



GONÇALO MIGUEL VEREDAS VALENTIM

Licenciado em Ciências da Engenharia Eletrotécnica e de
Computadores

SIMULADOR CIBER-FÍSICO DE SISTEMA INDUSTRIAL COM TECNOLOGIA SIEMENS

MESTRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

Universidade NOVA de Lisboa
Abril, 2024



SIMULADOR CIBER-FÍSICO DE SISTEMA INDUSTRIAL COM TECNOLOGIA SIEMENS

GONÇALO MIGUEL VEREDAS VALENTIM

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Orientador: Luís Filipe Figueira Brito Palma
Professor Doutor, Universidade NOVA de Lisboa

Coorientador: Vasco Brito
Mestre, Universidade NOVA de Lisboa

Simulador ciber-físico de sistema industrial com tecnologia Siemens

Copyright © Gonçalo Miguel Veredas Valentim, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Dedicado à minha Família.

AGRADECIMENTOS

Gostaria de começar por agradecer ao professor Luís Filipe Figueira Brito Palma por ter aceite ser o meu orientador, e por por todo o apoio enquanto balançava a vida laboral com o desenvolvimento desta dissertação.

Ao meu co-orientador, Vasco da Silva Brito, gostaria de agradecer pela paciência e a sempre presente disponibilidade.

Gostaria também de agradecer aos meus amigos, que sempre me apoiaram e motivaram a concluir este marco importante na minha vida.

Gostaria também de deixar um agradecimento muito profundo e significativo à minha família, nomeadamente aos meus pais, pela motivação, apoio e afeto incondicionais e por nunca terem deixado de acreditar em mim, mesmo nos tempos mais difíceis.

Por último agradeço à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa por me ter moldado na pessoa que sou hoje e por me ter proporcionado o rumo e experiências de carreira já vivenciadas e ainda as que estão para vir.

“So it goes.” (Kurt Vonnegut)

RESUMO

Neste trabalho de dissertação foi proposto o desenvolvimento de um protótipo ciber-físico de ensino, de um processo industrial de enchimento e esvaziamento de um carrinho à escala laboratorial. Este protótipo encontra-se presente no laboratório de automação departamento de engenharia eletrotécnica da Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa e tem como foco o ensino de automação aos alunos do mesmo, recorrendo a um dos autómatos mais utilizados no mercado.

No presente trabalho de dissertação os maiores focos foram o desenvolvimento de todo o circuito de controlo através da instalação do hardware, a programação do autómato para o funcionamento desejado do processo, a programação da consolo HMI para um controlo e visualização dos alunos, dando assim ênfase à relação Homem-Máquina e a incorporação de controlo do sistema em anel aberto e em anel fechado.

A fim de testar todas as componentes físicas e virtuais do protótipo, foram realizados diversos testes ao processo para verificar o seu funcionamento, em anel aberto e em anel fechado.

Palavras-chave: Protótipo de Ensino, Processo Cíber-Físico, Processo Industrial, Autómato, Sinópticos, Automação.

ABSTRACT

In this dissertation work the development of a teaching cyber-physical prototype of an industrial process of filling and emptying of a trolley at a laboratory scale was proposed. This prototype is located in the automation laboratory of the department of eletrotechnical engineering of FCT-UNL and has the main focus of teaching the students automation using the most used automaton in the market.

The main focuses of this dissertation work were the development of all the control circuit, mainly through the the installation of the hardware, the programming of the controller to have the desired functioning, the programming of the Human Machine Interface to provide control and visualization of the processo to the students and the incorporation of control in an open and in a closed loop.

With the goal of testing all the physical and virtual components of the prototype, different tests were performed to the process, verifying it's functioning in an open and closed loop.

Keywords: Teaching Prototype, Cyber-Physical Process, Industrial Process, Automaton, Synoptics, Automation.

ÍNDICE

Índice de Figuras	x
Índice de Tabelas	xiv
Glossário	xv
Siglas	xvi
1 Introdução	1
1.1 Enquadramento e Motivação	1
1.2 Objetivos e Contribuições	2
1.3 Estrutura da Dissertação	3
2 Estado da Arte	4
2.1 História dos PLCs	4
2.2 Tecnologia dos PLCs	7
2.2.1 Hardware	7
2.2.2 Programação dos controladores	11
2.3 Tecnologia a Implementar	16
2.3.1 Hardware	16
2.3.2 Software	18
2.4 Comunicações	19
2.4.1 IEC61131-5	19
2.4.2 Interligação entre os controladores S7-1200 e S7-300	20
2.4.3 Integração de IOT	20
3 Projeto do Simulador Cíber-Físico	23
3.1 Arquiteturas de Alto Nível	24
3.2 Projeto de Hardware	26

3.2.1	Descrição do Protótipo do Processo Industrial de Transporte de Produtos	26
3.2.2	Descrição do Hardware	26
3.3	Projeto de Software	43
3.3.1	Necessidades a Nível de Software	43
3.3.2	Programação Lógica	44
3.3.3	Programação de Sinópticos	58
4	Resultados Experimentais	62
4.1	Descrição do Processo Cíber-Físico	62
4.2	Teste do Controlo em Anel Aberto	65
4.3	Teste do Controlo em Anel Fechado	66
4.3.1	Teste do Sensor	73
5	Conclusões e Trabalho Futuro	92
5.1	Conclusões	92
5.2	Trabalhos Futuros	93
5.2.1	TZID	93
5.2.2	IOT	94
	Bibliografia	95
	Apêndices	
	Anexos	

ÍNDICE DE FIGURAS

2.1	Painel de controlo de relés da década de 1960 [41].	5
2.2	Modicon 084 [11].	6
2.3	Rack [42].	7
2.4	Fonte de Alimentação Schneider [8].	8
2.5	Terminal de Programação Allen-Bradley [7].	8
2.6	Módulos I/O Mitsubishi [23].	9
2.7	CPU Omron [13].	11
2.8	Exemplo de um script em ST [35].	12
2.9	Exemplo de um script em IL [36].	12
2.10	Exemplo de um FBD [19].	13
2.11	Diagrama Geral de um SFC [5].	14
2.12	Número semanal de pesquisas relacionadas no motor de pesquisa Google [18].	14
2.13	Exemplo de um LD [30].	15
2.14	Motor SIEMENS D-91056 Erlangen [38].	16
2.15	Variador de Velocidade SIEMENS G110 com painel de operação básico [45].	17
2.16	SIEMENS SIMATIC S7-1200 [12].	18
2.17	SIEMENS SIMATIC S7-300 [49].	18
2.18	Painel SIEMENS HMI KTP400 Basic Color PN [3].	19
2.19	Ilustração da comunicação entre os PLCs S7-1200 e S7-300 [26].	20
2.20	Módulo SIMATIC IOT2000 [44].	21
2.21	PLC Schneider Modicon M340 [48].	22
3.1	Diagrama da Arquitetura de Alto Nível.	24
3.2	Processo Industrial de Transporte de Produtos AEG Logistat ET 705 [4]. . .	27
3.3	Autómato Siemens S7-1200 [32].	29
3.4	Linguagens de Programação suportados pelo Siemens S7-1200 [1].	30
3.5	Entradas Digitais do PLC e descrição das mesmas.	31
3.6	Saídas Digitais do PLC e descrição das mesmas.	32
3.7	Saída Analógica do PLC e descrição da mesma.	32

3.8	Módulo de Saídas Analógicas Siemens [22].	33
3.9	Siemens Simatic HMI KTP600 [21].	34
3.10	Módulo Switch Siemens para conexões Profinet [14].	34
3.11	Motor Siemens [14].	35
3.12	Dados para a instalação e utilização do motor [2].	35
3.13	Variador de Velocidade G110 [46].	36
3.14	BOP - Basic Operater Panel / Painel de Operação [46].	36
3.15	Fluxograma de Comissionamento do Sistema Variador/Motor [46].	37
3.16	Diagrama de Funcionamento do Variador [46].	38
3.17	Botoneira de marcha Telemecanique [9].	39
3.18	Botão de Emergência Telemecanique [17].	40
3.19	Sinal Sonoro de Alarme "Buzzer"[10].	40
3.20	Sinal de Alarme Luminoso - Hager SVN 131 [15].	41
3.21	Fonte de Alimentação PULS 24 Vdc [33].	41
3.22	Grafcet do Processo.	42
3.23	DataBlock - Conjunto de Tags [37].	44
3.24	Função [37].	45
3.25	Bloco de Organização (OB) e os vários processos associados [37].	46
3.26	Função e declaração de variável de saída[37].	47
3.27	Bloco de Função com chamada de um bloco de dados associado a este[37].	47
3.28	Estrutura da interface de um bloco de função [37].	48
3.29	Variáveis da Função ON/OFF.	49
3.30	Programação Lógica da Função ON/OFF.	50
3.31	Bloco de Função ON/OFF.	50
3.32	Variáveis da Função Mover para a Direita.	51
3.33	Programação Lógica da Função Mover para a Direita.	51
3.34	Variáveis da Função Mover para a Esquerda.	52
3.35	Programação Lógica da Função Mover para a Esquerda.	52
3.36	Variáveis da Função Abertura da Válvula de Enchimento.	53
3.37	Programação Lógica da Função Abertura da Válvula de Enchimento.	53
3.38	Variáveis da Função Abertura da Válvula Hidráulica de Esvaziamento.	53
3.39	Programação Lógica da Função Abertura da Válvula Hidráulica de Esvaziamento.	54
3.40	Variáveis da Função Sinal Sonoro de Alarme "Buzzer".	54
3.41	Programação Lógica da Função Sinal Sonoro de Alarme "Buzzer".	54
3.42	Variáveis da Função Sinal Luminoso de Alarme.	55
3.43	Programação Lógica da Função Sinal Luminoso de Alarme.	55
3.44	Variáveis da Função Inverter Rotação do Motor.	55
3.45	Programação Lógica da Função Inverter Rotação do Motor.	56
3.46	Programação Lógica dos Temporizadores.	56
3.47	Escrita de valores analógicos para o motor - conversão de V para Hz.	57

3.48	Escrita de valores analógicos para o motor - conversão de Hz para valor inteiro até 27648.	57
3.49	Leitura do autómato de valores analógicos (velocidade do motor) - conversão de valor inteiro até 27648 para Hz.	57
3.50	Menu inicial de sinópticos.	58
3.51	Ecrã de estado do processo em tempo real.	59
3.52	Ecrã de controlo do motor.	60
3.53	Ecrã de velocidade do motor ao longo do tempo.	61
4.1	Foto do Protótipo de Ensino Cíber-Físico.	63
4.2	Foto do Sinal "Buzzer"ativo aquando da pressão Botão de Emergência.	65
4.3	Foto do Sinal Luminoso ativo no momento em que se encontra aberta a válvula de enchimento.	66
4.4	Controlo do Processo em Anel Fechado.	67
4.5	Amplificador Operacional TL082 [28].	68
4.6	Amplificador Operacional TL084 [28].	68
4.7	Circuito Amplificador Inversor.	69
4.8	Valores de simulação com o motor em rotação no sentido horário.	70
4.9	Comportamento do sensor com o motor em rotação no sentido horário.	71
4.10	Valores de simulação com o motor em rotação no sentido anti-horário.	72
4.11	Comportamento do sensor com o motor em rotação no sentido anti-horário.	73
4.12	Placa de aquisição de dados USB 6008 e respetivos pinos de ligação [16].	74
4.13	Resultados experimentais de leitura do sensor em rotação anti-horária.	75
4.14	Resultados experimentais de leitura do sensor em rotação horária.	75
4.15	Funções para escalar valores para a gama desejada.	76
4.16	Comportamento em Regime Estacionário - 1 Hz.	77
4.17	Comportamento em Regime Estacionário - 2 Hz.	78
4.18	Comportamento em Regime Estacionário - 3 Hz.	79
4.19	Comportamento em Regime Estacionário - 4 Hz.	80
4.20	Comportamento em Regime Estacionário - 5 Hz.	81
4.21	Comportamento em Regime Estacionário - 6 Hz.	82
4.22	Comportamento em Regime Estacionário - 7 Hz.	83
4.23	Offsets de leitura relativamente aos setpoints para a gama de trabalho definida.	84
4.24	Comportamento em Regime Estacionário com Carga no Veio- 1 Hz.	85
4.25	Comportamento em Regime Transitório de 1 para 7 Hz.	86
4.26	Sobre-elevação em Regime Transitório de 1 para 7 Hz.	87
4.27	Sub-elevação em Regime Transitório de 1 para 7 Hz.	87
4.28	Comportamento em Regime Transitório de 1 para 4 Hz.	88
4.29	Sobre-elevação em Regime Transitório de 1 para 4 Hz.	88
4.30	Sub-elevação em Regime Transitório de 1 para 4 Hz.	89
4.31	Comportamento em Regime Transitório de 4 para 7 Hz.	89

4.32 Sobre-elevação em Regime Transitório de 4 para 7 Hz.	90
4.33 Sub-elevação em Regime Transitório de 4 para 7 Hz.	90
5.1 TZID [6].	93
5.2 Módulo IOT [34].	94
5.3 Módulo Arduino IOT [27].	94

ÍNDICE DE TABELAS

2.1	Fabricantes e percentagem dos seus equipamentos no mercado [50].	7
-----	--	---

GLOSSÁRIO

SIGLAS

INTRODUÇÃO

Neste capítulo são apresentados o enquadramento, motivação e objetivos que levaram ao desenvolvimento desta dissertação, dando ainda a conhecer a estrutura proposta para a mesma.

1.1 Enquadramento e Motivação

O presente trabalho de dissertação foca-se no desenvolvimento de um protótipo de ensino com incorporação Hardware/Software em automação com tecnologia SIEMENS, tendo o propósito de dar a conhecer, não só o equipamento e tecnologia como tipos de integração dos mesmos, aos alunos da área de Automação e Controlo do Mestrado Integrado de Engenharia Electrotécnica e de Computadores.

'Com o aumento significativo da necessidade de automação, um sistema de controlo tem a necessidade de ser facilmente programável, flexível, duradouro, robusto e economicamente eficiente.' [39]

O PLC é utilizado em vários processos e aplicações, em diversas áreas da Automação. Tendo em conta uma sondagem consideravelmente recente, estes controladores são aplicados predominantemente nas seguintes áreas:

- Controlo de Máquinas - 87%;
- Controlo de Processos - 58%;
- Controlo de Movimento - 40%;
- Batch Control* - 26%;
- Diagnóstico - 18%;
- Outras Aplicações - 3%.

O *Batch Control* é o controlo de documentos e dados electrónicos que estão em espera para processamento, depois de serem catalogados e agrupados.

É de referir que um sistema de controlo, por norma, tem diversas aplicações, daí advém a soma das percentagens não ser o valor esperado de 100% [41].

Tendo em consideração o predominante uso deste tipo de controlador na Indústria da Automação, torna-se fundamental a introdução e adaptação dos alunos a este tipo de equipamento. Esta dissertação pretende proporcionar uma ferramenta que permita uma fácil percepção e manuseamento da tecnologia a estes mesmos alunos.

1.2 Objetivos e Contribuições

O principal objetivo desta dissertação consiste na elaboração de um sistema de controlo de um motor conectado a um variador de velocidade, constituído pelo processo físico e implementação do mesmo em software, com troca de informação entre diferentes modelos de equipamento de automação com tecnologia SIEMENS.

Para o efeito, serão usados um motor SIEMENS D-91056 Erlangen de 0,12kW trifásico, um variador de velocidade SIEMENS G110 com um painel de operação básico, os controladores PLC SIMATIC SIEMENS S7-1200 e S7-300 e um painel SIEMENS HMI KTP400 Basic Color PN.

No que toca à partilha de dados, inicialmente será feita a troca de informação entre o controlador de gama mais recente (S7-1200) e de gama mais antiga (S7-300), demonstrando a versatilidade de comunicação dos controladores deste fabricante.

Devido à constante necessidade de adaptabilidade e conexão de instalações, tanto ao nível industrial como ao nível do ensino, irão ser implementados métodos de troca de informação com outros protótipos de ensino presentes no laboratório de Automação e Controlo, ilustrando assim a flexibilidade dos controladores seleccionados para o projecto.

Relativamente ao controlo e visualização do processo por software, este será realizado através de um painel HMI KTP400. Os softwares necessários para o desenvolvimento do sistema são SIEMENS SIMATIC STEP7 e SIEMENS SIMATIC WinCC, ambos do portal de automação deste fabricante, o TIA PORTAL - "Totally Integrated Automation Portal".

O STEP 7 será utilizado para a programação e configuração dos controladores S7-1200 e S7-300, sendo necessárias diferentes versões do software para cada. O WinCC permitirá a visualização e controlo do processo no painel HMI KTP400.

1.3 Estrutura da Dissertação

O presente trabalho de dissertação apresenta a seguinte proposta de estrutura:

Capítulo 1: Introdução

O primeiro capítulo, como foi previamente referido, é constituído pelos sub-capítulos Enquadramento e Motivação, Objectivos e Contribuição e Estrutura da Dissertação.

Capítulo 2: Estado da Arte

O capítulo relativo ao Estado da Arte compreende quatro sub-capítulos: um sub-capítulo inicial dedicado à história dos controladores lógicos programáveis; o segundo sub-capítulo abrange a tecnologia dos controladores PLC, tanto ao nível do hardware como do software, a sua programação e protocolos adjacentes e aplicações dos mesmos; um terceiro sub-capítulo dedicado à tecnologia utilizada na implementação deste protótipo, tanto ao nível de hardware como ao nível de software; e um sub-capítulo final referente às comunicações que iram ser implementadas neste trabalho de dissertação, nomeadamente a comunicação entre os controladores da SIEMENS e integração de IOT (Gateways, Módulos IOT, entre outros).

Capítulo 3: Proposta e Planeamento

No presente capítulo serão apresentadas as etapas pertinentes para o desenvolvimento desta dissertação e uma estimativa do espaço temporal que é considerado necessário para a conclusão das mesmas.

2.1 História dos PLCs

Um PLC é uma unidade de programação lógica que executa funções de controlo em diversos tipos de instalações [41].

O método de controlo de máquinas utilizado antes da existência dos PLCs era o controlo por relés. Os relés funcionam através de uma lógica ON/OFF, quando estimulados por uma corrente dá-se a comutação da posição do interruptor para a posição contrária à de repouso (ON ou OFF), quando essa corrente deixa de ser induzida o interruptor remota à posição de repouso.

Tendo em consideração a contínua evolução dos processos de controlo, necessitando de cada vez mais componentes para desafios de engenharia mais complexos, seria um processo bastante moroso, propício a uma quantidade significativamente maior de falhas e de bastante manutenção, usar relés para o controlo deste tipo de instalações. Pois bastava ocorrer apenas uma falha num dos relés do processo, para este deixar de funcionar [20]. Com base nisto, os sistemas de controlo eram constituídos por um número bastante elevado de fios, terminais e relés. Os painéis tinham um aspecto semelhante ao apresentado na figura 2.1:



Figura 2.1: Painel de controlo de relés da década de 1960 [41].

Para solucionar este problema os engenheiros da empresa General Motors propuseram as seguintes condições para o desenvolvimento de um novo sistema de controlo generalizado [20] :

- 1) Permitir a expansão e troca de componentes através da sua arquitectura modular;
- 2) Funcionar num ambiente industrial, tendo em conta os factores associados ao mesmo, como poeira, sujidade, humidade, vibrações e electromagnetismo;
- 3) Ser um sistema de estado sólido que funcione com uma lógica de controlo à base de relés, flexível, de custo moderado e de fácil manutenção;
- 4) Programável de acordo com a linguagem de programação lógica de relés Ladder.

O primeiro controlador lógico programável a cumprir esses requisitos foi o Modicon 084, desenvolvido por Dick Morely em 1969, na sua empresa Modicon.



Figura 2.2: Modicon 084 [11].

Desde esse progresso na área de Automação e Controle, os PLCs têm sido desenvolvidos com cada vez mais capacidade de processamento, flexibilidade, durabilidade e conectividade.

Cerca de 80% dos controladores usados na indústria são distribuídos por sete fabricantes. Sendo a contribuição de cada um desses fabricantes expressa na tabela abaixo [50]:

Fabricante	Presença no Mercado [%]
SIEMENS	30,7
Rockwell Automation	21,6
Mitsubishi	13,9
Schneider Electric	8,9
Omron	6,6
GE Fanuc	4,0
Moeller	2,3

Tabela 2.1: Fabricantes e percentagem dos seus equipamentos no mercado [50].

Existem ainda outros fabricantes não representados na tabela 2.1 que compõem a restante porção do mercado.

2.2 Tecnologia dos PLCs

Neste sub-capítulo será apresentada a constituição de um PLC, os elementos que constituem um controlador deste tipo, a linguagem de programação utilizada e aplicações do mesmo.

2.2.1 Hardware

A arquitectura dos sistemas controlados por PLC é feita através da integração de vários módulos com funções diversificadas, sendo estes:

1) Rack

Tendo em conta os diversos módulos que são imprescindíveis para o devido funcionamento do PLC, torna-se importante a localização próxima dos módulos. A rack do PLC tem o propósito de manter os módulos na devida posição e ainda proporcionar contactos eléctricos entre estes. A última função é possível através de uma placa de circuito impresso que se encontra na parte de trás da rack.



Figura 2.3: Rack [42].

2) Fonte de Alimentação

A fonte de alimentação fornece energia em corrente contínua (DC) para os módulos instalados na rack, encontra-se um exemplar abaixo:



Figura 2.4: Fonte de Alimentação Schneider [8].

3) Terminal de Programação

A programação da Unidade Central de Processamento é feita através do Terminal de Programação. Dependendo do tipo e condições da instalação, preferência de equipamento do cliente, fabricante e outros factores, serão implementados diversas variantes de terminais. Estes podem variar de pequenos módulos com uma ou poucas linhas de código, a monitores de visualização incorporados com teclados e ainda a computadores portáteis [39]. Abaixo encontra-se um exemplo de um terminal de programação:



Figura 2.5: Terminal de Programação Allen-Bradley [7].

4) Zona de Input/Output

A interface e conexão dos dispositivos de campo com a unidade de processamento

central é feita através da zona de I/O. A interface desta zona com os dispositivos exteriores pode ser feita de um par de maneiras [39]:

1) Do tipo fixo, em que todos os componentes estão em um único módulo, sem possibilidade de alteração do número de entradas e/ou saídas. Predominantemente implementados em sistemas de PLC de escala reduzida;

2) Do tipo modular, incorporados na rack, permitindo assim a alteração do número de entradas e/ou saídas de acordo com a evolução de uma dada instalação. Permitindo uma maior adaptabilidade e flexibilidade quando comparando com o tipo fixo.



Figura 2.6: Módulos I/O Mitsubishi [23].

Estes módulos I/O diversificam em termos de aplicação, podendo ser dos seguintes tipos:

1) Módulo de Entrada - Input

Devido às diferentes tensões de operação entre o terminal de programação da unidade de processamento central e os dispositivos externos, é necessária a precaução de separar as tensões para evitar danos à electrónica do PLC e a filtragem de picos de tensão e ruído eléctrico [39].

2) Módulo de Saída - Output

Este módulo tem o propósito de conectar a Unidade de Processamento Central à carga e isolar a linha entre o mesmo e o circuito exterior. Como o módulo de entrada, este também filtra ruído eléctrico e picos de tensão [39].

3) Módulo Discreto Entrada/Saída - I/O

O módulo discreto Entrada/Saída proporciona, com o controlo de entrada a interligação entre dispositivos de entrada de sinais discretos e com o controlo de saída a

interligação entre dispositivos de saída da mesma natureza.

4) Módulo Analógico Entrada/Saída - I/O

Com a introdução deste tipo de módulo, os PLCs passaram a conseguir processar sinais de natureza não discreta, sinais analógicos. Este avanço permitiu o controlo de processos, não só com sinais de natureza binária, mas como processos que necessitam de valores quantificados, tendo como exemplo o nível de líquido num tanque, a temperatura do óleo de um motor, o peso presente no tapete rolante de uma linha de montagem, entre outros [39].

5) Módulos Especiais Entrada/Saída - I/O

Tornou-se necessário o desenvolvimento de módulos especiais I/O para tipos específicos de controlo, como o módulo PID (Controlo Proporcional Integral), o módulo ASCII (código binário, padrão de codificação), o módulo de saída BCD (outro código binário, padrão de codificação), o módulo de controlo de posição e movimento, os módulos de comunicação e outros [39].

5) Módulo de Processamento

O módulo de processamento tem a função de gerir todo o processo do sistema de controlo programável. Este contém um cartão de memória, no mínimo um microprocessador e circuitos que proporcionam o armazenamento e recolha de dados na memória. Este módulo é constituído pela CPU (Unidade de Processamento Central), ALU (Unidade Lógica e Aritmética) e a memória [39].

1) CPU - tem a função de aceder à memória do processador para executar os programas que lá se encontram armazenados e aceder à informação dos mesmos, coordenar quando e o que será feito na ALU e memória, diagnósticos de rotinas e comunicar com outros processadores e dispositivos periféricos.

2) ALU - tem o propósito de realizar operações matemáticas (de adição) e operações lógicas (lógica booleana AND)

3) Memória - esta encontra-se dividida de acordo com a função de cada uma das suas secções e armazena dados e programas que irão ser acedidos pela CPU.



Figura 2.7: CPU Omron [13].

2.2.2 Programação dos controladores

As linguagens estabelecidas para a programação de PLCs estão definidas pelo padrão IEC61131-3 da Comissão Electrotécnica Internacional (IEC). No sub-capítulo seguinte são apresentadas as linguagens de programação apresentadas neste padrão, sendo a linguagem utilizada predominantemente a linguagem Ladder.

2.2.2.1 IEC61131-3

Os requerimentos dos sistemas de PLC, tanto ao nível do hardware como ao nível do software, encontram-se normalizados no padrão IEC61131. O padrão IEC61131-3 é uma directriz para a programação de PLCs, que inclui 62 tabelas de requerimentos, permitindo a verificação da proximidade do controlador ao padrão.

Devido à constante evolução das instalações e dos seus sistemas de controlo, é fundamental conseguir um equilíbrio entre a expansão do sistema e os custos associados ao mesmo. A padronização permite assim reduzir custos, tanto ao nível dos fabricantes como dos consumidores. Os fabricantes reduzem os seus custos: no processo de desenvolvimento de software, podendo partilhar editores de código, usar softwares previamente testados para o teste de um novo, entre outros pontos; no processo de desenvolvimento de hardware com a utilização de componentes já existentes no mercado, cortando os gastos no desenvolvimento desse componente. Os consumidores reduzem nos seus custos: através de uma formação padronizada que corta a generalidade dos custos necessários para a formação em equipamentos de vários fabricantes; com a portabilidade e flexibilidade dos controladores PLC, podendo ser adaptados a novas extensões adicionadas a um sistema de controlo prévio, ou mesmo a novos sistemas de controlo.

A harmonização e compatibilidade das linguagens de programação dos PLCs, permite uma idealização de sistemas de controlo não só para as necessidades actuais como para futuras [40].

2.2.2.2 Linguagens de Programação

Existem cinco linguagens de programação de PLCs que se encontram padronizadas pelo IEC61131-3, em que duas são linguagens de programação em texto e três são linguagens de programação gráficas.

Dentro das linguagens em texto encontram-se:

1) Texto Estruturado (Structured Text - ST)

Linguagem de programação de alto nível semelhante a C++, geralmente usada para o processamento de dados complexos [51].

```
IF %M0 THEN
  FOR %MW99:=0 TO 31 DO
    IF %MW100[%MW99]<>0 THEN
      %MW10:=%MW100[%MW99];
      %MW11:=%MW99;
      %M1:=TRUE;
      EXIT; (* exit the loop *)
    ELSE
      %M1:=FALSE;
    END_IF;
  END_FOR;
ELSE
  %M1:=FALSE;
END_IF;
```

Figura 2.8: Exemplo de um script em ST [35].

2) Lista de Instruções (Instruction List - IL)

Código representado numa lista de instruções, semelhante à linguagem de programação Assembly. Permite acesso directo ao equipamento com codificação rápida e compacta [51].

```
LD      [%MW50>10]
ST      %Q2.2
LD      %I1.0
        [%MW10:=%KW0+10]
LDF     %I1.2
        [INC%MW100]
```

Figura 2.9: Exemplo de um script em IL [36].

As linguagens gráficas são:

3) Diagrama de Blocos (Function Block Diagram - FBD)

Linguagem de programação gráfica com foco I/O, em que o código é representado por blocos de funções - FBD. Como é possível verificar na figura 2.10, a linguagem FBD apresenta blocos de funções com lógica entre entradas e saídas e conexões entre estes blocos. Esta lógica interior pode ser codificada para executar processos de controlo ainda mais complexos [51].

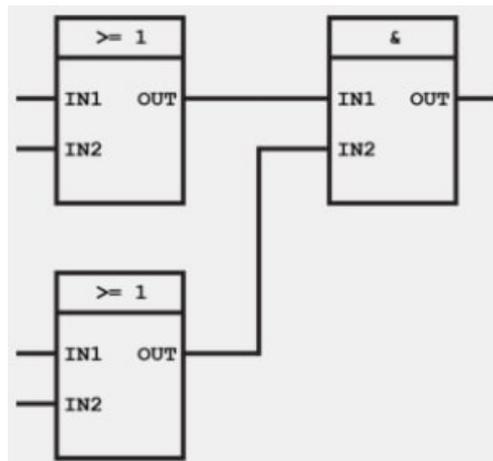


Figura 2.10: Exemplo de um FBD [19].

4) Diagrama de Fluxo (Sequential Function Chart - SFC)

Linguagem de programação representada graficamente com um aspecto semelhante a um fluxograma, que formula a sua saída utilizando estados e transições. Os estados são funções que englobam condições e específicos eventos que necessitam destas mesmas para serem iniciados. As transições têm por base o valor lógico verdadeiro ou falso para permitir ou não a transição de um estado para outro. É possível verificar esta arquitetura na figura 2.11. Vários estados podem ser executados simultaneamente utilizando ramificações, estas inicializam vários estados num determinado instante previamente determinado. Esta linguagem permite ainda uma monitorização visual do processo [51].

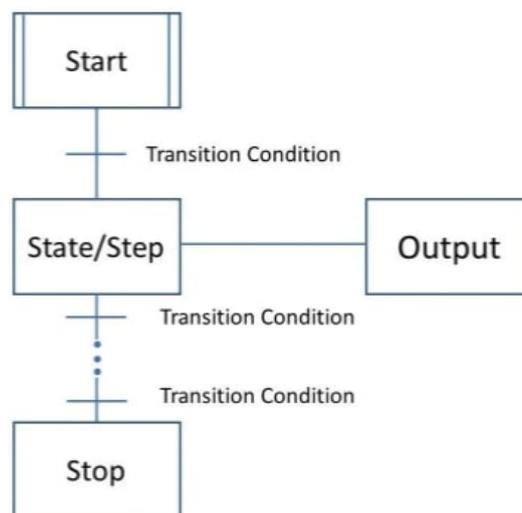


Figura 2.11: Diagrama Geral de um SFC [5].

5) Diagrama Ladder (Ladder Diagram - LD)

Através da análise do histórico de pesquisas do Google, dados disponíveis em fonte aberta que representam uma boa referência de popularidade para o estudo da utilização destas linguagens de programação, foi formalizada uma representação da razão entre as pesquisas semanais e as diferentes linguagens, como é possível verificar na figura 2.12. Com o propósito de normalização, foi definido um tecto máximo para o valor da representação da frequência de pesquisa semanal destas linguagens de 100. Para uma comparação viável, os dados foram todos normalizados à mesma escala [18].

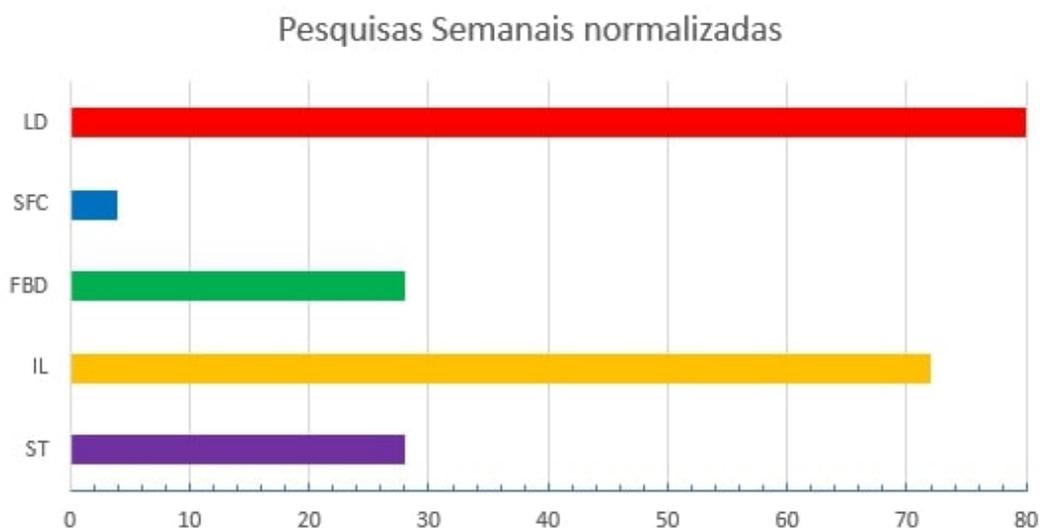


Figura 2.12: Número semanal de pesquisas relacionadas no motor de pesquisa Google [18].

Com a análise da figura 2.12 verifica-se que as linguagens com maior interesse no

mercado são a LD e IL, de representação gráfica e em texto respectivamente, sendo então as com maior uso em instalações. De referir ainda que a linguagem que apresenta um menor interesse no mercado é a SFC, sendo esta a que apresenta um menor uso em instalações.

Para além de a linguagem LD ser a mais utilizada, também é a linguagem que se encontra há mais tempo em uso no que toca à programação de PLCs. Como referido no capítulo 2.1, a arquitectura dos sistemas de controlo anteriores aos PLCs era à base de lógica de relés. Para facilitar a transição para o novo tipo de sistemas de controlo, esta linguagem foi modelada à base dessa mesma lógica [51].

Por ser uma linguagem gráfica de programação que recorre a uma notação simbólica para representar operações lógicas e apresentar uma aparência semelhante aos circuitos electrónicos de lógica de relé, torna-se mais acessível a aprendizagem desta linguagem [29].

As conexões electrónicas são representadas por régua de contactos e patamares, em que as régua simbolizam a fonte de alimentação e os patamares o número de circuitos de controlo [51]. Cada um destes patamares é basicamente uma linha de lógica com um resultado final binário ON/OFF. É possível observar as características descritas na figura 2.13.

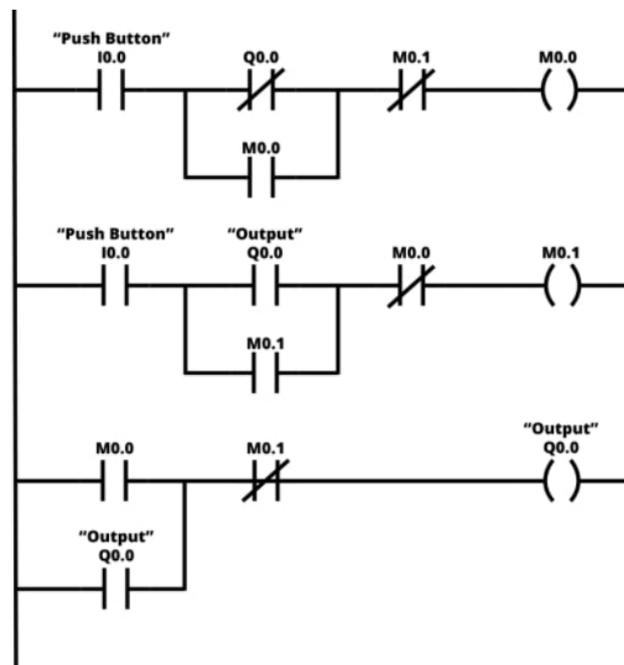


Figura 2.13: Exemplo de um LD [30].

2.3 Tecnologia a Implementar

Neste sub-capítulo serão apresentados os equipamentos e tecnologia a implementar na elaboração do protótipo de ensino, primeiramente ao nível do hardware e no momento seguinte ao nível do software.

2.3.1 Hardware

Como referido no sub-capítulo 1.2, para o desenvolvimento deste protótipo serão utilizados os seguintes equipamentos:

Motor SIEMENS D-91056 Erlangen

O processo que irá ser o foco do sistema de controlo é um motor trifásico conectado a um variador de velocidade, como referido no sub-capítulo 1.2. O motor seleccionado para o efeito é um motor trifásico SIEMENS D-91056 Erlangen de 0,12kW. As especificações e um exemplo do modelo deste motor, com diferentes especificações, encontram-se abaixo:



Figura 2.14: Motor SIEMENS D-91056 Erlangen [38].

- 1) Frequência - 50Hz
- 2) Potência - 0,12kW
- 3) Tensão - 230/400 V
- 4) Intensidade de Corrente - 0,73/0,42 A
- 5) Factor de Potência - 0,75
- 6) rpm - 1350

De referir que estes níveis de tensão são relativos à configuração do motor em Δ Y e que a totalidade dos valores das especificações são apenas apresentados para a frequência da rede em uso, 50 Hz.

Variador de Velocidade SIEMENS G110 com painel de operação básico

Para o controlo da velocidade do motor trifásico irá ser utilizado um variador de velocidade SIEMENS G110 com painel de operação básico. Tendo em conta a potência debitada pelo motor, 0,12kW, esta gama mais baixa de variador com painel de operação básico é perfeitamente capaz do controlo pois opera num espectro entre [0.12-0.37]kW [52]. Estes equipamentos são controlados por microprocessadores e englobam uma vasta gama de aplicações de controlo de motores. Na figura 2.15 é possível observar o modelo que irá ser utilizado na implementação do protótipo de ensino.



Figura 2.15: Variador de Velocidade SIEMENS G110 com painel de operação básico [45].

PLC SIMATIC SIEMENS S7-1200 e S7-300

Os controladores lógicos programáveis que irão ser utilizados para o desenvolvimento deste protótipo são o S7-1200, visível na figura 2.16 e o S7-300, visível na figura 2.17. Ambos operam com um funcionamento de lógica de relés, utilizando a linguagem LD. De seguida, são referidas características compreendidas como relevantes para a escolha da funcionalidade de cada PLC.

Relativamente à comunicação, o protocolo MODBUS é o único disponível no S7-1200. Tem uma maior eficiência quando comparado com o S7-300, mas em contrapartida tem um processamento mais lento. Além de ter um número limitado de entradas e saídas. Estes controladores foram desenhados para ambientes mais protegidos, sem elementos como poeiras, sujidade e outros, logo têm uma menor resistência quando comparados com os S7-300 [24].

Os controladores S7-300 trabalham com o protocolo PROFINET para a sua comunicação. Estes são mais robustos e resistentes quando comparados ao S7-1200. Têm uma maior expansão no que toca ao número de entradas e saídas, maior número de funções operacionais e são projectados para controlar processos mais completos, em qualquer condição ambiental [25].



Figura 2.16: SIEMENS SIMATIC S7-1200 [12].



Figura 2.17: SIEMENS SIMATIC S7-300 [49].

Painel SIEMENS HMI KTP400 Basic Color PN

O painel representado na figura 2.18 permitirá a visualização e controlo do processo em sinópticos de compreensão acessível. A programação visual de todo o processo será feito no software WinCC, como referido no sub-capítulo 1.2.

As características relevantes ao desenvolvimento do protótipo desta interface homem-máquina são as seguintes:

- 1) Painel básico;
- 2) Operação com painel e botões;
- 3) Painel de 4 polegadas;
- 4) Interface PROFINET.

2.3.2 Software

O programa que será utilizado na programação e configuração dos controladores e, por consequência, no desenvolvimento da lógica do processo que irá estar sobre o controlo do

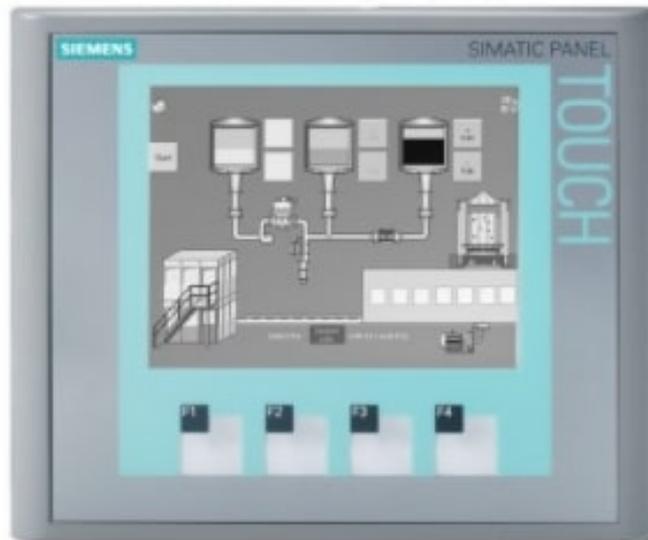


Figura 2.18: Painel SIEMENS HMI KTP400 Basic Color PN [3].

sistema é o SIMATIC STEP 7 (do TIA Portal). Este software permite a configurar tanto o hardware que constitui o processo como as redes que este tem incorporadas.

Para a visualização e controlo do processo na HMI KTP400 será utilizado o programa WinCC. Este software permite o desenvolvimento de um sistema de supervisão, controlo, aquisição de dados e interface entre homem e máquina.

2.4 Comunicações

Neste sub-capítulo serão apresentados os métodos de comunicação dos PLCs, a norma que os rege, comunicação entre PLCs do mesmo fabricante, neste caso a interligação entre os controladores SIEMENS S7-1200 e S7-300 e comunicação entre PLCs de outros fabricantes.

2.4.1 IEC61131-5

O padrão IEC61131-5 é uma directriz para a comunicação de PLCs, que especifica como um dispositivo externo pode conectar-se a um PIC actuando como servidor e como se dá a comunicação entre um PLC e um qualquer dispositivo externo. De referir que a comunicação dependerá das capacidades de cada sistema de comunicação individual. Este padrão demonstra como actua o PLC quando está a providenciar serviços a dispositivos externos e como o PLC requisita dispositivos externos para determinados processos. Um PLC providencia funções específicas ao resto do sistema de controlo e pode requisitar funções de outros PLCs, inclusivé pode comunicar com dispositivos externos que se comportem como um PLC utilizando a função de cliente [47].

2.4.2 Interligação entre os controladores S7-1200 e S7-300

A Interligação entre os controladores S7-1200 e S7-300 é feita utilizando a Comunicação S7. Como referido no capítulo 2.3, a programação e configuração dos PLCs para este efeito será feita no software STEP 7, com versões diferentes para cada um dos controladores.

Será necessário estabelecer duas comunicações S7. Para se estabelecer a primeira conexão, o controlador S7-1200 funciona como servidor e o controlador S7-300 como cliente, sendo este último o que efectivamente estabelece a comunicação. Isto significa que o S7-300 recebe informação do S7-1200 e escreve neste. Para a segunda conexão dá-se a inversão de funções, o controlador S7-300 actua como servidor e o controlador S7-1200 como cliente, sendo este último que estabelece a comunicação. Pela lógica anterior, o S7-1200 recebe informação do S7-300 e escreve neste [26].

Na figura 2.19 encontram-se representados a comunicação entre estes controladores e o processo descrito acima.

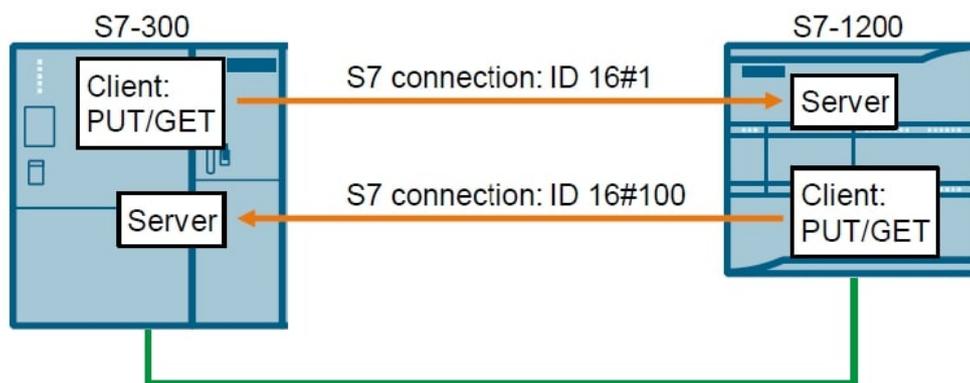


Figura 2.19: Ilustração da comunicação entre os PLCs S7-1200 e S7-300 [26].

2.4.3 Integração de IOT

O presente sub-capítulo tem o propósito de apresentar as diferentes formas de comunicação com outros dispositivos que irão ser estudadas para implementação no protótipo de ensino. Pretende-se implementar um módulo IOT, permitindo assim a partilha de informação em cloud, aproximando o protótipo a um ambiente de Indústria 4.0. Outra possível solução será a implementação de uma gateway para a troca de informação com outros projectos de dissertação presentes no laboratório de Automação e Controlo.

Com um módulo IOT tornar-se-ia acessível a comunicação entre uma vasta gama de equipamentos que se encontra no laboratório, proporcionando inúmeras possibilidades para a incorporação de vários equipamentos, protótipos e a implementação de instalações cada vez mais complexas. Existem várias possibilidades para a escolha de um módulo deste tipo, mas para uma melhor conectividade com o protótipo em desenvolvimentos, irão ser estudadas soluções dentro do mesmo fabricante. Tendo como referência o módulo

SIEMENS SIMATIC IOT2000 representado na figura 2.20, serão descritas as características deste tipo de módulos.



Figura 2.20: Módulo SIMATIC IOT2000 [44].

O módulo SIEMENS SIMATIC IOT2000 transfere, recebe e analisa informação de e para a cloud ou as tecnologias de informação implementadas na instalação. Relativamente à recolha de dados, este módulo recebe informação de vários dispositivos, como sensores e controladores e tem suporte para vários protocolos através de diversas interfaces. No que toca ao processamento, armazenamento e visualização desses dados, estes processos podem ser efectuados de inúmeras formas utilizando várias linguagens de alto nível. Todas as transferências de dados são feitas de uma maneira flexível e segura, de acordo com os protocolos apropriados [43].

Implementando uma gateway, esta será dimensionada para uma comunicação entre os controladores SIEMENS S7-1200 e S7-300 e o controlador Schneider Modicon M340. A interligação será feita com o controlador M340 (um exemplar do mesmo é representado na figura 2.21), pois este encontra-se a ser utilizado no desenvolvimento de outros projectos de dissertação presentes no laboratório de Automação e Controlo. Promovendo assim a interactividade entre diferentes processos presentes no laboratório.



Figura 2.21: PLC Schneider Modicon M340 [48].

Para ser possível a comunicação entre os diferentes PLCs é necessário ter em consideração os diferentes padrões de comunicação de cada controlador. Como referido no sub-capítulo 2.3, os controladores SIEMENS S7-1200 e S7-300 comunicam de através de PROFINET. O controlador Schneider M340 comunica através de MODBUS (um diferente protocolo industrial de troca de dados). Será então necessário uma gateway PROFINET/MODBUS para permitir a comunicação entre os diferentes projectos de dissertação.

Poderão ainda ser implementados outros métodos de comunicação que tragam uma maior interligação no laboratório de Automação e Controlo.

PROJETO DO SIMULADOR CÍBER-FÍSICO

Neste capítulo são explicitadas as arquiteturas e metodologias implementadas para o desenvolvimento deste trabalho de dissertação, tanto a nível físico, como a nível de programação do mesmo. Este trabalho tem o propósito de funcionar como um protótipo de ensino, correlacionado o ambiente industrial com o ambiente laboratorial, para a programação de automação, nomeadamente ao nível de linguagens utilizadas na indústria em PLC's, para os alunos do laboratório de Automação do departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade NOVA de Lisboa.

Este capítulo encontra-se dividido em três sub-capítulos. No primeiro sub-capítulo são desenvolvidas as Arquiteturas de Alto Nível, tanto ao nível de controlo, como de comunicação. No segundo sub-capítulo são explicitadas as arquiteturas de implementação, tanto ao nível de hardware, como ao nível de software. No sub-capítulo final são enumerados os componentes utilizados ao nível de hardware, as suas funcionalidades e especificações, assim como é feita a descrição da programação e configuração ao nível de software.

3.1 Arquiteturas de Alto Nível

A ideia para a implementação deste projeto surgiu da necessidade da incorporação de aprendizagem, relativa aos equipamentos Siemens, no percurso curricular dos alunos de Automação. Remetendo à Figura 2.12, é possível verificar que este fabricante tem o domínio do mercado no que toca a instalações industriais e fabris, sendo assim pertinente a familiarização dos alunos com estes equipamentos para futuras oportunidades de carreira.

É de mencionar que este protótipo tem a possibilidade de expansão de funcionalidades. Funcionalidades estas que se encontram desenvolvidas no capítulo seguinte, no sub-capítulo Trabalhos Futuros.

Na Figura 3.1 encontra-se o Diagrama da Arquitetura de Alto Nível do sistema.

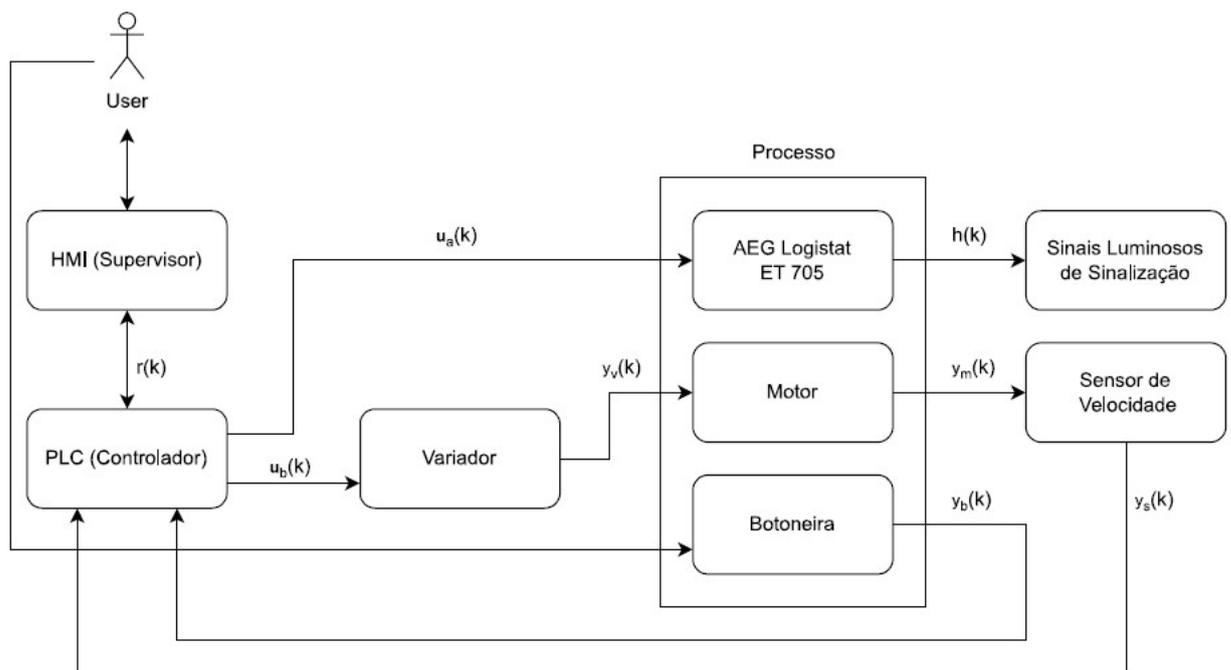


Figura 3.1: Diagrama da Arquitetura de Alto Nível.

Os sinais representados na Figura 3.1 são os seguintes:

- $r(k)$ - Comunicação bidirecional entre Controlador e Supervisor;
- $u_a(k)$ - Sinais Digitais de Saída do Controlador;
- $u_b(k)$ - Sinais Analógicos de Saída do Controlador;
- $y_v(k)$ - Sinal Analógico de Saída do Variador para controlo do Motor;
- $y_m(k)$ - Sinal Analógico de Saída do Motor para leitura do Sensor de Velocidade;
- $y_b(k)$ - Sinais Digitais de Saída da Botoneira para processamento no Controlador;
- $y_s(k)$ - Sinal Analógico de Saída do Sensor de Velocidade para processamento no Controlador;
- $h(k)$ - Sinal Digital para Ativação de Alarmes.

A Figura 3.1 não incorpora o Switch utilizado no protótipo, pois este funciona apenas como gateway para comunicação entre dispositivos, podendo ainda permitir a comunicação com mais dois dispositivos, sendo um deles o computador responsável pela programação do autómato e dos respetivos sinópticos.

A comunicação entre Switch, Controlador, Supervisor, computador de programação e outro possível dispositivo é feita através de cabo Ethernet, numa rede Profinet, que é o protocolo de comunicação Siemens. A restante comunicação no sistema é feita através de sinais digitais e de sinais analógicos. Os sinais digitais estão presentes na comunicação do Controlador para o Processo AEG Logistat ET 705, deste Processo para os Sinais Luminosos de Sinalização de alarme e nos Sinais enviados da Botoneira (Botoneira de Marcha e Botão de Emergência) para o Controlador. Por último, os sinais analógicos estão incorporados no protótipo com a comunicação do Controlador para o Variador de Velocidade, deste Variador para o Motor, do Motor para o Sensor de Velocidade e do Sensor de Velocidade para o Controlador.

Tornou-se necessário o desenvolvimento da arquitetura deste Sensor, isto porque os Sinais Analógicos do Motor variam de sinal de acordo com a rotação do mesmo, no sentido horário a tensão lida é negativa e no sentido anti-horário a tensão lida é positiva. Como o Controlador permite uma leitura na gama de [0-10] V, é pertinente que os valores sejam todos positivos e dentro dessa gama. A implementação e eletrónica deste Sensor são explicitadas neste capítulo. É de referir que este Sensor encontra-se desenvolvido, mas ainda não implementado para comunicar com o Controlador. Sendo assim, neste momento, o sistema tem um Controlo em Anel Aberto. Posteriormente à sua instalação, o sistema terá um Controlo em Anel Fechado.

3.2 Projeto de Hardware

3.2.1 Descrição do Protótipo do Processo Industrial de Transporte de Produtos

O sistema desenvolvido no âmbito desta dissertação é constituído por diversos componentes de Hardware, alguns essenciais para o funcionamento do mesmo, outros incluídos para uma maior aproximação à realidade do ambiente industrial e para uma melhor compreensão dos alunos relativamente ao processo.

Os componentes essenciais são o PLC Siemens S7-1200, o Processo Industrial de Transporte de Produtos AEG Logistat ET 705 e a Fonte de Alimentação PULS 24 Vdc. Com estes é possível o controlo do processo, mas a vertente didática do protótipo seria muito reduzida.

Os componentes incorporados para o complemento do sistema são o Módulo de Saídas Analógicas Siemens, a HMI Siemens Simatic KTP600, o Módulo Switch Siemens, o Motor Siemens, a Botoneira de marcha e Botão de Emergência Telemecanique e o Sensor de Velocidade.

No sub-capítulo seguinte são descritas os equipamentos referidos e as suas funcionalidades.

3.2.2 Descrição do Hardware

3.2.2.1 Processo Industrial de Transporte de Produtos AEG Logistat ET 705

Na Figura 3.2 encontra-se representado o processo industrial, de transporte de produtos, que foi utilizado nesta dissertação para futuro efeito pedagógico de controlo de processos e linguagens de programação complementares ao mesmo.

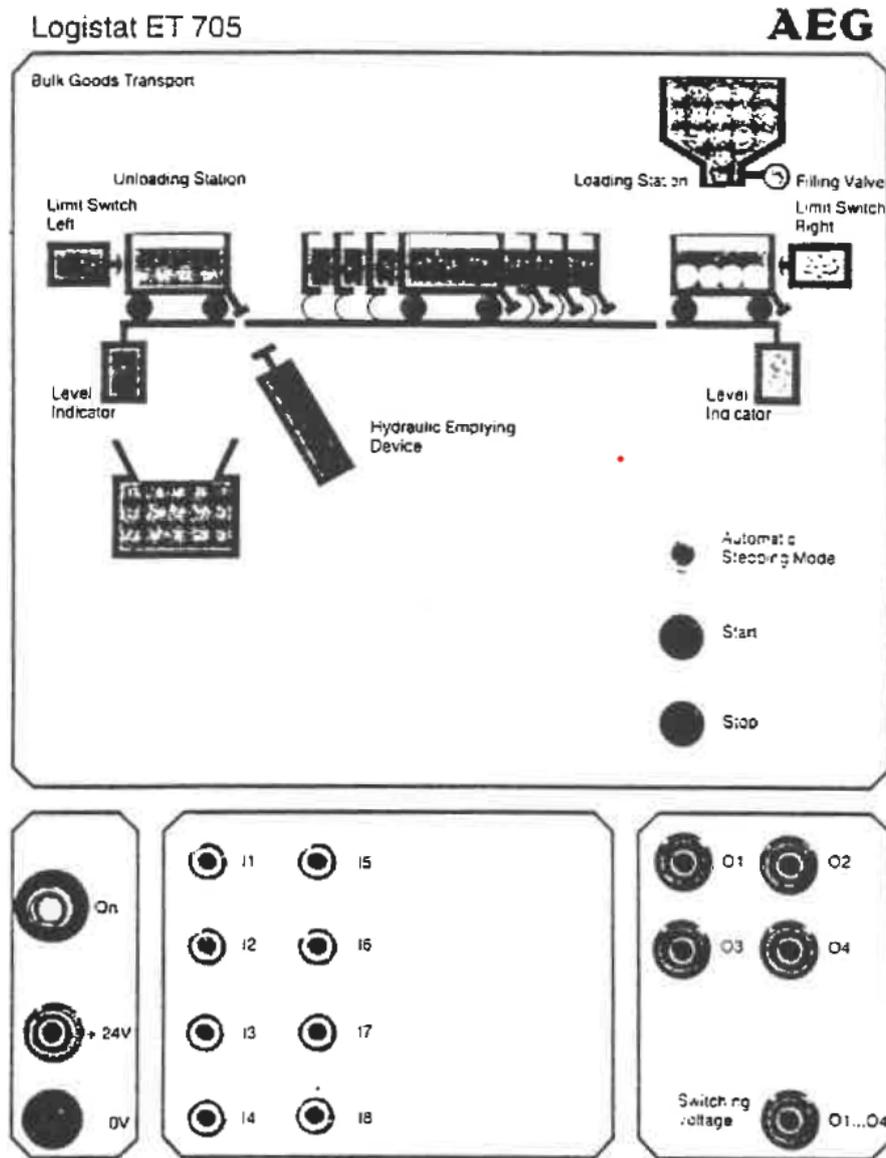


Figura 3.2: Processo Industrial de Transporte de Produtos AEG Logistat ET 705 [4].

O Processo ET705 reage ao sinais de entrada através de LEDs coloridos que dão a aparência de movimento ao processo. Este processo, por si, não incorpora funções de segurança, como um botão de emergência, o que motivou à adição dessas mesmas funções através de outros componentes. A alimentação do mesmo é feita a 24 Vdc [4].

O Processo é controlado através de 4 entradas e 8 saídas digitais, entradas e saídas para o simulador do processo, o inverso relativamente ao PLC. As mesmas estão descritas abaixo:

Entradas Digitais [0;24]Vdc:

- Q1 - Motor para a direita;
- Q2 - Motor para a esquerda;
- Q3 - Abertura da Válvula de Enchimento;
- Q4 - Abertura da Válvula Hidráulica de Esvaziamento.

Saídas Digitais [0;24]Vdc:

- I1 - Botão de Pressão "Start";
- I2 - Sensor de Limite Esquerdo;
- I3 - Indicador de Nível Esquerdo;
- I4 - Sensor de Limite Direito;
- I5 - Indicador de Nível Direito;
- I6 - Botão de Pressão "Stop";
- I7 - Interruptor para funcionamento em modo Automático;
- I8 - Interruptor para funcionamento em modo Manual.

Para uma melhor compreensão do sistema é necessário descrever as etapas do processo. Para dar início ao mesmo é necessário pressionar o botão "Start", depois disso o carrinho percorre o tapete até se encontrar na zona de enchimento, indicada pelo Sensor de Limite Direito e é cheio até o Indicador de Nível Direito sinalizar que este já se encontra a máxima capacidade. O enchimento é feito através da abertura da válvula de enchimento. Posteriormente, o carrinho percorre o tapete até chegar à zona de esvaziamento, indicada pelo Sensor de Limite Esquerdo. Na etapa seguinte, a válvula hidráulica de esvaziamento é aberta até o Indicador de Nível Esquerdo sinalizar que o carrinho já se encontra vazio.

Em modo automático, o processo continua até existir uma interrupção forçada. Em modo manual, é necessário indicar ao carrinho para se mover na direção indicada e isto é feito através de uma botoneira externa ao processo.

3.2.2.2 PLC Siemens S7-1200

O autômato utilizado para o controlo deste processo é o Siemens S7-1200 CPU 1214C, com referência 6ES7214-1BE30-0XB0, que se encontra representado na Figura 3.3. Este é alimentado a uma tensão de 230 Vac, incorpora quatorze entradas e dez saídas digitais de 24 Vdc e duas entradas analógicas que funcionam numa gama de [0-10] V. Tem ainda uma porta Profinet para comunicação.

As entradas e saídas do controlador são representadas da seguinte forma:

- DI - Variável de Entrada Digital;
- DQ - Variável de Saída Digital;
- AI - Variável de Entrada Analógica;
- AQ - Variável de Saída Analógica.

De referir que as saídas analógicas só podem ser adicionadas com a incorporação do módulo respetivo representado no sub-capítulo seguinte.

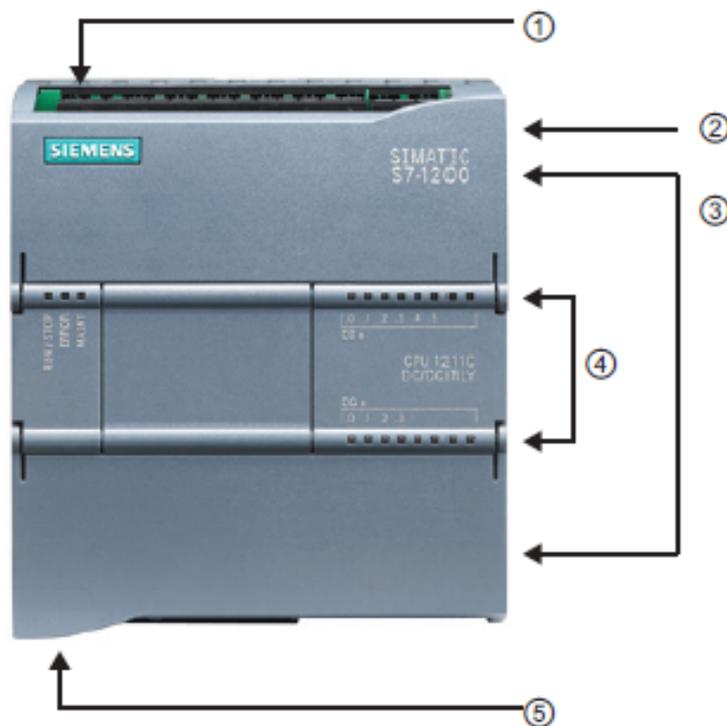


Figura 3.3: Autômato Siemens S7-1200 [32].

Os componentes físicos do controlador são:

- 1 - Conector de Alimentação;
- 2 - Ranhura para Cartão de Memória;
- 3 - Conectores Amovíveis para Ligações;
- 4 - LEDs de estado;
- 5 - Conector de Profinet.

As linguagens de programação suportadas pelo autómato são:

Programming language	S7-1200
Ladder diagram (LAD)	yes
Function block diagram (FBD)	yes
Structured Control Language (SCL)	yes
Graph	no
Statement list (STL)	no

Figura 3.4: Linguagens de Programação suportados pelo Siemens S7-1200 [1].

A linguagem utilizado foi o Ladder Diagram e o software foi o TIA Portal V 13. A descrição deste é feita no sub-capítulo 3.3.2.

Os endereços de ligações de entradas e saídas do PLC e as suas respetivas descrições e características relativamente ao processo estão representados nas Figuras 3.5, 3.6 e 3.7:

Entradas Digitais

Entradas Digitais	Sigla	Descrição	Tipo de Contacto	Endereço
DI a.0	B_EM	Emergency Stop Button / Botão de Paragem de Emergência	B/P N/F	%I0.0
DI a.1	I1_PS	Pushbutton Start / Botão de Pressão "Start"	B/P N/A	%I0.1
DI a.2	I2_L	Limit Switch Left / Sensor de Limite Esquerdo	N/F	%I0.2
DI a.3	I3_IL	Level Indicator Left / Indicador de Nível Esquerdo	N/A	%I0.3
DI a.4	I4_R	Limit Switch Right / Sensor de Limite Direito	N/F	%I0.4
DI a.5	I5_IR	Level Indicator Right / Indicador de Nível Direito	N/A	%I0.5
DI a.6	I6_ST	Pushbutton Stop / Botão de Pressão "Stop"	N/F	%I0.6
DI a.7	I7_SA	Switch Automatic / Interruptor para modo Automático	Interruptor	%I0.7
DI b.0	I8_SM	Stepping Mode / Interruptor para modo Manual	Interruptor	%I1.0
DI b.1	START	Button to start the trolley / Botão para mover o carrinho	B/P N/A	%I1.1
DI b.2	B_LEFT	Button to move trolley left / Botão para mover carrinho para a esquerda	B/P N/A	%I1.2
DI b.3	B_RIGHT	Button to move trolley right / Botão para mover carrinho para a direita	B/P N/A	%I1.3
DI b.4	-	-	-	%I1.4
DI b.5	-	-	-	%I1.5

Figura 3.5: Entradas Digitais do PLC e descrição das mesmas.

Saídas Digitais

Saídas Digitais	Sigla	Descrição	Tipo de Contacto	Endereço
DQ a.0	ON_OFF_M	ON/OFF Motor / Ligar/Desligar Motor	B/P N/F	%Q0.0
DQ a.1	Q1_R	Motor Right / Motor para a Direita	B/P N/A	%Q0.1
DQ a.2	Q2_L	Motor Left / Motor para a Esquerda	N/F	%Q0.2
DQ a.3	Q3_FV	Filling Valve / Abertura da Válvula de Enchimento	N/A	%Q0.3
DQ a.4	Q4_EV	Hydraulic Emptying Valve / Abertura da Válvula Hidráulica de Esvaziamento	N/F	%Q0.4
DQ a.5	S_S	Sound Signal / Sinal Sonoro de Alarme	N/A	%Q0.5
DQ a.6	L_S	Light Signal / Sinal Luminoso de Aviso	N/F	%Q0.6
DQ a.7	R_M	Reverse Motor / Inverter Rotação do Motor	Interruptor	%Q0.7
DQ b.0	-	-	-	%Q1.0
DQ b.1	-	-	-	%Q1.1

Figura 3.6: Saídas Digitais do PLC e descrição das mesmas.

De notar que as siglas representam acrónimos das descrições das entradas e saídas em inglês. Os tipos de contactos são definidos por B/P - Botão de Pressão -, N/F - Normalmente Fechado - e N/A - Normalmente Aberto. Os sinais digitais (entradas e saídas) representados nas Figuras 3.5 e 3.6 são variáveis do tipo booleano, com capacidade de 1 bit, ou seja, apenas assumem o valor de 0 ou 1. As entradas DI b.4 e DI b.5 e as saídas DQ b.0 e DQ b.1 não foram utilizadas para possibilitar adições de funcionalidades e equipamentos em possíveis trabalhos futuros.

Saída Analógica

Saída Analógica	Sigla	Descrição	Endereço
AQ a.0	A_O_S	Analog Output Speed / Velocidade de Saída Analógica	%QW80

Figura 3.7: Saída Analógica do PLC e descrição da mesma.

O sinal de saída analógico representado na Figura 3.7 é uma variável do tipo Word, ou seja tem uma capacidade de 8 bits, ou 1 byte, o necessário para debitar valores na gama de trabalho do controlador, [0-10] V e tem endereço %QW80 pois é o definido em memória pelo fabricante para este mesmo efeito.

3.2.2.3 Módulo de Saídas Analógicas

Para possibilitar um controlo mais dinâmico do motor, através do variador, ou seja, do processo, foi necessária a incorporação de um módulo de saídas analógicas. Este, que se encontra representado na Figura 3.8, permite adicionar o controlo de frequência, ou velocidade, além do controlo digital de ligar/desligar e de inverter o sentido de rotação.



Figura 3.8: Módulo de Saídas Analógicas Siemens [22].

O módulo inclui apenas uma saída analógica que opera na gama de [0-10] V e a sua referência é 6ES7232-4HA30-0XB0.

3.2.2.4 Consola HMI Siemens SIMATIC KTP600

Um painel de operação, ou HMI, foi instalado e programado para permitir uma visualização e controlo aos alunos, sem a necessidade de modificar a programação lógica do PLC. Assim, é possível ter uma leitura do estado em que se encontra o processo e fazer o controlo da velocidade do motor através do seu ecrã tátil. A HMI foi programada, também, com o software TIA Portal V 13.



Figura 3.9: Siemens Simatic HMI KTP600 [21].

O equipamento, representado na Figura 3.9, é alimentado a uma tensão de 24 Vdc, tem a descrição SIMATIC HMI KTP600 Basic Color PN, tem 6 polegadas de display, 256 cores, uma interface PROFINET e referência 6AV6647-0AD11-3AX0.

3.2.2.5 Módulo Switch Siemens

Foi incorporado um módulo switch na instalação para possibilitar aos alunos uma melhor acessibilidade ao processo, utilizando um computador pessoal para se ligarem diretamente a todos os equipamentos que o permitam, sem a necessidade de desligar cabos/conexões do processo.



Figura 3.10: Módulo Switch Siemens para conexões Profinet [14].

O módulo, representado na Figura 3.10, é alimentado a uma tensão de 24 Vdc, tem a descrição Compact Switch Module CSM 1277, tem 4 portas RJ45, uma velocidade de transferência de 100 Mbits/s e referência 6GK7277-1AA00-0AA0.

3.2.2.6 Motor Siemens

Para uma simulação mais realista do movimento do carrinho no tapete do processo industrial, foi introduzido um motor ao protótipo de ensino, que se encontra representado na Figura 3.11. O controlo deste é feito através de um variador de velocidade, possibilitando ligar e desligar o motor, inverter o sentido de rotação (representado o movimento para a direita e para a esquerda do carrinho) e controlar a velocidade de rotação do mesmo.

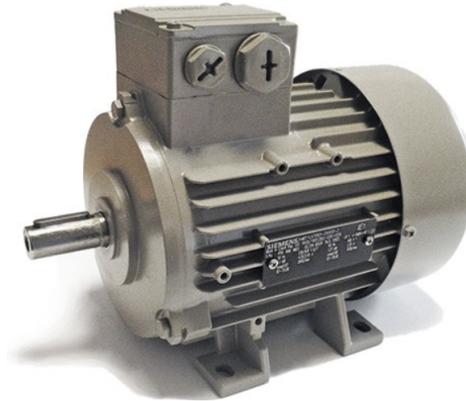


Figura 3.11: Motor Siemens [14].

O motor tem as características representadas na Figura 3.12, para uma frequência de 50 Hz, pois é a frequência da rede elétrica em Portugal, e referência 1LA7060-4AB10.

Electrical data:						
Rated voltage :	(1) 230 VD/400 VY, 50 Hz, 460 VY, 60 Hz					
Frequency :	50 Hz		60 Hz			
Rated power :	0.12 kW		0.14 kW			
Rated speed :	1350 1/ min		1650 1/ min			
Rated torque :	0.8 Nm		0.8 Nm			
Rated current (IE) :	VD	VY	VY			
	0.73 A	0.42 A	0.43 A			
Starting / rated current :	2.8		3.1			
Breakdown / rated torque :	2.0		2.1			
Starting / rated torque :	1.9		2.0			
	4/4	3/4	2/4	4/4	3/4	2/4
Efficiency %	53.6%	52.1%	47.6%	56.0%	54.5%	50.0%
Power factor :	0.75	0.69	0.56	0.74	0.68	0.55
Efficiency class :	-		-			

Figura 3.12: Dados para a instalação e utilização do motor [2].

3.2.2.7 Variador de Velocidade Siemens Sinamics G110

Como referido no sub-capítulo 3.2.2.6, o variador de velocidade Siemens Sinamics G110 permite o controlo de velocidade/frequência do motor referenciado na Figura 3.11. Este equipamento é controlado através de um microprocessador que permite um controlo praticamente instantâneo do motor e incorpora funções de proteção para o variador e motor. Na Figura 3.13 verifica-se que existe uma patilha para a escolha de frequência de trabalho do variador, o que permite uma utilização universal do mesmo.



Figura 3.13: Variador de Velocidade G110 [46].

Na Figura 3.14 está representado o Painel de Operação amovível. Este permite a visualização da frequência a que o motor está a operar e um controlo manual do mesmo. O Painel de Operação foi utilizado para fazer o comissionamento rápido do sistema motor/variador, através dos parâmetros específicos do motor, representados na Figura 3.12.



Figura 3.14: BOP - Basic Operator Panel / Painel de Operação [46].

O comissionamento é feito através da configuração de parâmetros do variador que estão correlacionados com as características do motor, como a região de operação (a frequência de trabalho), a tensão, corrente e potência do motor, o setpoint de frequência, o tempo de aceleração e desaceleração, entre outros. A ordem em que este que ser feito está representada no Fluxograma da Figura 3.15.

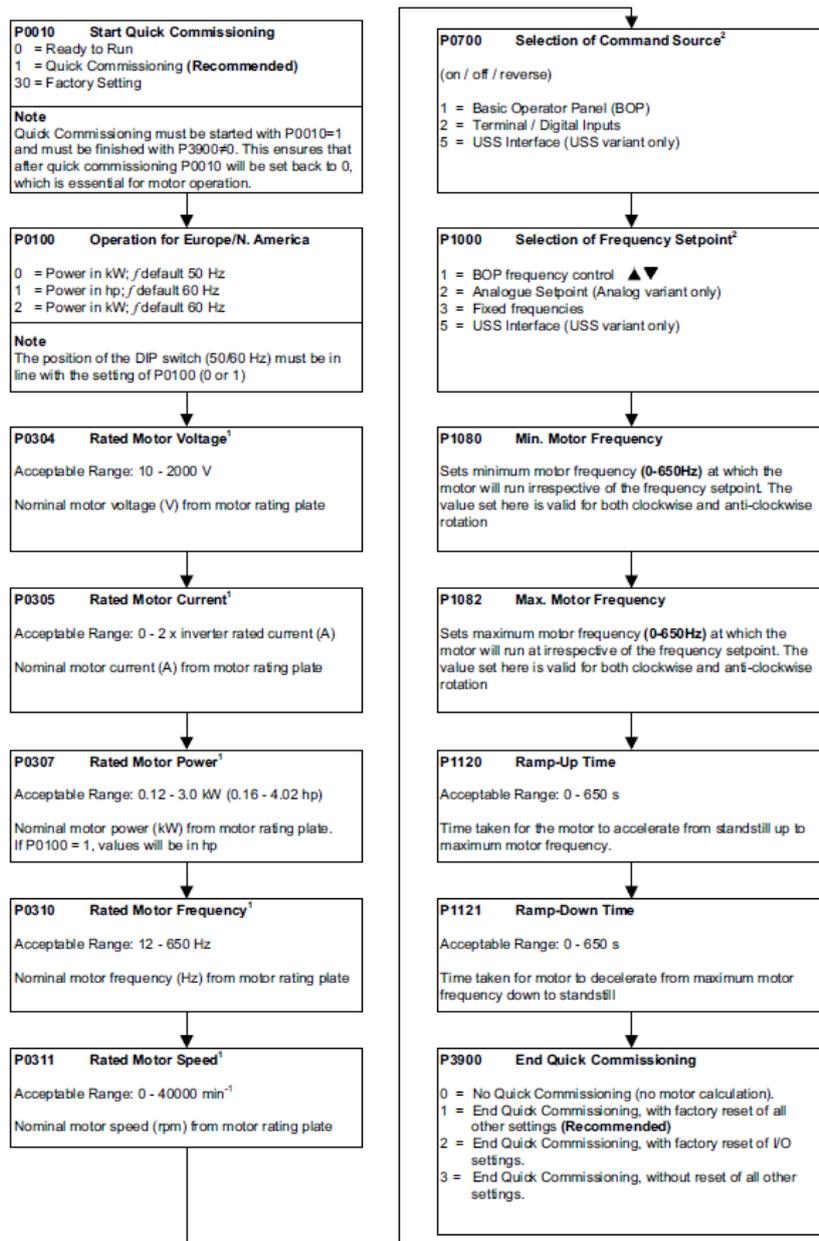


Figura 3.15: Fluxograma de Comissionamento do Sistema Variador/Motor [46].

Na Figura 3.16 encontra-se representado o diagrama de funcionamento do variador Siemens Sinamics G110, sendo que a variante utilizada do mesmo é a analógica.

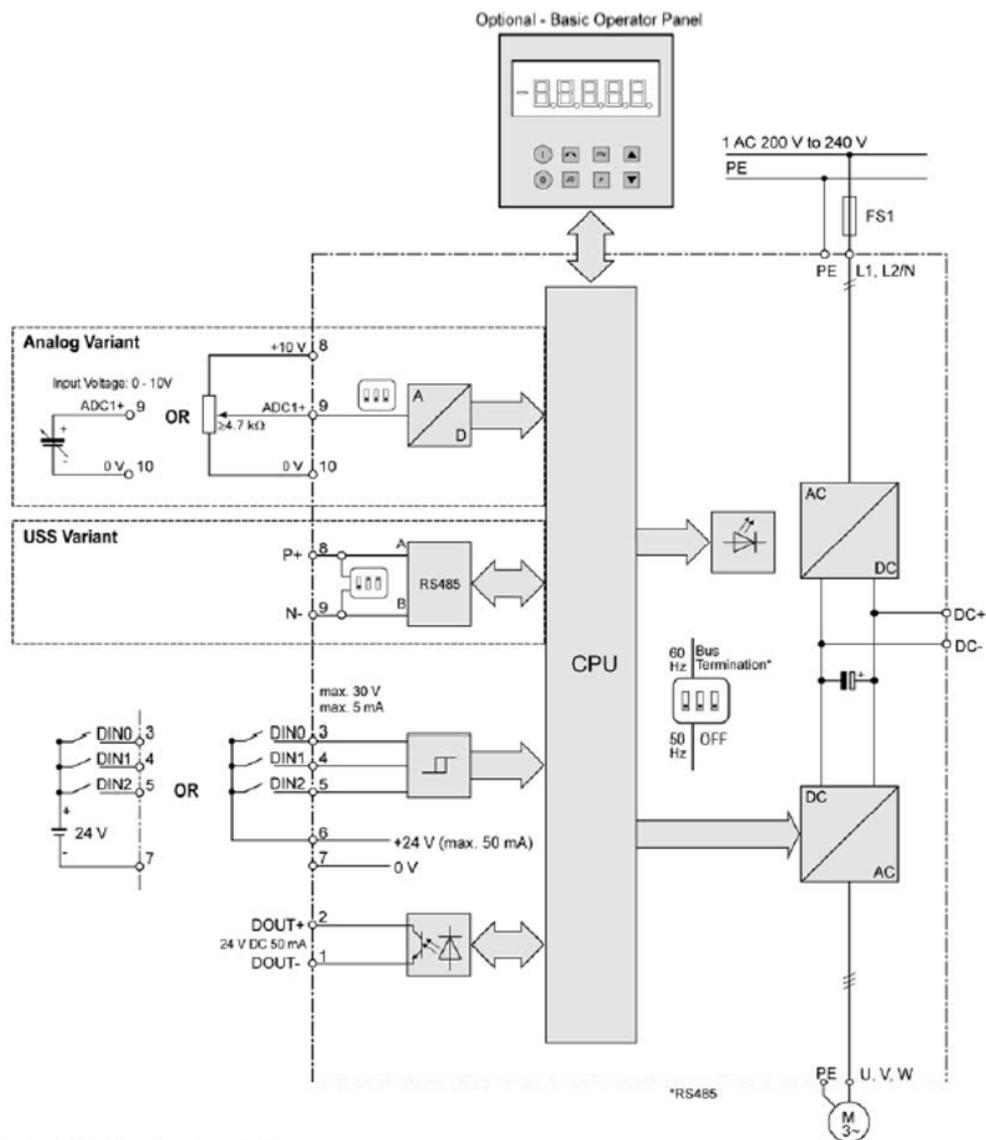


Figura 3.16: Diagrama de Funcionamento do Variador [46].

Como é possível verificar, o variador tem 10 entradas. Neste projeto foram utilizadas apenas 5 delas, a entrada 3 para ligar e desligar o motor, a entrada 4 para alternar o sentido de rotação do mesmo, a entrada 7 para fornecer a tensão base (0V) para as entradas digitais anteriormente referidas e as entradas 9 e 10 para fornecer a velocidade ou frequência de rotação do motor através de sinal analógica, ou seja, de uma tensão a variar na gama de [0-10] V.

3.2.2.8 Botoneiras

Como referido anteriormente, foram adicionadas botoneiras (botoneira de marcha, representada na Figura 3.17, e botoneira de emergência, representada na Figura 3.18) para permitir um controlo manual aos alunos, assumindo que o interruptor se encontra no modo manual e não automático, dando a ênfase requerida ao ambiente industrial. Estas botoneiras incorporam contactos do tipo normalmente aberto.

A botoneira de marcha permite 3 diferentes funcionalidades, iniciar ou parar a marcha, movimentar o carrinho para a direita ou movimentar o carrinho para a esquerda, assumindo que este não se encontra na estação de enchimento ou esvaziamento, nesse caso o processo prossegue até que seja finalizado o objetivo de cada estação.



Figura 3.17: Botoneira de marcha Telemecanique [9].

A botoneira de emergência apresenta uma única funcionalidade, a de parar completamente o processo, simulando assim um alarme. Ao premir este botão, todo o processo é imediatamente parado e é acionado um alarme denominado para este processo de "buzzer", simulando assim uma paragem ao nível fabril ou industrial. Apenas "corrigindo" a falha, correção essa representada pela pressão do botão de marcha, é que o processo volta a prosseguir a partir do estado em que se encontrava.



Figura 3.18: Botão de Emergência Telemecanique [17].

3.2.2.9 Equipamentos para Sinal de Alarme

Para este protótipo de ensino estar em concordância com um ambiente industrial, foram adicionados dois elementos que representam tipos de avisos e/ou alarmes presentes em fábrica. Na Figura 3.19, é possível verificar um sinal luminoso laranja de emergência, utilizado para representar as sirenes que se podem ouvir num ambiente fabril em caso de paragem de emergência, dado a denominação de "Buzzer". Na Figura 3.20, encontra-se um LED Verde de indicação de uma fase específica do processo.



Figura 3.19: Sinal Sonoro de Alarme "Buzzer"[10].

Este sinal de alarme apenas é ativado quando o Botão de Emergência é pressionado e só deixa de estar ativo quando se "resolve o alarme", ou seja, se pressiona o botão de marcha.



Figura 3.20: Sinal de Alarme Luminoso - Hager SVN 131 [15].

Seja para o processo de enchimento, ou de esvaziamento do carrinho, é acionado um alarme luminoso, representado pelo equipamento Hager SVN 131, que ilumina um LED Verde. Este aviso é emitido para se ter em atenção que está a ocorrer um processo potencialmente perigoso para qualquer operador que esteja presente.

3.2.2.10 Fonte de Alimentação

Para a alimentação de todos os equipamentos do protótipo de ensino, à exceção do controlador, variador e motor, é necessária uma tensão de 24 Vdc. Esta tensão é fornecida através da fonte de alimentação PULS 24 Vdc, representada na Figura 3.21 e a mesma é alimentada a 230 Vac.



Figura 3.21: Fonte de Alimentação PULS 24 Vdc [33].

3.2.2.11 Funcionamento do Processo

Para o desenvolvimento deste protótipo de ensino, foi definido um processo sequencial de funcionamento, também denominado de Grafcet, que se encontra representado na Figura 3.22. As nomenclaturas de cores representadas são as seguintes: as etapas do processo a verde, as entradas digitais do autômato a amarelo, as saídas digitais do autômato a azul e os sinais de alarme "buzzer" e luminoso a creme.

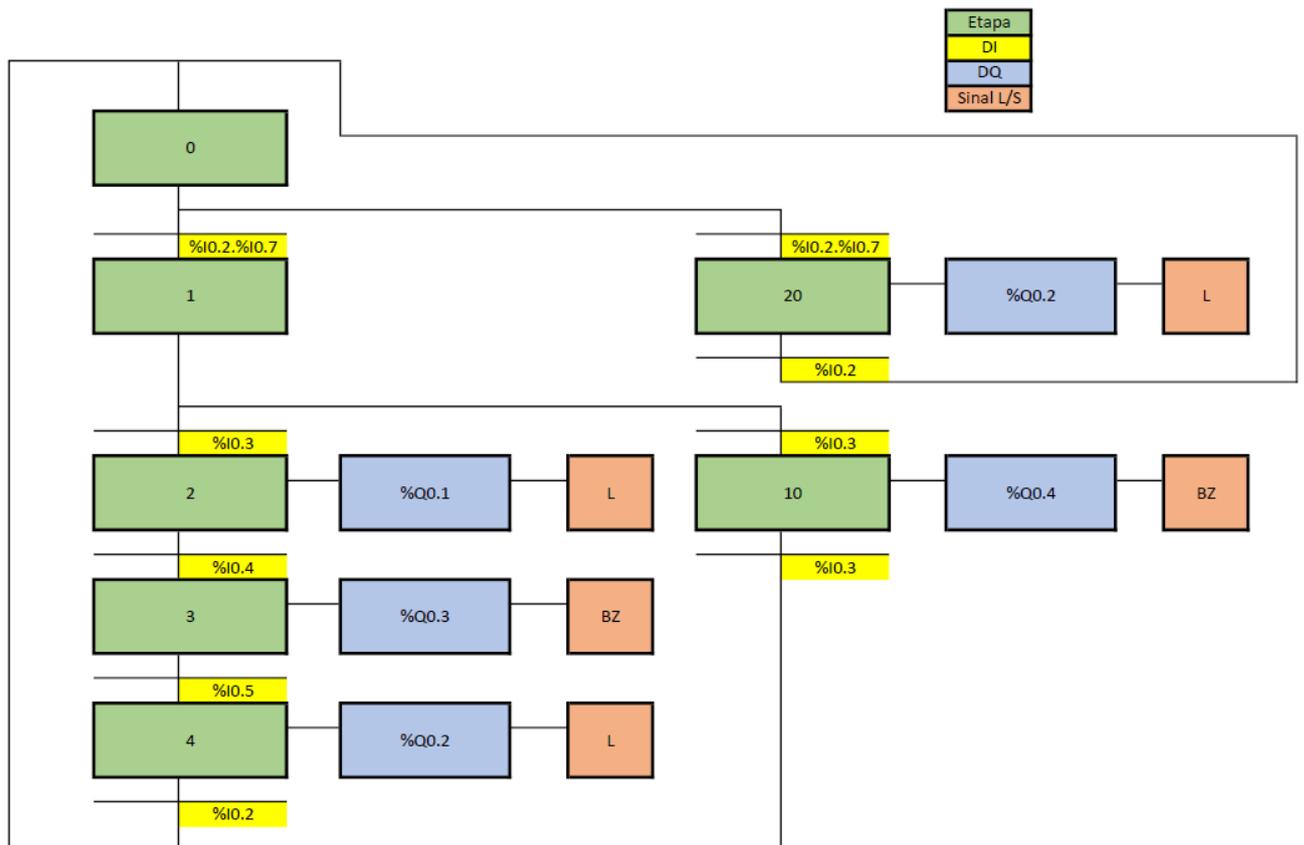


Figura 3.22: Grafcet do Processo.

3.3 Projeto de Software

3.3.1 Necessidades a Nível de Software

A programação do autômato foi realizada em duas fases distintas.

Uma primeira fase em que se procedeu à programação lógica do mesmo, de acordo com o funcionamento desejado para o processo, referenciado na Figura 3.22. Tendo em atenção as seguintes funcionalidades:

- No caso de existência de alarme, representado pela pressão do botão de paragem de emergência, o processo deve parar até existir uma validação da "reparação" desse alarme. "Reparação" essa que é feita pressionando o botão de marcha;

- Na ocorrência do processo de esvaziamento, ou no processo de enchimento, processos que podem representar perigos para o operador, o carrinho não poderá ter qualquer tipo de movimento. Estes processos só podem ser interrompidos com a ativação do botão de emergência (alarme) e, posteriormente, só podem retomar depois da falha que gerou o alarme ser reparada;

- O processo deve ter formas distintas de funcionamento, nomeadamente, um modo automático em que o processo continuará em ciclo até ser dada uma ordem em contrário, um modo manual em que o processo só funcionará se for dada uma ordem pelo operador através da botoneira e ainda um controlo de velocidade do motor, permitindo a paragem do mesmo se for necessário, através do painel táctil (HMI);

- A programação do controlador foi feita em linguagem LAD para permitir uma acessível compreensão aos alunos e futuras adições ao sistema de fácil implementação.

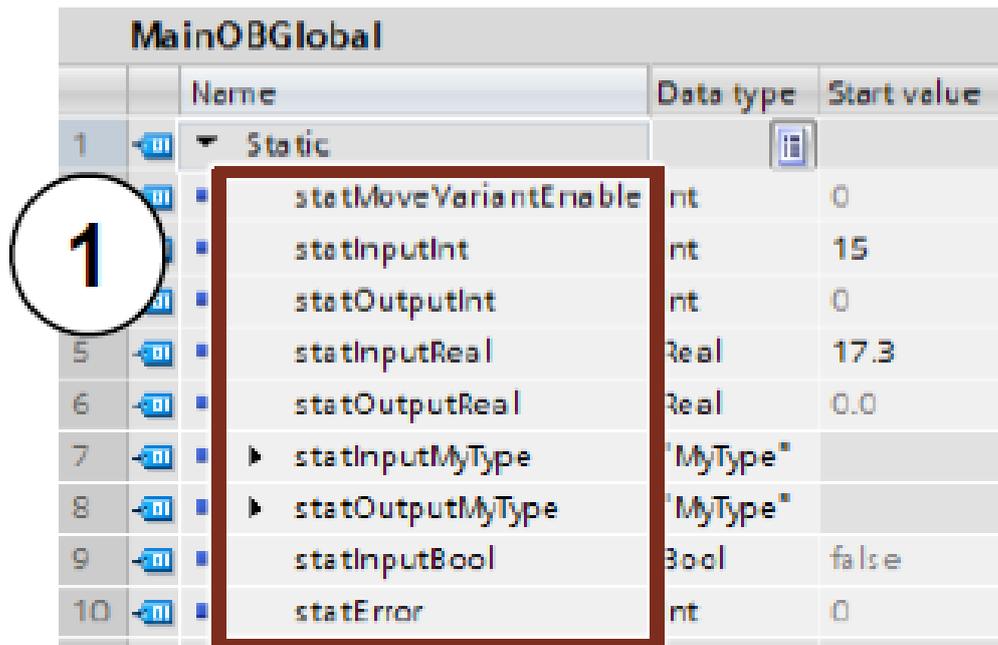
Uma segunda fase de programação visual, em que foram desenvolvidos sinópticos, depois de feita toda a programação lógica, com o intuito de permitir de demonstrar um controlo de sala de comando normalmente presente em grandes instalações industriais, fornecendo ainda outro tipo de controlo ao operador, neste caso o aluno, com a utilização de botões touch screen e visualização gráfica do estado do processo, as variáveis ativas e velocidade em tempo real do motor em funcionamento.

3.3.2 Programação Lógica

3.3.2.1 Definição de Variáveis e Processos

Para uma melhor compreensão da programação do controlador é necessário definir vários conceitos.

Na Figura 3.23 encontra-se representado um conjunto de Tags.



	Name	Data type	Start value
1	Static		
	statMoveVariantEnable	nt	0
	statInputInt	nt	15
	statOutputInt	nt	0
5	statInputReal	Real	17.3
6	statOutputReal	Real	0.0
7	statInputMyType	"MyType"	
8	statOutputMyType	"MyType"	
9	statInputBool	Bool	false
10	statError	nt	0

Figura 3.23: DataBlock - Conjunto de Tags [37].

1) Tags - Variáveis classificadas por um nome identificador ou nome que usam um endereço de memória do controlador. Estas podem ser de vários tipos, booleano, inteiro, real, word, doubleword, entre outros. A um conjunto de Tags dá-se o nome de DataBlock.

Na Figura 3.24 está representada uma Função. Esta pode ser criada pelo utilizador através de programação lógica ou podem ser usadas funções inerentes ao programa TIA Portal.

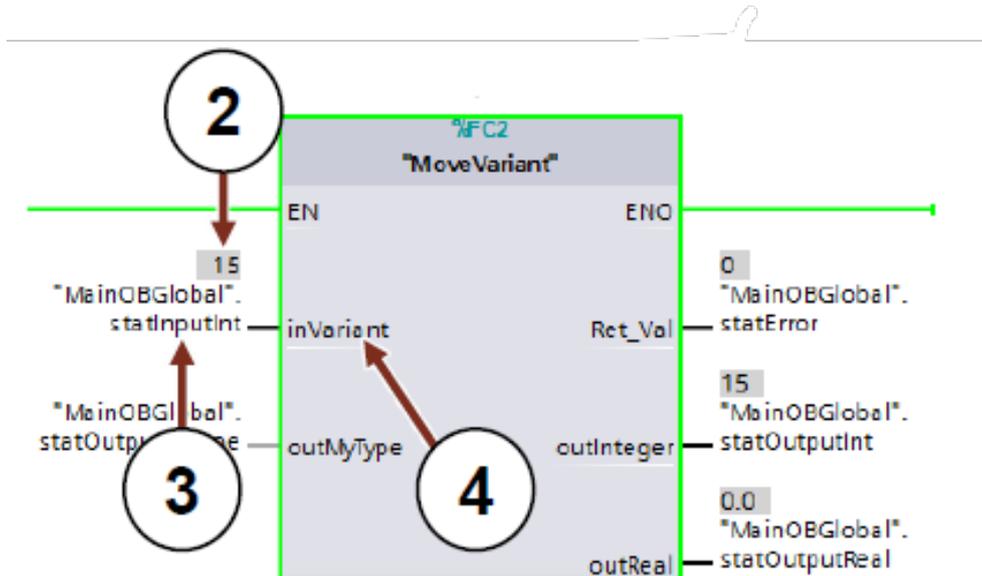


Figura 3.24: Função [37].

2) Valor da Tag - Valor guardado na Tag, que pode ser variável ou constante.

3) Parâmetro efetivo - Tags interconectadas à interface de instruções, funções ou blocos de função.

4) Parâmetro formal - Parâmetro de interface entre instruções, funções ou blocos de função. Estes podem ser do tipo Input, Output ou InOut.

No programa TIA PORTAL existem vários tipos de blocos para estruturar e reutilizar de acordo com a programação desejada. Estes blocos são:

1) Blocos de Organização - OB (Organization Block) - funcionam como interface entre o sistema operativo e o programa e apresentam funções variadas como:

- Comportamento de inicialização do autómato;
- Processamento cíclico do programa;
- Interrupção controlada do processamento do programa;

- Manuseamento de Erros.

Dentro de um programa, podem ser criados vários Blocos de Organização e estes são processados sequencialmente pelo número que lhes foi atribuído. De referir que são chamados pelo sistema operativo do controlador. Na Figura 3.25 encontra-se representada a seleção de diferentes tipos de blocos, em que se seleciona o Bloco de Organização e é possível visualizar uma lista com os vários processos associados ao mesmo.

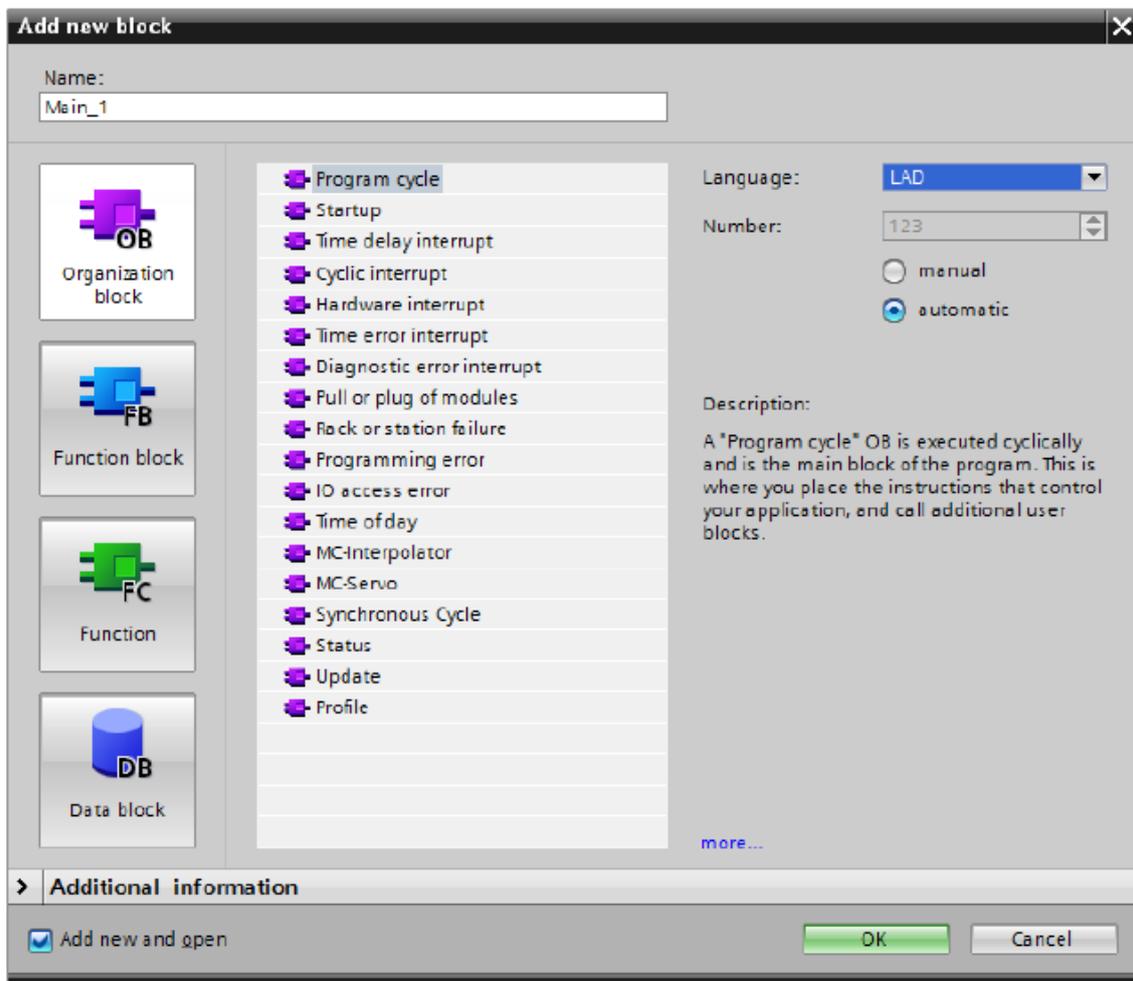


Figura 3.25: Bloco de Organização (OB) e os vários processos associados [37].

2) Funções - FC - blocos sem armazenamento de dados cíclico, portanto os parâmetros presentes nos mesmos só conseguem ser guardados no ciclo seguinte. Estes podem ter diversas variáveis de entrada e saída e a funcionalidade que lhes for atribuída através de programação lógica. Na Figura 3.26 encontra-se um exemplo de uma função com a variável de saída declarada como "return" para esta poder ser diretamente reutilizada em qualquer instância do programa.

The screenshot shows the 'CircularSegmentReturn' function block declaration in the SIMATIC Manager. The table below represents the data shown in the interface:

Name	Data type	Default value	Supervision	Comment
Input				
h	LReal			
r	LReal			
Output				
InOut				
Temp				
Constant				
Return				
CircularSegmentReturn	LReal			

Below the table, the function declaration code is shown:

```
1 #CircularSegmentReturn := SQR(#r) * ACOS(1-#h/#r) - SQR(2*#r*#h-SQR(#h)) * (#r-#h);
```

Figura 3.26: Função e declaração de variável de saída[37].

3) Blocos de Funções - FB (Function Blocks) - blocos com armazenamento cíclico de dados, em que os valores são permanentemente guardados num bloco de dados - DB (Data Block). Um exemplo de um bloco de funções é representado na Figura 3.27, em que se verifica a chamada de um bloco de dados associado ao bloco de funções.

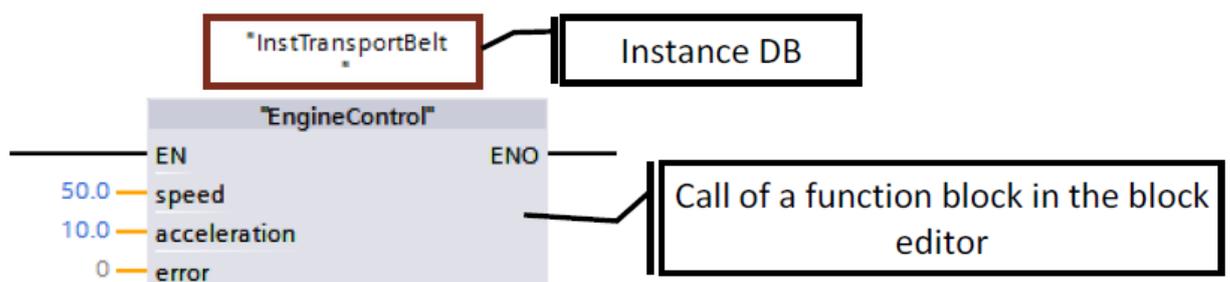


Figura 3.27: Bloco de Função com chamada de um bloco de dados associado a este[37].

4) Instâncias - chamada de um bloco de função. Os dados com que o bloco de função está a funcionar são guardados num bloco de dados, estes são sempre criados instantaneamente de acordo com as especificações da interface do bloco de função, consequentemente não podem ser alteradas posteriormente e não necessitam de ser criadas manualmente. A estrutura de um bloco de dados é especificada no bloco de função correspondente e só pode ser alterada aí. Na Figura 3.28 observa-se a estrutura da interface de um bloco de função e as variáveis correspondentes ao bloco de dados devidamente assinaladas.

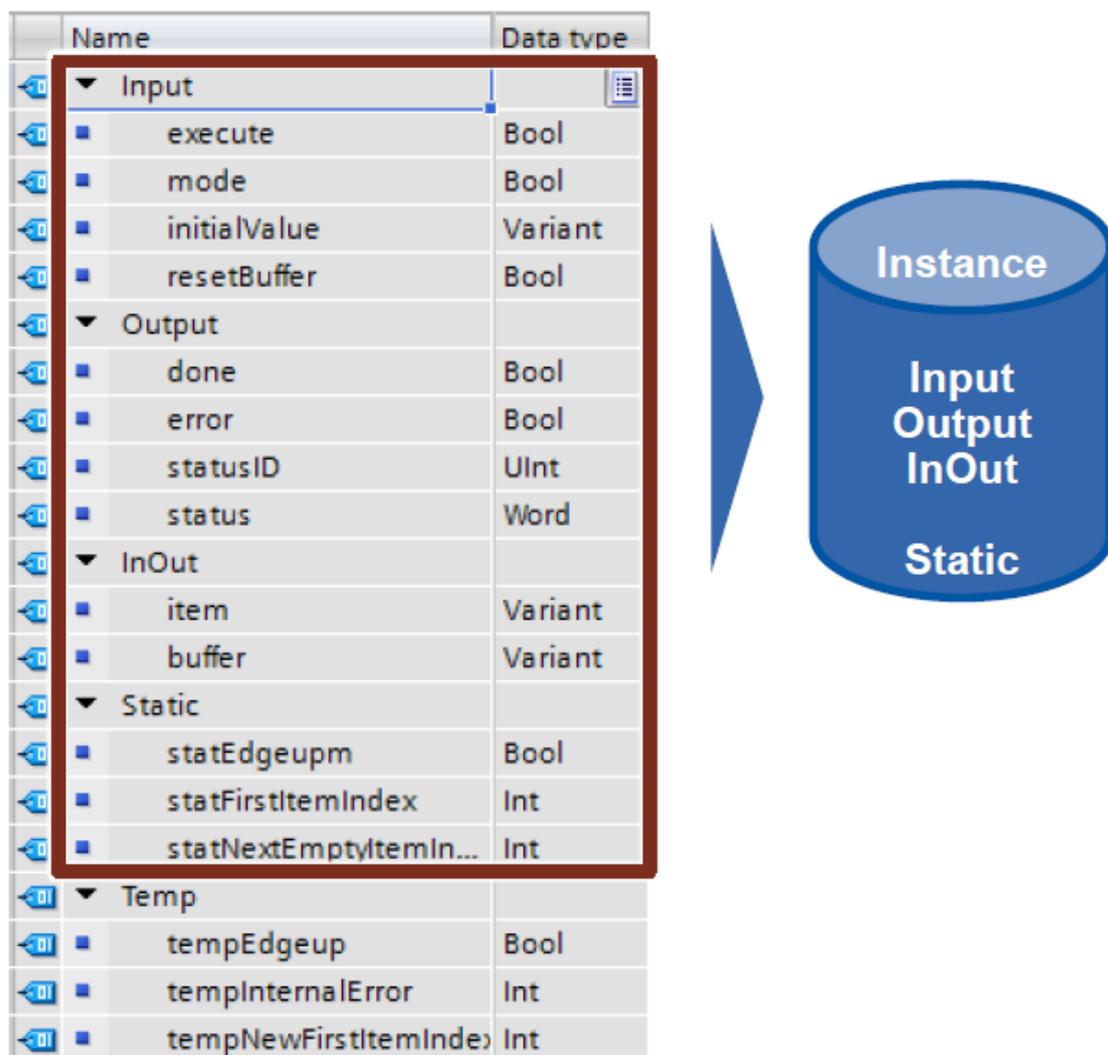


Figura 3.28: Estrutura da interface de um bloco de função [37].

A instância do bloco de dados consiste em interfaces de variáveis do tipo input, output, InOut e estático (invariável para todo o processo), armazenadas em memória permanente. Relativamente às variáveis do tipo temporário, estas são armazenadas em memória volátil, ou seja, necessitam de ser inicializadas em cada novo ciclo.

3.3.2.2 Variáveis e Processos Utilizados

Em concordância com o apresentado no sub-capítulo anterior foram definidas variáveis e funções necessárias para o desejável funcionamento do processo. De seguida, apresentam-se essas funções e variáveis correspondentes e a sua finalidade. As funções implementadas e as variáveis digitais respetivas são:

- 1) ON/OFF - Ligar e Desligar o Motor;

	Descrição	Endereço
Entradas	Emergency Stop Button / Botão de Paragem de Emergência	%I0.0
	Pushbutton Start / Botão de Pressão "Start"	%I0.1
	Pushbutton Stop/ Botão de Pressão "Stop"	%I0.6
	Switch Automatic / Interruptor para modo Automático	%I0.7
	Stepping Mode / Interruptor para modo Manual	%I1.0
	Button to start the trolley / Botão para mover o carrinho	%I1.1
Saída	ON/OFF Motor / Ligar/Desligar Motor	%Q0.0

Figura 3.29: Variáveis da Função ON/OFF.

Com a observação da Figura 3.30, verifica-se que o funcionamento do motor depende de vários fatores. Se for acionado o botão de paragem de emergência (%I0.0), ou o botão de paragem de marcha (%I0.6), este deixa imediatamente de funcionar. Para se dar início à marcha são necessárias certas condições, o botão de pressão "Start" tem que ser pressionado e para retomar a marcha, o botão para mover o carrinho também tem que ser pressionado. A saída ligar/desligar motor também é entrada da função para ser possível a realimentação da mesma, tendo uma marcha contínua em modo automático.

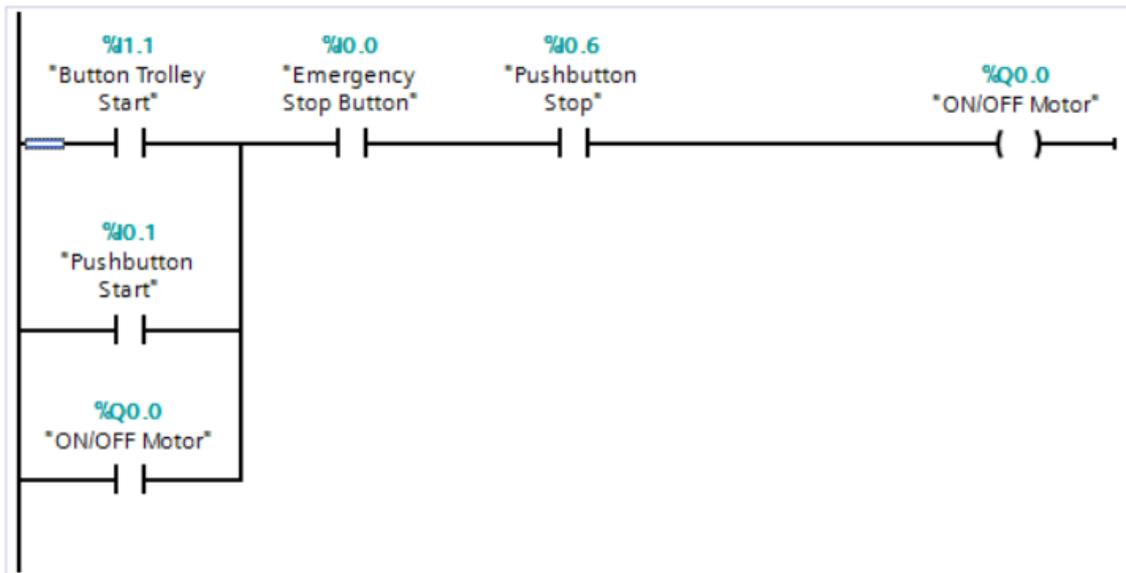


Figura 3.30: Programação Lógica da Função ON/OFF.

Para exemplificar a visualização de uma função em bloco, apresenta-se na Figura 3.31 a função de ligar/desligar o motor.

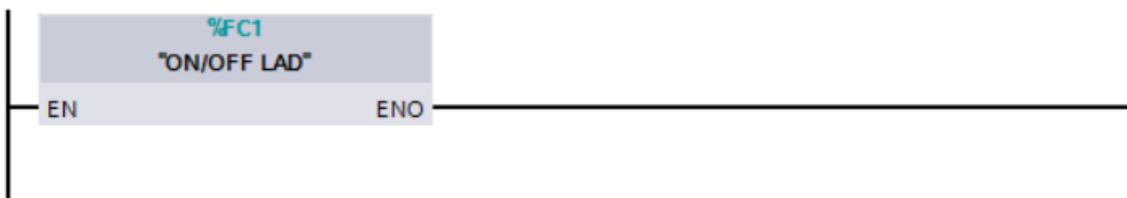


Figura 3.31: Bloco de Função ON/OFF.

2) Move Right - Mover o Carrinho para a Direita

	Descrição	Endereço
Entradas	Level Indicator Left / Indicador de Nível Esquerdo	%I0.3
	Limit Switch Right / Sensor de Limite Direito	%I0.4
	Button to move trolley right / Botão para mover carrinho para a direita	%I1.2
	Hydraulic Emptying Valve / Abertura da Válvula Hidráulica de Esvaziamento	%Q0.4
Saídas	Motor Right / Motor para a Direita	%Q0.1
	Reverse Motor / Inverter Rotação do Motor	%Q0.7

Figura 3.32: Variáveis da Função Mover para a Direita.

A programação da movimentação do carrinho para a direita observa-se na Figura 3.33 e vai depender do tipo de controlo, se automático (%I0.7) ou manual (neste caso o modo automático negado), sendo que em ambos os casos o carrinho não se move se estiver a ocorrer o processo de esvaziamento (%Q0.4) ou o processo de enchimento (%Q0.3) e se já estiver no fim de curso à direita, ou seja, com o sensor de limite direito (%I0.4) ativado. Para o motor se mover para a direita é necessário acabar o processo de esvaziamento (%Q0.4), sendo que foi adicionado um temporizador de 5s após o esvaziamento para ser possível a visualização no processo físico.

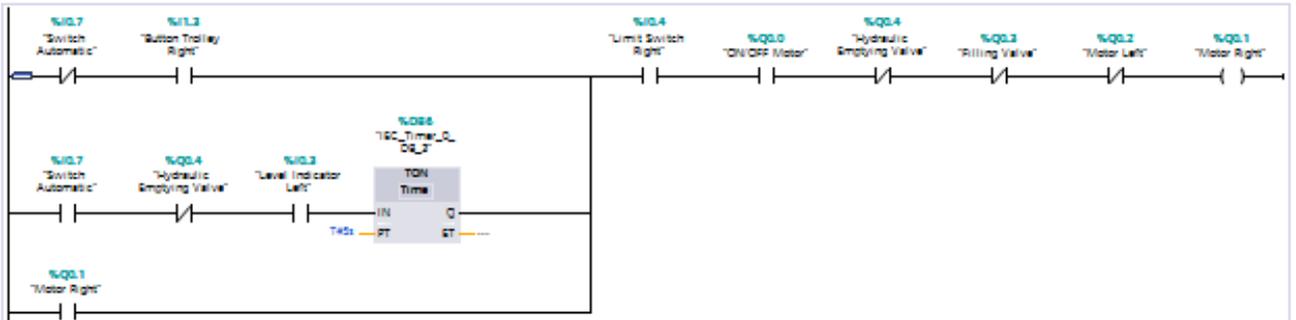


Figura 3.33: Programação Lógica da Função Mover para a Direita.

3) Move Left - Mover o Carrinho para a Esquerda;

	Descrição	Endereço
Entradas	Limit Switch Left / Sensor de Limite Esquerdo	%I0.2
	Level Indicator Right / Indicador de Nível Direito	%I0.5
	Button to move trolley left / Botão para mover carrinho para a esquerda	%I1.1
	Filling Valve / Abertura da Válvula de Enchimento	%Q0.3
Saídas	Motor Left / Motor para a Esquerda	%Q0.2
	Reverse Motor / Inverter Rotação do Motor	%Q0.7

Figura 3.34: Variáveis da Função Mover para a Esquerda.

A programação da movimentação do carrinho para a esquerda observa-se na Figura 3.35 e, semelhante à movimentação para a direita, vai depender do tipo de controlo, se automático (%I0.7) ou manual (neste caso o modo automático negado), sendo que em ambos os casos o carrinho não se move se estiver a ocorrer o processo de esvaziamento (%Q0.4) ou o processo de enchimento (%Q0.3) e se já estiver no fim de curso à esquerda, ou seja, com o sensor de limite esquerdo (%I0.2) ativado. Para o motor se mover para a esquerda é necessário acabar o processo de enchimento (%Q0.3), sendo que foi adicionado um temporizador de 5s após o enchimento para ser possível a visualização no processo físico.

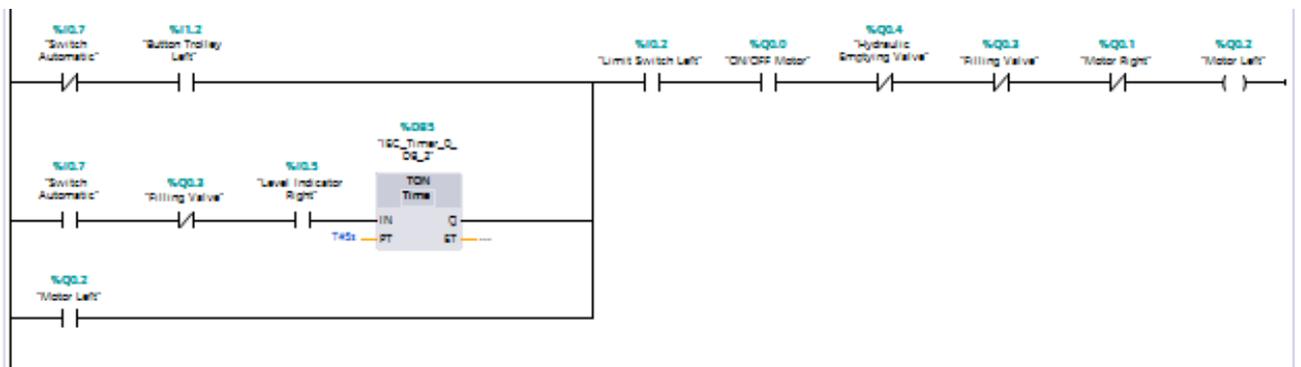


Figura 3.35: Programação Lógica da Função Mover para a Esquerda.

4) Filling Valve - Válvula de Enchimento;

	Descrição	Endereço
Entradas	Limit Switch Right / Sensor de Limite Direito	%I0.4
	Level Indicator Right / Indicador de Nível Direito	%I0.5
Saída	Filling Valve / Abertura da Válvula de Enchimento	%Q0.3

Figura 3.36: Variáveis da Função Abertura da Válvula de Enchimento.

A abertura da válvula de enchimento (%Q0.3) só ocorre quando o sensor de limite direito estiver ativado (%I0.4) e se o indicador de nível direito (%I0.5) não estiver ativo, como é possível verificar na Figura 3.37.

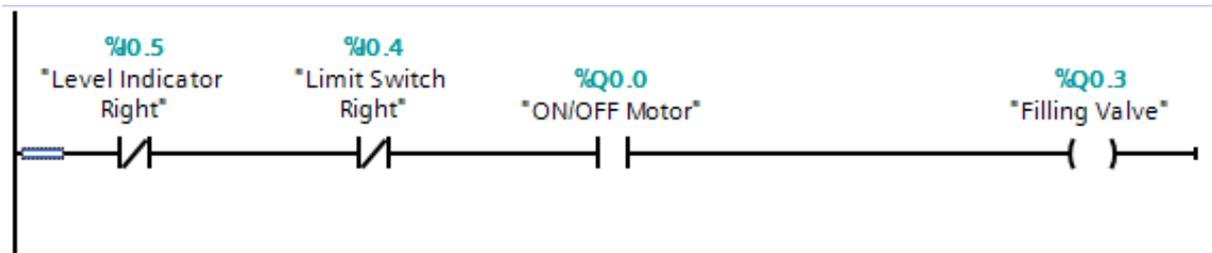


Figura 3.37: Programação Lógica da Função Abertura da Válvula de Enchimento.

5) Hydraulic Emptying Valve - Válvula de Esvaziamento;

	Descrição	Endereço
Entradas	Limit Switch Left / Sensor de Limite Esquerdo	%I0.2
	Level Indicator Left / Indicador de Nível Esquerdo	%I0.3
Saída	Hydraulic Emptying Valve / Abertura da Válvula Hidráulica de Esvaziamento	%Q0.4

Figura 3.38: Variáveis da Função Abertura da Válvula Hidráulica de Esvaziamento.

A abertura da válvula hidráulica de esvaziamento (%Q0.4) só ocorre quando o sensor de limite esquerdo estiver ativado (%I0.2) e se o indicador de nível esquerdo (%I0.3) não estiver ativo, como é possível verificar na Figura 3.37.

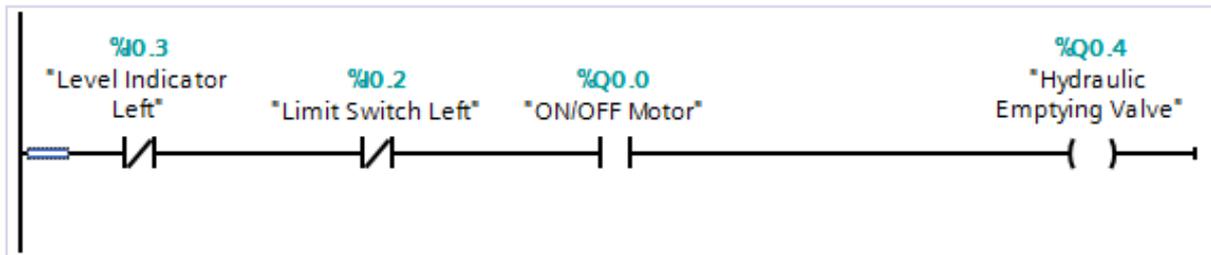


Figura 3.39: Programação Lógica da Função Abertura da Válvula Hidráulica de Esvaziamento.

6) Sound Signal - Sinal Sonoro "Buzzer"de Alarme;

	Descrição	Endereço
Entradas	Emergency Stop Button / Botão de Paragem de Emergência	%I0.0
	Button to start the trolley / Botão para mover o carrinho	%I1.3
Saída	Sound Signal / Sinal Sonoro de Alarme "Buzzer"	%Q0.5

Figura 3.40: Variáveis da Função Sinal Sonoro de Alarme "Buzzer".

O sinal sonoro de alarme "buzzer"(%Q0.5) é ativado quando o botão de paragem de emergência (%I0.0) é pressionado e desativado quando se tiver retificado o erro que deu origem ao alarme através da pressão do botão para mover o carrinho (%I1.3), como se observa na Figura 3.41.

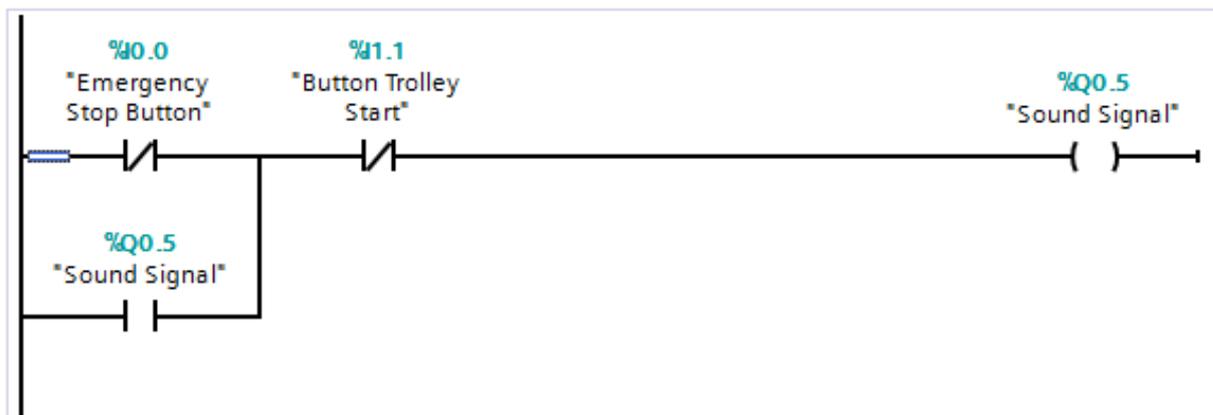


Figura 3.41: Programação Lógica da Função Sinal Sonoro de Alarme "Buzzer".

7) Light Signal - Sinal Luminoso de Alarme;

	Descrição	Endereço
Entradas	Filling Valve / Abertura da Válvula de Enchimento	%Q0.3
	Hydraulic Emptying Valve / Abertura da Válvula Hidráulica de Esvaziamento	%Q0.4
Saída	Light Signal / Sinal Luminoso de Aviso	%Q0.6

Figura 3.42: Variáveis da Função Sinal Luminoso de Alarme.

O sinal luminoso de alarme (%Q0.6) é ativado quando se dá a abertura da válvula de enchimento (%Q0.3), ou a abertura da válvula hidráulica de esvaziamento (%Q0.4), como se observa na Figura 3.43. Foi incorporado um sinal luminoso de alarme pois estes processos, em ambiente industrial, são de algum perigo para os operadores, tornado-se necessário um aviso relativamente os mesmos.



Figura 3.43: Programação Lógica da Função Sinal Luminoso de Alarme.

8) Reverse Motor - Inversão do Sentido de Marcha do Motor.

	Descrição	Endereço
Entrada	Motor Left / Motor para a Esquerda	%Q0.2
Saída	Reverse Motor / Inverter Rotação do Motor	%Q0.7

Figura 3.44: Variáveis da Função Inverter Rotação do Motor.

O sinal para a inversão da rotação do motor (%Q0.7) é ativado quando o sinal de marcha de motor para a esquerda (%Q0.2) é ativado, como se observa na Figura 3.45. Foi estipulado que o sentido horário de rotação do motor representa o movimento do carrinho para a direita e o sentido anti-horário (sentido inverso) representa o movimento do carrinho para a esquerda.

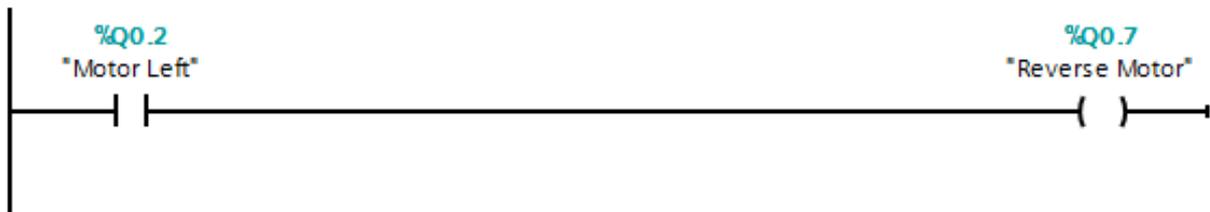


Figura 3.45: Programação Lógica da Função Inverter Rotação do Motor.

Para proporcionar outro tipo de controlo e visualização de processo ao alunos, foram adicionados temporizadores para a contagem de tempo que os processos demoram, sendo estes processos o movimento do carrinho para a direita e para a esquerda, a abertura da válvula hidráulica de esvaziamento e a abertura da válvula de enchimento. A programação lógica destes encontra-se representada na Figura 3.46. É de referir que foi utilizada uma variável de memória de frequência 1 Hz (%M0.5) para contar cada segundo e que o temporizador tem uma contagem máxima de 60 segundos, mas é possível expandir esse tempo com programação lógica adicional.

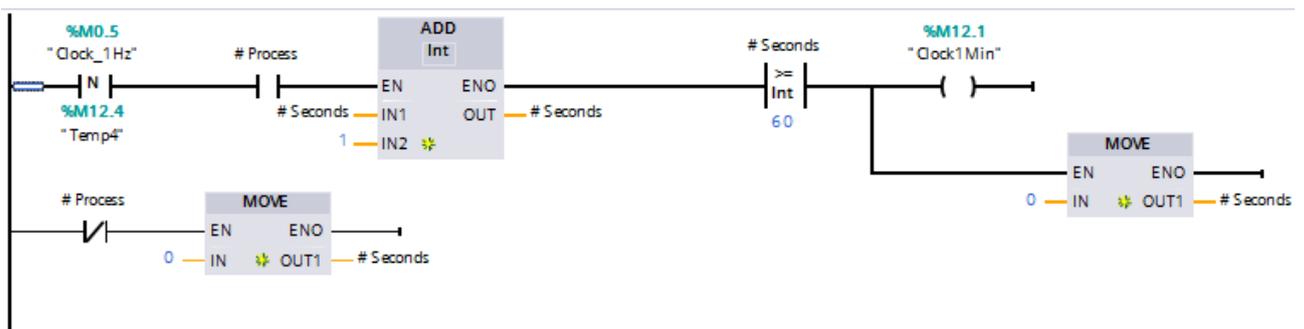


Figura 3.46: Programação Lógica dos Temporizadores.

A leitura e escrita de valores analógicos é feita numa gama de valores definida pelo fabricante [0 - 27648], sendo que, com este modelo de autómato, só é possível trabalhar com valores positivos. Torna-se pois necessário implementar funções para a conversão destes valores nas gamas desejadas. Sendo que o autómato permite leitura e escrita de valores numa gama [0-10] V, o valor 27648 representa o valor máximo da gama, 10 V. A correlação da velocidade do motor (máxima a 50 Hz) com os restantes valores é: $27648 = 10 \text{ V} = 50 \text{ Hz}$.

A Figura 3.47 apresenta, inicialmente, a conversão do valor analógico pretendido, na gama de [0-10] V, para o valor de velocidade do motor, em Hz, posteriormente normalizado.

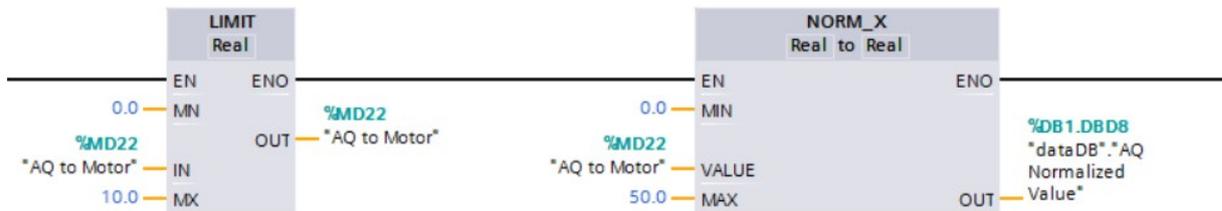


Figura 3.47: Escrita de valores analógicos para o motor - conversão de V para Hz.

Na Figura 3.48 é possível visualizar a conversão do valor definido em Hz para os valores definidos pelo fabricante, em variável do tipo Real e posteriormente, a conversão desse valor Real para Inteiro, para que seja possível a escrita do mesmo.

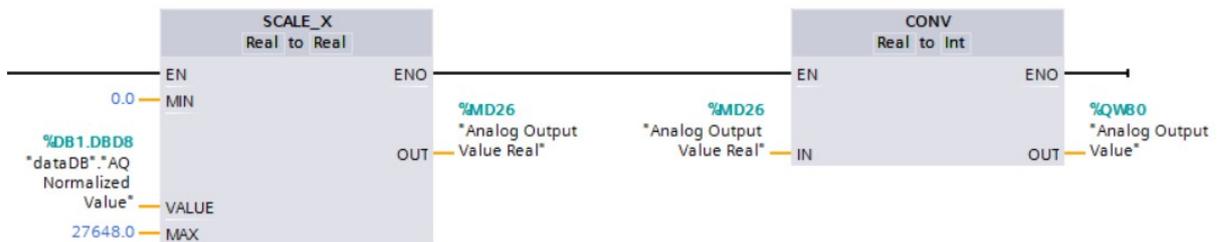


Figura 3.48: Escrita de valores analógicos para o motor - conversão de Hz para valor inteiro até 27648.

A leitura de valores analógicos, representada na Figura 3.49, é feita através da conversão do valor analógico Inteiro lido pelo autômato para Real e posteriormente, da conversão desse valor para Hz.

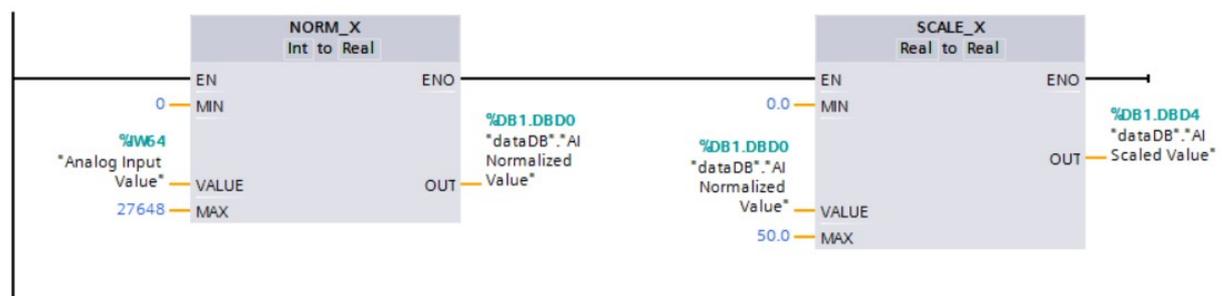


Figura 3.49: Leitura do autômato de valores analógicos (velocidade do motor) - conversão de valor inteiro até 27648 para Hz.

3.3.3 Programação de Sinópticos

Com o intuito de proporcionar uma visualização e controlo do processo semelhante ao que existe nas salas de controlo em ambiente industrial, desenvolveram-se ecrãs com elementos pertinentes a esse mesmo controlo. É possível a troca de ecrãs através dos botões físicos "Fx" da consola HMI.

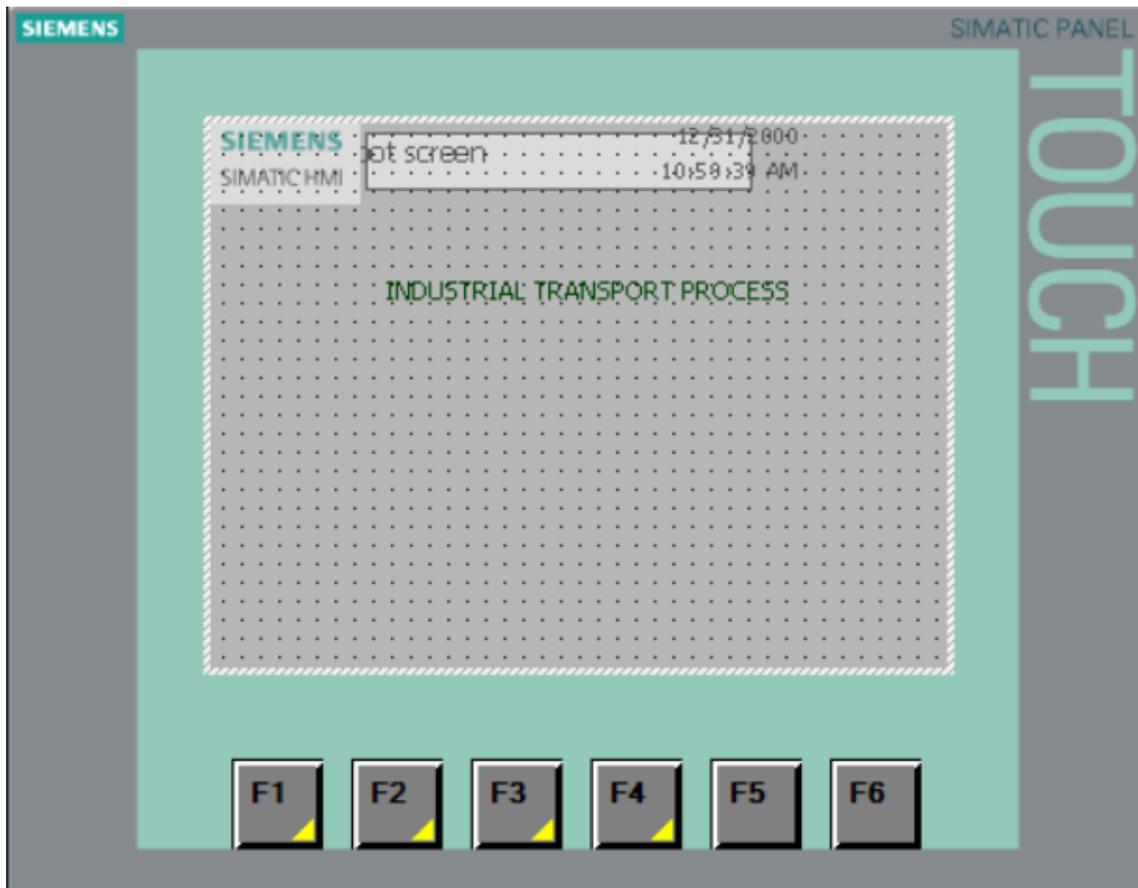


Figura 3.50: Menu inicial de sinópticos.

O ecrã representado na Figura 3.50 ilustra o menu de seleção inicial do processo. O botão referente a este ecrã é o F1.

No ecrã de estado do processo em tempo real, representado na Figura 3.51, é possível verificar o estado de todas as variáveis do processo, o tempo de atuação das mesmas e o sentido de rotação do motor. O botão referente a este ecrã é o F2.

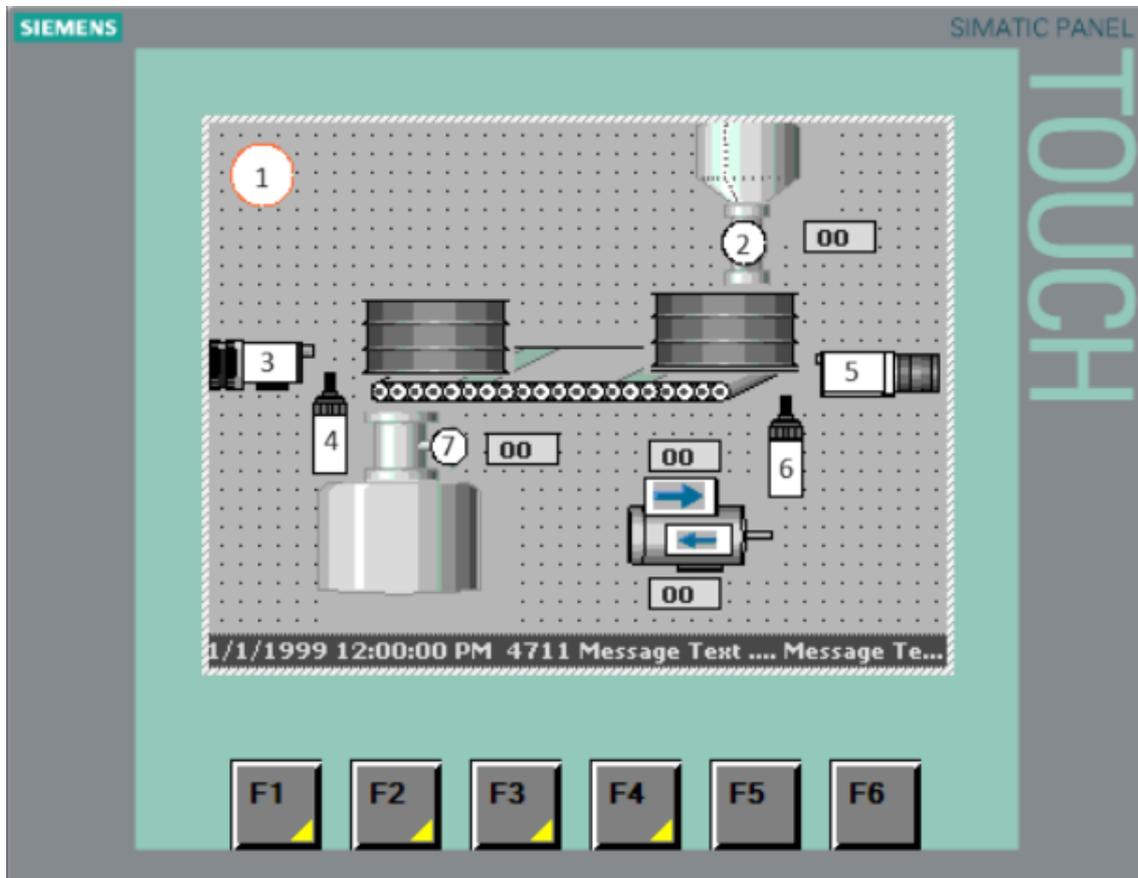


Figura 3.51: Ecrã de estado do processo em tempo real.

As variáveis do processo encontram-se representadas pelos números ilustrados na Figura 3.51 e iluminam-se quando se dá a ativação das mesmas.

As variáveis de processo representadas são:

- 1) Sinal Sonoro de Alarme "Buzzer";
- 2) Sinal Luminoso de Alarme, referente à abertura da válvula de enchimento;
- 3) Sensor de Limite Esquerdo;
- 4) Indicador de Nível Esquerdo;
- 5) Sensor de Limite Direito;

6) Indicador de Nível Direito;

7) Sinal Luminoso de Alarme, referente à abertura da válvula hidráulica de esvaziamento.

É possível ainda verificar os temporizadores referidos no sub-capítulo da anterior da programação lógica e setas a indicar o sentido de rotação do motor e, conseqüentemente, o sentido em que o carrinho se movimenta.

No ecrã de controlo do motor, representado na Figura 3.52, é feito o controlo manual da velocidade do motor (limitado até uma frequência de 10 Hz), através dos botões táteis apresentados no ecrã. É ainda possível visualizar a velocidade do motor no gráfico de barra posicionado do lado esquerdo do ecrã e foi adicionado um outro gráfico do lado direito para a leitura proveniente do sensor de leitura que será incorporado, permitindo um controlo em anel fechado e um fator para corroborar o controlo desejado do motor. O botão referente a este ecrã é o F3.

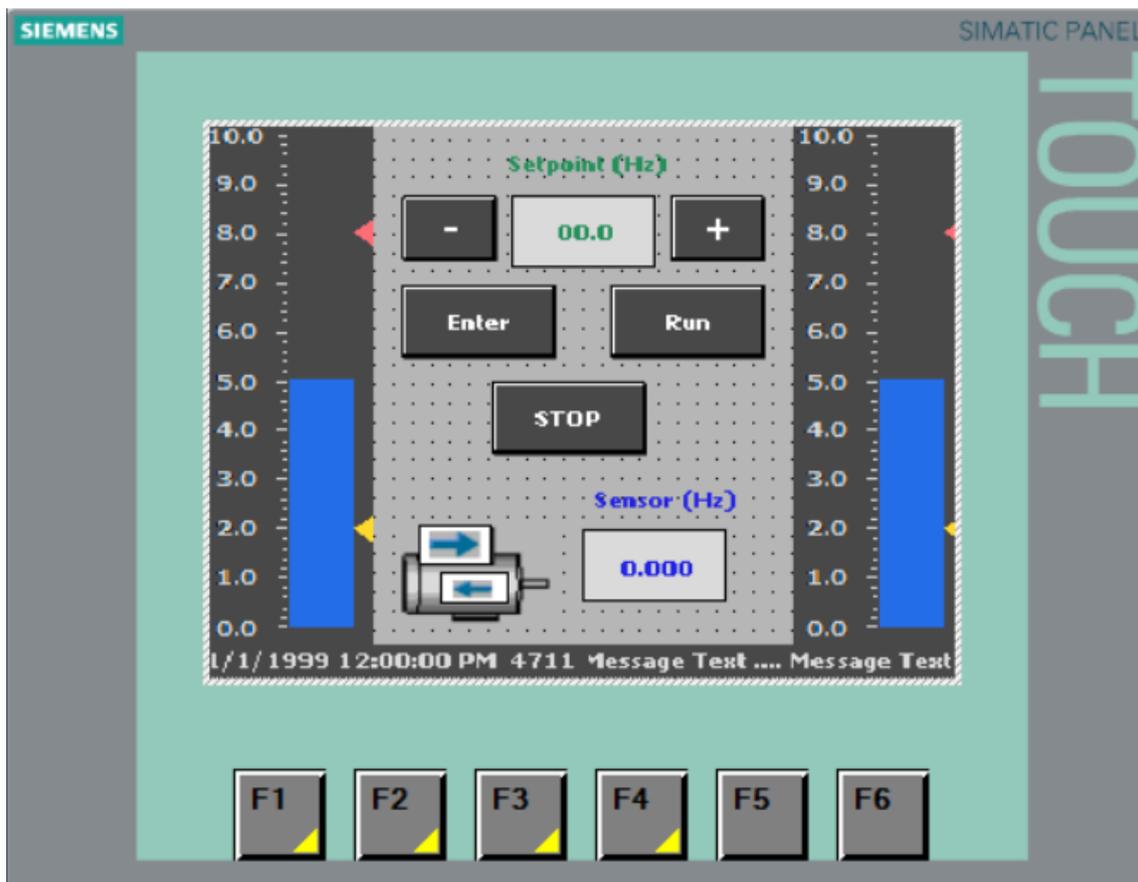


Figura 3.52: Ecrã de controlo do motor.

No ecrã de velocidade do motor ao longo do tempo, representado na Figura 3.53, verifica-se a velocidade a que o motor está a operar ao longo de um intervalo de tempo e todas as mudanças que tenham sido efetuadas a esta velocidade, como por exemplo a mudança de setpoint de velocidade. O botão referente a este ecrã é o F4.

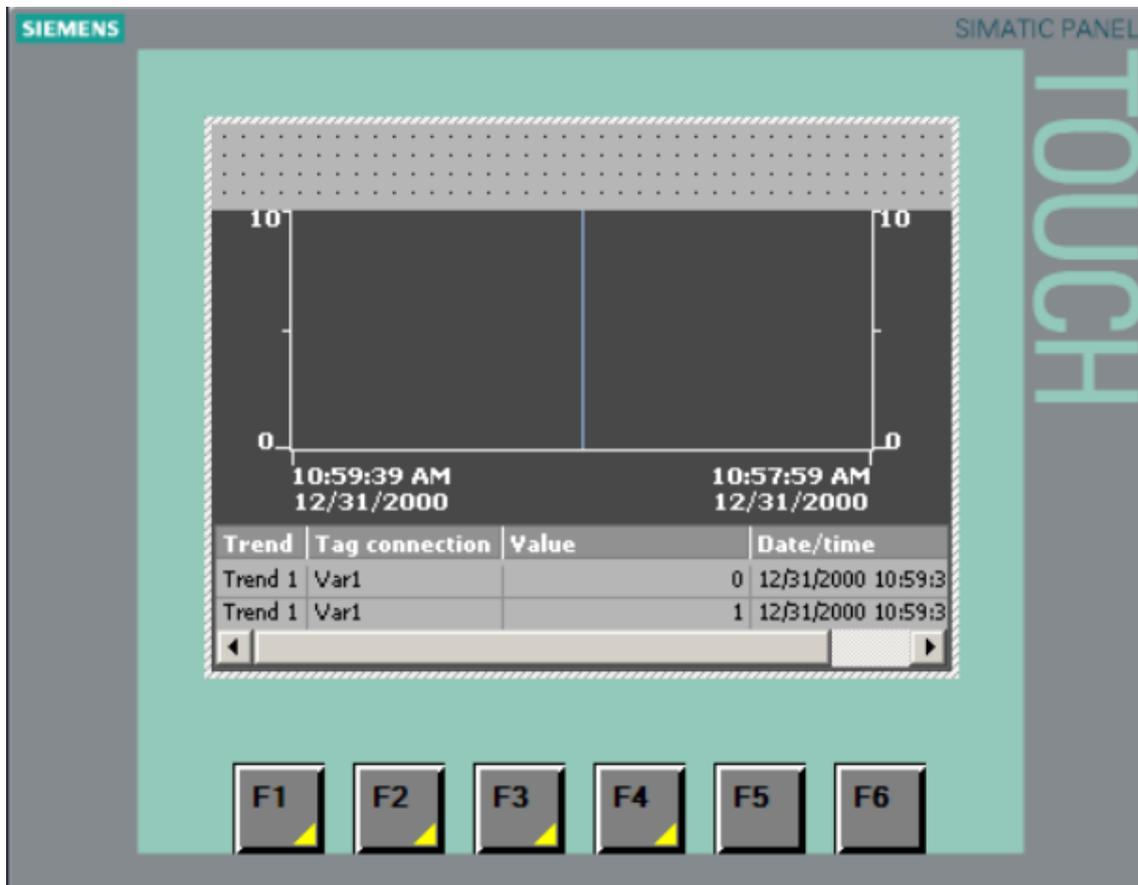


Figura 3.53: Ecrã de velocidade do motor ao longo do tempo.

RESULTADOS EXPERIMENTAIS

O presente capítulo encontra-se dividido em 3 partes.

Numa primeira parte é feita a descrição do processo ciber-físico real instalado.

Na segunda parte são exemplificados testes feitos para corroborar o funcionamento devido do protótipo de ensino em anel aberto.

Na última parte é descrita a implementação necessário para um controlo em anel fechado.

4.1 Descrição do Processo Ciber-Físico

Para efetuar a instalação física do protótipo de ensino, foi necessário incorporar os diferentes componentes de hardware e realizar a ligação elétrica de todos estes. Inicialmente, foi criada uma estrutura para suportar o processo, de seguida foi estudado e implementado o posicionamento de cada componente para o manuseamento e visualização do processo desejados, por fim foram feitas todas as ligações necessárias. O processo encontra-se representado na Figura 4.1, com os componentes de hardware em funcionamento.

4.1. DESCRIÇÃO DO PROCESSO CÍBER-FÍSICO

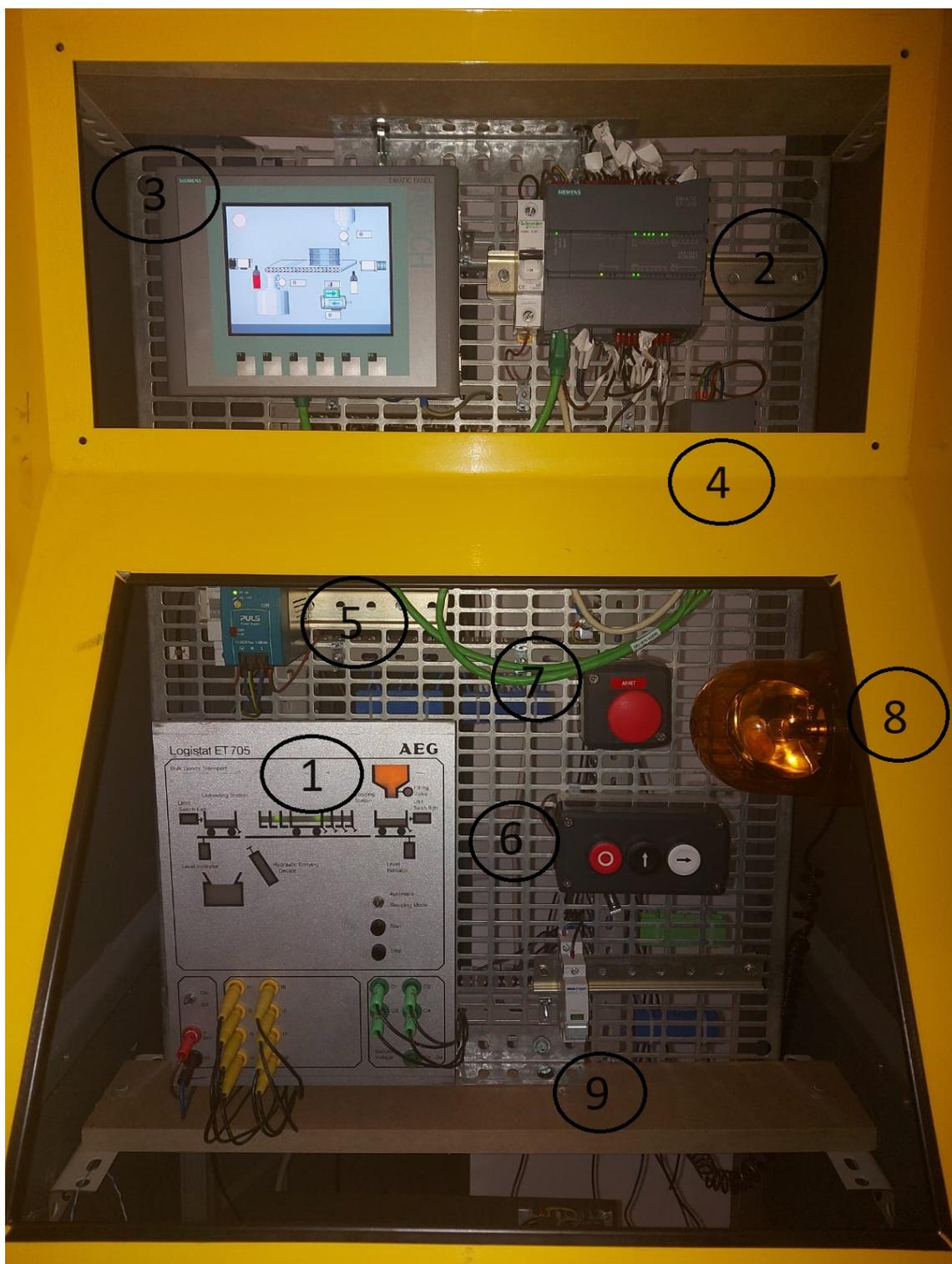


Figura 4.1: Foto do Protótipo de Ensino Cíber-Físico.

Os componentes de hardware representados na imagem são:

- 1) Processo Industrial de Transporte de Produtos AEG Logistat ET 705;
- 2) PLC Siemens S7-1200 com Módulo de Saídas Analógicas incorporado;
- 3) Consola HMI Siemens SIMATIC KTP600;
- 4) Módulo Switch Siemens;
- 5) Fonte de Alimentação PULS 24 Vdc;
- 6) Botoneira de Marcha;
- 7) Botão de Emergência;
- 8) Sinal de Alarme Sonoro "Buzzer" representado por um sinal luminoso laranja de emergência;
- 9) Sinal de Alarme Luminoso representado pelo equipamento Hager SVN 131.

O motor Siemens e o variador de velocidade Siemens Sinamics G110 encontram-se abaixo da estrutura apresentada na Figura 4.1. Isto foi decidido pois o equipamento é demasiado pesado para ser incorporado igualmente aos outros componentes de hardware apresentados e para evitar o contacto e manuseamento dos alunos com partes em movimento.

4.2 Teste do Controlo em Anel Aberto

Os testes realizados tiveram o intuito de corroborar e validar a programação lógica e a programação de sinópticos realizadas.

Numa primeira instância foi verificado que os sinais ativos na programação lógica correspondem com os sinais iluminadas do processo industrial de transporte de produtos AEG Logistat ET 705.

De seguida foram correlacionados os sinais ativos na programação lógica e os sinais iluminados no processo com os sinópticos desenvolvidos e verificou-se que a correspondência é a desejada, incluindo a velocidade e direção de rotação do motor representadas no ecrã de controlo do motor e no ecrã de velocidade do motor ao longo do tempo.

Numa terceira fase, foi testado o processo em modo manual através do controlo do mesmo com as botoneiras, o que também foi validado.

Por último, foram testados os sinais físicos de alarme, como se verifica na Figura 4.2, relativamente ao sinal luminoso de alarme e na Figura 4.3, relativamente ao sinal sonoro de alarme "Buzzer".

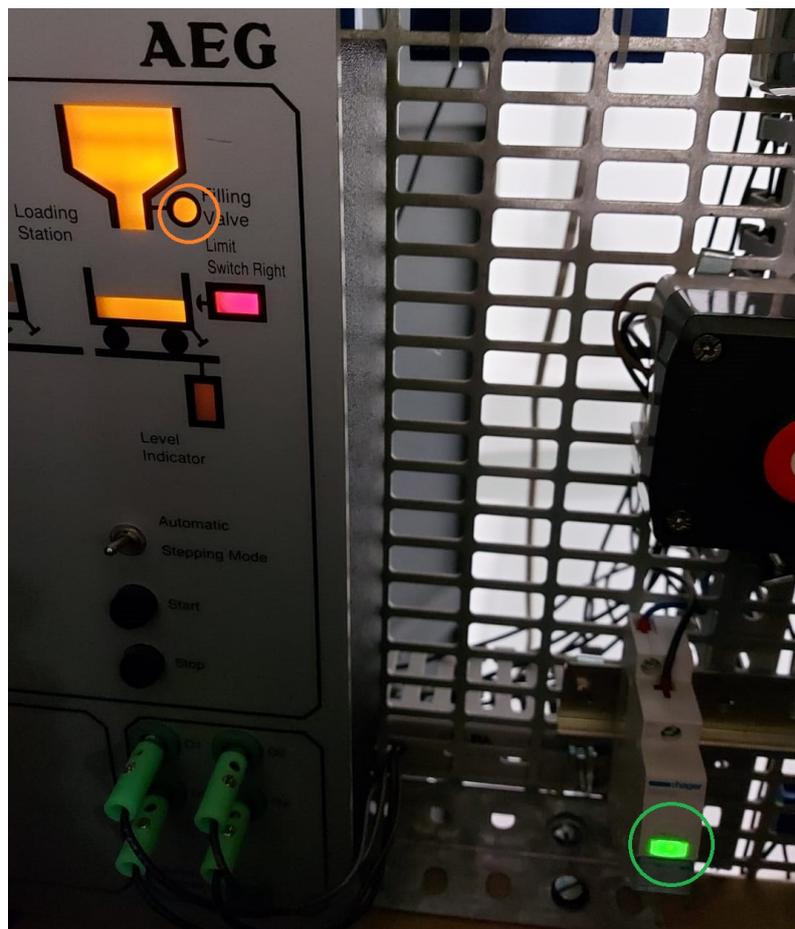


Figura 4.2: Foto do Sinal "Buzzer" ativo aquando da pressão Botão de Emergência.

Como é possível verificar na Figura 4.2, o sinal luminoso de alarme, representado com um LED verde, fica ativo quando o sinal da abertura da válvula de enchimento é ativado, representado com um LED laranja.



Figura 4.3: Foto do Sinal Luminoso ativo no momento em que se encontra aberta a válvula de enchimento.

Da mesma forma que acontece com o sinal luminoso de alarme, é possível verificar na Figura 4.3, que o sinal sonoro de alarme "Buzzer", representado pelo sinal laranja, fica ativo imediatamente a seguir à pressão do botão de emergência, representado a vermelho.

4.3 Teste do Controlo em Anel Fechado

Para ser possível um controlo do processo em anel fechado, representado na Figura 4.4, é necessário validar o controlo solicitado com o sinal que é recebido pelo autómato. Para isto, é necessário conseguir obter uma leitura da velocidade de rotação do motor para corroborar com a velocidade inserida no programa, ou seja, equiparar a saída analógica do autómato com a entrada no mesmo.

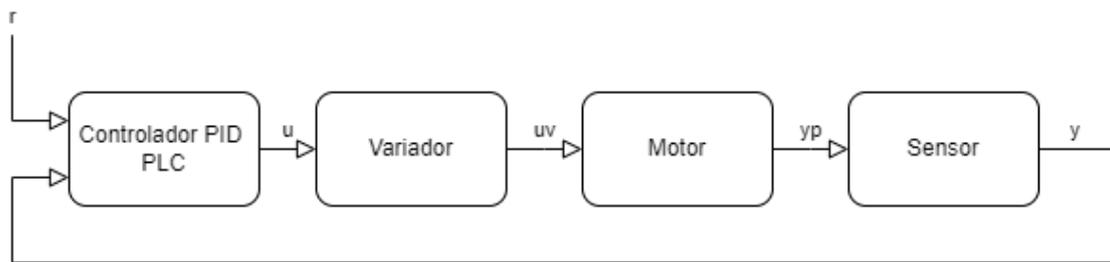


Figura 4.4: Controlo do Processo em Anel Fechado.

Os sinais representados na Figura 4.4 são os seguintes:

$r(k)$ - Sinal de comando de Setpoint programado pelo utilizador;

$u(k)$ - Sinal Analógico de Saída do Controlador;

$u_v(k)$ - Sinal Analógico de Saída do Variador para controlo do Motor;

$y_p(k)$ - Sinal Analógico de Saída do Motor para leitura no Sensor;

$y(k)$ - Sinal Analógico de Saída do Sensor para leitura no Controlador;

Sendo assim, foi necessário o desenvolvimento de um sensor de velocidade, pois a compra de um sensor que incorpora-se as especificações necessárias não era viável. Este sensor tem que devolver uma leitura de tensão na gama de [0-10] V e foi testado o seu dimensionamento no programa de simulação Electronics Workbench.

Com o desenvolvimento deste sensor foram verificados dois problemas. O primeiro era relativo aos valores de tensão lidos pelo tacómetro (instrumento utilizado para a medição) variarem apenas numa gama de [0-1] V, o que iria prejudicar a sensibilidade da leitura, portanto o circuito teve que ser implementado com função de amplificador. O segundo era relativo aos valores de tensão lidos enquanto o motor está em rotação no sentido horário serem negativos, o que o autómato não interpreta, sendo assim foi também necessário incorporar a função de inversor ao circuito, quando este se encontra neste sentido de rotação.

Os amplificadores operacionais utilizados, mais adequados à implementação do circuito encontram-se na Figura 4.5 e na Figura 4.6.

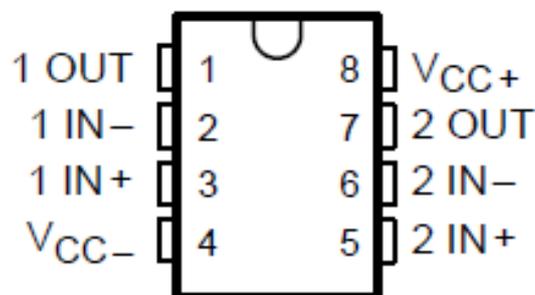


Figura 4.5: Amplificador Operacional TL082 [28].

O TL082 incorpora 2 amplificadores operacionais.

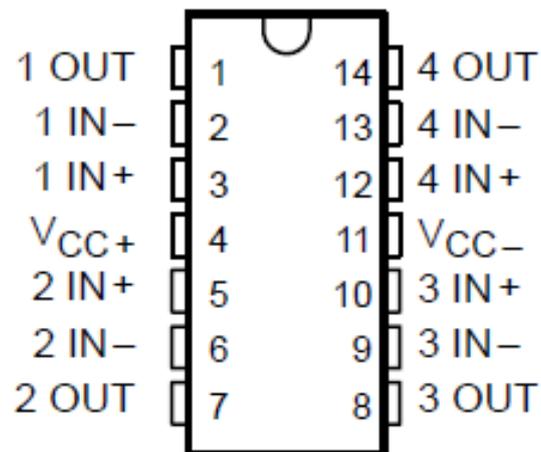


Figura 4.6: Amplificador Operacional TL084 [28].

O TL084 incorpora 4 amplificadores operacionais.

Em suma, ao todo foram necessários 6 amplificadores operacionais para o dimensionamento do circuito amplificador inversor, circuito este que se encontra representado na Figura 4.7.

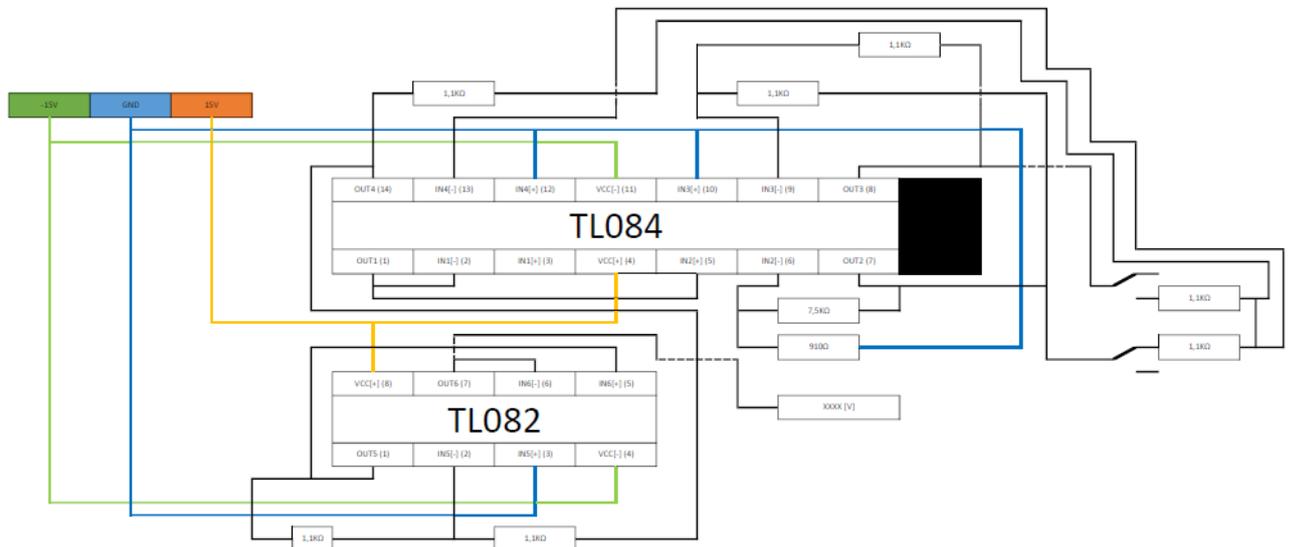


Figura 4.7: Circuito Amplificador Inversor.

Para a validação deste circuito foram realizados testes de simulação ao mesmo, incorporando os valores de tensão lidos no tacómetro, os valores simulados e os valores obtidos no circuito. De referir que a tensão ao motor foi limitada a uma gama de [0-2] V para não exceder a velocidade de rotação desejada.

Na Figura 4.8 é possível observar toda a gama de valores de simulação e os erros comparativos entre a simulação e o circuito real, no sentido horário. De notar que os erros com maior significado são obtidos a baixas velocidades.

Sentido Horário				
Um [V]	Utac [V]	Usim [V]	Ucirc [V]	Erro (Usim/cir) [%]
0	0	0	0	-
0.1	-0.0412	0.44	0.3798	13.68181818
0.2	-0.0872	0.88	0.8409	4.443181818
0.3	-0.1413	1.32	1.305	1.136363636
0.4	-0.1949	1.76	1.8	2.272727273
0.5	-0.2308	2.2	2.132	3.090909091
0.6	-0.2803	2.64	2.589	1.931818182
0.7	-0.3251	3.08	3.003	2.5
0.8	-0.3665	3.52	3.386	3.806818182
0.9	-0.4237	3.96	3.914	1.161616162
1	-0.4483	4.4	4.142	5.863636364
1.1	-0.5068	4.84	4.682	3.26446281
1.2	-0.5712	5.28	5.277	0.056818182
1.3	-0.6185	5.72	5.714	0.104895105
1.4	-0.6574	6.16	6.074	1.396103896
1.5	-0.7142	6.6	6.599	0.015151515
1.6	-0.7569	7.04	6.993	0.667613636
1.7	-0.7991	7.48	7.383	1.296791444
1.8	-0.8447	7.92	7.805	1.452020202
1.9	-0.8873	8.36	8.198	1.937799043
2	-0.9537	8.8	8.812	0.136363636

Figura 4.8: Valores de simulação com o motor em rotação no sentido horário.

Na Figura 4.9 verifica-se o comportamento do circuito em sentido horário relativamente ao medido no tacómetro. De notar que o circuito tem um comportamento praticamente linear, com um desvio no valor de 0.0047.

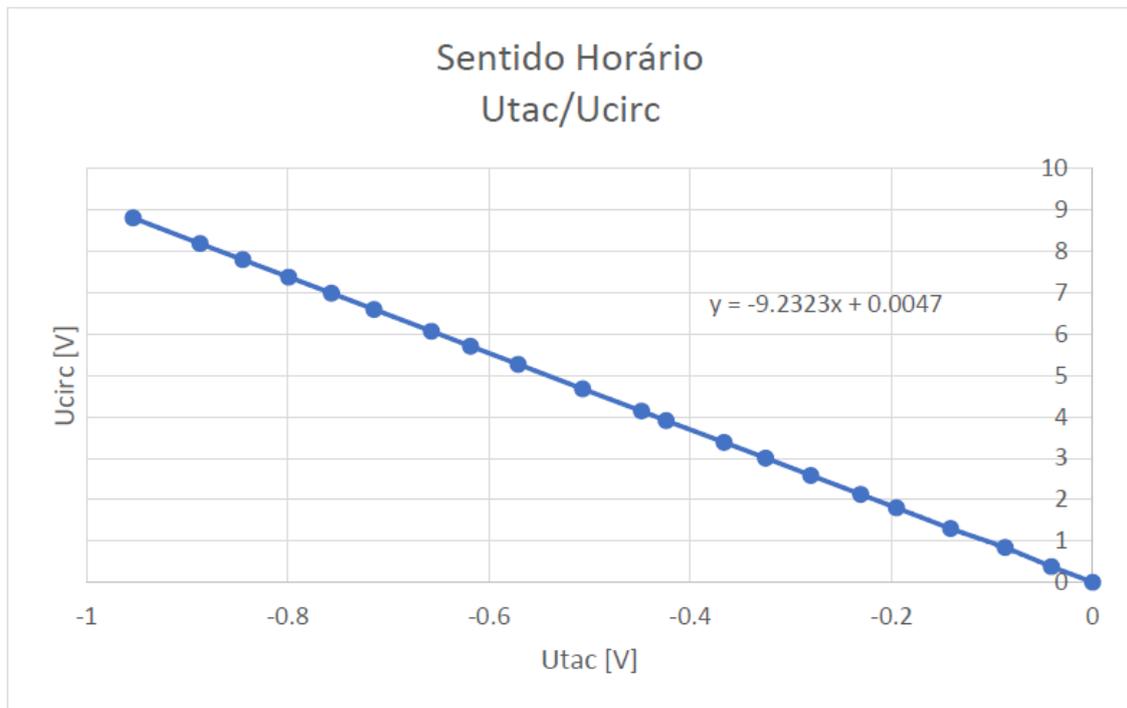


Figura 4.9: Comportamento do sensor com o motor em rotação no sentido horário.

Na Figura 4.10 é possível observar toda a gama de valores de simulação e os erros comparativos entre a simulação e o circuito real, no sentido anti-horário. De notar que os erros com maior significado são obtidos a baixas velocidades, à semelhança do sentido horário.

Sentido Anti-Horário				
Um [V]	Utac [V]	Usim [V]	Ucirc [V]	Erro (Usim/cir) [%]
0	0	0	0	-
0.1	0.0416	0.44	0.3853	12.43181818
0.2	0.0995	0.88	0.9203	4.579545455
0.3	0.1157	1.32	1.07	18.93939394
0.4	0.1897	1.76	1.754	0.340909091
0.5	0.2165	2.2	2.001	9.045454545
0.6	0.2708	2.64	2.503	5.189393939
0.7	0.3061	3.08	2.829	8.149350649
0.8	0.3629	3.52	3.354	4.715909091
0.9	0.4258	3.96	3.936	0.606060606
1	0.4673	4.4	4.319	1.840909091
1.1	0.5347	4.84	4.942	2.107438017
1.2	0.5465	5.28	5.051	4.337121212
1.3	0.6197	5.72	5.727	0.122377622
1.4	0.6378	6.16	5.895	4.301948052
1.5	0.6783	6.6	6.269	5.015151515
1.6	0.7474	7.04	6.907	1.889204545
1.7	0.7819	7.48	7.226	3.395721925
1.8	0.8442	7.92	7.802	1.48989899
1.9	0.8935	8.36	8.257	1.232057416
2	0.9427	8.8	8.712	1

Figura 4.10: Valores de simulação com o motor em rotação no sentido anti-horário.

Na Figura 4.11 verifica-se o comportamento do circuito em sentido anti-horário relativamente ao medido no tacómetro. De notar que o circuito também apresenta um comportamento praticamente linear, com um desvio no valor de 0.0007, à semelhança do sentido horário.

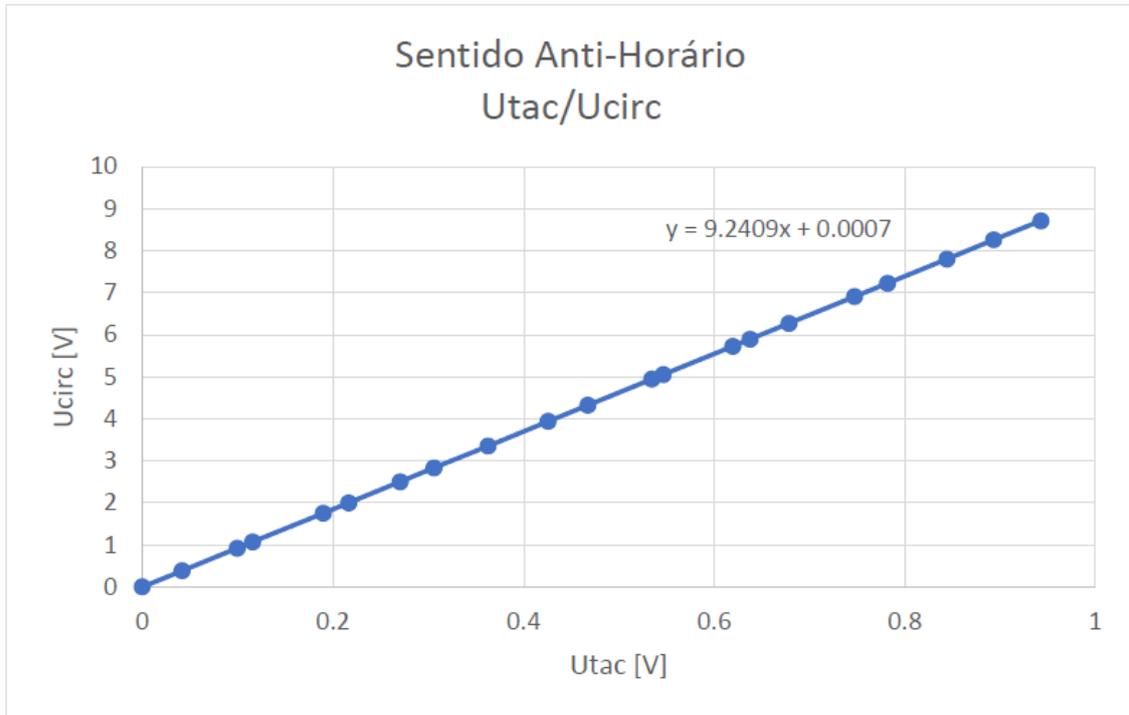


Figura 4.11: Comportamento do sensor com o motor em rotação no sentido anti-horário.

4.3.1 Teste do Sensor

Para o teste do Sensor foram considerados dois regimes distintos: Estacionário e Transitório. Isto torna possível a análise do comportamento do mesmo com um setpoint constante e com a variação de setpoints. É ainda de notar que a calibração do sensor foi feita para ter um offset mínimo a uma velocidade de 5 Hz.

Através dos testes referidos é feito o cálculo do offset (em regime estacionário), da sobre-elevação e sub-elevação (em regime transitório) entre o setpoint e o valor lido pelo sensor, o que permite concluir relativamente à fiabilidade do sensor.

De referir que os valores experimentais foram obtidos através da placa de aquisição de dados USB 6008 representada na Figura 4.12:

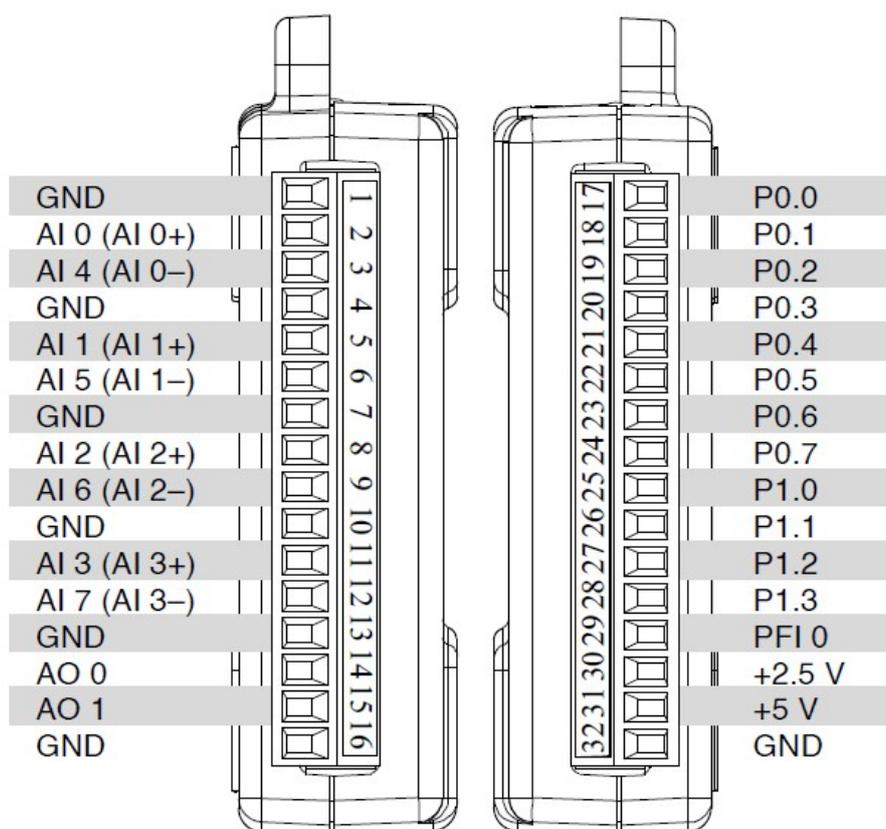


Figura 4.12: Placa de aquisição de dados USB 6008 e respectivos pinos de ligação [16].

A placa opera com valores na gama de [0-10] V e os pinos utilizados para a obtenção dos dados foram o 1 (valor GND de referência), o 2 (valor analógico do setpoint) e o 5 (valor analógico de leitura do sensor).

Tendo em consideração que a leitura de valores da placa é na grandeza Volt e os setpoints serem definidos na grandeza Hertz é necessário ter em consideração a conversão entre grandezas. O valor máximo de leitura possível é de 10 V e a frequência máxima de trabalho é de 50 Hz, logo um valor equivale a outra nas respectivas grandezas. Tomando como exemplo o valor pretendido para um offset mínimo, 5 Hz tem equivalência a 1 V.

4.3. TESTE DO CONTROLO EM ANEL FECHADO

É necessário ter em consideração a conversão dos valores lidos pelo sensor para a escala de valores desejada. Para isso foram realizadas 2 experiências, no sentido de rotação horário e anti-horário, em que se retirou medições, em V, dos valores obtidos pelo sensor na gama de velocidade de funcionamento do motor, considerando o valor mínimo para o início de rotação do motor [0.6-7] Hz.

Os valores obtidos para o sentido de rotação anti-horário são os representados na Figura 4.13:

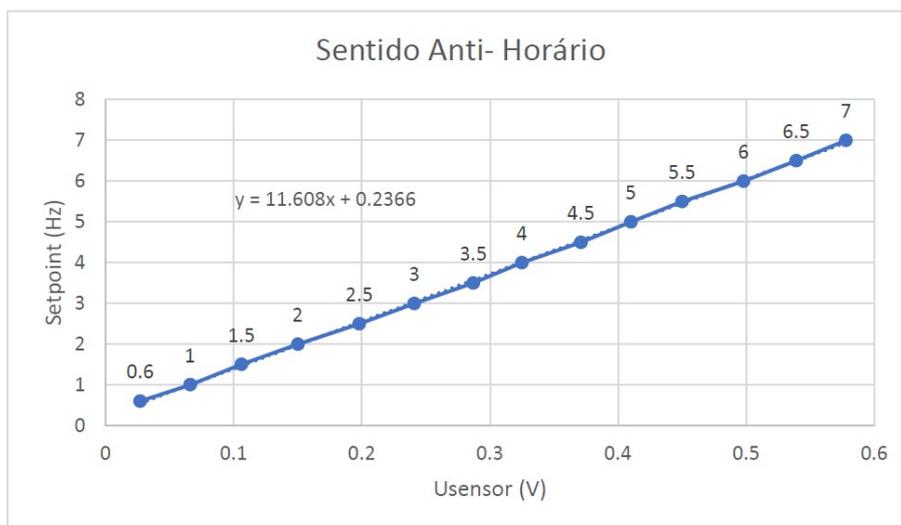


Figura 4.13: Resultados experimentais de leitura do sensor em rotação anti-horária.

Os valores obtidos para o sentido de rotação horário são os representados na Figura 4.14:

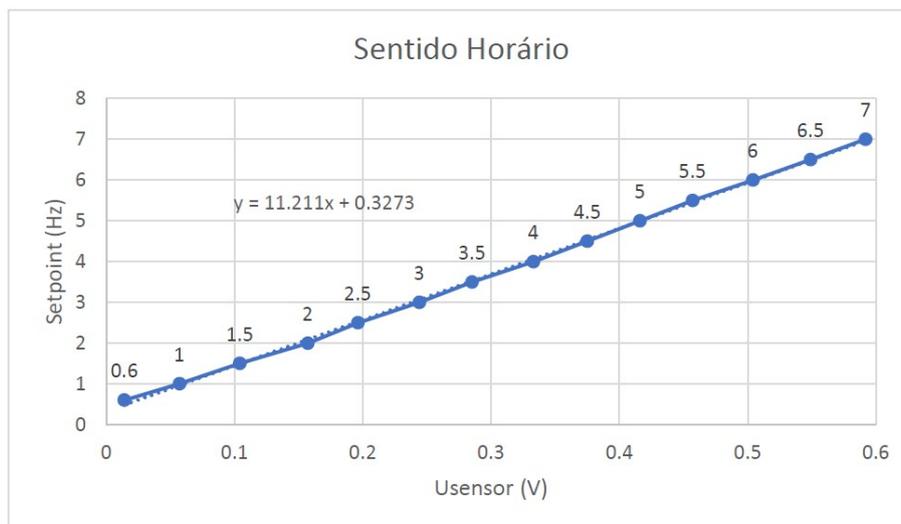


Figura 4.14: Resultados experimentais de leitura do sensor em rotação horária.

Através das duas experiências referidas é possível definir uma equação de reta para o dimensionamento dos valores transmitidos do sensor para o autômato com o cálculo da média de ambas:

$$y_{ah} = 11.608x + 0.2366$$

$$y_h = 11.211x + 0.3273$$

$$y_{final} = \left(\frac{11.608+11.211}{2} \right) x + \left(\frac{0.2366+0.3273}{2} \right) = 11.4095x + 0.28195$$

Com esta equação é possível definir uma função para a leitura dos valores, como apresentada na Figura 4.15:

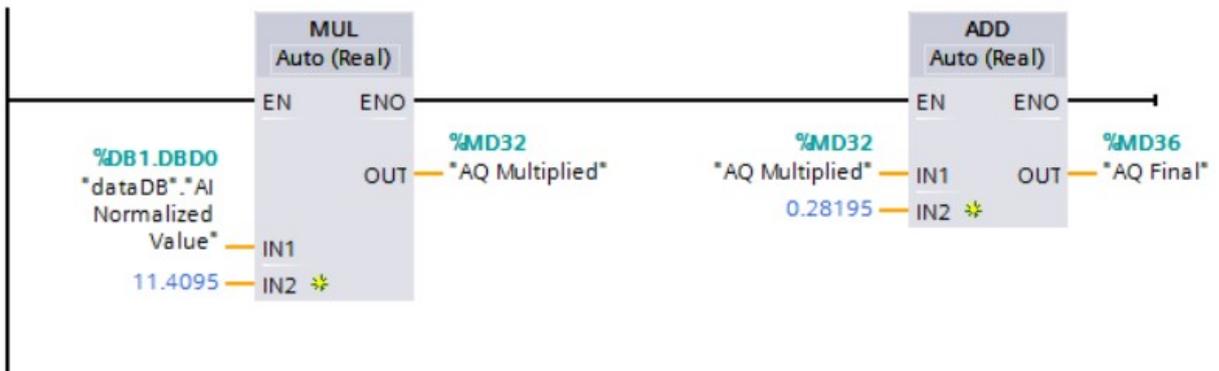


Figura 4.15: Funções para escalar valores para a gama desejada.

4.3.1.1 Testes em Regime Estacionário

Para os testes em regime Estacionário foram selecionado 7 setpoints distintos, todos com intervalo de 1 Hz entre eles, representado a gama de velocidade definida para o funcionamento do motor - [1-7] Hz, ou seja [0,2-1,4] V. Tendo como referência a Figura 3.12, a rotação do motor a 50 Hz é de 1350 rpm, portanto a 7 Hz a rotação do motor é de 189 rpm ou 3.15 rps.

De referir que o valor experimental mínimo de velocidade do motor para a obtenção de leituras fidedignas através do sensor é de 0.6 Hz.

Estes setpoints são apresentados em forma ascendente de valor, sendo o setpoint representado a vermelho e o valor de leitura do sensor a verde, $u(k)$ e $y(k)$ respetivamente, representados na Figura 4.4.

Na Figura 4.16 verifica-se um offset bastante significativo, no valor de 0.073621 V. Torna-se possível concluir que a leitura a frequências significativamente baixas não é de elevada fidelidade.

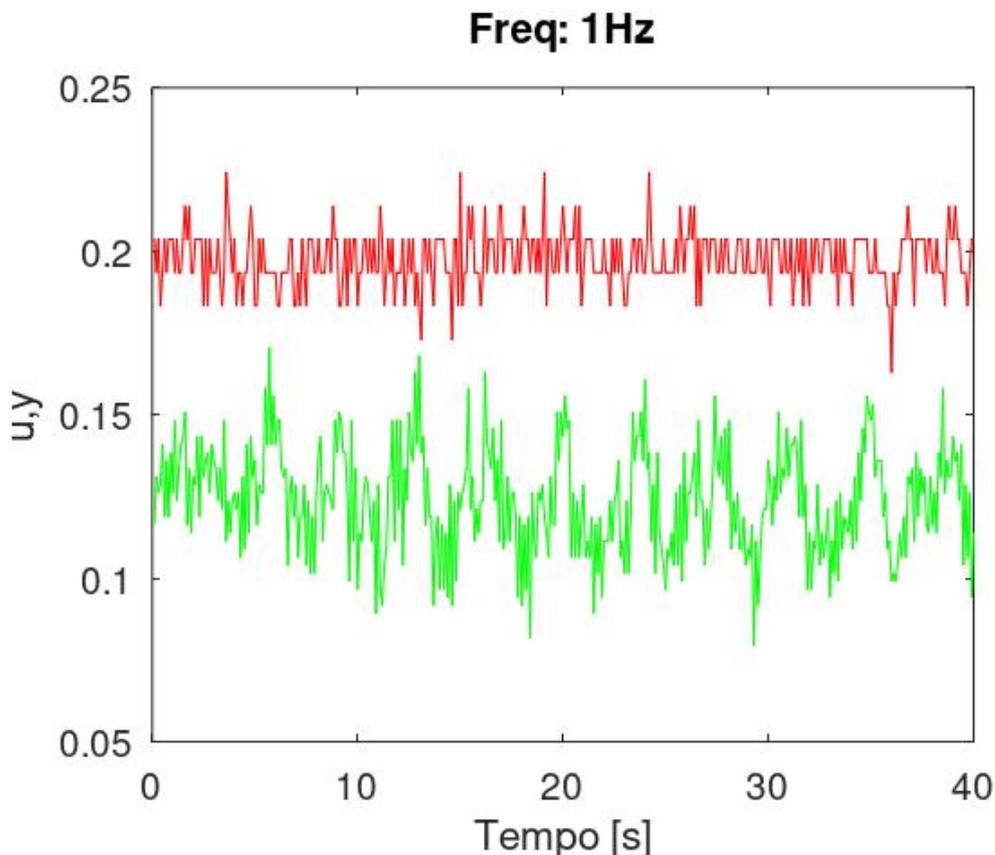


Figura 4.16: Comportamento em Regime Estacionário - 1 Hz.

Na Figura 4.17 verifica-se um offset reduzido relativamente ao setpoint anterior, mas ainda algo significativo, no valor de 0.042447 V .

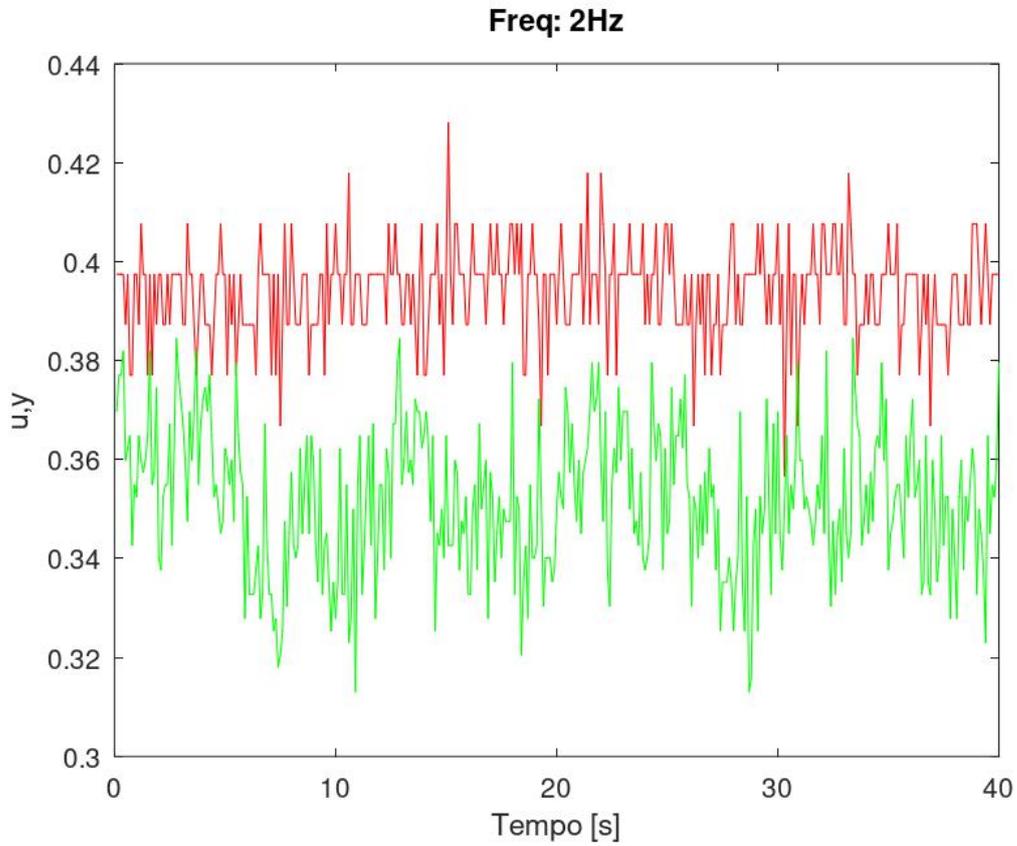


Figura 4.17: Comportamento em Regime Estacionário - 2 Hz.

Com a subida de velocidade, observa-se na Figura 4.18 um offset reduzido relativamente ao setpoint anterior, no valor de 0.025684 V e uma aproximação à leitura de valores pretendida.

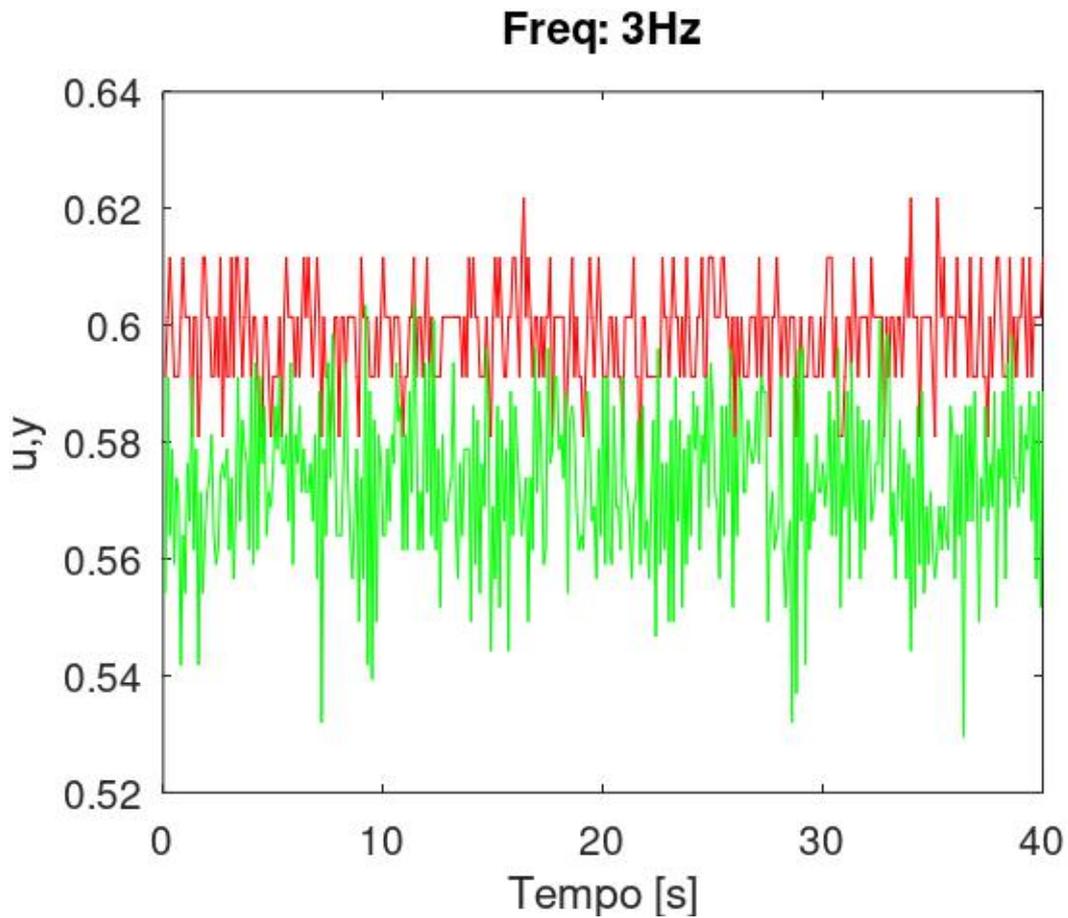


Figura 4.18: Comportamento em Regime Estacionário - 3 Hz.

Na Figura 4.19 observa-se um offset bastante reduzido, no valor de 0.0056897 V e uma aproximação quase ideal à leitura de valores pretendida.

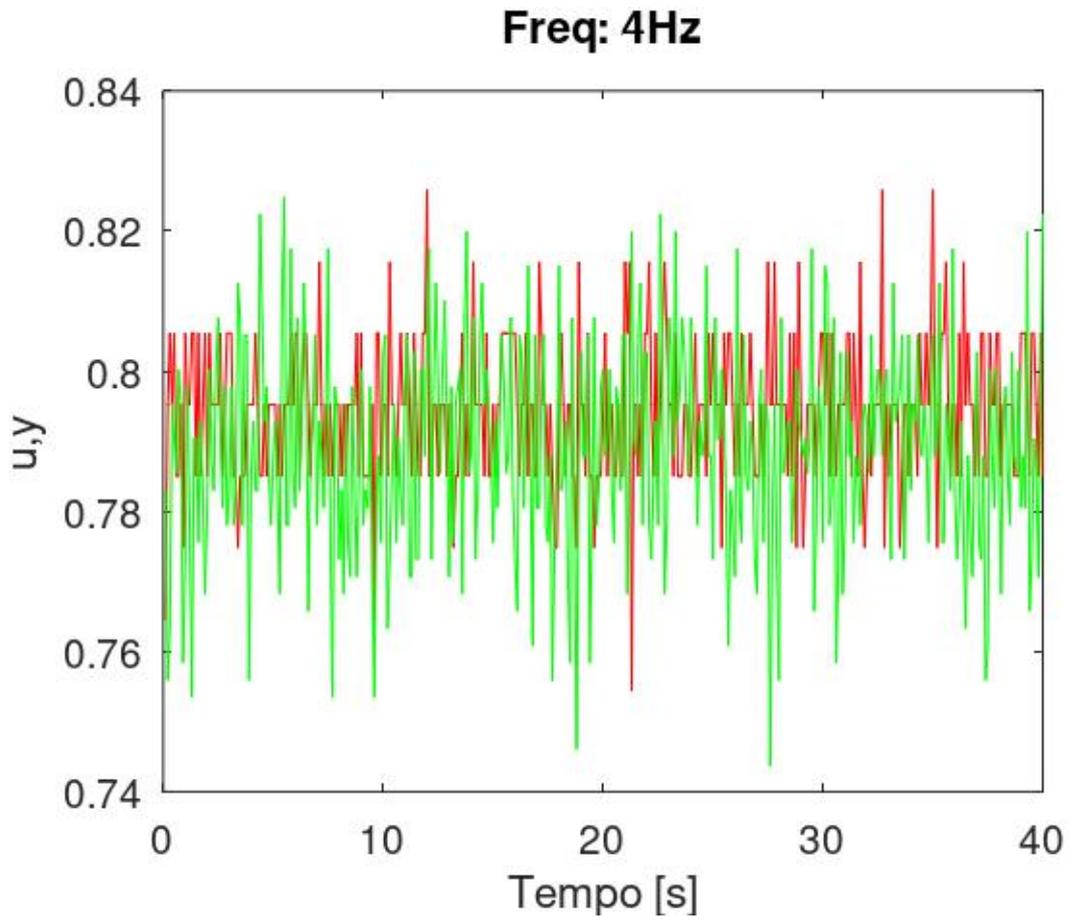


Figura 4.19: Comportamento em Regime Estacionário - 4 Hz.

Na Figura 4.20 é apresentado o setpoint ideal de trabalho do motor, com o offset mais reduzido, no valor de 0.0037324 V e a melhor aproximação possível à leitura de valores pretendida.

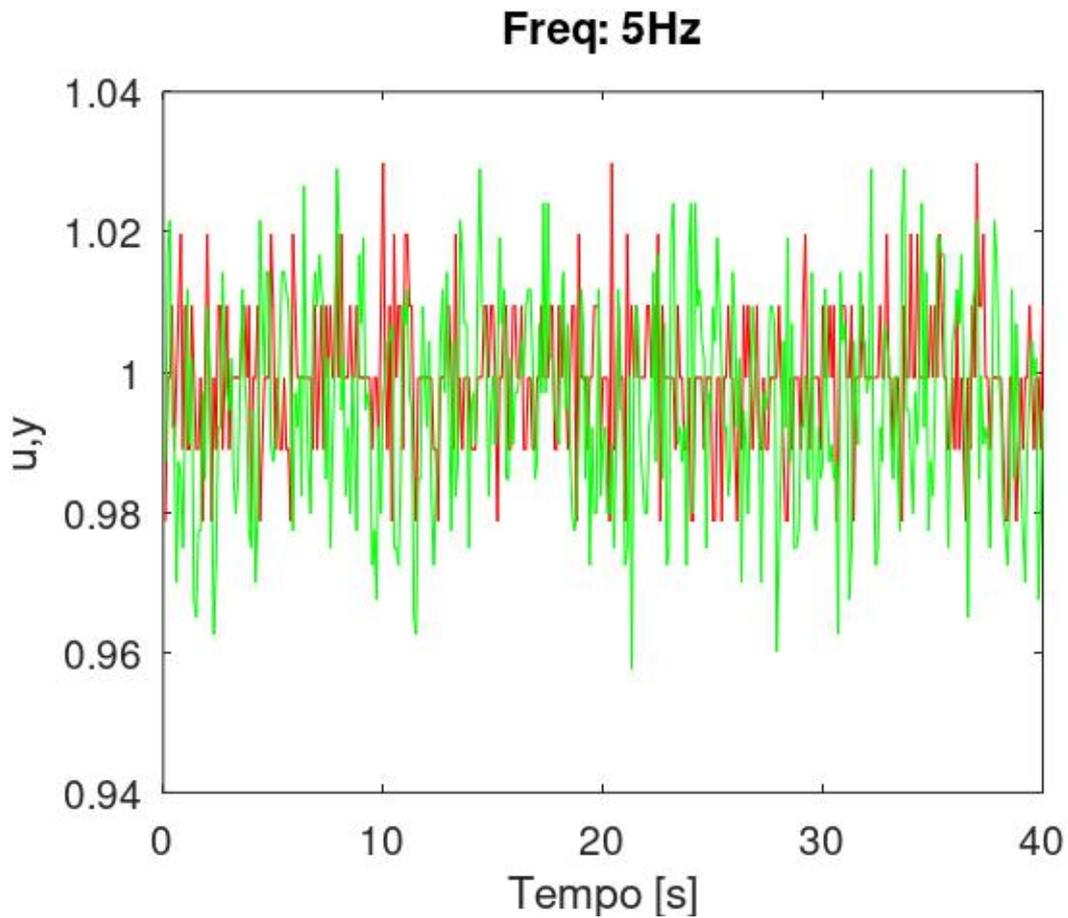


Figura 4.20: Comportamento em Regime Estacionário - 5 Hz.

Na Figura 4.21 é possível verificar um ligeiro aumento de offset relativamente ao set-point anterior, no valor de 0.008748 V.

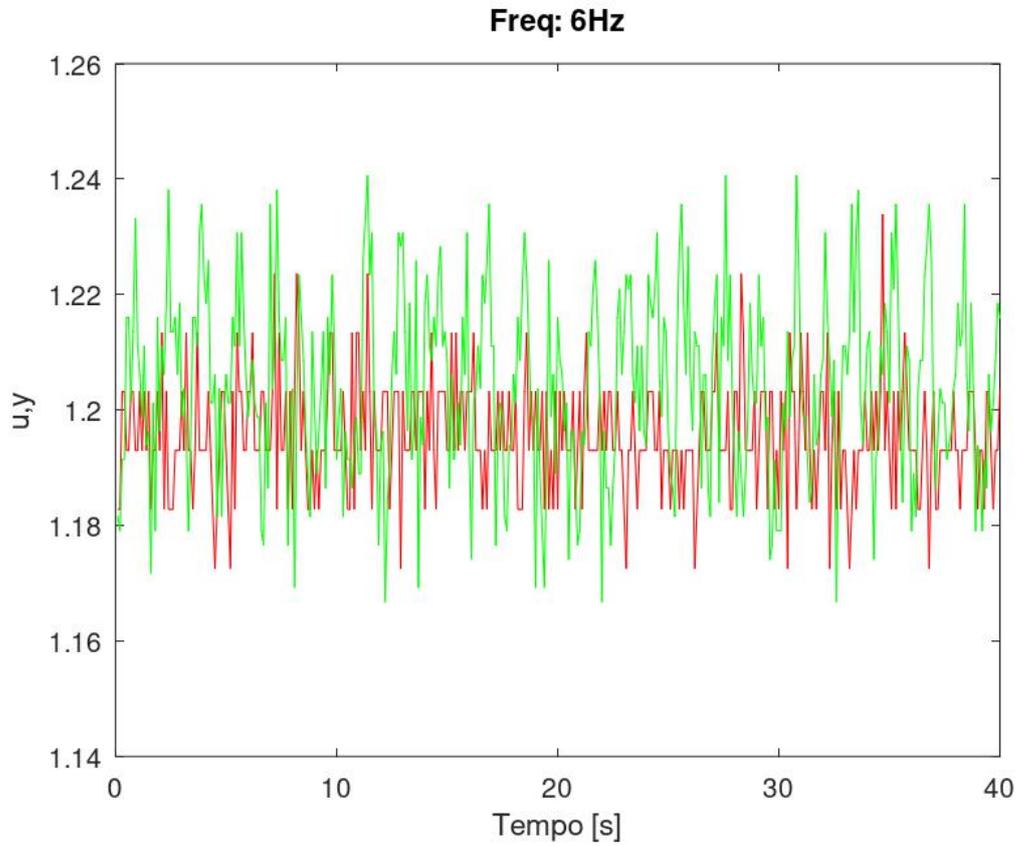


Figura 4.21: Comportamento em Regime Estacionário - 6 Hz.

Por último, verifica-se na Figura 4.22 outro ligeiro aumento de offset relativamente ao setpoint anterior, no valor de 0.011731 V.

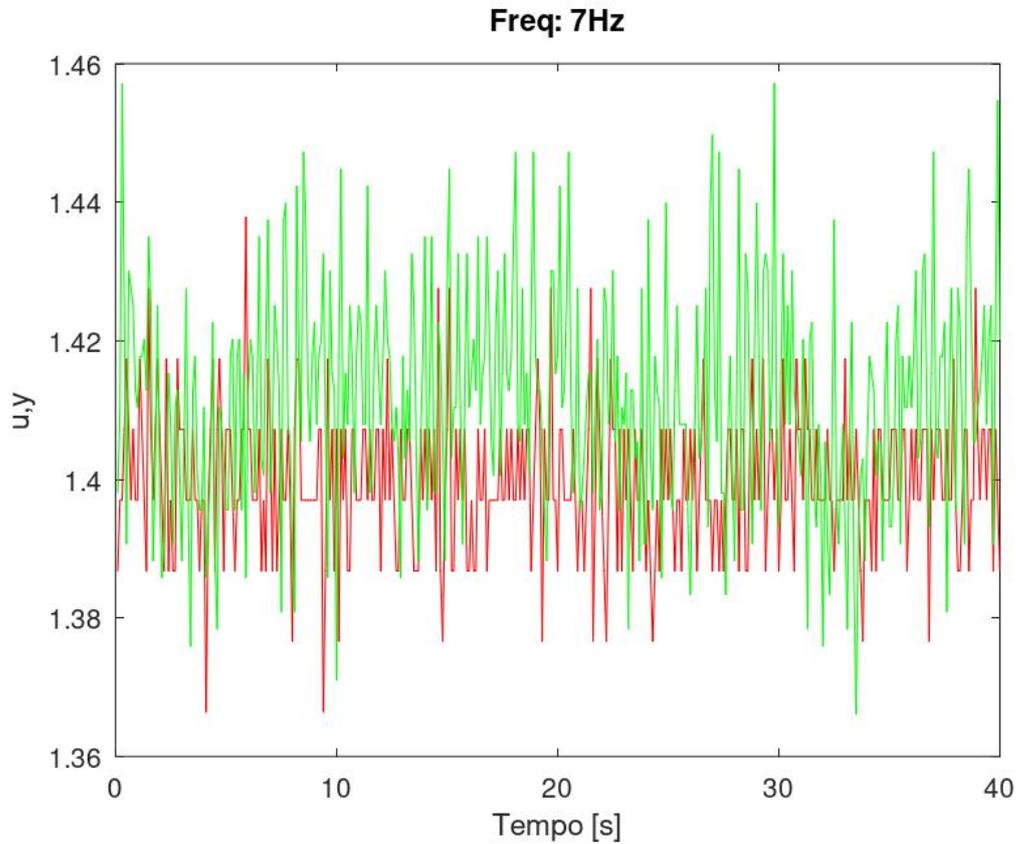


Figura 4.22: Comportamento em Regime Estacionário - 7 Hz.

A análise destes valores é necessária para definir qual a melhor gama de trabalho para futuros testes experimentais de controladores.

De referir que não foram testados valores de setpoint acima de 7 Hz pois o sensor saturava.

Na Figura 4.23 são apresentados os valores de offset obtidos para os diferentes setpoints:

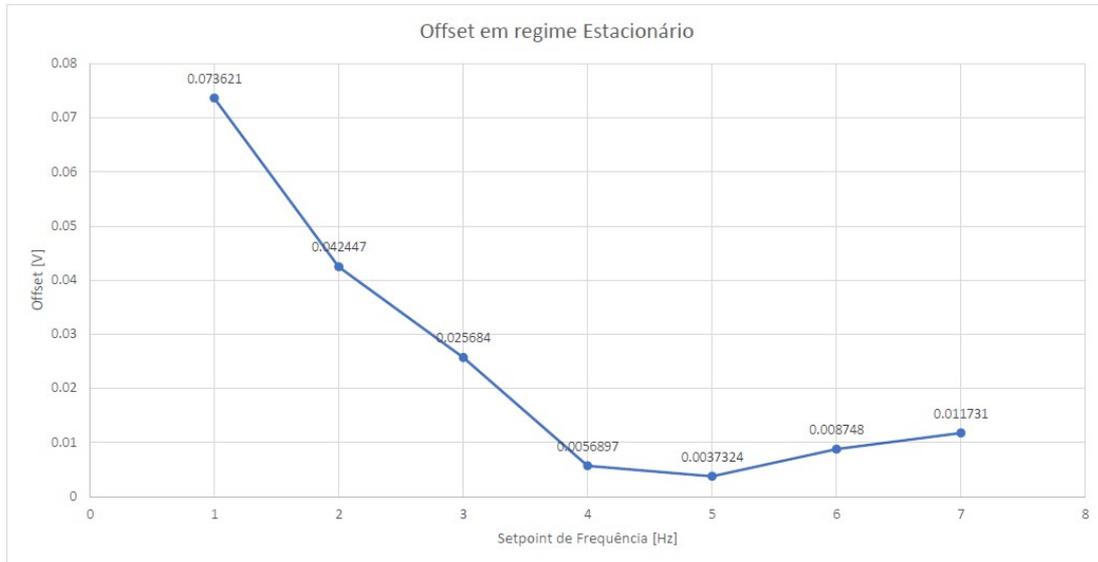


Figura 4.23: Offsets de leitura relativamente aos setpoints para a gama de trabalho definida.

Através da visualização da Figura 4.23 é possível concluir que a gama de setpoints mais viável para a realização de testes experimentais é de [4-6] Hz, sendo que todos os offsets nesta gama estão abaixo de 0.01 e que o setpoint com menor valor de offset é 5 Hz, como pretendido.

Para a deteção de variações do controlador do variador, foi realizada uma experiência com dinâmica mais lenta em que o veio do motor foi submetido a uma carga, utilizando a menor velocidade de trabalho possível, 1 Hz. Os resultados desta experiência encontram-se na Figura 4.24:

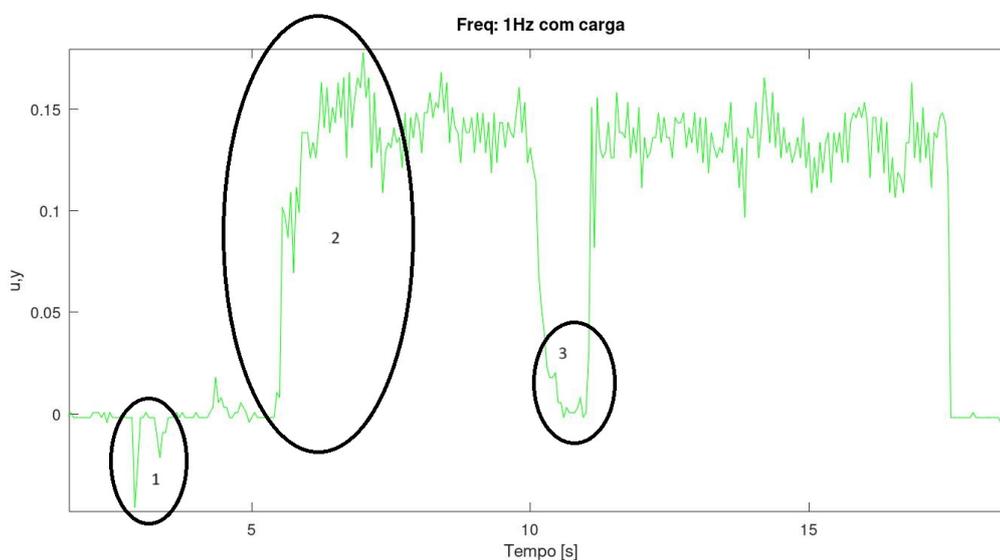


Figura 4.24: Comportamento em Regime Estacionário com Carga no Veio- 1 Hz.

Na instância de tempo marcada com o número 1, observa-se a aplicação de uma carga antes do arranque do motor. Inclusivamente verifica-se que o esforço feito traduz-se em valores de tensão negativos lidos no sensor.

No momento 2, é possível verificar o arranque do motor para o setpoint de 1 Hz. Este arranque é lento, sendo necessários praticamente 2 segundos para se dar a estabilização do controlador.

O último momento, marcado com o número 3, representa uma grande carga aplicada no veio depois do controlador estabilizar e de se ter retirado a anterior carga. Verifica-se que o motor praticamente parou durante 1 segundo, concluindo que o mesmo, assim como o controlador, é bastante afetado pelas cargas induzidas, sendo que a baixas velocidade, até deixa de operar.

4.3.1.2 Testes em Regime Transitório

Para os testes em regime Transitório foram selecionadas variações do setpoint de valor mínimo (1 Hz) para o de valor máximo (7 Hz) e do valor máximo (7 Hz), para o valor mínimo (1 Hz), para o valor médio (4 Hz) e deste para o de valor máximo (7 Hz), ou seja um incremento de 3 Hz em cada experiência.

Estes testes têm a finalidade de permitir calcular a sobre-elevação e a sub-elevação do controlador, dados pertinentes para a avaliação de eficiência do mesmo. O cálculo destes valores é realizado através da seguinte fórmula:

$$S[\%] = \left(\frac{V_p - V_F}{V_F - V_I} \right) * 100\%$$

As variáveis representadas na fórmula são:

V_p - Valor de Pico de Tensão;

V_F - Valor Final de Tensão após estabilização do sinal;

V_I - Valor Inicial de Tensão.

A Figura 4.25 representa o comportamento do controlador aquando se dá o maior salto possível relativamente aos valores de setpoint, de 1 Hz para 7 Hz e vice-versa.

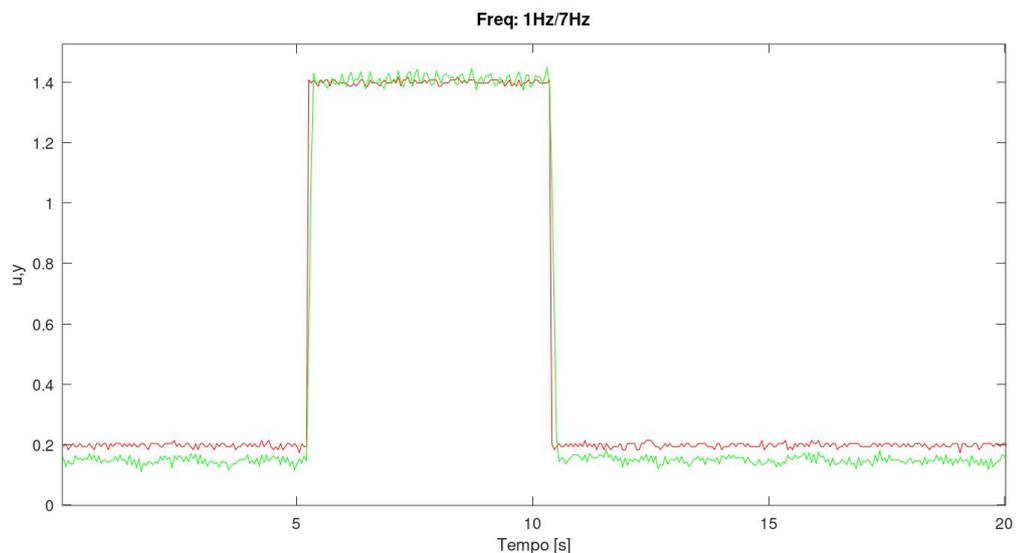


Figura 4.25: Comportamento em Regime Transitório de 1 para 7 Hz.

A Figura 4.26 permite calcular a sobre-elevação ocorrida na mudança de setpoint de 1 Hz para 7 Hz e visualizar o tempo de subida.

4.3. TESTE DO CONTROLO EM ANEL FECHADO

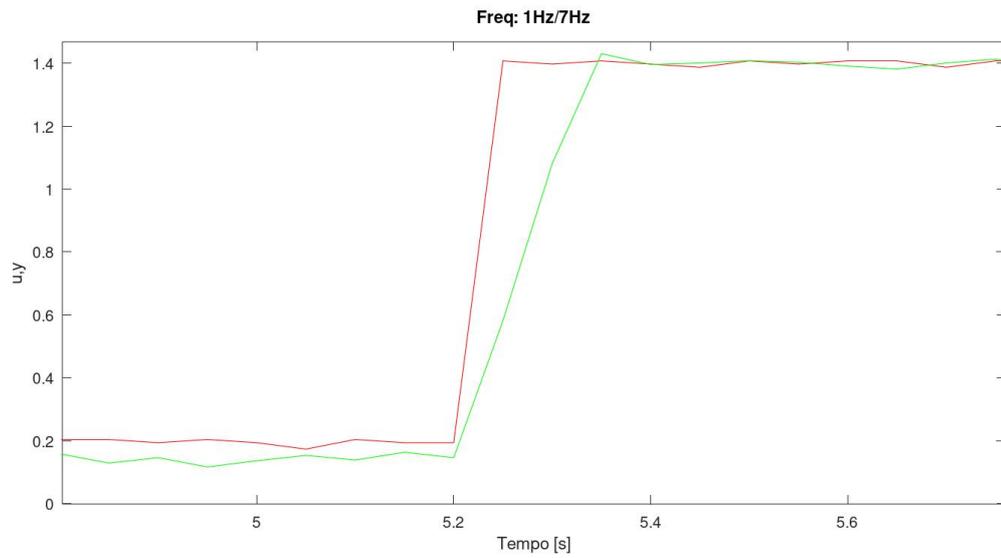


Figura 4.26: Sobre-elevação em Regime Transitório de 1 para 7 Hz.

$$Sob_{1/7}[\%] = \left(\frac{1.45 - 1.4}{1.4 - 0.18} \right) * 100\% = 4.1\%, \text{ com } T_s = 0.15s$$

A Figura 4.27 permite calcular a sub-elevação ocorrida na mudança de setpoint de 7 Hz para 1 Hz e visualizar o tempo de descida.

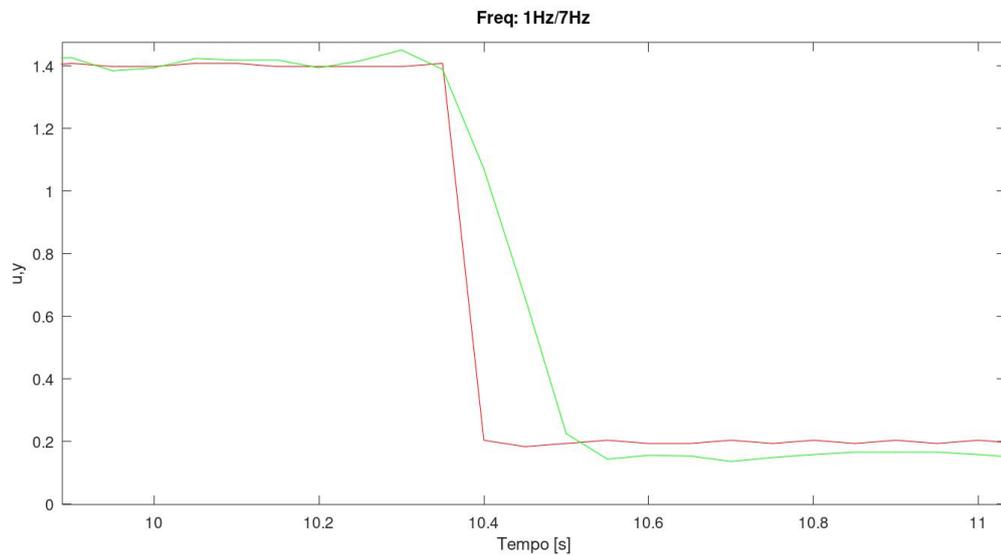


Figura 4.27: Sub-elevação em Regime Transitório de 1 para 7 Hz.

$$Sub_{7/1}[\%] = \left(\frac{0.15 - 0.18}{0.18 - 1.4} \right) * 100\% = 2.46\%, \text{ com } T_d = 0.25s$$

A Figura 4.28 representa o comportamento do controlador quando se dá o maior salto possível relativamente aos valores de setpoint, de 1 Hz para 4 Hz e vice-versa.

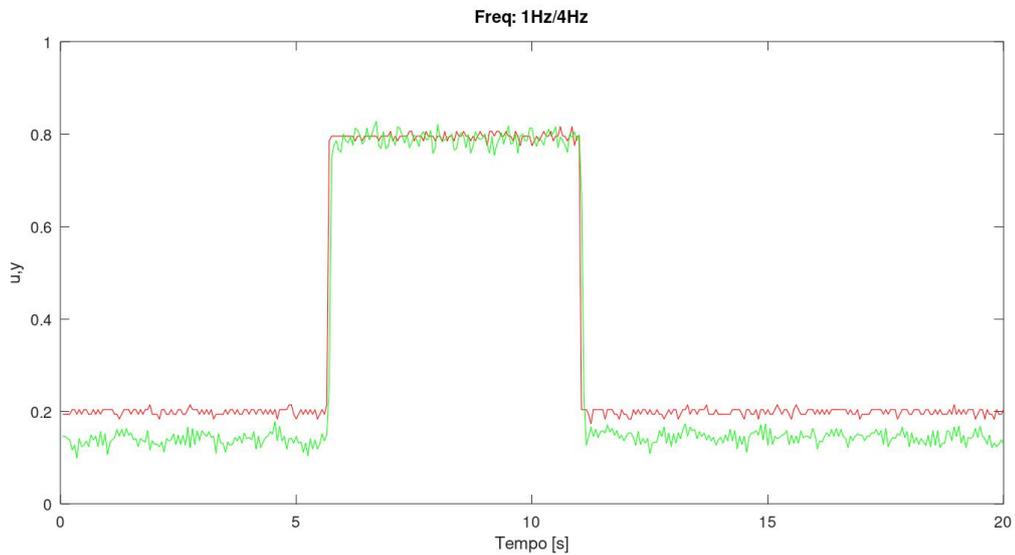


Figura 4.28: Comportamento em Regime Transitório de 1 para 4 Hz.

A Figura 4.29 permite calcular a sobre-elevação ocorrida na mudança de setpoint de 1 Hz para 4 Hz e visualizar o tempo de subida.

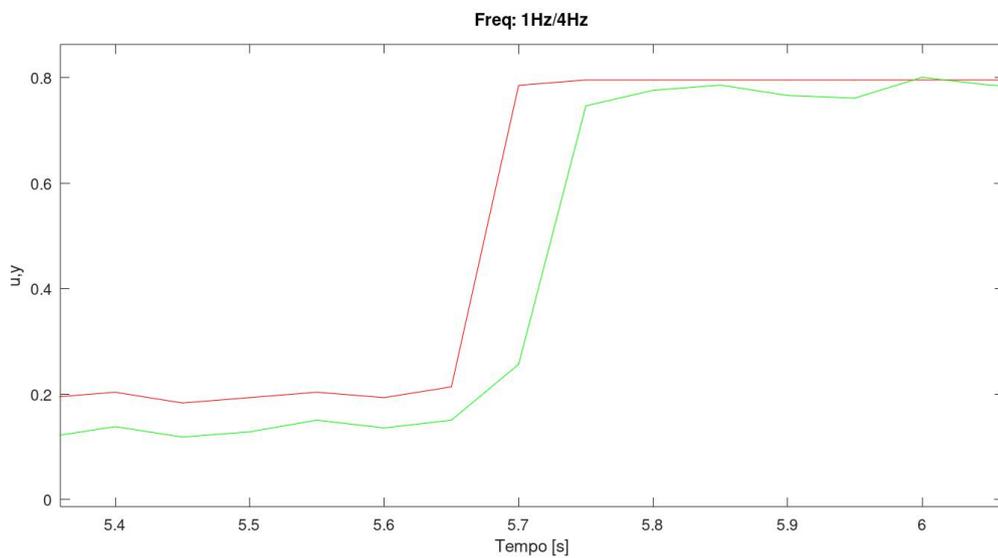


Figura 4.29: Sobre-elevação em Regime Transitório de 1 para 4 Hz.

$$Sob_{1/4}[\%] = \left(\frac{0.81 - 0.8}{0.8 - 0.16} \right) * 100\% = 1.56\%, \text{ com } T_s = 0.35s$$

A Figura 4.30 permite calcular a sub-elevação ocorrida na mudança de setpoint de 4 Hz para 1 Hz e visualizar o tempo de descida.

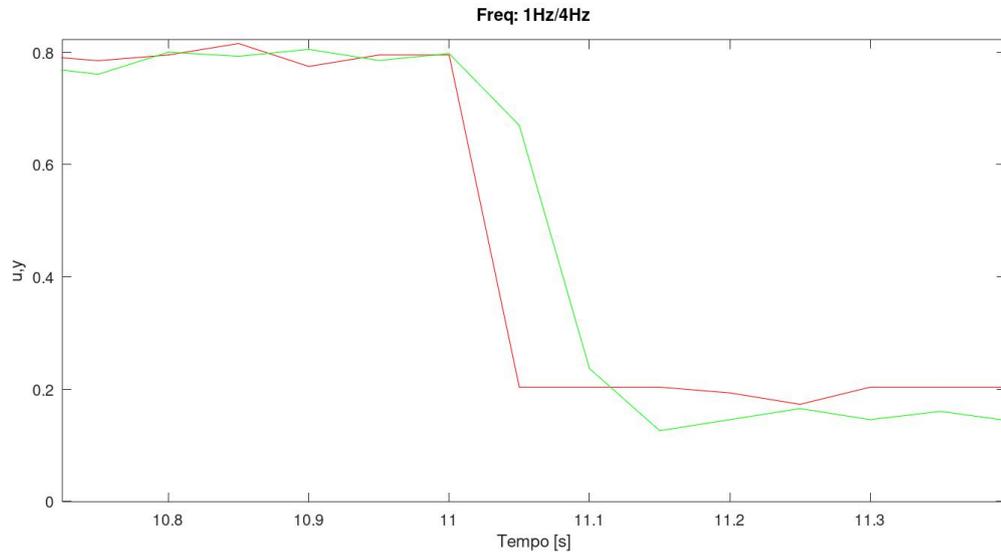


Figura 4.30: Sub-elevação em Regime Transitório de 1 para 4 Hz.

$$Sub_{4/1}[\%] = \left(\frac{0.12 - 0.18}{0.18 - 0.8} \right) * 100\% = 9.68\%, \text{ com } T_d = 0.15s$$

A Figura 4.31 representa o comportamento do controlador quando se dá o maior salto possível relativamente aos valores de setpoint, de 4 Hz para 7 Hz e vice-versa.

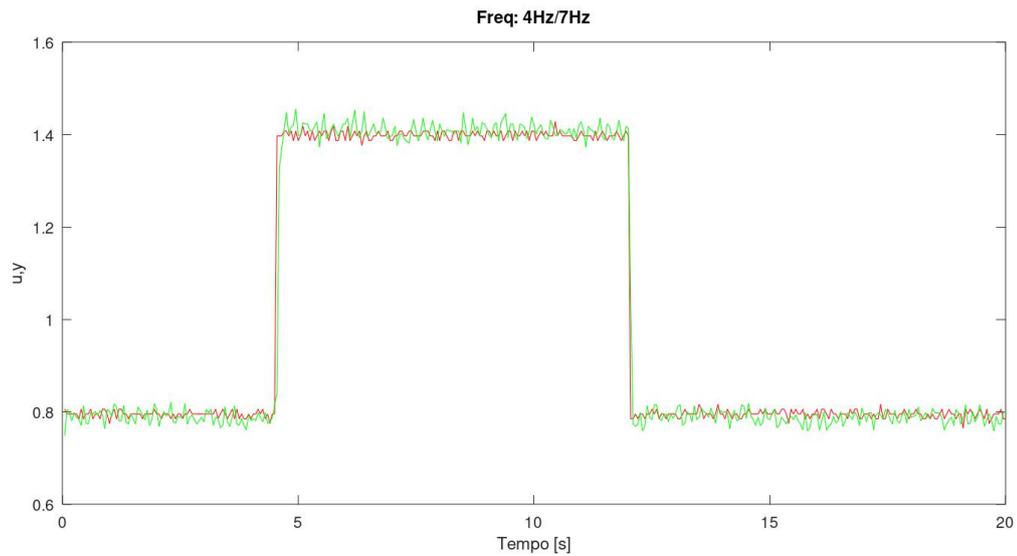


Figura 4.31: Comportamento em Regime Transitório de 4 para 7 Hz.

A Figura 4.32 permite calcular a sobre-elevação ocorrida na mudança de setpoint de 4 Hz para 7 Hz e visualizar o tempo de subida.

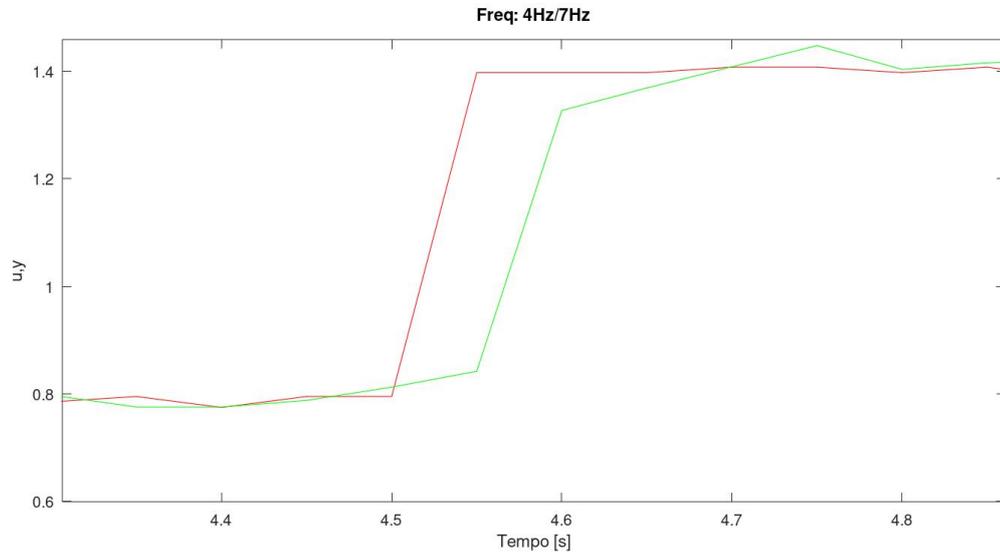


Figura 4.32: Sobre-elevação em Regime Transitório de 4 para 7 Hz.

$$Sob_{4/7}[\%] = \left(\frac{1.44 - 1.4}{1.4 - 0.82} \right) * 100\% = 6.9\%, \text{ com } T_s = 0.2s$$

A Figura 4.33 permite calcular a sub-elevação ocorrida na mudança de setpoint de 7 Hz para 4 Hz e visualizar o tempo de descida.

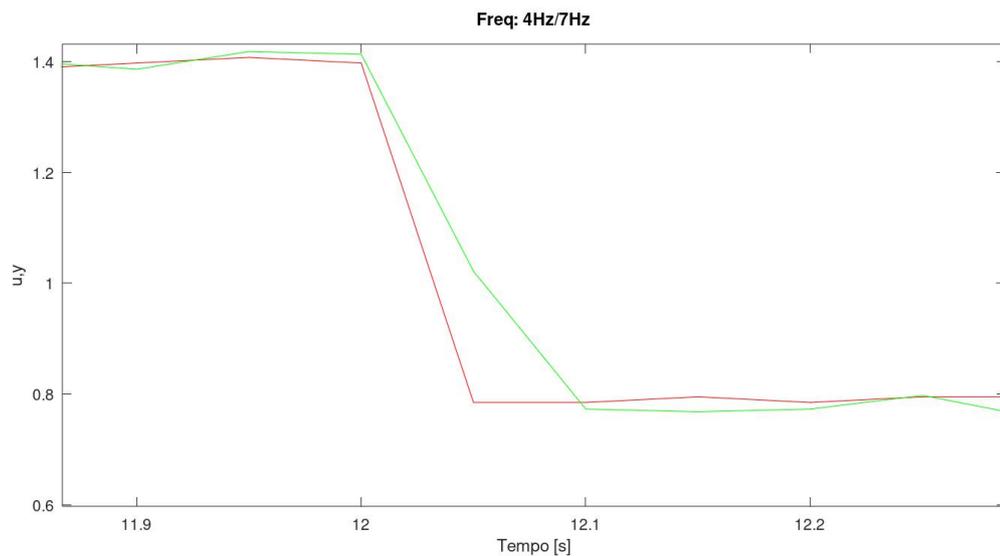


Figura 4.33: Sub-elevação em Regime Transitório de 4 para 7 Hz.

$$Sub_{7/4}[\%] = \left(\frac{0.78 - 0.8}{0.8 - 1.4} \right) * 100\% = 3.33\%, \text{ com } T_d = 0.1s$$

Através dos resultados obtidos, é possível verificar que os tempos de subida e descida são reduzidos e dentro do desejado, sendo que o mais elevado é o Tempo de Subida de 0.35s na transição do setpoint de 1 Hz para 4 Hz.

É ainda de notar que os valores de sobre-elevação e sub-elevação também são reduzidos e dentro do desejável (abaixo de 5%), excluindo o de sub-elevação na transição de setpoint de 4 Hz para 1 Hz e o de sobre-elevação na transição de setpoint de 4 Hz para 7 Hz.

É então possível concluir que, apesar de não ser ideal, o sensor dimensionado vai de encontro aos objetivos desejados para a realização do controlo em anel fechado.

CONCLUSÕES E TRABALHO FUTURO

5.1 Conclusões

Neste trabalho de dissertação o principal objetivo consistiu no desenvolvimento de um prototipo de ensino, através de um processo ciber-físico, para atividade letivas no âmbito da automação, nomeadamente no laboratório de automação do departamento de engenharia eletrotécnica e de computadores. Prototipo este que permite não só a programação lógica de processos, como a programação visual de sinópticos relativos a estes. Este objetivo foi cumprido resultando num processo adicional de ensino, tanto para as cadeiras da área de Controlo e Automação, como para futuros projetos de dissertação.

Um objetivo adicional correspondeu a uma aproximação de um ambiente industrial à escala laboratorial. Isto foi conseguido através da adição de componentes normalmente utilizados na indústria, nomeadamente as botoneiras, os sinais de alarme, o controlo do processo através da HMI, movimentação de componentes físicos (representada pela rotação do motor) e, inclusive, os equipamentos e materiais utilizados.

Outro objetivo seria a incorporação de diversos tipos de linguagens de programação, para uma compreensão mais ampla das mesmas dos alunos, mas foi decidido programar utilizando a linguagem mais visual (Ladder) para uma melhor acessibilidade de assimilação do funcionamento da programação lógica do processo.

Um último objetivo seria a implementação de um controlador externo em cascata com o controlador do variador de velocidade. Este controlador poderia ser implementado no próprio autómato, programado em linguagem SCL - Structured Control Language, através de um bloco de função ou num Arduino conectado ao autómato . Permitindo assim corroborar a funcionalidade do sistema ciber-físico como protótipo de ensino para os alunos do laboratório de Automação.

5.2 Trabalhos Futuros

Certas funcionalidade adicionais ao processo não foram desenvolvidas como seria desejado devido a limitações temporais e do hardware disponível no laboratório de automação.

Funcionalidades estas que estão descritas nos seguintes tópicos.

5.2.1 TZID

Para uma representação física da válvula hidráulica de esvaziamento, seria pertinente a incorporação no processo de um posicionador de válvula. Assim, além da visualização do funcionamento do motor, os alunos poderiam observar a abertura e fecho pneumáticos. O posicionados representado na Figura 5.1 encontra-se no laboratório de automação, mas para a sua incorporação no processo seria necessária a adição de um módulo de entradas e saídas analógicas, visto que o autômato já se encontra com todas as suas preenchidas. Este tipo de módulo acarreta um custo não viável.

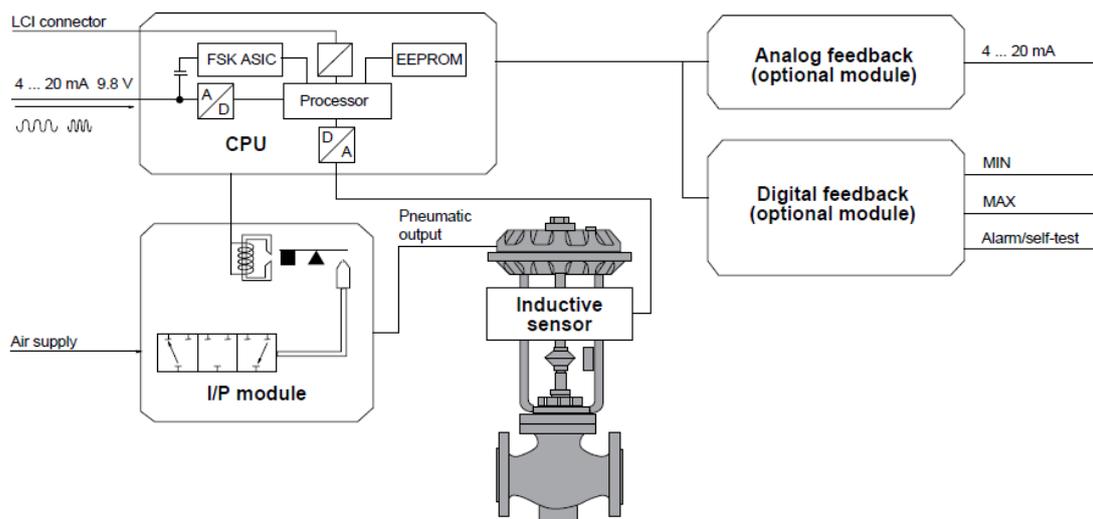


Figura 5.1: TZID [6].

5.2.2 IOT

Com o intuito de permitir a comunicação entre processos no laboratório de automação, dando assim ênfase ao ambiente industrial (vários processos a ocorrer na mesma rede local), seria pertinente a incorporação de um módulo IOT, representado na Figura 5.2. Esta incorporação não foi possível pois o autômato utilizado no processo tem uma versão de firmware inferior à necessária para a comunicação entre estes equipamentos. A compra de um novo autômato seria ainda mais dispendiosa que a compra do módulo analógico referido a cima.



Figura 5.2: Módulo IOT [34].

Com a incorporação deste módulo Iot seria ainda possível acoplar um módulo de comunicação com arduino ao mesmo. Permitindo assim ainda mais funcionalidades, como controlo do processo através do arduino, implementação de diversos tipos de controladores, entre outros. Este módulo encontra-se representado na Figura 5.3.



Figura 5.3: Módulo Arduino IOT [27].

BIBLIOGRAFIA

- [1] • *without battery Yes* (ver p. 30).
- [2] “1LA7060-4AB10_{datasheet_e}n”. Em: () (ver p. 35).
- [3] 6AV6647-0AA11-3AX0 - Siemens Simatic HMI Basic Panel - PLC-City. URL: <https://www.plc-city.com/shop/en/siemens-simatic-hmi-basic-panels-1st-generation/6av6647-0aa11-3ax0.html> (acedido em 05/03/2021) (ver p. 19).
- [4] “AEG-Process-705”. Em: () (ver p. 27).
- [5] *An Overview of Sequential Function Chart (SFC) Programming - Technical Articles*. URL: <https://control.com/technical-articles/an-overview-of-sequential-function-chart-sfc-programming/> (acedido em 04/03/2021) (ver p. 14).
- [6] F. Asic e E. D. Lci. *Construction and mode of operation Inductive sensor I/P module Analog feedback (optional module) Digital feedback (optional module) TZID Intelligent Positioner The concept* (ver p. 93).
- [7] *Basic Instructions of a PLC | PLC Technician*. URL: <https://www.plctechnician.com/news-blog/basic-instructions-and-operation-plc-part-1> (acedido em 02/03/2021) (ver p. 8).
- [8] *BMXCPS3522 - redundant power supply module X80 - 125 V DC | Schneider Electric Global*. URL: <https://www.se.com/ww/en/product/BMXCPS3522/redundant-power-supply-module-x80---125-v-dc/> (acedido em 02/03/2021) (ver p. 8).
- [9] *Botoneira*. URL: <https://www.arc-components.com/0-657-03-three-button-single-pole-two-open-no-one-closed-nc-contacts-6451.html> (ver p. 39).
- [10] *Buzzer*. URL: <https://www.amazon.co.uk/12v-Orange-Revolving-Beacon-Light/dp/B076F5T5KP> (ver p. 40).
- [11] *Celebrating 50 years of innovation and leadership | Schneider Electric*. URL: <https://www.se.com/my/en/about-us/events/modicon50years.jsp> (acedido em 01/03/2021) (ver p. 6).

- [12] *CLP SIMATIC S7-1200 | Controladores SIMATIC | Siemens*. URL: <https://new.siemens.com/br/pt/produtos/automacao/controladores/s7-1200.html> (acedido em 05/03/2021) (ver p. 18).
- [13] *CP1L | OMRON, Portugal*. URL: <https://industrial.omron.pt/pt/products/cp11> (acedido em 03/03/2021) (ver p. 11).
- [14] *Data sheet 6GK7277-1AA00-0AA0*. URL: <http://www.siemens.com/tia-selection-tool> (ver pp. 34, 35).
- [15] *Degree of pollution according to IEC 60664 / IEC 60947-2 3 Altitude 2000 m Storage/transport temperature-40... 80 °C Subject to technical modifications* (ver p. 41).
- [16] S. Descriptions. *Cont Content ents s USB-6008 and USB-6009 Features 2 ni.com*. 2024 (ver p. 74).
- [17] *EmergencyButton*. URL: <https://twineagleimports.com.au/products/telemecanique-emergency-stop-x-1> (ver p. 40).
- [18] E. Enoiu. "Programming languages popularity and implications to testing programmable logic controllers". Em: *PeerJ* 3.2013 (2015). DOI: [10.7287/peerj.preprints.879](https://doi.org/10.7287/peerj.preprints.879) (ver p. 14).
- [19] *Function Block Diagram (FBD) Programming Tutorial | PLC Academy*. URL: <https://www.plcacademy.com/function-block-diagram-programming/> (acedido em 04/03/2021) (ver p. 13).
- [20] *History of the PLC | Library.AutomationDirect.com |*. URL: <https://library.automationdirect.com/history-of-the-plc/> (acedido em 01/03/2021) (ver pp. 4, 5).
- [21] "HMI₆AV66470AD113AX0_{datasheet_e}n". Em: () (ver p. 34).
- [22] *Hoja de datos 6ES7232-4HA30-0XB0* (ver p. 33).
- [23] *I/O module | MITSUBISHI ELECTRIC FA*. URL: <https://eu3a.mitsubishielectric.com/fa/en/products/cnt/plcq/items/io/> (acedido em 02/03/2021) (ver p. 9).
- [24] S. Industry. "S7- 1200 Programmable controller S7-1200 Programmable controller". Em: (2012) (ver p. 17).
- [25] S. Industry. "S7- 300 Programmable controller". Em: (2003) (ver p. 17).
- [26] S. Industry e O. Support. "S7 Communication between SIMATIC S7-1200 and S7-300". Em: () (ver p. 20).
- [27] *Input ranges • Voltage Yes; 0 to 10V* (ver p. 94).
- [28] T. Instruments. • *Low-Power Consumption • Wide Common-Mode and Differential Voltage Ranges • Low Input Bias and Offset Currents • Output Short-Circuit Protection • Low Total Harmonic*. 1992 (ver p. 68).
- [29] *Ladder Logic Examples and PLC Programming Examples*. URL: <https://www.plcacademy.com/ladder-logic-examples/> (acedido em 04/03/2021) (ver p. 15).

-
- [30] *Ladder Logic Examples and PLC Programming Examples*. URL: <https://www.plcacademy.com/ladder-logic-examples/> (acedido em 04/03/2021) (ver p. 15).
- [31] J. M. Lourenço. *The NOVAthesis L^AT_EX Template User's Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/master/template.pdf> (ver p. ii).
- [32] S. Manual. *S7-1200 Programmable controller*. 2012. URL: <http://www.siemens.com/automation/> (ver p. 29).
- [33] *Mini is more*. URL: <http://www.puls-power.com> (ver p. 41).
- [34] *Number of PROFINET interfaces 2 USB port 2x USB 2.0 Connection for keyboard/mouse USB serial interface 1x COM (1x RS 232 / 422 / 485) Video interfaces • Graphics interface 1x DisplayPort Industrial Ethernet • Industrial Ethernet interface 2x Ethernet (RJ45)* (ver p. 94).
- [35] J. Oliveira, Paulo Jorge; Gaspar. *PLC Programming Languages (Structured Text)*. 2013 (ver p. 12).
- [36] O. Paulo Jorge e G. José. “PLC Programming Languages (Instruction List)”. Em: (2013) (ver p. 12).
- [37] *Programming Guideline for S7-1200/1500 TIA Portal*. URL: <https://www.siemens.com/industrialsecurity>. (ver pp. 44–48).
- [38] K. L. Richards e K. L. Richards. “Squirrel-Cage Motors”. Em: *Design Engineer's Sourcebook* (2020) (ver p. 16).
- [39] E. Ryan e M. Omar. “A review on the applications of programmable logic controllers (PLCs)”. Em: *Renewable and Sustainable Energy Reviews* (2016) (ver pp. 1, 8–10).
- [40] R. T. Sataloff, M. M. Johns e K. M. Kost. “IEC61131-3: Programming Industrial Automation Systems”. Em: () (ver p. 11).
- [41] V. R. Segovia e A. Theorin. “History of Control History of PLC and DCS”. Em: (2013) (ver pp. 1, 4, 5).
- [42] *Siemens - Mounting Rail for use with SIMATIC S7-300 Series | RS Components*. URL: <https://au.rs-online.com/web/p/plc-accessories/4532567/> (acedido em 02/03/2021) (ver p. 7).
- [43] Siemens AG. “SIMATIC IOT2000 Operating Instructions”. Em: (2016) (ver p. 21).
- [44] *SIMATIC IOT2000 | SIMATIC IOT gateways | Siemens Global*. URL: <https://new.siemens.com/global/en/products/automation/pc-based/iot-gateways/iot2000.html> (acedido em 05/03/2021) (ver p. 21).
- [45] Sinamics. “Sinamics G110”. Em: (2002) (ver p. 17).
- [46] *sinamics SINAMICS G110* (ver pp. 36–38).

- [47] I. Standard 11266. *International Standard International Standard*. Vol. 2014. 2014 (ver p. 19).
- [48] *Standard environment - Modicon M340 | Schneider Electric Global*. URL: <https://www.se.com/ww/en/product-range/1468-modicon-m340/> (acedido em 05/03/2021) (ver p. 22).
- [49] *Take Benefits of Siemens PLC S7-300 Product Specification and Supporting Device for innovative system solutions in Manufacturing Industries*. URL: <https://program-plc.blogspot.com/2016/01/take-benefits-of-siemens-plc-s7-300.html> (acedido em 05/03/2021) (ver p. 18).
- [50] *The top most used PLC Systems around the world*. URL: <http://engineering-electrical-equipment.org/electrical-distribution/the-top-most-used-plc-systems-around-the-world.html> (acedido em 01/03/2021) (ver pp. 6, 7).
- [51] *Top 5 PLC programming languages. Before we go through the 5 PLC... | by Raji Ayinla | The Open Manuel | Medium*. URL: <https://medium.com/the-open-manuel/top-5-plc-programming-languages-edf4de2dc3e4> (acedido em 03/03/2021) (ver pp. 12, 13, 15).
- [52] S. Variator. "Operating Instructions Edition 04/2005". Em: (2005) (ver p. 17).

