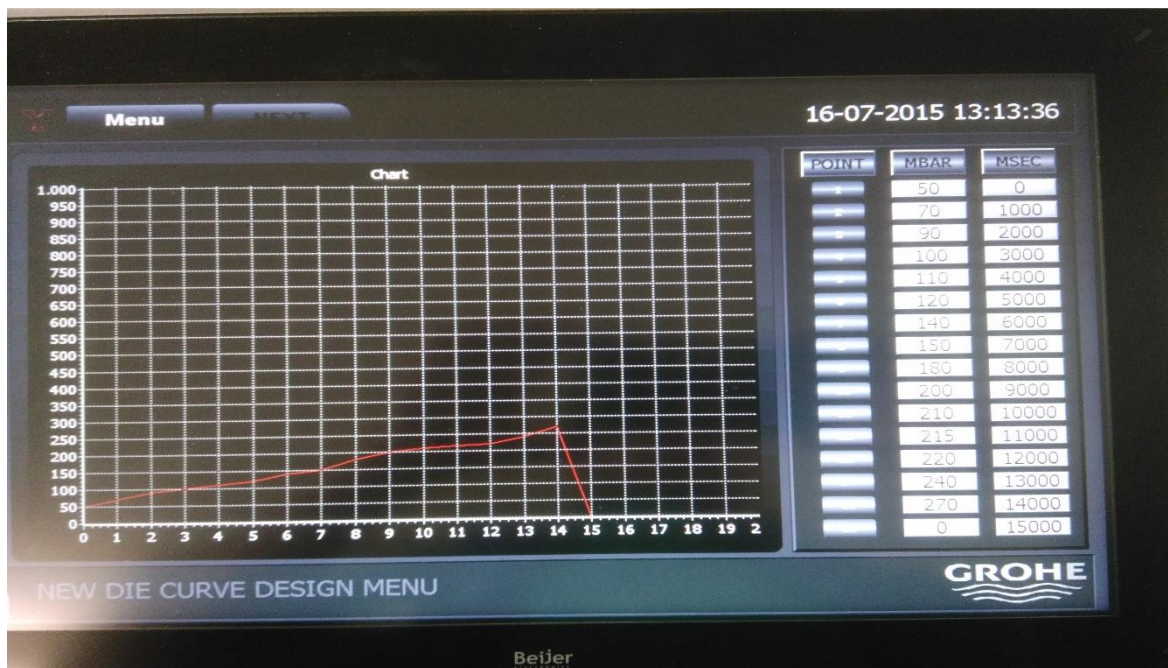


Figura 103 Definição de uma curva com valores aleatórios.

Após a definição da curva a máquina foi colocada em funcionamento, e quando uma das cabeças estivesse na posição do forno iniciava-se o processo de vazamento.

Aplicando as constantes PI obtidas na Secção 7.2 ($K_c = 0,6816$ e $T_i = 0,208$), o resultado obtido é mostrado na Figura 104.

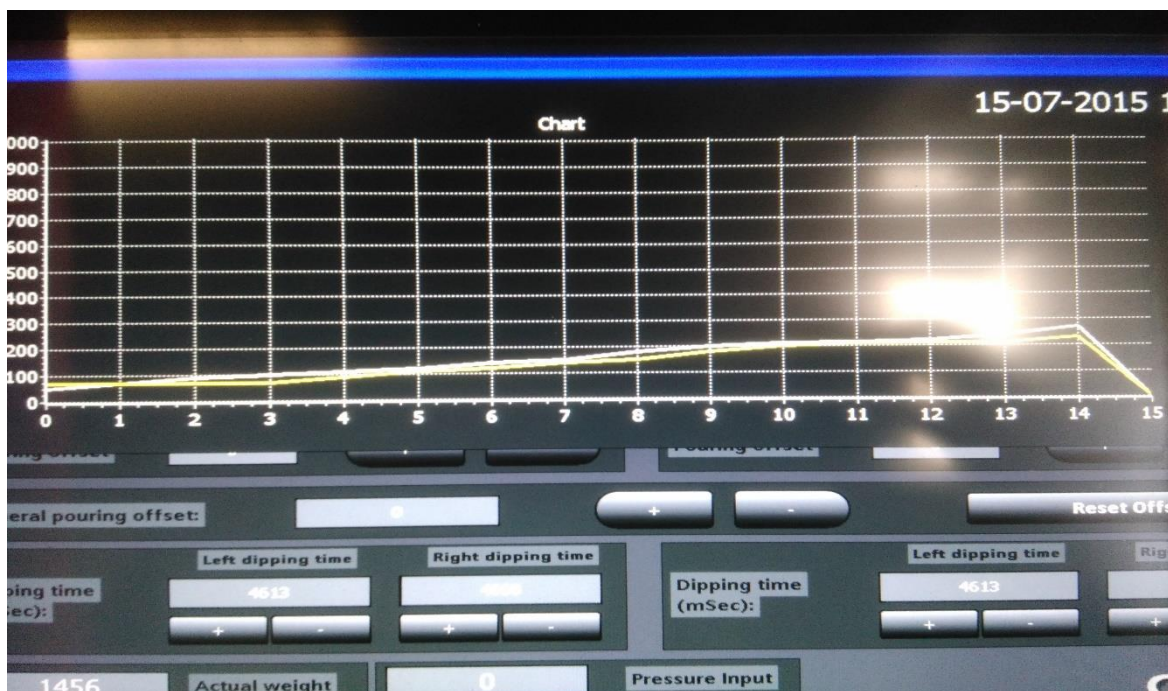


Figura 104 Resposta do sistema de controlo com as constantes $K_c = 0,6816$ e $T_i = 0,208$.

Aplicando as constantes PI calculadas na Secção 7.3 ($K_c = 0,978$ e $T_i = 0,233$), obteve-se o resultado apresentado na Figura 105.

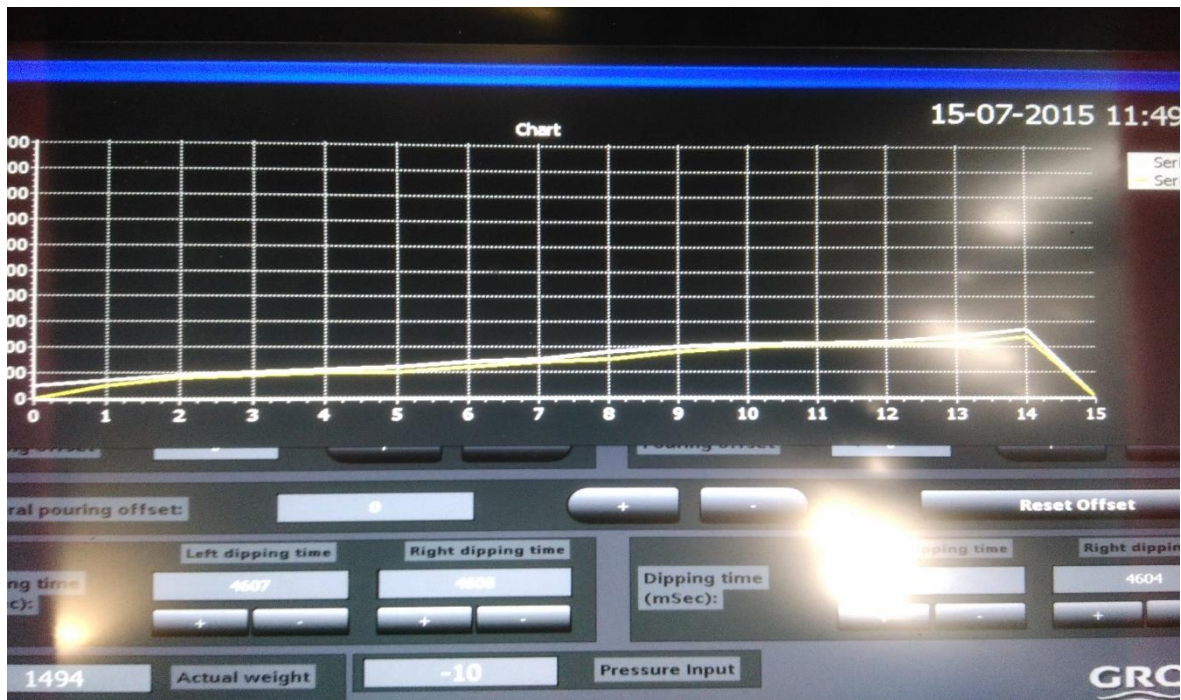


Figura 105 Resposta do sistema de controlo com as constantes $K_c = 0,978$ e $T_i = 0,233$.

De acordo com os resultados apresentados, em que a linha branca representa a curva padrão (*setpoint*) e a linha a amarelo a resposta lida pelo transdutor, de um modo geral a resposta obtida é semelhante para ambos os casos. Apenas existe uma ligeira diferença no tempo de resposta da Figura 105, uma vez que esta começou a responder com um valor próximo de zero, enquanto a resposta da Figura 104 aproximou-se muito mais do *setpoint* estabelecido. Tal facto deve-se à constante de tempo integral, uma vez que para o primeiro caso este valor é menor.

Com os resultados obtidos, e após vários testes para a mesma curva, alternando apenas as constantes (ou seja, trocando os valores das contantes entre $K_c = 0,978$ e $T_i = 0,233$ e $K_c = 0,6816$ e $T_i = 0,208$), concluiu-se que os resultados para os valores de $K_c = 0,6816$ e $T_i = 0,208$ aproximavam-se mais dos valores da curva pré-definida. Por este motivo, na realização dos testes seguintes foram usados estes valores.

Para a realização dos testes apresentados a seguir, foram definidos novos valores para a curva estabelecida, aproximando-se mais dos valores usados para a realização do vazamento, para assim verificar se o sistema de controlo desenvolvido consegue acompanhar curvas

diferentes. A Figura 106 mostra a definição de uma curva para a cabeça Y com diferentes valores.

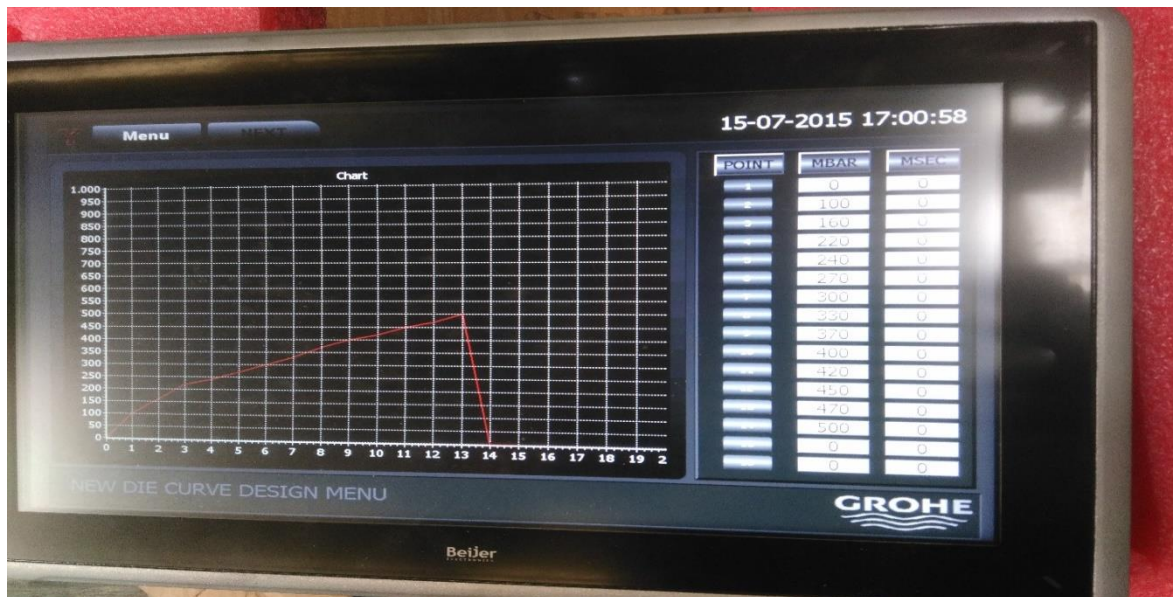


Figura 106 Criação de uma nova curva (teste na máquina).

Apesar dos valores das constantes usadas levarem a bons resultados, alterou-se ligeiramente o valor da constante T_i para 0,198 s para que a resposta fosse ligeiramente mais rápida, melhorando assim os resultados como se pode comprovar na Figura 107.



Figura 107 Resposta do sistema de controlo (teste) com as constantes $K_c = 0,6816$ e $T_i = 0,198$.

Mais uma vez o comportamento do sistema de controlo foi o previsto e, com a ligeira alteração do valor $T_i = 0,198$ s, a resposta obtida melhorou, visto que a curva estabelecida e a curva da resposta são bastante semelhantes, como é perceptível na Figura 107.

Para a cabeça X da máquina foi definida uma curva “real”, ou seja, os valores introduzidos foram os mesmos usados para vaziar um modelo de peça produzido. A Figura 108 apresenta a definição da curva de vazamento real de uma determinada peça.

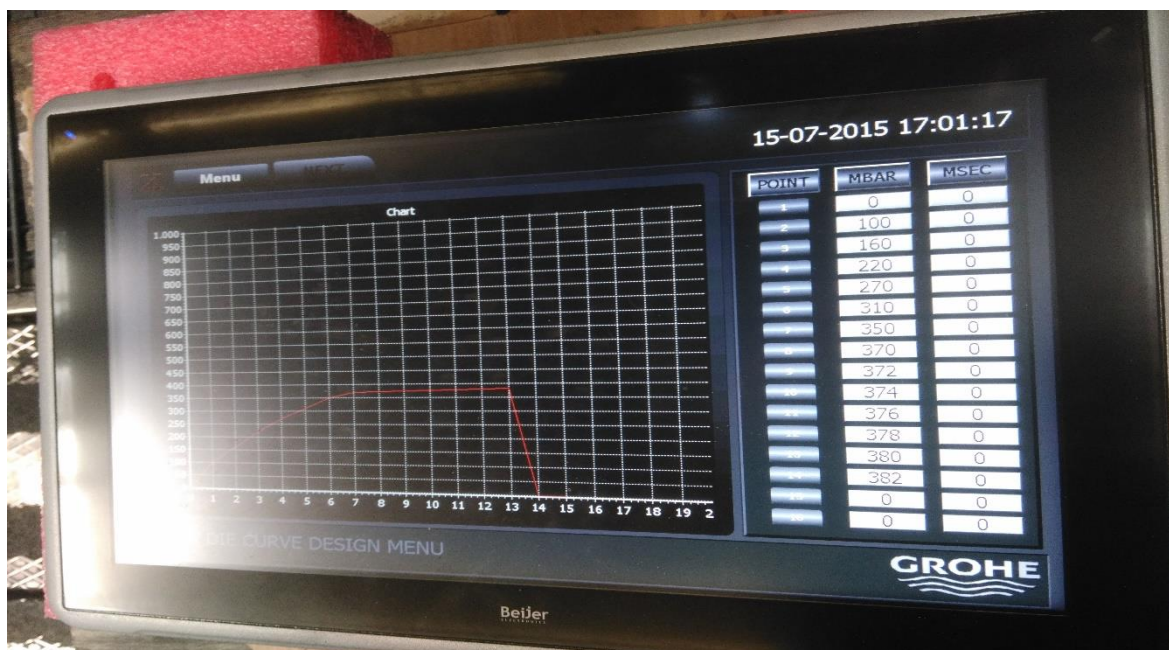


Figura 108 Definição de uma nova curva (curva real).

Os parâmetros utilizados para este teste foram os mesmos que no caso anterior (teste na máquina), devido aos bons resultados apresentados. De salientar que estes dois testes foram realizados um após outro, ou seja, a máquina foi colocada em ciclo automático e mal terminou o primeiro teste de vazamento começou o segundo teste, isto para verificar se o sistema de controlo é capaz de se adaptar a valores de curvas diferentes.

Assim, após a realização do primeiro ensaio (teste na máquina), começou o segundo de acordo com os valores definidos (e apresentados na Figura 108), sendo os resultados obtidos apresentados na Figura 109.

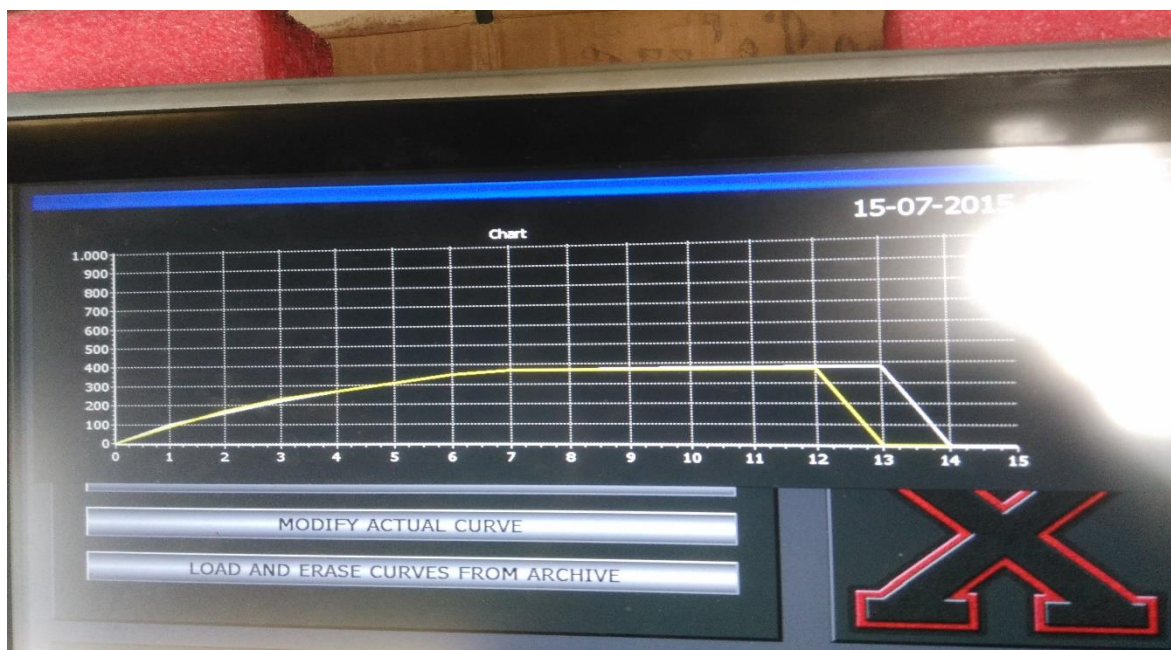


Figura 109 Resposta do sistema de controlo (curva real) com as constantes $K_c = 0,6816$ e $T_i = 0,198$.

Mais uma vez os resultados obtidos foram considerados positivos, pois a resposta obtida seguiu os valores de *setpoint* estabelecidos. De notar que na Figura 109 o sistema de controlo ainda não havia terminado quando a imagem foi obtida. Por este motivo, a linha que no ponto 12 do eixo do x tende para zero (no ponto 13), não significa que a pressão caiu para zero. Simplesmente diz respeito à forma como os valores são apresentados no gráfico, pois um *array* tem que ser definido inicialmente com os parâmetros todos a zero, e ao longo do tempo os vários valores nessas posições do *array* vão sendo atualizados. Neste caso, ao fim de 13 segundos (ponto 13 do eixo x) o valor ainda ia ser atualizado de acordo com o transdutor, que neste caso particular iria ser aproximadamente 382 milibar.

O último teste que foi possível realizar foi um processo de vazamento real, usando para isso a curva já definida na cabeça X, e verificar o resultado final.

Como mostra a Figura 110, o macho de areia foi inserido na coquilha e foram efetuados todos os procedimentos de acordo com uma produção normal: as coquilhas foram aquecidas, o bico do forno foi destapado e o metal fundido colocado no forno.



Figura 110 Inserção do macho de areia na coquilha para a realização de um teste real.

Neste último teste, excepcionalmente, a janela de visualização que apresenta o gráfico foi alterada para uma visualização mais detalhada dos valores do transdutor. Assim, a Figura 111 mostra o resultado obtido para este teste de vazamento real.

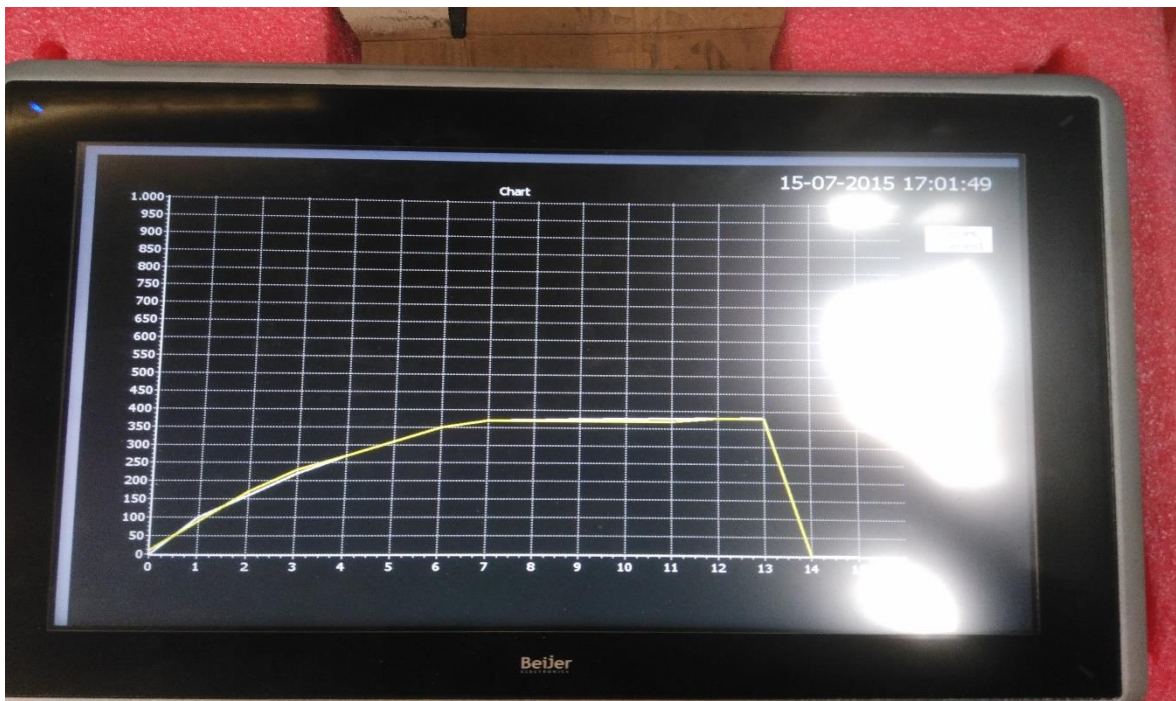


Figura 111 Resposta do sistema de controlo (teste real) com as constantes $K_c = 0,6816$ e $T_i = 0,198$.

A peça final foi produzida, como é apresentado na Figura 112, provando assim que o sistema de controlo desenvolvido funciona. No entanto, é necessário proceder à realização de mais testes, uma vez que os já realizados não são suficientes para tirar conclusões definitivas. É necessário verificar se o sistema de controlo desenvolvido irá responder sempre da mesma maneira, ao longo do tempo e após vários processos de vazamento. Um outro aspeto a verificar seria a resposta do PI mediante o consumo de metal no forno, ou seja, perceber se o controlador é capaz de compensar essa alteração, principalmente para condições extremas (quando o metal fundido no forno atinge o valor mínimo estabelecido).

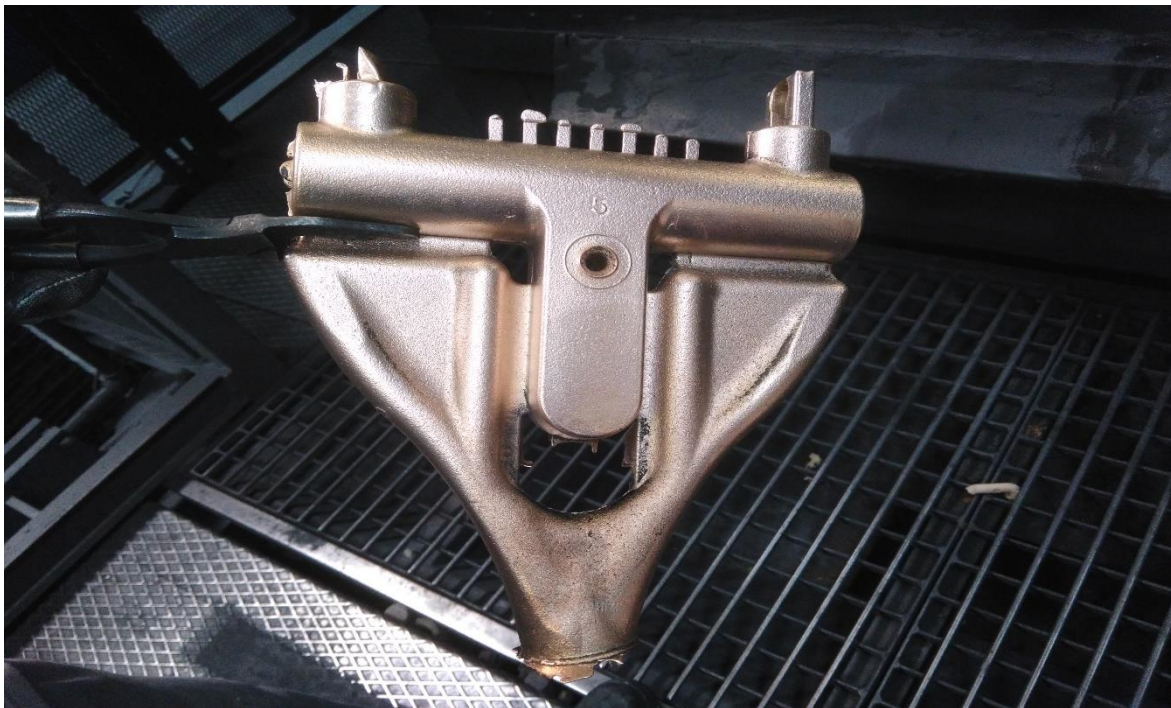


Figura 112 Peça final obtida com o novo sistema de controlo de vazamento.

Comparativamente ao controlo anteriormente efetuado, o novo sistema de controlo desenvolvido tem perspectivas para funcionar melhor visto que é realizado mediante os valores lidos pelo transdutor, e vai-se ajustando de acordo com o *setpoint* estabelecido. Conforme é demonstrado na Figura 113, o controlo efetuado pelo sistema antigo não é o mais adequado visto que a linha amarela (valores lidos pelo transdutor) não acompanha a curva pré-estabelecida: os valores de pressão são muito superiores aos definidos e no início do processo a evolução não é feita gradualmente uma vez que no início a resposta não é adequada, pois apresenta um tipo de “*overshoot*” inicial e só passado algum tempo é que

começa a comportar-se da forma esperada. De notar que a resposta obtida com o sistema de controlo antigo seguiu as mesmas condições que a resposta obtida pelo novo sistema de controlo, ou seja, mesma curva definida, mesma peça a ser produzida e forno cheio (com quantidade de metal máxima).



Figura 113 Resposta do sistema de controlo (teste real) com o sistema antigo.

8. CONCLUSÕES

O presente trabalho teve como objetivos realizar a HMI e controlar o processo de vazamento de peças de uma máquina de vazamento de metal. Para tal foi efetuado o estudo de todos os componentes envolvidos, tanto de *hardware* como de *software*.

No que diz respeito ao *hardware*, os componentes substituídos foram a CPU (S7-314) e a consola com sistema SCADA. Também tinha sido delineado a substituição da válvula; no entanto, devido a demoras e atrasos na entrega, a alteração não foi possível.

Em relação ao *software*, foram consideradas duas hipóteses para a resolução do problema: a criação de uma interface SCADA através de programação orientada a objetos, como por exemplo C#, que após a sua criação seria instalada num PC, e outra que passaria por ser o PLC a unidade de controlo “total”, ou seja, o PLC, para além de controlar a máquina, passaria também a controlar o processo de vazamento, enviando depois a informação para uma consola SCADA entretanto adquirida.

Considerou-se como sendo a melhor solução o controlo efetuado pelo PLC, pois de acordo com a experiência dos profissionais da área industrial o PLC dura mais tempo do que um computador no que toca a questões de atualizações e manutenção.

Foram encontrados diversos obstáculos na realização deste projeto. Para começar a grande diferença que existe entre o mundo académico e o mundo industrial. Cada empresa tem métodos de trabalho próprios e nem tudo o que se aprende na universidade é suficiente para conseguir entrar na indústria a 100 %. No entanto, é no meio industrial que de facto se

aprende grande parte da prática, mediante a experiência que se obtém na resolução dos problemas. A falta de conhecimentos “sobre tudo” o que envolve PLC Siemens, que não foram estudados ao longo do percurso académico, levou à necessidade de um estudo mais alargado e prolongado no tempo, mas que se revelou positivo e que se tornará uma vantagem no futuro. Foi também adquirida muita experiência pois foram realizados e/ou resolvidos vários outros pequenos projetos na Grohe, envolvendo a programação de PLC Siemens, que ajudaram muito na realização deste projeto.

Em relação ao *software* SCADA adquirido à Beijer Electronics, a experiência foi bastante positiva: verificou-se que é uma ferramenta fácil de usar e bastante intuitiva, o que possibilitou a criação da comunicação do HMI com o PLC com sucesso.

Os resultados obtidos foram bastante positivos, uma vez que toda a programação realizada para o sistema de controlo proporcionou que a curva de vazamento cumprisse as especificações pretendidas. No entanto, os testes realizados não foram suficientes, uma vez que é necessário verificar se o sistema de controlo desenvolvido irá responder sempre da mesma maneira, ao longo do tempo, e após vários processos de vazamento.

Um outro aspeto a verificar seria a resposta do PI mediante o consumo de metal no forno, ou seja, entender se o controlador é capaz de compensar essa alteração, principalmente para condições extremas (quando o metal fundido no forno atinge o valor mínimo estabelecido).

Comparativamente ao sistema de controlo existente anteriormente, o novo controlador desenvolvido tem perspectivas para funcionar melhor visto que recebe *feedback* dos valores lidos pelo transdutor, e vai-se ajustando de acordo com o *setpoint* estabelecido.

No entanto, o sistema ainda não foi implementado na máquina, devido à falta de testes em número suficiente cuja realização não foi permitida, devido a limitações de disponibilidade da máquina. O método de sintonia PID pode ser alterado para outro com mais rigor no cálculo das constantes, como por exemplo o método Kappa-Tau.

Como melhorias a implementar no futuro, será necessário proceder à realização de mais testes para obter conclusões mais sólidas.

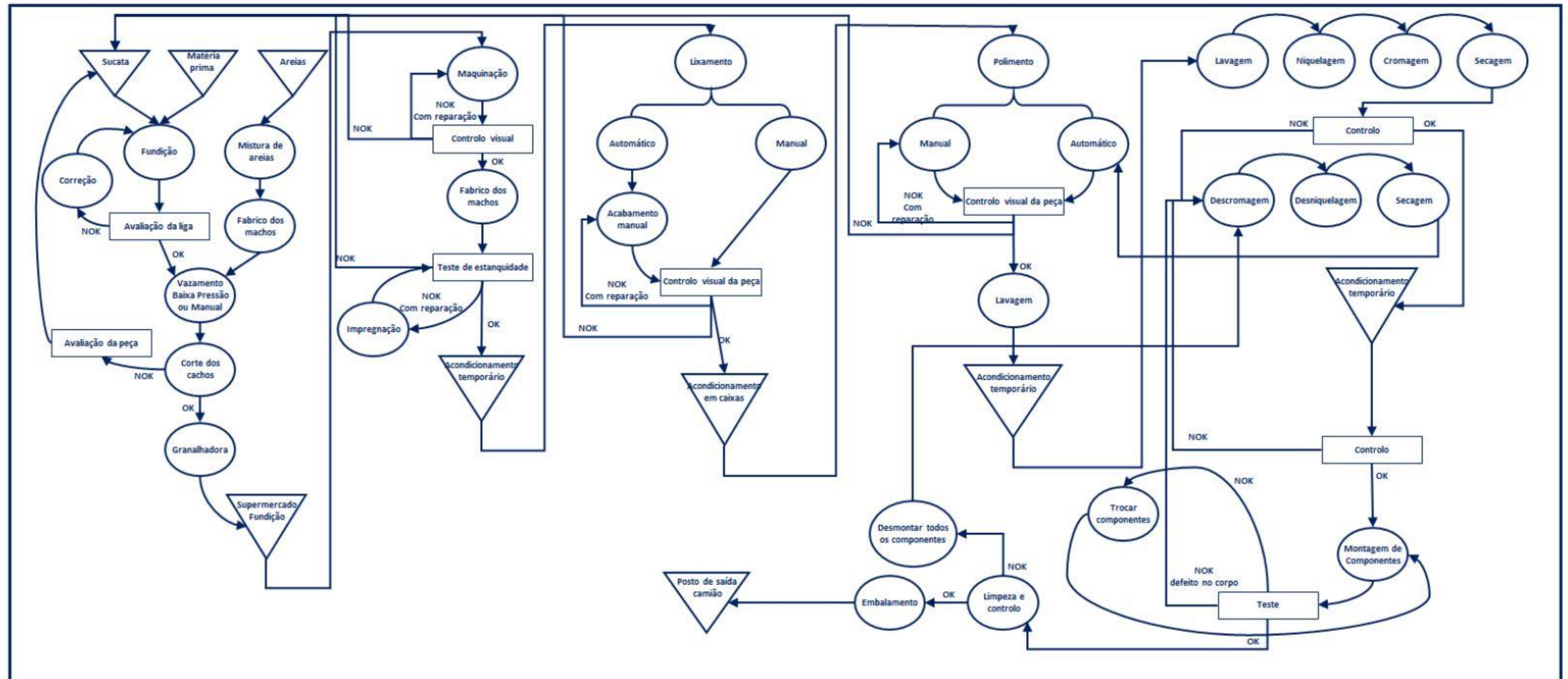
Mediante as necessidades que possam surgir, também podem ser efetuadas melhorias na programação do HMI.

Referências Documentais

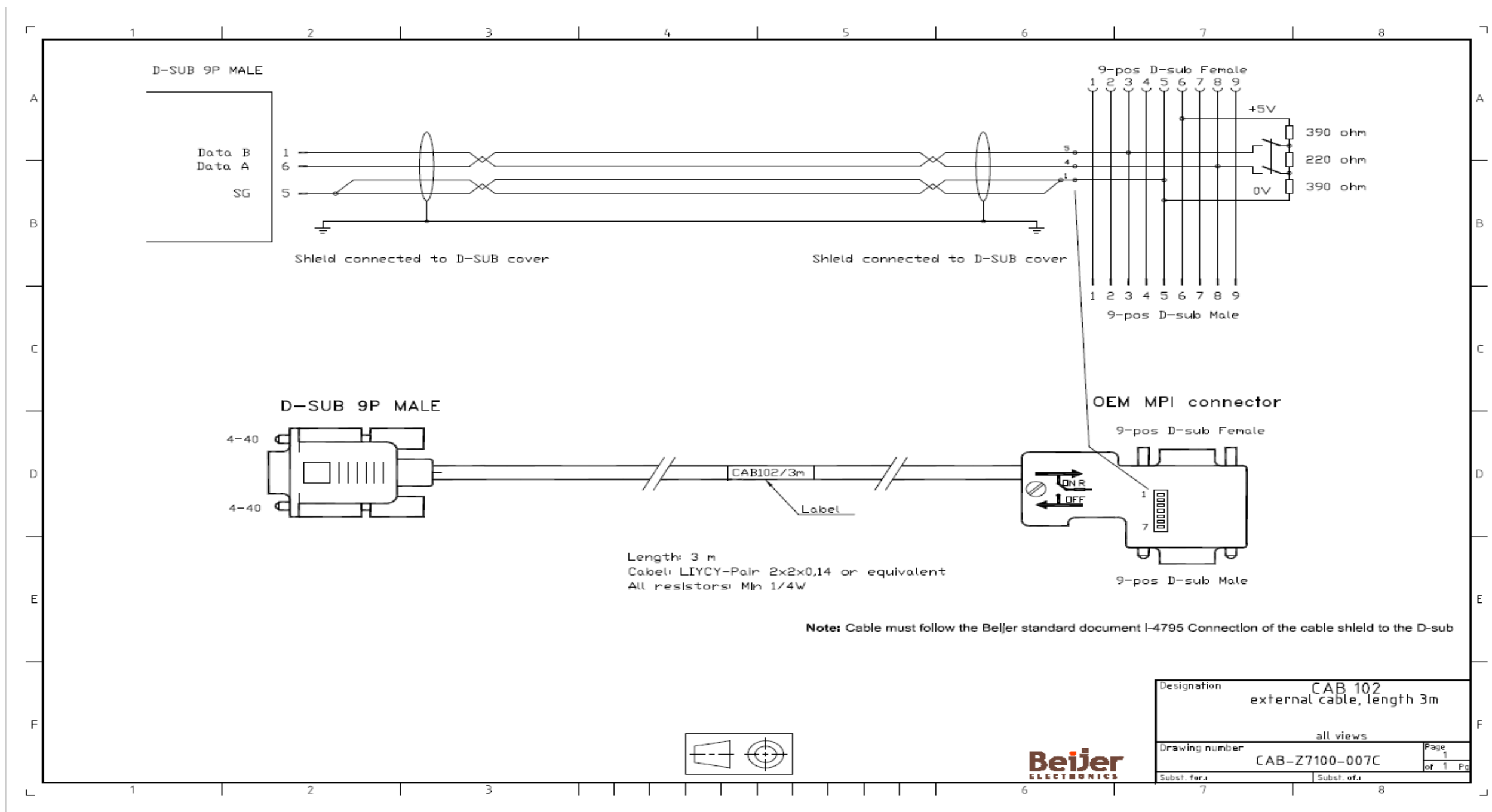
- [1] A. DANEELS , W. SALTER.—*What is SCADA ?*,. *International Conference on Accelerator and Large Experimental Physics Control Systems, 1999, Trieste, Italy.*
- [2] Acedido em 20 de Outubro de 2014, no Web site da: Caledonya Montagens Elétricas Ltda: <http://www.caledonya.com/wp/wp-content/uploads/2008/12/dsc00198.jpg>
- [3] A. THAKUR. – *SCADA Systems*. Acedido em 28 de Outubro de 2014, em: <http://www.engineersgarage.com/articles/scada-systems/>
- [4] Revista Mecatrônica Atual. - *CLP – Evolução e Tendências Ano 4, Edição 25, 2006, Brasil*. Acedido em 29 de Outubro de 2014, em: <http://www.mecatronicaatual.com.br/educacao/1389-clp-evolucao-e-tendencias?start=1>
- [5] ISA – The Instrumentation, Systems, and Automation Society (2000). *Sensors Guide*. Acedido em 30 de Outubro de 2014, em: <http://www.isa.org>.
- [6] NCS TIB 04-1 (2004). *Supervisory Control and Data Acquisition (SCADA) Systems*. National Communications System, Technical Information Bulletin 04-01, Outubro.
- [7] HONEYWELL – *PlantScape*. Acedido em 13 de Novembro de 2014, em: <http://honeywell.com/Pages/Home.aspx>
- [8] ROCKWELL. *Making Sense of e-Manufacturing: A Roadmap for UK Industry*. Acedido em 15 de Novembro de 2014, em: <http://www.rockwellautomation.co.uk>
- [9] SIEMENS. The SINUMERIK System DOC ON CD. 04/2000 Edition, electronic manual. [CD-ROM]
- [10] CASSIOLATO, C. *Redes Industriais*. Acedido em 20 de Dezembro de 2014, em: <http://www.smar.com/brasil/artigo-tecnico/redes-industriais>
- [11] ROCHA, I. R. *Redes de comunicação industriais: solução ou opção?*.
- [12] INTERBUS. *INTERBUS BASICS (Englische Version)*. Acedido em 28 de Dezembro de 2014, em: http://interbus.com/downloads_en.html
- [13] OPC Foudation. *OPC Technical Overview*. Acedido em 2 de Fevereiro de 2015, em: <http://www.opcfoudation.org>
- [14] Dr. H.K. VERMA – *Functions of Supervisory Control & Data Acquisition (Scada) System*. Department of Electrical and Electronics Engineering School of Engineering and Technology SHARDA UNIVERSITY Greater Noida, India.
- [15] MARTEM – Telecontrol Systems, *TELEM SCADA SYSTEM*. Acedido em 13 de Fevereiro de 2015, em: <http://www.martem.ee/telem-scada-system/>
- [16] Acedido a 3 de Março de 2015, no Web site da: Universidade Federal do Paraná, Engenharia Industrial Madeireira – *Controladores Lógicos Programáveis – CLP*: <http://www.madeira.ufpr.br/disciplinasivan/AULACLP.pdf>
- [17] SILVA FILHO (2002), Isair Silva Sousa. *Controladores Lógico Programável*. Acedido em 10 de Março de 2015, em: <http://www.ebah.pt/content/ABAAAQnEAB/controlador-logico-programavel#>
- [18] ENG° JORDÃO SILVA DA ROCHA (2013). *Como funciona o CLP ?*. Acedido em 15 de Março de 2015, em: <http://saladaautomacao.com.br/como-funciona-o-clp/>

- [19] W. Bolton, *Programmable Logic Controllers*, Fourth Edition, Newnes, 2006 ISBN 978-0-7506-8112-4, Chapter 1.
- [20] Acedido em 15 de Março de 2015, no *Web site* da: AliExpress:
<http://www.aliexpress.com/item/USB-MPI-PC-Adapter-USB-for-Siemens-S7-200-300-400PLC-MPI-DP-PPI-Programming-64bit/32254625908.html>
- [21] ROMANHOLO (2014, 21 de Abril). *TIA Portal hands on – básico v11-v2*. Acedido em 10 de Maio de 2015, em: <http://pt.slideshare.net/marcosromanholo7/01-tia-portal-hands-on-basico-v11v2?related=1>
- [22] Ferramenta help do TIA Portal v13
- [23] SMUTS J., *Process Control for Practitioners*. Acedido em 24 de Maio de 2015, em: <http://blog.opticontrols.com/archives/383>

Anexo A. Fluxo Produtivo da Grohe-Portugal



Anexo B. *Datasheet* do cabo de comunicação entre a consola HMI (Beijer) e o PLC Siemens e características da consola iX T15B

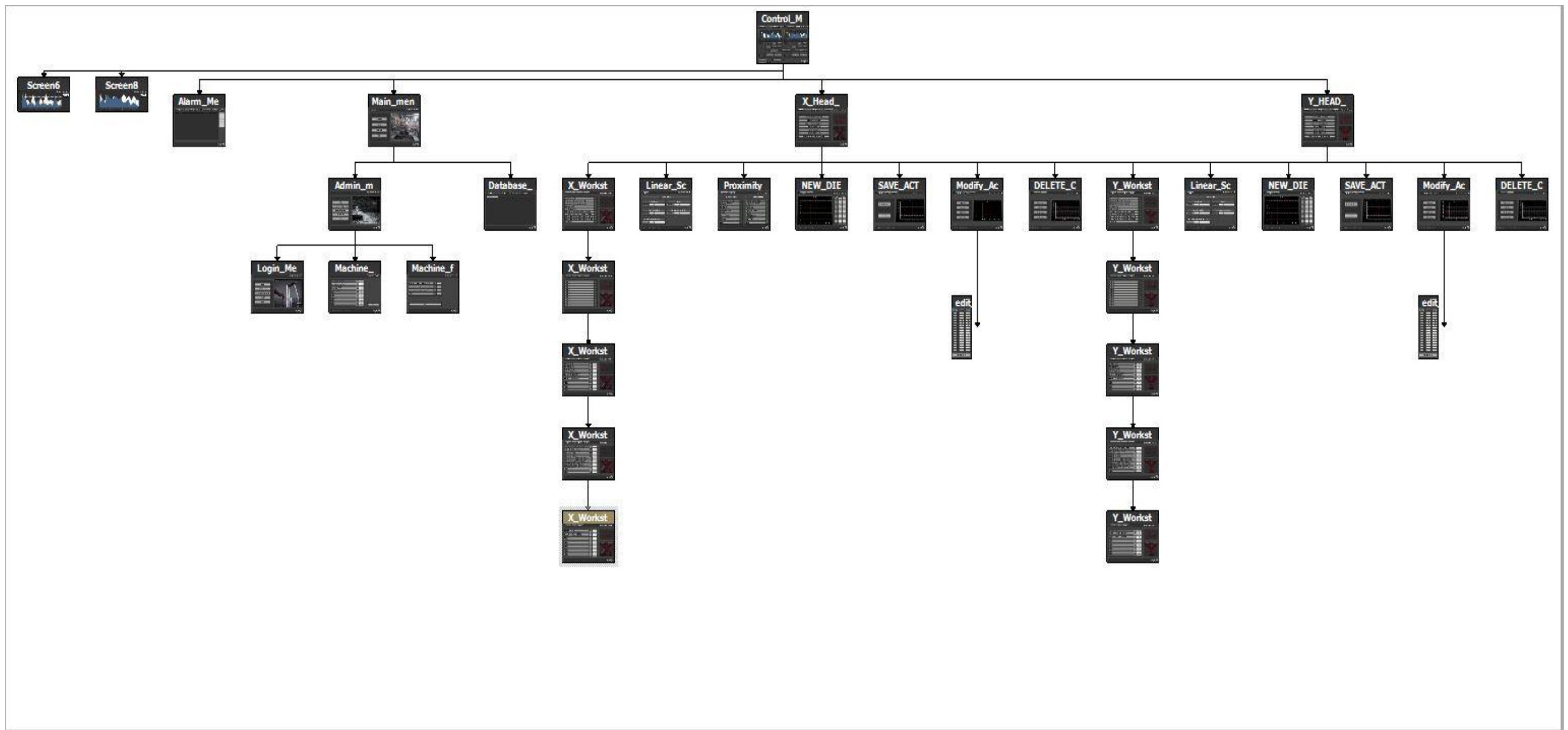


iX T15B

General Description		Serial Communication	
Article number	64000302	Com 1	RS232 (RTS/CTS)
Part status	Active	Com 2	RS422/ RS485
HMI development software	iX Developer	Com 3	RS232
System Info		Com 4	RS422 (RTS)/ RS485
RAM	1 GB	Ethernet	
CPU	Intel® Atom	Ethernet 1	1 x 10/100 Mbit/s (shielded RJ 45)
Application	1.4 GB	Ethernet 2	1 x 10/100/1000 Mbit/s (shielded RJ 45)
Display		Expansion	
Type	TFT-LCD	USB	3x USB Host 2.0, max output current 500mA
Size	15" 1280 X 800 pixels	Memory expansion	1xSD card
Backlight	LED backlight	iX Field buss expansion	yes
Backlight	50000 h	Certifications	
Brightness	450 cd/m2	UL	UL508
Interaction		DNV	N/A
External	USB	CE	Noise tested according to EN61000-6-4 emission and EN61000-6-2 immunity.
Front LED	Multi-color software programmable		Environmental
Touch screen type	Resistive touch	Sealing - back panel	IP 20
Power		Humidity	5% - 85% non-condensed
Power consumption at rated voltage	24 W	Storage temperature	-20° to +70°C
Voltage	+24 V DC (18-32 V DC)	Operating temperature	-10° to +50°C
Fuse	Internal DC fuse, 3.15A slow	Sealing - front panel	IP 65
Mechanical			
Size	410 x 286 x 83mm		
Weight	3,6 Kg		
Housing material	Powder-coated aluminum		



Anexo C. Árvore de menus da nova consola Beijer



Anexo D. Endereçamento indireto [22]

ANY

Description

An ANY type parameter points to the start of a data area and specifies its length. An ANY pointer occupies 10 bytes of memory and may contain the following information:

- Data type:

Data type of the elements of the data area

- Repetition factor:

Number of elements of the data area

- DB number:

Data block that contains the declaration of data area elements.

- Memory area:

Memory area of the CPU that stores the data area elements.

- Start address of the data in the format "byte.bit":

Data area start identified by the ANY pointer.

- Zero pointer:

Use the zero pointer to indicate a missing value. A missing value may indicate that no value exists, or that the value is not yet known. A zero value represents the absence of a value, but is also a value.

Note

Memory area

For an S7-1500 CPU, the ANY pointer can also only point to memory areas with "Standard" access mode.

The following figure shows the structure of the ANY pointer:

An ANY pointer cannot identify structures. It can only be assigned to local tags.

Tabela 13 Formats for the declaration of an ANY pointer.

Representation	Format	Example of value input	Description
Symbolic	P#DataBlock.Memory Area DataAddress Type Number	"MyDB".StructTag.InitialComponents	Area with 10 words in global DB11 starting with DBB20.0
	P#MemoryArea DataAddress Type Number	"MyMarkerTag"	Area with 4 bytes starting with MB 20.0
		"MyTag"	Input I1.0
	P#Zero value	-	Zero value
Absolute	P#DataBlock.Memory Area DataAddress Type Number	P#DB11.DBX20.0 INT 10	Area with 10 words in global DB11 starting with DBB20.0
	P#MemoryArea DataAddress Type Number	P#M20.0 BYTE 10	Area with 10 bytes starting with MB 20.0
		P#E1.0 BOOL 1	Input I1.0
	P#Zero value	P#0.0 VOID 0, ZERO	Zero value

Coding of data types

Tabela 14 Coding of data types for the ANY pointer.

Hexadecimal code	Data type	Description
B#16#00	NIL	Null pointer
B#16#01 ¹⁾	BOOL	Bits
B#16#02	BYTE	bytes, 8 bits
B#16#03	CHAR	8-bit characters
B#16#04	WORD	16-bit words
B#16#05	INT	16-bit integers
B#16#06	DWORD	32-bit words
B#16#07	DINT	32-bit integers
B#16#08	REAL	32-bit floating-point numbers
B#16#0B	TIME	Time duration
B#16#0C	S5TIME	Time duration
B#16#09	DATE	Date
B#16#0A	TOD	Date and time
B#16#0E	DT	Date and time
B#16#13	STRING	Character string
B#16#17 ¹⁾	BLOCK_FB	Function block

B#16#18 ¹⁾	BLOCK_FC	Function
B#16#19 ¹⁾	BLOCK_DB	Data block
B#16#1A ¹⁾	BLOCK_SDB	System data block
B#16#1C ¹⁾	COUNTER	Counter
B#16#1D ¹⁾	TIMER	Timer

¹⁾ These data types can only be used for the ANY pointer on a CPU S7-300/400.

Coding of the memory area

Tabela 15 Coding of the memory areas for the ANY pointer.

Hexadecimal code	Area	Description
B#16#80 ¹⁾	P	I/O
B#16#81	I	Memory area of inputs
B#16#82	Q	Memory area of outputs
B#16#83	M	Memory area of bit memory
B#16#84	DBX	Data block
B#16#85	DIX	Instance data block
B#16#86	L	Local data
B#16#87	V	Previous local data

¹⁾ These memory areas can only be used for the ANY pointer on an S7-300/400 CPU.



ANY

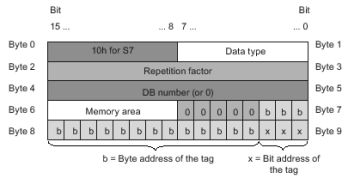
- Start address of the data in the format "byte.bit":
Data area start identified by the ANY pointer.
- Zero pointer:
Use the zero pointer to indicate a missing value. A missing value may indicate that no value exists, or that the value is not yet known. A zero value represents the absence of a value, but is also a value.

Note

Memory area

For an S7-1500 CPU, the ANY pointer can also only point to memory areas with "Standard" access mode.

The following figure shows the structure of the ANY pointer:



An ANY pointer cannot identify structures. It can only be assigned to local tags.

The following table shows the formats for the declaration of an ANY pointer:

Figura 114 **Ponteiro Any.**

Anexo E. Função bloco CONT_C (PID) [22]

Description CONT_C

The CONT_C instruction is used on SIMATIC S7 automation systems to control technical processes with continuous input and output variables. You can assign parameters to enable or disable sub-functions of the PID controller and adapt it to the process. In addition to the functions in the setpoint and process value branches, the instruction implements a complete PID controller with continuous output value output and the option of manually influencing the value of the output value.

Application

You can use the controller as a PID fixed setpoint controller, or in multi-loop control systems, also as a cascade, blending or ratio controller. The functions of the controller are based on the PID control algorithm of the sampling controller with an analog signal, if necessary extended by including a pulse shaper stage to generate pulse-width modulated output signals for two or three step controllers with proportional actuators.

Call

The CONT_C instruction has an initialization routine that is run through when input parameter COM_RST = TRUE is set. During initialization, the integral action is set to the initialization value I_ITVAL. All the signal outputs are set to zero. COM_RST = FALSE has to be set after the initialization routine has been completed.

The calculation of the values in the control blocks is only correct if the block is called at regular intervals. You should therefore call the control blocks in a cyclic interrupt OB (OB 30 to OB 38). Enter the sampling time in the CYCLE parameter.

If you call the instruction CONT_C as a multiple instance DB, no technology object is created. No parameter assignment interface or commissioning interface is available. You must assign parameters for CONT_C directly in the multiple instance DB and commission it via a watch table.

Error information

The error message word RET_VAL is not evaluated by the block.

How CONT C works

Setpoint branch

The setpoint is entered in floating-point format at the SP_INT input.

Process value branch

The process value can be input in I/O or floating-point format. The function CRP_IN converts the I/O value PV_PER to a floating-point format -100 to +100 % in accordance with the following rule:

$$\text{Output of CRP_IN} = \text{PV_PER} * 100 / 27648$$

The PV_NORM function scales the output of CRP_IN according to the following rule:

$$\text{Output of PV_NORM} = (\text{output of CRP_IN}) * \text{PV_FAC} + \text{PV_OFF}$$

PV_FAC has a default of 1 and PV_OFF a default of 0.

Forming the error signal

The difference between the setpoint and process value is the error signal. To suppress a minor sustained oscillation due to manipulated variable quantization (e.g. with a pulse width modulation with PULSEGEN), the error signal is applied to a dead band (DEADBAND). With DEADB_W = 0, the dead band is switched off.

PID Algorithm

The PID algorithm operates as a position algorithm. The proportional, integral (INT), and differential (DIF) actions are connected in parallel and can be activated or deactivated individually. This allows P, PI, PD, and PID controllers to be configured. Pure I controllers are also possible.

Manual value processing

It is possible to switch over between manual and automatic mode. In manual mode, the manipulated variable is corrected to a manually selected value.

The integral action (INT) is set internally to $LMN - LMN_P - DISV$ and the derivative action (DIF) is set to 0 and synchronized internally. Changeover to automatic mode is therefore bumpless.

Manipulated value processing

You can use the LMNLIMIT function to limit the manipulated value to selected values. Alarm bits indicate when a limit is exceeded by the input variable.

The LMN_NORM function normalizes the output of LMNLIMIT according to the following rule:

$$LMN = (\text{output of LMNLIMIT}) * LMN_FAC + LMN_OFF$$

LMN_FAC has a default of 1 and LMN_OFF a default of 0.

The manipulated value is also available in I/O format. The CRP_OUT function converts the LMN floating-point value to a I/O value according to the following rule:

$$LMN_PER = LMN * 27648 / 100$$

Feedforward control

A disturbance variable can be added at the DISV input.

CONT_C block diagram

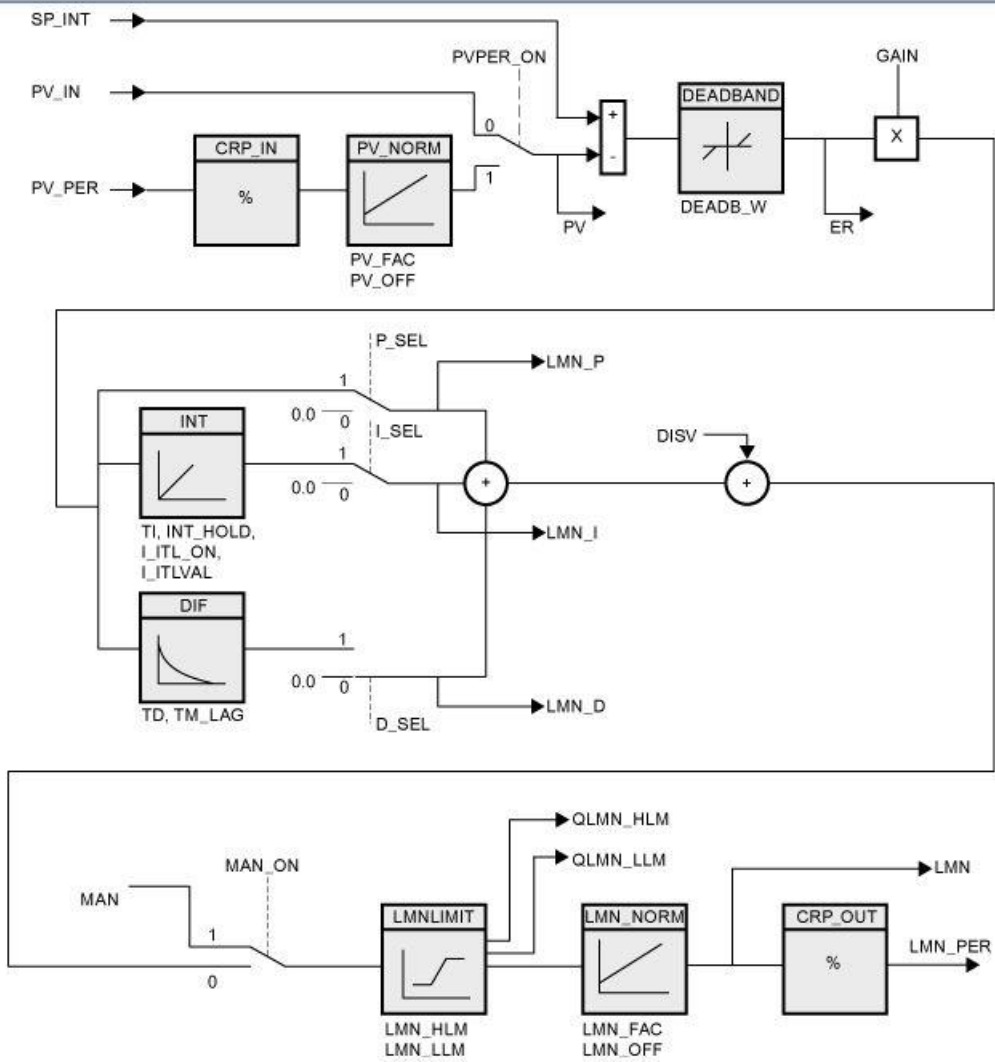


Figura 115 Diagrama de blocos do CONT_C.

Tabela 16 Input parameter CONT_C.

Parameters	Data type	Default	Description
COM_RST	BOOL	FALSE	The instruction has an initialization routine that is processed when the "Restart" input is set.
MAN_ON	BOOL	TRUE	If the input "Enable manual mode" is set then the control loop is interrupted. A manual value is set as the manipulated value.
PVPER_ON	BOOL	FALSE	If the process value is to be read in from the I/Os, the PV_PER input must be interconnected with the I/Os and the "Enable process value I/Os" input must be set.
P_SEL	BOOL	TRUE	The PID actions can be switched on and off individually in the PID algorithm. P-action is on when the "Enable P-action" input is set.
I_SEL	BOOL	TRUE	The PID actions can be switched on and off individually in the PID algorithm. I action is on when the input "I-action on" is set.
INT_HOLD	BOOL	FALSE	The output of the integral action can be frozen. For this the input "I-action hold" must be set.
I_ITL_ON	BOOL	FALSE	The output of the integral action can be set at the I_ITLVAL input. For this the input "Set I-action" must be set.
D_SEL	BOOL	FALSE	The PID actions can be switched on and off individually in the PID algorithm. D-action is on when the input "Enable D-action" is set.
CYCLE	TIME	T#1s	The time between block calls must be constant. The "Sampling time" input specifies the time between block calls. CYCLE >= 1ms
SP_INT	REAL	0.0	The input "Internal setpoint" is used to specify a setpoint. Permissible are values from -100 to 100 % or a physical variable 1).
PV_IN	REAL	0.0	At the "Process value input" you can assign parameters to a commissioning value or you can interconnect an external process value in floating-point format. Permissible are values from -100 to 100 % or a physical variable 1).
PV_PER	WORD	W#16#0000	The process value in I/O format is interconnected with the controller at the "Process value I/O" input.

MAN	REAL	0.0	The "Manual value" input is used to set a manual value using the operator interface functions. Permissible are values from -100 to 100 % or a physical variable 2).
GAIN	REAL	2.0	The "Proportional gain" input specifies controller amplification.
TI	TIME	T#20s	The "Integration time" input determines the time response of the integral action. TI >= CYCLE
TD	TIME	T#10s	The "Derivative action time" input determines the time response of the derivative action. TD >= CYCLE
TM_LAG	TIME	T#2s	Time lag of the D-action The algorithm of the D-action contains a delay for which parameters can be assigned at the input "Time lag of the D-action". TM_LAG >= CYCLE/2
DEADB_W	REAL	0.0	A dead band is applied to the system deviation. The "Dead band width" input determines the size of the dead band. DEADB_W >= 0.0 (%) or a physical variable 1)
LMN_HLM	REAL	100.0	The manipulated value is always restricted to a high limit and low limit. The "High limit of manipulated value" input specifies the high limit. Permissible are real values starting at LMN_LLM or a physical variable 2).
LMN_LLM	REAL	0.0	The manipulated value is always restricted to a high limit and low limit. The "Low limit of manipulated value" input specifies the low limit. Permissible are real values up to LMN_HLM or a physical variable 2).
PV_FAC	REAL	1.0	The "Process value factor" input is multiplied by the process value. The input is used to scale the process value range.
PV_OFF	REAL	0.0	The input "Process value offset" is added to the process value. The input is used to scale the process value range.
LMN_FAC	REAL	1.0	The "Manipulated value factor" input is multiplied with the manipulated value. The input is used to scale the manipulated value range.

LMN_OFF	REAL	0.0	The input "Manipulated value offset" is added to the process value. The input is used to scale the manipulated value range.
I_ITLVAL	REAL	0.0	The output of the integral action can be set at the I_ITL_ON input. The initialization value is applied to the input "Initialization value of the I-action." Permissible are values of -100.0 to 100.0 (%) or a physical variable 2).
DISV	REAL	0.0	For feedforward control, the disturbance variable is interconnected to the "Disturbance variable" input. Permissible are values of -100.0 to 100.0 (%) or a physical variable 2).

1) Parameters in the setpoint and process value branches with the same unit

2) Parameters in the manipulated value branch with the same unit

Tabela 17 Output parameters CONT_C.

Parameter	Data type	Default	Description
LMN	REAL	0.0	The effective "Manipulated value" is output in floating point format at the "Manipulated value" output.
LMN_PER	WORD	W#16#0000	The manipulated value in I/O format is interconnected on the input "Manipulated value I/O" with the controller.
QLMN_HLM	BOOL	FALSE	The manipulated value is always restricted to a high limit and low limit. The output "High limit of manipulated value reached" indicates that the high limit has been reached.
QLMN_LLM	BOOL	FALSE	The manipulated value is always restricted to a high limit and low limit. The output "Low limit of manipulated value reached" indicates that the low limit has been reached.
LMN_P	REAL	0.0	The "P-action" output contains the proportional action of the manipulated variable.
LMN_I	REAL	0.0	The "I-action" output contains the integral action of the manipulated variable.
LMN_D	REAL	0.0	The "D-action" output contains the derivative action of the manipulated variable.
PV	REAL	0.0	The effective process value is output at the "Process value" output.
ER	REAL	0.0	The effective system deviation is output at the "Error signal" output.

Anexo F. Cohen-Coon Tuning Rules [23]

Based on the number of Google searches in 2010, the Cohen-Coon tuning rules are second in popularity only to the Ziegler-Nichols tuning rules. Cohen and Coon published their tuning method in 1953, eleven years after Ziegler and Nichols published theirs.

More Flexible than Ziegler-Nichols

The Cohen-Coon tuning rules are suited to a wider variety of processes than the Ziegler-Nichols tuning rules. The Ziegler-Nichols rules work well only on processes where the dead time is less than half the length of the time constant.

The Cohen-Coon tuning rules work well on processes where the dead time is less than two times the length of the time constant (and you can stretch this even further if required).

Cohen-Coon provides one of the few sets of tuning rules that has rules for PD controllers – should you ever need this.

Quarter-Amplitude Damping

Like the Ziegler-Nichols tuning rules, the Cohen-Coon rules aim for a quarter-amplitude damping response. Although quarter-amplitude damping-type of tuning provides very fast disturbance rejection, it tends to be very oscillatory and frequently interacts with similarly-tuned loops. Quarter-amplitude damping-type tuning also leaves the loop vulnerable to going unstable if the process gain or dead time doubles in value. However, the easy fix for both problems is to reduce the controller gain by half. E.g. if the rule recommends using a controller gain of 1.8, use only 0.9. This will prevent the loop from oscillating around its set point as described above, and will provide an acceptable stability margin.

Target PID Controller Algorithm

There are three types of PID controller algorithms: Interactive, Non interactive, and Parallel. The Cohen-Coon tuning rules were designed for controllers with the no interactive controller algorithm. If you are not using the derivative control mode (i.e. using P, PI, or PD control), the rules will also work for the interactive algorithm. However, if you are using derivative

(i.e. PID control) on an interactive controller, or if your controller has a parallel algorithm, you should convert the calculated tuning settings to work on your controller.

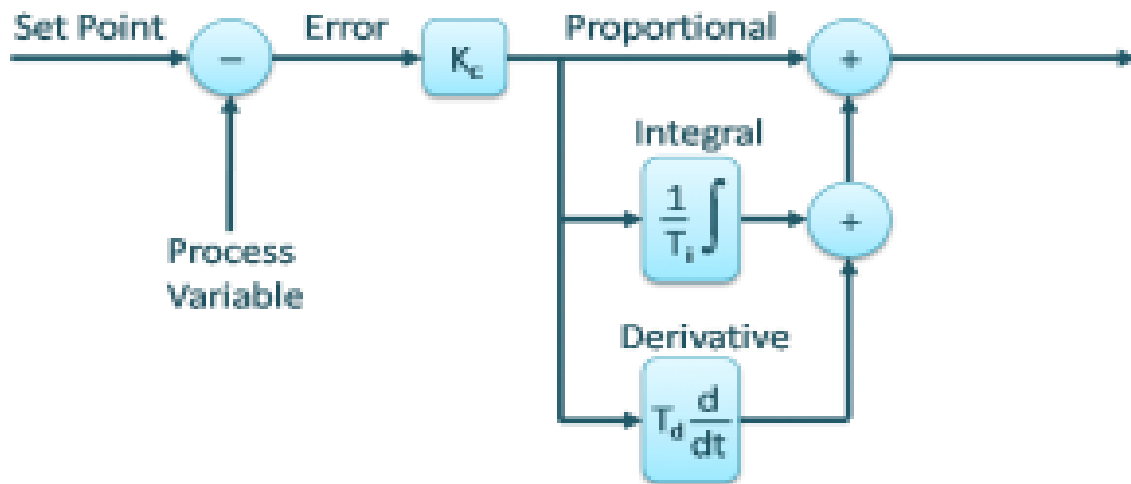


Figura 116 Esquema do Controle PID.

The original Cohen-Coon paper expressed the tuning constant for the integral control mode in terms of reset rate (or integral gain) in repeats per minute. Virtually all the modern texts on process control use integral time, and so do most control systems (DCS & PLC). Also, this blog generically uses integral time and not integral gain. Therefore, the tuning rules below use integral time (the reciprocal of what Cohen-Coon used). If your controller uses integral gain or reset rate, you'll have to invert the calculated integral time (use $1/T_i$).

Also, if your controller's integral time unit is in minutes, you must make your measurements of dead time and time constant in minutes. Likewise if your controller uses seconds, make your measurements in seconds.

When to use the Cohen-Coon Tuning Rules

The Cohen-Coon tuning rules are suitable for use on self-regulating processes if the control objective is having a fast response, but I recommend you divide the calculated controller gain by two, as described above.

If the control objective is to have a very stable, robust control loop that absorbs disturbances, rather use the Lambda tuning rules.

Tuning Procedure

Assuming the control loop is linear and the final control element is in good working order, you can continue with tuning the controller. The Cohen-Coon tuning rules use three process characteristics: process gain, dead time, and time constant. These are determined by doing a step test and analyzing the results.

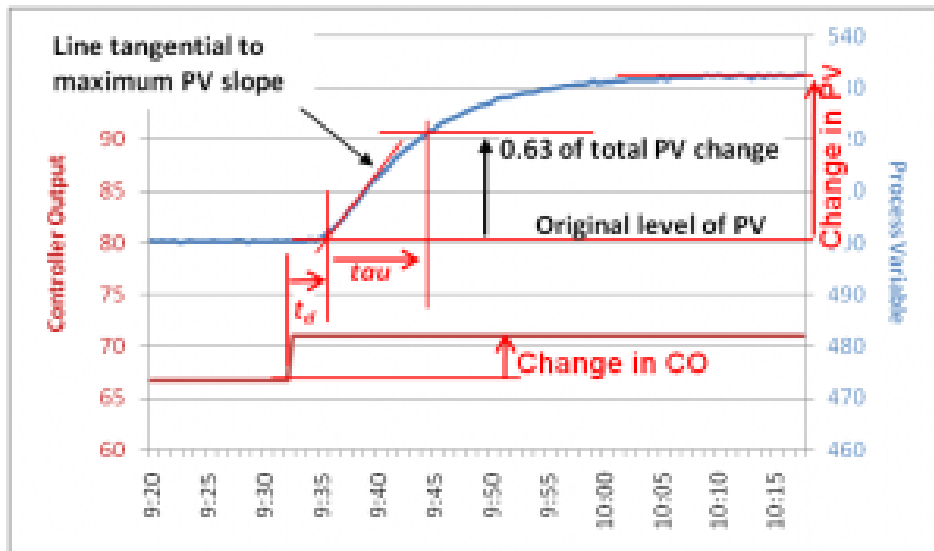


Figura 117 Aquisição de valores através do gráfico.

1. Place the controller in manual and wait for the process to settle out.
2. Make a step change of a few percent in the controller output (CO) and wait for the process variable (PV) to settle out at a new value. The size of this step should be large enough that the process variable moves well clear of the process noise/disturbance level. A total movement of five times the noise/disturbances on the process variable should be sufficient.
3. Convert the total change obtained in PV to a percentage of the span of the measuring device.
4. Calculate the process gain (gp) as follows:

$$gp = \text{change in PV [in \%]} / \text{change in CO [in \%]}$$

5. Find the maximum slope on the PV response curve. This will be at the inflection point (where the PV stops curving upward and begins curving downward). Draw a line tangential to the PV response curve through the point of inflection. Extend this

line to intersect with the original level of the PV (before the step change in CO). Take note of the time value at this intersection.

6. Measure the dead time (t_d) as follows:

t_d = time difference between the change in CO and the intersection of the tangential line and the original PV level.

7. Calculate the value of the PV at 63% of its total change. On the PV reaction curve, find the time value at which the PV reaches this level.

8. Measure the time constant (Greek symbol tau) as follows:

τ = time difference between intersection at the end of dead time, and the PV reaching 63% of its total change.

9. Convert your measurements of dead time and time constant to the same time-units your controller's integral mode uses. E.g. if your controller's integral time is in minutes, use minutes for this measurement.

10. Do two or three more step tests and calculate process gain, dead time, and time constant for each test to obtain a good average of the process characteristics. If you get vastly different numbers every time, do even more step tests until you have a few step tests that produce similar values. Use the average of those values.

11. Calculate new tuning settings using the Cohen-Coon tuning rules below. Note that these rules produce a quarter-amplitude damping response. See the next step.

www.gaticontrols.com	Controller Gain	Integral Time	Derivative Time
P Controller:	$K_c = \frac{1.03}{K_p} \left(\frac{\tau}{t_d} + 0.34 \right)$		
PI Controller:	$K_c = \frac{0.9}{K_p} \left(\frac{\tau}{t_d} + 0.092 \right)$	$T_i = 3.33 t_d \frac{\tau + 0.092 t_d}{\tau + 2.22 t_d}$	
PD Controller:	$K_c = \frac{1.24}{K_p} \left(\frac{\tau}{t_d} + 0.129 \right)$		$T_D = 0.27 t_d \frac{\tau - 0.324 t_d}{\tau + 0.129 t_d}$
PID Controller: (Noninteracting)	$K_c = \frac{1.35}{K_p} \left(\frac{\tau}{t_d} + 0.185 \right)$	$T_i = 2.5 t_d \frac{\tau + 0.185 t_d}{\tau + 0.611 t_d}$	$T_D = 0.37 t_d \frac{\tau}{\tau + 0.185 t_d}$

Figura 118 Regras Cohen-Coon.

12. Divide the calculated controller gain by two to reduce oscillations and improve loop stability.
13. Compare the newly calculated controller settings with the ones in the controller, and ensure that any large differences in numbers are expected and justifiable.
14. Make note of the previous controller settings, the new settings, and the date and time of change.
15. Implement and test the new controller settings. Ensure the response is in line with the overall control objective of the loop.
16. Leave the previous controller settings with the operator in case he/she wants to revert back to them and cannot find you to do it. If the new settings don't work, you have probably missed something in one or more of the previous steps.
17. Monitor the controller's performance periodically for a few days after tuning to verify improved operation under different process conditions.

Anexo G. Diferenças entre HMI antigo e novo HMI Beijer

A máquina tem uma consola implementada que está ligada a um PC, e ambos controlam o processo de vazamento. A consola serve para monitorizar as curvas de vazamento de ambas as cabeças e também para controlar certas variáveis, visto que é necessário introduzir os valores nas variáveis de configuração da máquina, forno e ambas as cabeças.

Neste anexo são comparados os menus mais relevantes da consola (PC), com o novo sistema desenvolvido.

Uma das evoluções do novo sistema diz respeito à interação do utilizador com a consola, visto que o sistema anterior era controlado através de um rato, e o novo utiliza o *touch-screen*.

MAIN MENU

A consola passa a ter um menu principal como o apresentado em baixo, com botões que apresentam os ecrãs que o utilizador desejar, conforme as suas necessidades, contrariamente à consola anterior, em que a interface principal era logo o menu de controlo.

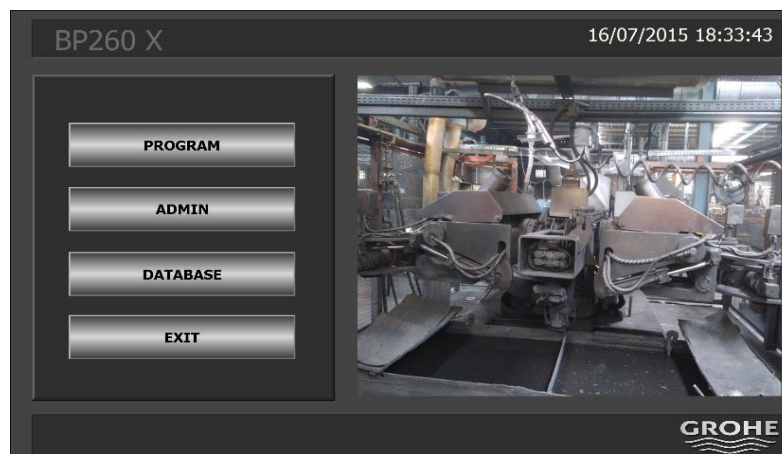


Figura 119 Main menu.

MENU PROGRAM

Através do menu de programa, Figura 120, é possível aceder a todos os menus, tendo sido introduzidos botões na parte superior capazes de levar o utilizador a todos os restantes

menus. Relativamente ao menu da consola antiga não são verificadas grandes diferenças entre ambos, sendo a única diferença, os botões enquadrados na nova consola com indicações específicas para onde levam.

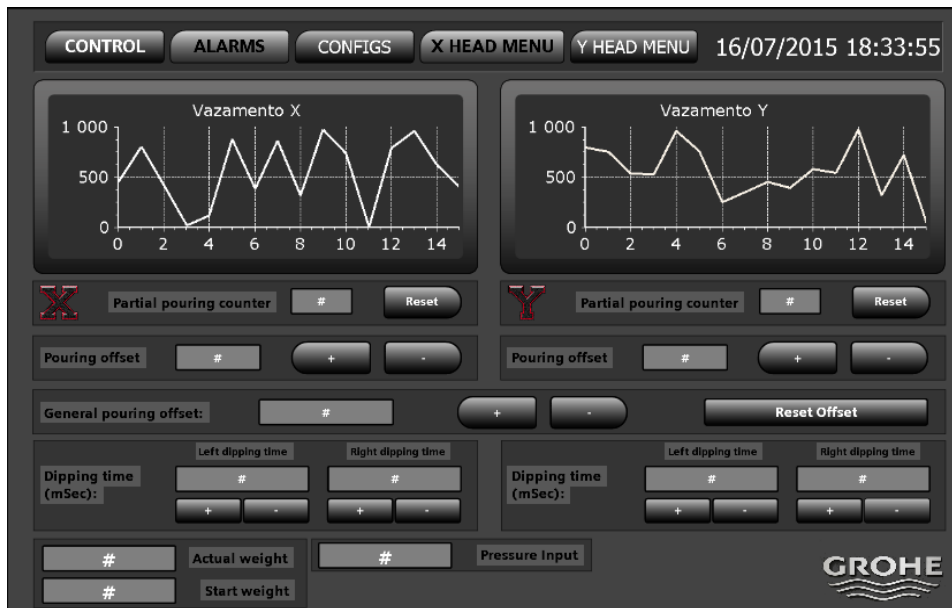


Figura 120 Menu do programa (controlo).



Figura 121 Menu do programa (HMI antigo).

MENUS DE CONFIGURAÇÃO

Anteriormente os menus de configuração apareciam diante do utilizador numa janela *popup*, para o novo sistema, esse menu passou a preencher o ecrã total. Outra grande diferença diz respeito ao facto de os parâmetros encontrarem-se dispersos ao longo do menu, ou seja, pelo meio dos vários parâmetros iam sendo encontradas variáveis “available” como é mostrado

na Figura 122. Na nova interface tudo foi organizado deixando os endereços “available” na mesma, mas no final e apenas visíveis com direitos de administrador (Figura 123).

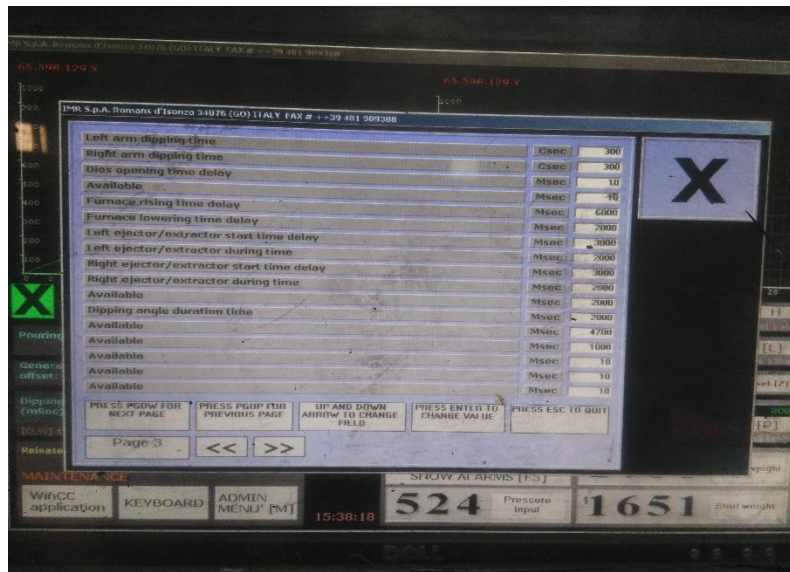


Figura 122 Menu configuração das cabeças (HMI antigo).

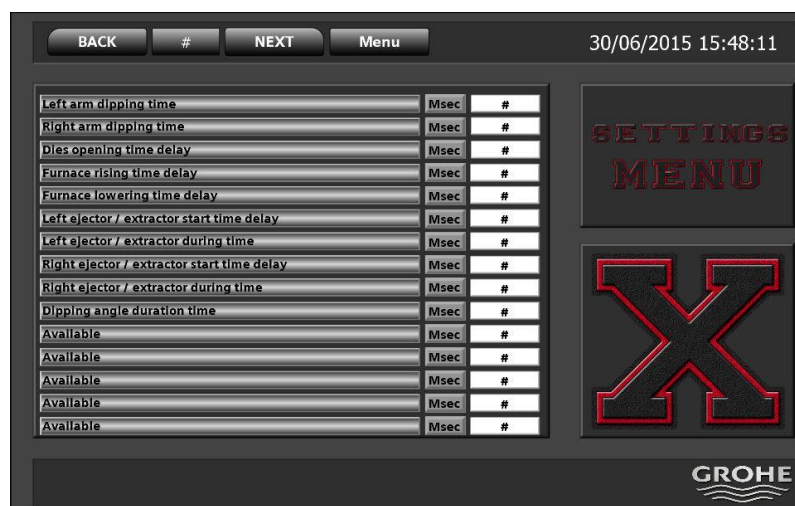


Figura 123 Menu configuração das cabeças (HMI novo).

MENU CRIAÇÃO DE CURVAS

A maior parte das diferenças encontram-se nos menus de configuração das curvas, onde no HMI antigo foram encontradas algumas limitações que foram corrigidas na nova interface.

Relativamente ao menu de criação das curvas, o modo de funcionamento é igual pois são introduzidos os valores de tempo e pressão, e é mostrado o gráfico criado com a inserção desses valores (Figura 124 e Figura 125).

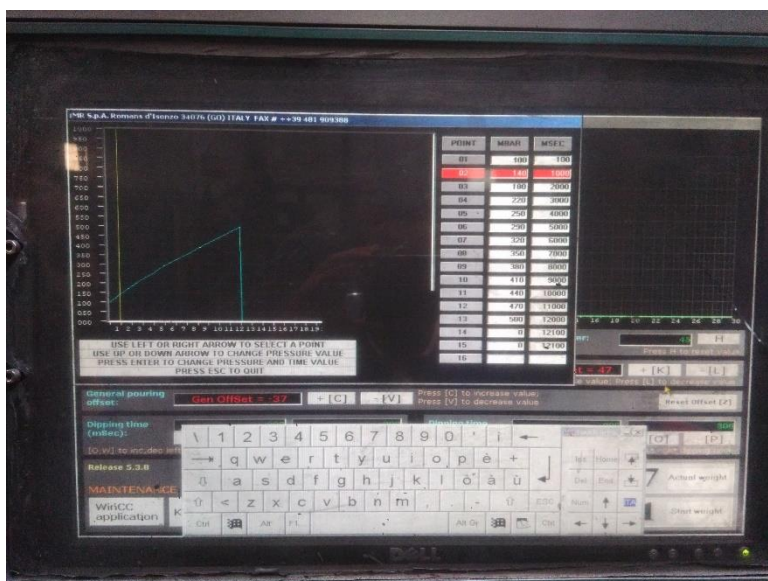


Figura 124 Criação de uma nova curva de vazamento.

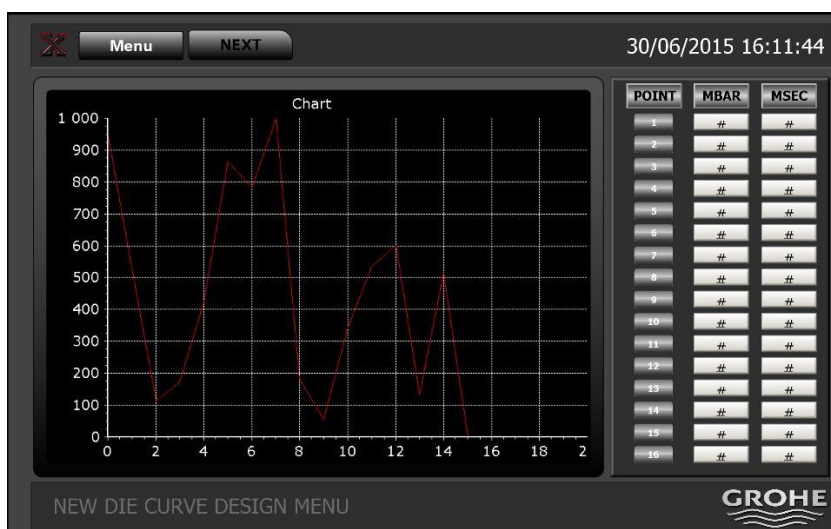


Figura 125 Criação de uma nova curva de vazamento.

No entanto na consola antiga este menu é único para ambas as cabeças, que podem ter curvas diferentes, e na consola nova, passam a existir dois menus para a criação das curvas, ou seja, foram criados dois menus, um para a curva da cabeça X e outro para a cabeça Y, onde os dados introduzidos são guardados em *recipes* diferentes.

Após a criação da curva na consola antiga, surge um menu com várias possibilidades para carregar a curva para a cabeça X, carregar para a cabeça Y e/ou apagar a curva (Figura 126).



Figura 126 Menu para guardar, carregar e/ ou apagar as curvas.

Apesar de no menu da consola antiga existir 3 botões com nomes diferentes (Figura 127), o “SAVE ACTUAL CURVE AS”, “MODIFY ACTUAL CURVE” e o “LOAD AND ERASE CURVES FROM ARCHIVE”, todos eles estão associados a um único menu, o da Figura 126, ou seja, acaba por não fazer sentido esta distinção.

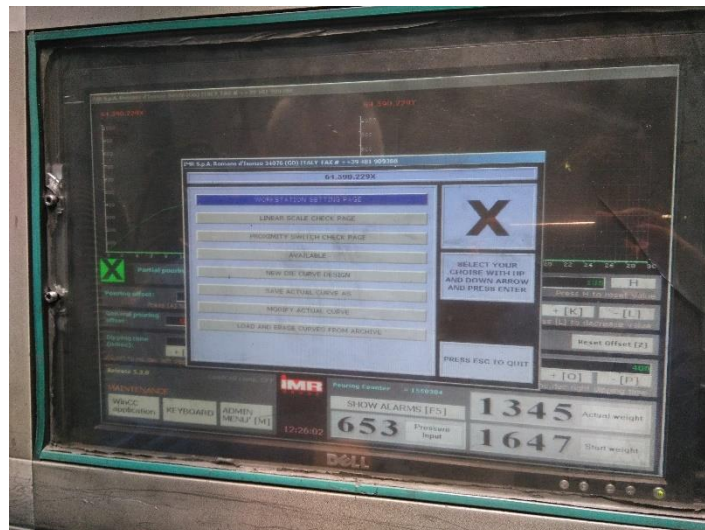


Figura 127 Menu da cabeça X (HMI antigo).

Na nova interface foram criados menus específicos para estas funções. Ao carregar no botão “NEXT” da parte superior do ecrã, é permitido o acesso a outro menu, “SAVE ACTUAL CURVE AS”, presente na Figura 128 com botões que permitem guardar a curva anteriormente criada, ou simplesmente carregar uma curva guardada em memória.

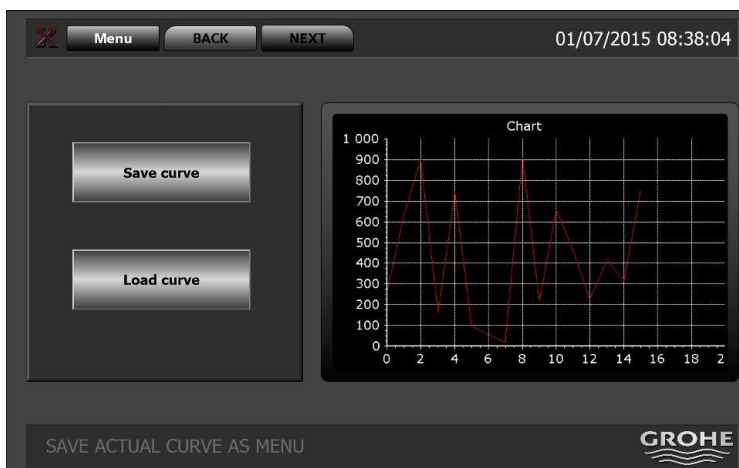


Figura 128 Carregar e guardar curva de vazamento.

Carregando novamente no botão “NEXT”, aparece diante do utilizador o menu para modificar curvas, o “MODIFY ACTUAL CURVE MENU”, apresentado na Figura 129.

Este menu permite fazer-se alteração de uma curva anteriormente criada. Para tal existem botões que permitem carregar curvas guardadas como o “Load curve”, e curvas guardadas em outros dispositivos como uma PEN, através do botão “Import curve”. Do mesmo modo o botão “Export curve” permite enviar as curvas criadas para outro dispositivo de armazenamento.

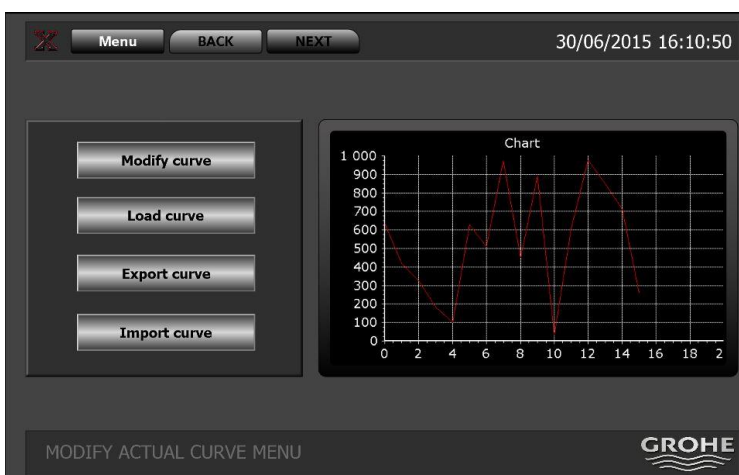


Figura 129 Modificar curva de vazamento.

Como referido anteriormente, este menu serve para modificar curvas já criadas. Carregando então no botão “Modify curve”, surgirá uma nova janela *popup*, Figura 130, que disponibiliza a introdução de novos valores de pressão e tempo para a configuração da curva.

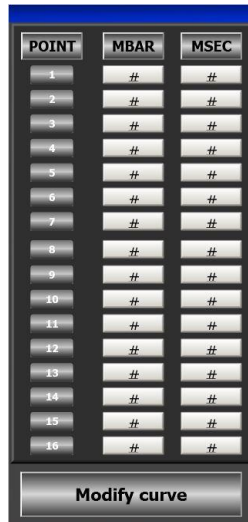


Figura 130 Atribuição de novos valores.

O último menu de configuração de curvas é o “LOAD AND ERASE CURVES FROM ARCHIVE MENU” (Figura 131), este menu serve para apagar as curvas criadas de modo que contem os botões necessário para fazer o carregamento quer da memória interna da consola quer de um dispositivo de armazenamento externo.

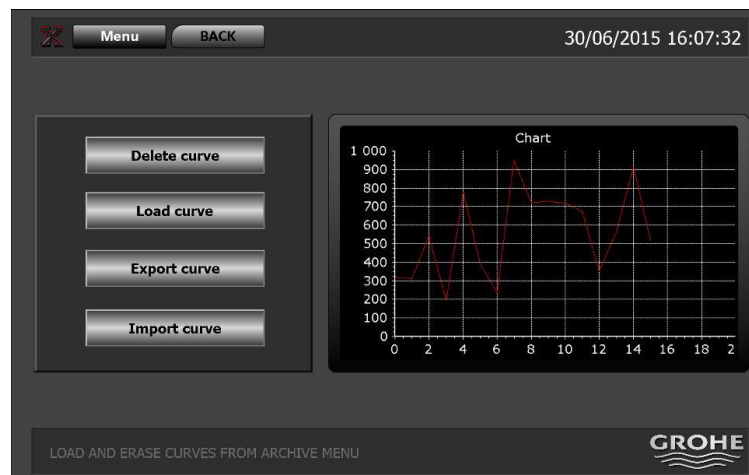


Figura 131 Apagar curvas de vazamento.

Uma outra limitação da interface antiga diz respeito ao facto de as configurações da máquina, forno e cabeças estarem associadas as curvas, ou seja, mediante a curva seleccionada todas as configurações são alteradas, este que acaba por ser um aspeto não desejado por parte da empresa.

Na nova consola todas as configurações e parâmetros estão divididos, não dependem da curva selecionada, fazendo com que a seleção da curva não altere qualquer tipo de configurações.

Anexo H. Componentes Informáticos

Foi gravado um CD-ROM com o conteúdo necessário para completar este projeto. Assim segue a lista dos dados contidos no CD:

- Programa da consola Beijer itX15B
- Programa STEP 7 da máquina em TIA Portal
- Ficheiros EXCEL (sintonização PID)
 - 1 sobre sistema do bidão;
 - 2 sobre sistema com o forno.