

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

António Mário Costa Ribeiro
Marcelo Amaral Martins

Automatização de um processo industrial



julho de 2023

António Mário Costa Ribeiro
Marcelo Amaral Martins

Automatização de um processo industrial

Relatório da UC Projeto
Licenciatura em Engenharia Eletrotécnica

Professor Doutor António Ferrolho
Professor Mestre João Ôlas

julho de 2023



“Nenhum vento sopra a favor para quem não sabe para onde ir”
Sêneca

RESUMO

O nosso trabalho consiste na automatização de um processo industrial. O aumento da procura por parte do consumidor originou a necessidade de aumento da produção e a necessidade de reduzir custos e minimizar o impacto ambiental. Com o avanço da tecnologia, surgiram as Células Flexíveis de Fabrico (CFF) que são sistemas automatizados que utilizam autómatos industriais, robôs, computadores, sensores e atuadores para gerir e controlar o processo de fabrico sem a necessidade de intervenção humana por longos períodos de tempo.

O relatório descreve o desenvolvimento de um processo industrial automatizado no Laboratório de Automação e Robótica da Escola Superior de Tecnologia e Gestão de Viseu (ESTGV) utilizando as várias ferramentas de software e hardware disponíveis.

ABSTRACT

Our work consists of automating an industrial process. The increase in demand on the part of the consumer originated the need to increase production and the need to reduce costs and minimize the environmental impact. With the advancement of technology, Flexible Manufacturing Cells (CFF) have emerged, which are automated systems that use industrial automata, robots, computers, sensors and actuators to manage and control the manufacturing process without the need for human intervention for long periods of time.

The report describes the development of an automated industrial process at the Laboratory of Automation and Robotics at the Escola Superior de Tecnologia e Gestão de Viseu (ESTGV) using the various software and hardware tools available.

PALAVRAS-CHAVE

Célula Flexível de Fabrico

Buffer

Transportador

Produto

Produção

Robô

Autômato

Armazém

Fabrico

KEY WORDS

Flexible Manufacturing Cell

Buffer

Transporter

Product

Production

Robot

Automation

Storage

Manufacture

AGRADECIMENTOS

Este projeto conclui uma etapa importante no nosso percurso académico e não teria sido possível sem a colaboração e ajuda de muitas pessoas. Queremos expressar a nossa gratidão e solidariedade a todos os que, de forma direta ou indireta, partilharam connosco este ciclo. Agradecemos à Escola Superior de Tecnologia e Gestão de Viseu por fornecer todo o material necessário para colocarmos em prática os nossos conhecimentos.

Agradecemos ao Professor Doutor António Ferrolho, nosso orientador, e ao Mestre João Ôlas, nosso coorientador, pela disponibilidade, atenção, momentos de aprendizagem e críticas construtivas que nos ajudaram a evoluir.

Agradecemos também às nossas famílias, que nos motivaram e nos deram força para concluirmos esta tarefa com sucesso.

A todos, muito obrigado.

ÍNDICE GERAL

ÍNDICE DE FIGURAS	xv
Índice de tabelas	xvii
ABREVIATURAS E SIGLAS	xix
1. Introdução.....	1
1.1 Motivo da Escolha do Projeto.....	1
1.2 Objetivos do Projeto	1
2. Estado da Arte	3
2.1 Célula Flexível de Fabrico do Laboratório de Robótica e Automação.....	3
2.2 <i>Layout</i> CFF e respetiva localização de equipamentos	4
2.2.1 Autómato	6
2.2.2 Setor de fabrico.....	8
2.2.3 Setor de Armazém	11
2.2.4 Ferramenta de Trabalho.....	14
2.2.5 Transportador.....	15
3. Proposta de Solução	21
3.1 Descrição do funcionamento	21
3.2 Graficets	23
3.2.1 Tabelas de Processo.....	23
3.2.2 Equações	26
3.2.3 Programa em diagrama de contactos	27
3.2.4 Melhorias ao programa.....	30
3.3 Robôs IRB 1400 e IRB 140 da ABB	32
3.3.1 Dificuldades.....	32
3.3.2 Solução	32
3.3.3 Robô IRB 140.....	32
3.3.4 Robô IRB 1400.....	42
3.4 Registo Carregamento de Paletes.....	54
3.5 Alarmes	55
3.6 Sistema de Segurança	56

3.6.1	Fluxograma do Sistema de Segurança	56
3.6.2	Programação RobotStudio do Sistema de Segurança	57
4.	Conclusão e propostas para trabalhos futuros.....	61
4.1	Conclusão	61
4.2	Propostas para trabalhos futuros.....	61
	Referências.....	63
	Anexo 1 - <i>Grafcet</i> principal (FirstScan e Largada temporizada das Paletes	65
	Anexo 2 – <i>Grafcet</i> Ordem de Carregar Paleta MPT no Robô IRB 140.....	67
	Anexo 3 - <i>Grafcet</i> Paleta MPT Carregada no Robô IRB 1400.....	69

ÍNDICE DE FIGURAS

Figura 1- <i>Layout</i> CFF do LAR	4
Figura 2 - PLC SIMATIC S7-1200 CPU 1215C AC/DC/RLY	6
Figura 3 - Fresadora EMCO Concept MILL 155	8
Figura 4 - Torno EMCO Concept TURN 55	8
Figura 5 - Setor de Fabrico da CFF	9
Figura 6 - Volume de trabalho do Robô IRB 140 da ABB	10
Figura 7 - Controlador S-721 68 do robô IRB140 da ABB	11
Figura 8 - Setor Armazém da CFF e Descarga de Produtos Acabados	11
Figura 9 - Volume de trabalho do robô IRB 1400 da ABB.....	12
Figura 10 - Características dos robôs IRB 1400 e IRB 140	13
Figura 11 - Disposição dos elementos do setor de armazém.....	13
Figura 12 – Peças Matéria Prima Fresadora.....	14
Figura 13 - Peças Matéria Prima Torno	14
Figura 14 – Ferramenta de Trabalho do robô IRB 1400	15
Figura 15 - Transportador da CFF.....	16
Figura 16 - Representação das Paletes	17
Figura 17 - Disposição das chapas metálicas nas paletes.....	17
Figura 18 - Representação da localização dos sensores indutivos	18
Figura 19 - Representação da localização dos <i>Stoppers</i>	18
Figura 20 – <i>FirstScan</i> na <i>Network 2</i>	27
Figura 21 – Temporizador TON.....	28
Figura 22 – Pulse Timing Diagrame (Diagrama de tempo de pulso) TON.....	28
Figura 23 – Temporizador TP	29
Figura 24 - Pulse Timing Diagrame (Diagrama de tempo de pulso) TP.....	29
Figura 25 – <i>Network 2</i> : Largada de Paletes no Arranque	30
Figura 26 - Adição ao programa da rotina Carrega Paleta Torno	31
Figura 27 - Mensagem Falta Peças.....	31
Figura 28 - Fluxograma Programa Principal IRB 140	34
Figura 29 – Programação do Programa Principal RobotStudio IRB 140.....	35
Figura 30 - Fluxograma da rotina Descarrega Paleta Torno para o <i>buffer</i> da matéria-prima do IRB 140	36
Figura 31 – Programação da rotina Descarrega Paleta Torno para o <i>buffer</i> da matéria-prima do IRB 140	37
Figura 32 - Fluxograma da Rotina Transfere Peças do <i>buffer</i> matéria-prima para o <i>buffer</i> de produto acabado do IRB 140.....	38
Figura 33 - Programação da Rotina Transfere Peças do <i>buffer</i> matéria-prima para o <i>buffer</i> de produto acabado do IRB 140.....	39

Figura 34 – Fluxograma da rotina carrega peças do <i>buffer</i> do produto acabado para a palete PAT do IRB 140	40
Figura 35 - Programação RobotStudio da rotina carrega peças do <i>buffer</i> do produto acabado para a palete PAT do IRB 140	41
Figura 36 - Fluxograma Programa Principal IRB 1400.....	44
Figura 37 – Programação do Programa Principal RobotStudio IRB 1400	45
Figura 38 - Programação Principal RobotStudio IRB 1400	46
Figura 39 - Fluxograma da rotina Verifica Palete Torno IRB 1400	47
Figura 40 – Programação da rotina Verifica Palete Torno IRB 1400.....	48
Figura 41 – Fluxograma da rotina Descarrega Palete Torno para a rampa IRB 1400.....	49
Figura 42 – Programação da rotina Descarrega Palete Torno IRB 1400.....	50
Figura 43 - Fluxograma da rotina Carrega Palete Torno IRB 1400.....	51
Figura 44 – Programação da rotina Carrega Palete Torno IRB 1400	52
Figura 45 – Continuação da Programação da rotina Carrega Palete Torno IRB 1400	53
Figura 46 – Registo Carregamento das Paletes.....	54
Figura 47 – Alarmes.....	55
Figura 48 – Fluxograma do Sistema de Segurança.....	56
Figura 49 – Programação do Sistema de Segurança	57
Figura 50 – Menu Visualização Multitask.....	58
Figura 51 – Sensores Óticos dos Robôs.....	58
Figura 52 - Botoneira para executar a marcha e paragem.....	59
Figura 53 - Sinalizador luminoso em marcha c/ lâmpada verde acesa	59
Figura 54 – Sinalizador luminoso em paragem c/ lâmpada vermelha acesa	59

ÍNDICE DE TABELAS

Tabela 1 - Interface dos <i>Stoppers</i>	19
Tabela 2 - Interface Robótica	22
Tabela 3 - Identificação e descrição das recetividades externas	23
Tabela 4 - Identificação e descrição das recetividades internas	24
Tabela 5 - Identificação e descrição das ações externas.....	25
Tabela 6 - Multiplexagem das entradas do IRB140	33
Tabela 7 - Interface robótica das Entradas e Saídas do robô IRB 140	33
Tabela 8 - Interface robótica dos sensores no setor de fabrico.....	34
Tabela 9 - Multiplexagem das entradas do IRB1400	42
Tabela 10 - Interface robótica do IRB 1400	42
Tabela 11 - Interface robótica dos sensores no setor de Armazém	43

ABREVIATURAS E SIGLAS

CFF - Célula Flexível de Fabrico
IPV - Instituto Politécnico de Viseu
ESTGV - Escola Superior de Tecnologia e Gestão de Viseu
CNC - Controle Numérico Computadorizado
AGV - Veículo Guiado Automaticamente
TIA - Totally Integrated Automation Portal
GRAFCET - Gráfico Funcional de Comando Etapa Transição
PLC - Programmable Logic Controller
MPT - Matéria Prima Torno
MPF - Matéria Prima Fresadora
PAT - Produto Acabado Torno
PAF - Produto Acabado Fresadora
LAR - Laboratório de Automação e Robótica
DEE - Departamento de Engenharia Eletrotécnica
SCADA - Sistemas de Supervisão e Aquisição de Dados

1. Introdução

1.1 Motivo da Escolha do Projeto

Cada vez mais, as empresas do setor industrial estão interessadas em usar sistemas robóticos nas suas linhas de produção e investir recursos financeiros significativos no desenvolvimento e aquisição desses sistemas, pois são de baixa manutenção, podem operar em qualquer ciclo de horário (24/24 horas) e permite também uma baixa quantidade mão de obra.

Por esse motivo, focámo-nos na automatização de um processo industrial, pois existe uma procura crescente por profissionais especializados nessa área. Ao trabalhar neste projeto, adquirimos conhecimentos e habilidades para atender às necessidades atuais da indústria de forma criativa, inovadora e flexível.

1.2 Objetivos do Projeto

Os principais objetivos para este projeto foram executar a programação do autômato para que possa realizar a gestão no que toca a saber o tipo de palete nos pontos de paragem, tal como se as mesmas estão a deslocar-se carregadas ou vazias, bem como dar as ordens aos robôs IRB1400 e IRB140 de carregamento e descarregamento das respetivas paletes mediante o tipo de peça, e a transferência entre *buffers* das mesmas anulando assim o que seria a fresagem e o tornear nas respetivas máquinas.

Relativamente às ações de carregamento, descarregamentos, e transferência de peças são situações que cabem aos robôs, robôs esses que foram programados para que executem as ordens dadas pelo autômato.

Também tínhamos o objetivo de implementar medidas de segurança, fisicamente ao longo da própria Célula Flexível de Fabrico (CFF) através de sensores, para evitar a intrusão na zona de trabalho dos robôs.

2. Estado da Arte

2.1 Célula Flexível de Fabrico do Laboratório de Robótica e Automação

Uma célula flexível de fabrico (CFF) é um sistema automatizado de produção que combina vários dispositivos, como autómatos, computadores, sensores e atuadores, para gerir e controlar o processo de fabrico de um produto. As CFF são projetadas para fornecer uma produção sofisticada e eficiente para uma ampla variedade de produtos, sem a necessidade de intervenção humana por longos períodos de trabalho. Esses sistemas são capazes de funcionar continuamente, 24 horas por dia, sem cansaço comparativamente à produção com mão de obra humana, o que resulta em redução de custos e menor impacto para o ambiente.

As máquinas, por outro lado, podem trabalhar continuamente com a manutenção adequada e programada e podem levantar cargas pesadas sem auxílio humano.

A redução da intervenção humana através de um controlo adequado resulta na diminuição dos tempos de produção, maior precisão dos trabalhos executados e maior controlo da produção, isso é essencial para atender a procura exigida nos países desenvolvidos.

As CFF são amplamente utilizadas em vários setores da indústria e podem integrar diferentes equipamentos, de diferentes fabricantes, tais como:

- Robôs;
- Autómatos;
- Transportadores (tapetes, esteiras, AVG's);
- Máquinas CNC;
- Sistemas de Visão;
- Computadores;
- HDMI.

O grande desafio da integração destes equipamentos passa pelo desenvolvimento de uma interface compatível com diferentes fabricantes e de um *layout* adequado às necessidades.

2.2 Layout CFF e respetiva localização de equipamentos

No Laboratório de Automação e Robótica (LAR), do Departamento de Engenharia Eletrotécnica (DEE) da Escola Superior de Tecnologia e Gestão de Viseu (ESTGV), está implementada a CFF, representada na figura 1.

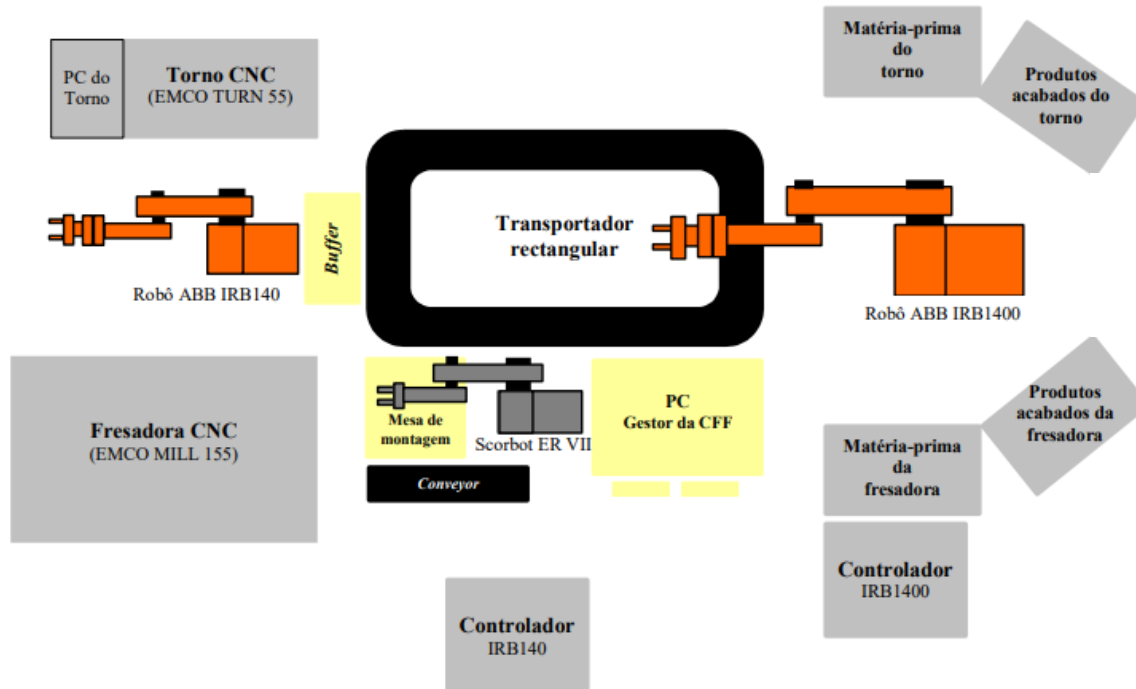


Figura 1- Layout CFF do LAR

- **Transportador rectangular**

Localizado no centro do *layout*, permite interligar fisicamente todos os setores envolvidos na CFF. Nele movimentam-se as paletes, cada uma com ranhuras adequadas a cada tipo de peça utilizada, permitindo transportar até 4 peças de cada vez, desde o setor de armazém ao setor de fabrico e vice-versa.

- **Setor de armazém**

Localizado à direita do *layout*, com o robô IRB 1400 da ABB no centro, facilitando todo o processo de descarga de matérias-primas para as paletes e descarga do produto acabado na rampa de descarga.

- **Buffer do setor de fabrico**

Instalado no centro das máquinas CNC junto ao robô IRB 140, permite armazenar provisoriamente as matérias-primas para o processo de fabrico e armazenar temporariamente os produtos acabados para transporte.

- Rampa de descarga

Encontra-se junto ao local de paragem das paletes no setor de armazém. Neste projeto os produtos acabados serão depositados na rampa.

- Controladores dos robôs

Estão instalados fora do volume de trabalho dos robôs, permitindo que os operadores trabalhem em segurança, com o campo de visão para todo o layout.

- Autómato

O Autómato é responsável pela integração, controlo e supervisão de todas as operações que ocorrem na CFF.

Controla toda a operação efetuada no *layout* pelo que a sua disposição se encontra ao lado do computador.

2.2.1 Autômato

A gestão completa de todo o processo da CFF é realizada por meio de um PLC que controla/monitoriza os vários setores individualmente, e os comanda para executar as tarefas específicas de cada um.

No nosso projeto tivemos à escolha 2 modelos de autômatos, o modelo SIMATIC S7-1200 e o SIMATIC S7-200 ambos da SIEMENS, entretanto optámos pela utilização do S7-1200 por ter uma interface mais recente e pelo facto de ter mais funcionalidades.

Na figura 2 podemos visualizar o autômato utilizado neste projeto, nas referências pode ser consultado o respetivo *Datasheet* com os seus detalhes, nomeadamente:

- Fabricante Siemens;
- Modelo CPU 1215C AC/DC/RLY;
- 14 entradas digitais;
- 10 saídas digitais;
- Alimentação a 230V AC.

O SIMATIC S7-1200 suporta diferentes opções de conectividade em rede, como *Ethernet* e *PROFIBUS/PROFINET*. Através dessas interfaces, é possível estabelecer comunicação com outros dispositivos e sistemas, com Sistemas de Supervisão e Aquisição de Dados (*SCADA*), outros PLCs, sistemas de controle de processos, robôs, entre outros.

O SIMATIC S7-1200 suporta uma variedade de protocolos de comunicação padrão da indústria, como Modbus TCP/IP, OPC, UA, MPI (Multi-Point Interface), entre outros.

A programação do autômato foi executada através do software de programação TIA PORTAL V18, fornecido pela Siemens.



Figura 2 - PLC SIMATIC S7-1200 CPU 1215C AC/DC/RLY

Programação do Autómato

A programação do Autómato foi executada através do software TIA Portal, que é o mais adequado para desenvolver programas de Autómatos da Siemens.

Inicialmente, desenvolvemos a programação no TIA Portal para a zona de armazém e de seguida a programação para a zona do fabrico.

Os métodos utilizados na programação foram os que aprendemos ao longo do curso:

- Método do *grafcet*;
- Elaboração das tabelas;
- Elaboração das equações de cada etapa;
- Desenvolvimento do programa em software TIA Portal através do diagrama de contactos.

Dentro da programação, definimos um método de indicar o estado das paletes e dos *buffers* (cheios ou vazios). Para isso, utilizámos *bits* de memória, ou seja, a cada paleta e a cada *buffer* atribuímos um *bit*, em que se o sinal lógico estiver a 1 indica que o estado é cheio, se estiver a 0 indica que o estado é vazio. Os *bits* de memória atribuídos encontram-se na tabela 4.

Dificuldades na Programação do Autómato

Inicialmente tivemos dificuldades na paragem das paletes pois estas passavam muito rápido nos sensores. Sensores estes que fazem atuar os *stoppers* e não permitiam que fosse realizada a leitura correta dos códigos das paletes, fazendo com que a paleta não parasse no local correto.

Para solucionar este problema utilizámos um bloco TON (temporizador ao trabalho), com 100ms para que, quando os 4 sensores estivessem ativos durante este tempo só assim fizesse atuar os *stoppers* evitando as leituras falsas que são efetuadas enquanto a paleta passa rapidamente nos sensores.

2.2.2 Setor de fabrico

O setor de fabrico é onde se realiza o processo de maquinação de matéria-prima em produto acabado. O processo de maquinação ocorre com recurso à utilização de uma fresadora e um torno, com o auxílio de um robô, utilizando um *buffer* como ponto de transição de produtos.

As máquinas-ferramentas presentes neste *layout* são de controlo numérico computadorizado, mais conhecidas como CNC.

Na figura 3 e 4 podemos observar as CNC's presentes no *layout* da CFF, não as usámos na realização do projeto, mas fizemos a simulação das mesmas transferindo as peças do *buffer* de matéria-prima para o *buffer* de produtos acabados.



Figura 3 - Fresadora EMCO Concept MILL 155



Figura 4 - Torno EMCO Concept TURN 55

Na figura 5 é possível visualizarmos o robô IRB 140 da ABB localizado no setor de fabrico, temos também presente as 2 CNC's (Fresadora e Torno), e os *buffers* da matéria-prima e do produto acabado.



Figura 5 - Setor de Fabrico da CFF

As suas funções do robô, no nosso projeto, são:

- 1º. Descarregar as peças da paleta do transportador para os respetivos *buffers* de matéria-prima, torno ou fresadora;
- 2º. Mover as peças do *buffer* da matéria-prima para o *buffer* do produto acabado.
- 3º. Carregar as peças do *buffer* do produto acabado para a respetiva paleta no transportador.

Características do robô IRB 140 da ABB

O robô IRB 140 é um modelo industrial de robô articulado, desenvolvido pela empresa ABB, está equipado com um controlador S4C+ que permite a utilização de instruções RAPID, tornando mais flexíveis e práticas as operações de programação.

Ele é capaz de realizar uma grande variedade de tarefas em linhas de produção, como manipulação de peças, montagem, embalagem, soldagem, entre outras. Ele possui alta precisão, velocidade e alcance, além de ser compacto e modular, permitindo sua adaptação a diferentes *layouts* de produção. O IRB 140 é um robô de 6 eixos, com capacidade de carga de até 6 kg, e é amplamente utilizado em diversos setores industriais, como automobilístico, metalúrgico, alimentício, entre outros.

A figura 6 apresenta o volume de trabalho do Robô IRB 140 da ABB

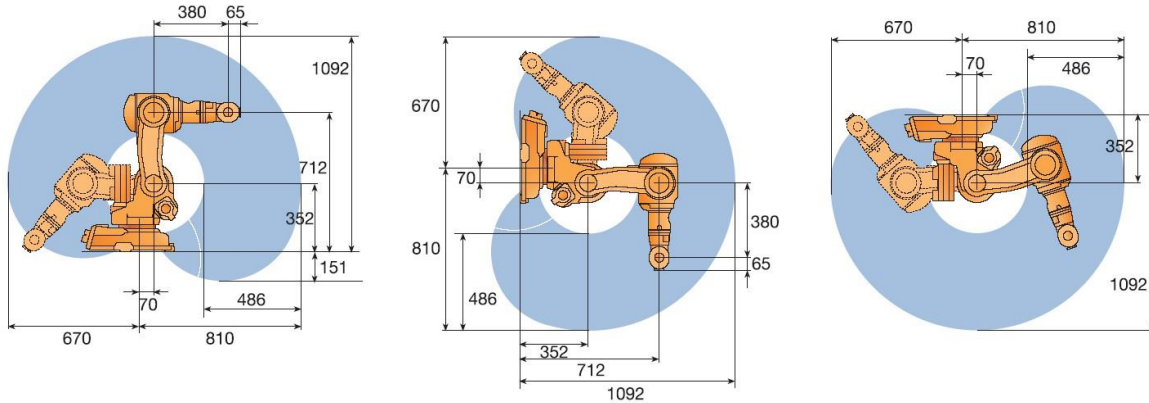


Figura 6 - Volume de trabalho do Robô IRB 140 da ABB

Como qualquer outro robô industrial, o ABB IRB 140 é controlado por um sistema eletrônico computadorizado, o controlador, responsável por controlar toda a estrutura física, coordenando a ação dos motores das juntas de acordo com o código do programa que está a ser executado, em conjunto com a informação captada pelos sensores de posição e velocidade. Além de controlar os movimentos do manipular, o controlador é também capaz de armazenar informação, bem como algoritmos de controlo e planeamento de trajetórias que descrevem a estrutura mecânica e o respetivo movimento.

São sistemas de utilização essencialmente local, que disponibilizam uma linguagem de programação estruturada (no caso Rapid), livrarias de funções de manipulação e controlo de posição avançadas, possibilidade de gestão de programas e ficheiros, bem como dispositivos de interface com o utilizador. Neste último, a interface é feita através de uma HMI, agregada ao controlador, dotada de um *display* de 16 linhas, um teclado, botão de paragem de emergência e um *joystick*. Através desta interface é possível controlar o manipulador manualmente e ler possíveis erros no código ou avisos internos do sistema.

O controlador associado ao robô IRB 1400 é um ABB S-721 68 representado na figura 7.



Figura 7 - Controlador S-721 68 do robô IRB140 da ABB

2.2.3 Setor de Armazém

O setor de armazém é onde se realiza o processo de reposição de matérias-primas nas paletes e o processo de descarregamento de produtos acabados na rampa. O processo ocorre com recurso à utilização do robô IRB 1400 da ABB, representado na figura 8.



Figura 8 - Setor Armazém da CFF e Descarga de Produtos Acabados

As suas funções, no nosso projeto, são:

- 1°. Carregar as peças do torno e da fresadora a partir do armazém da matéria-prima para a respetiva paleta do transportador;
- 2°. Verificar o correto carregamento das mesmas através das fibras óticas e criar um ficheiro com a respetiva informação;
- 3°. Verificar a presença de peças na paleta do produto acabado e descarregar as mesmas para a rampa.

Características do robô IRB 1400 da ABB

O robô IRB 1400 caracteriza-se por ser uma unidade com movimentação muito rápida de elevada precisão. Movimenta-se segundo 6 eixos e as suas dimensões permitem um alcance considerável, comparativamente ao robô que se encontra no setor de fabrico. Tem capacidade para movimentar cargas até 5 Kg, está também equipado com um controlador S4C+ que permitindo utilizar a mesma linguagem de programação do robô IRB 140.

Este robô apresenta uma grande variedade de aplicações na indústria devido à sua grande capacidade em operações de alto desempenho e à sua flexibilidade de utilização. Este é comumente utilizado em diversas aplicações, tais como estações de soldadura, estações de pintura, montagem de peças, manuseamento de materiais, carga e descarga de produtos.

A figura 9 mostra o volume de trabalho do robô IRB 1400 da ABB

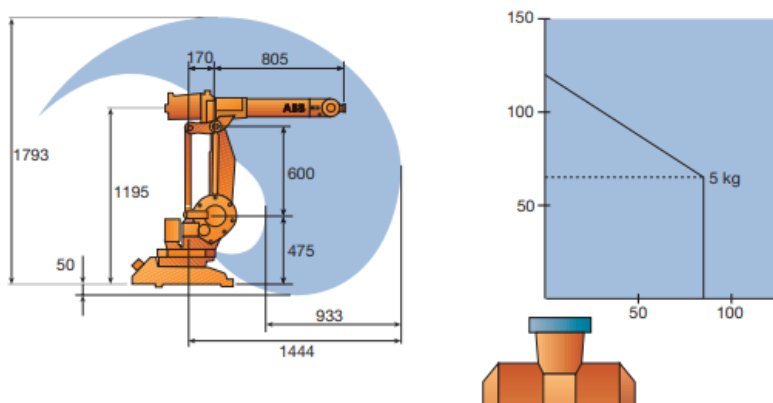


Figura 9 - Volume de trabalho do robô IRB 1400 da ABB

Na figura 10 são apresentadas as principais características dos robôs IRB 1400 e 140

Robô ABB IRB 1400	
Tipo	Antropomórfico
Número de Eixos	6
Movimento dos Eixos	
Eixo 1	+340° a -340°
Eixo 2	+140° a -140°
Eixo 3	+135° a -135°
Eixo 4	+300° a -300°
Eixo 5	+230° a -230°
Eixo 6	+600° a -600°
Velocidade dos eixos	
Eixo 1	110°/s
Eixo 2	110°/s
Eixo 3	110°/s
Eixo 4	280°/s
Eixo 5	280°/s
Eixo 6	280°/s
Dimensões da base	620 x 450 mm
Altura total	1310 mm
Peso	225 kg

Figura 10 - Características dos robôs IRB 1400 e IRB 140

O robô IRB 1400 utiliza o mesmo controlador que o IRB 140, portanto todas as características do controlador mencionado anteriormente mantêm-se neste.

Na figura 11 podemos observar a disposição dos armazéns no Laboratório de Automação e Robótica, no nosso projeto utilizámos os 2 Armazéns ilustrados abaixo (Armazém Peças Torno e o Armazém Peças Fresadora).

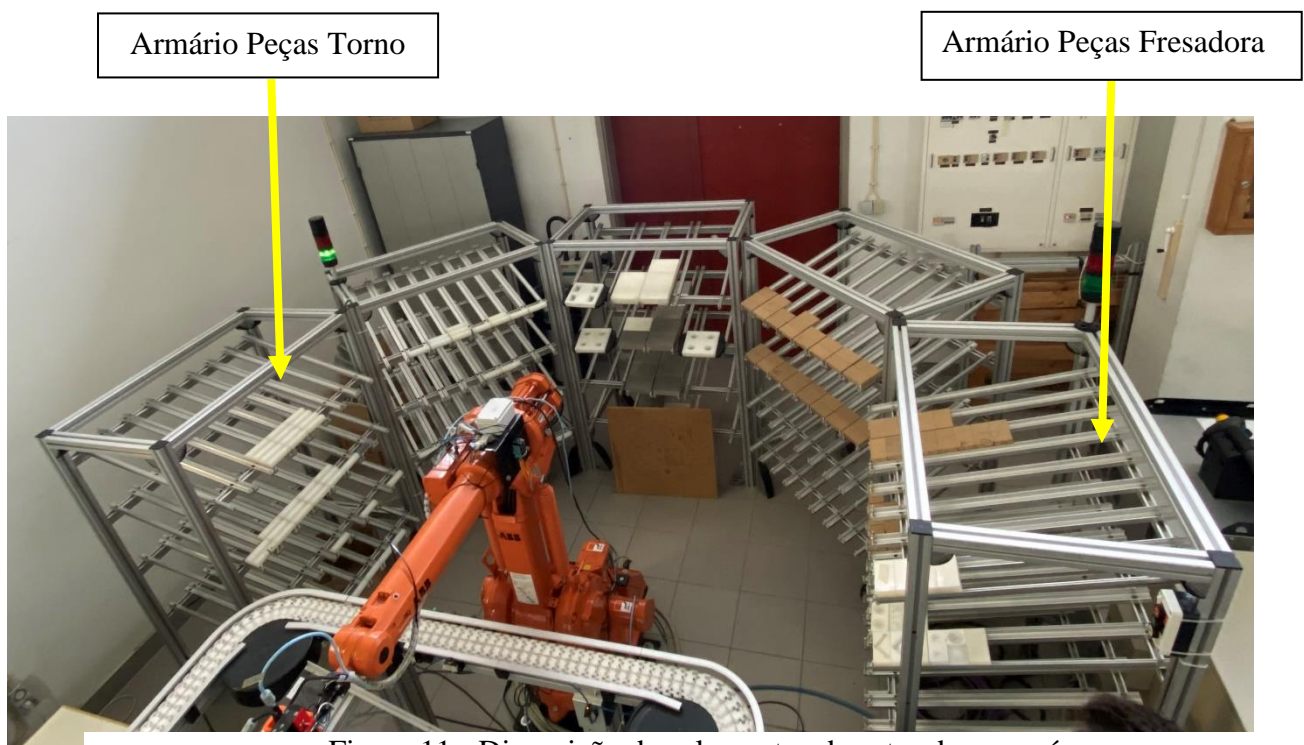


Figura 11 - Disposição dos elementos do setor de armazém

Neste setor temos, portanto, as peças da matéria-prima do torno e peças da matéria-prima da fresadora respectivamente ilustradas na figura 12 e 13.



Figura 13 - Peças Matéria Prima Torno



Figura 12 – Peças Matéria Prima Fresadora

2.2.4 Ferramenta de Trabalho

Tanto no manipulador do robô IRB 140 como no robô IRB 1400 está adicionada uma ferramenta de trabalho de seu nome pinça ou *gripper*.

A ferramenta que se adiciona ao manipulador deverá ser projetada conforme as funções para as quais o robô foi adquirido. No caso em estudo, a função é manipular peças com geometrias distintas, cilindros e paralelepípedos. Este tipo de ferramenta é acionada através de um sinal digital que faz atuar uma válvula pneumática. Quando o sinal está a 1, a ferramenta está fechada, quando o sinal está 0 esta está aberta. Na figura 14 é possível ver que na ferramenta instalada fazem parte sensores de fibra ótica e sensores indutivos. Estes sensores estão ligados ao controlador do robô permitindo ao utilizador usar os seus sinais nos programas que desenvolve. No caso em estudo são utilizados os sensores indutivos, através da combinação dos sinais destes sensores foi desenvolvido um algoritmo que permite após a abertura e fecho do gripper, saber se numa determinada posição do Armazém existe ou não uma peça, e as fibras óticas que nos permitem detetar a existência de peças nas paletes carregadas e guardar essa informação.

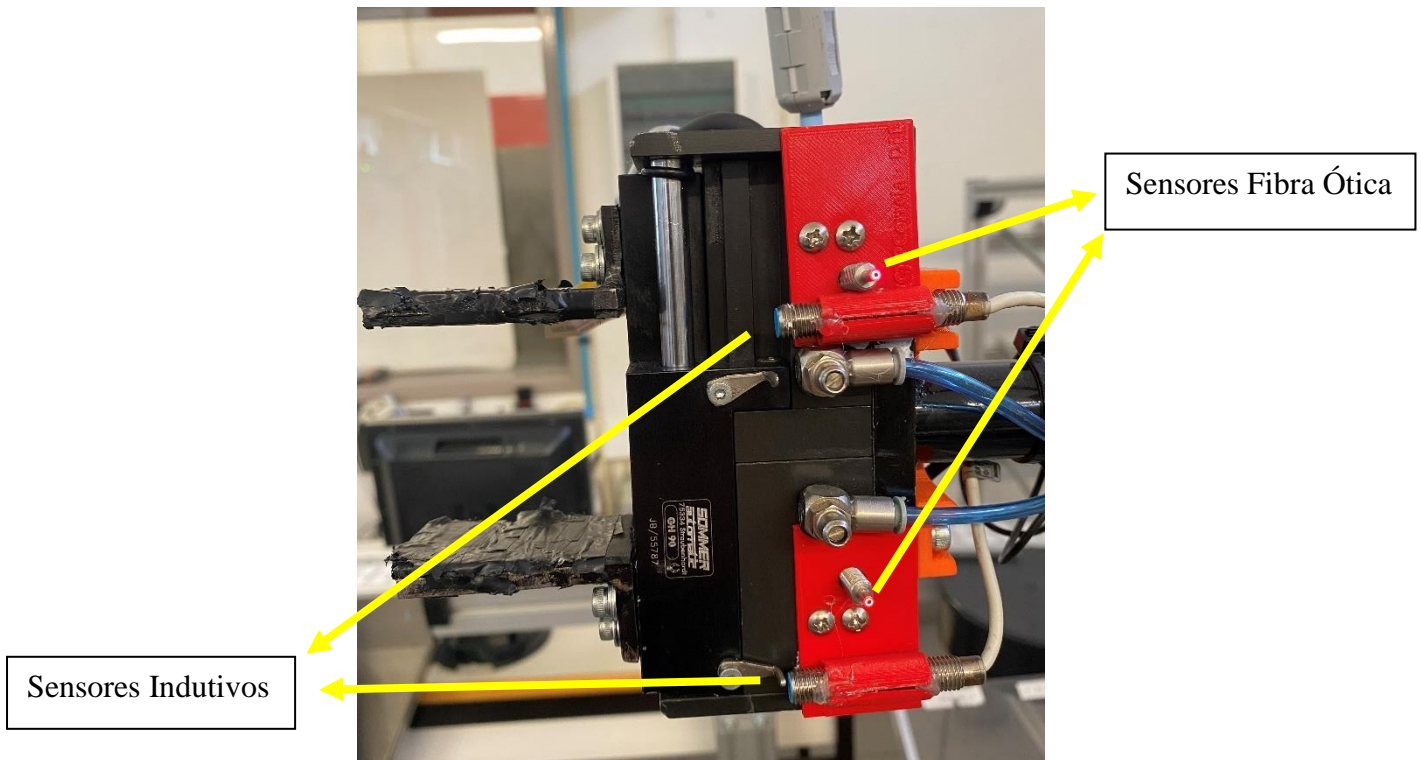


Figura 14 – Ferramenta de Trabalho do robô IRB 1400

A pinça do robô IRB 140 é exatamente igual só não dispõe das fibras óticas.

2.2.5 Transportador

A ligação entre os setores de armazém e de fabrico é garantida pelo transportador. Este pode ter diferentes características, dependendo das necessidades do projeto. Trata-se de um elemento extremamente importante, visto que o seu dimensionamento deve respeitar as necessidades dos diferentes setores do processo e as características dos materiais que vai transportar. Tanto a segurança, como a produtividade, dependem do funcionamento adequado dos sistemas de transporte, pois estão diretamente relacionados com a procura de matéria-prima e com o escoamento da produção.

Podemos afirmar que a eficiência e o funcionamento adequado do processo industrial se devem, em grande parte, à conceção de um design bem elaborado dos transportadores.

As soluções de transportadores que se encontram disponíveis na indústria são diversas e facilmente adaptáveis aos diferentes propósitos, sempre com o objetivo de um melhor aproveitamento em termos de *layout* com base na cadeia de produção e otimização dos tempos do processo.

Os principais tipos de transportadores podem ser classificados como:

- Esteiras rolantes;
- Transportadores de banda;
- Transportadores elevadores;
- AVG's.

No nosso trabalho, o transportador utilizado é uma esteira rolante, como podemos observar na figura 15. O transportador trabalha de modo contínuo com o auxílio de:

- Sensores indutivos;
- Atuadores;
- Setores de espera.



Figura 15 - Transportador da CFF

O transporte das peças é feito através de paletes ilustrado na figura 16. Cada uma das paletes que circula no transportador tem uma função diferente das demais. Apesar de aparentemente serem todas iguais entre si, encontram-se devidamente diferenciadas através de um código único atribuído por uns elementos metálicos que as constituem, demonstrado na Figura 17. Sendo assim, é possível colocar estas paletes de forma completamente aleatória no transportador. Cada uma das quatro paletes que circula de forma contínua no transportador é destinada a um único tipo de material: matéria-prima do torno, matéria-prima da fresadora e uma para cada um dos produtos acabados resultantes da maquinação das matérias-primas. Na Figura 16 é possível ver as quatro paletes utilizadas devidamente etiquetadas com a identificação de cada uma delas, PMPT (Paleta Matéria-Prima Torno), PMPF (Paleta Matéria-

Prima Fresadora), PPAT (Paleta Produto Acabado Torno) e PPAF (Paleta Produto Acabado Fresadora).



Figura 16 - Representação das Paletes



Figura 17 - Disposição das chapas metálicas nas paletes

A paragem das paletes nos devidos setores ocorre através da utilização de sensores indutivos, como podemos observar na figura 18, que quando alimentados, produzem um campo magnético constante que é alterado na presença de metal, alterando o estado das entradas do autómato a que se encontram ligados. Quando o autómato lê as entradas dos quatro sensores envia uma ordem para as saídas, e em função da combinação destes que são ativados à passagem das paletes é possível identificar qual delas é.

Na Tabela 8 apresentamos a Interface Robótica dos sensores e a respectiva combinação de cada palete.

Em função das paletes e do estado em que se encontram, cheias ou vazias, são executadas ações que as envolvem. As saídas estão ligadas aos atuadores, os *stoppers*, identificado na figura 19, que, quando atuadas, param as paletes na respectiva posição, permitindo que os robôs, nas respectivas estações, possam proceder ao carregamento e descarregamento das mesmas.

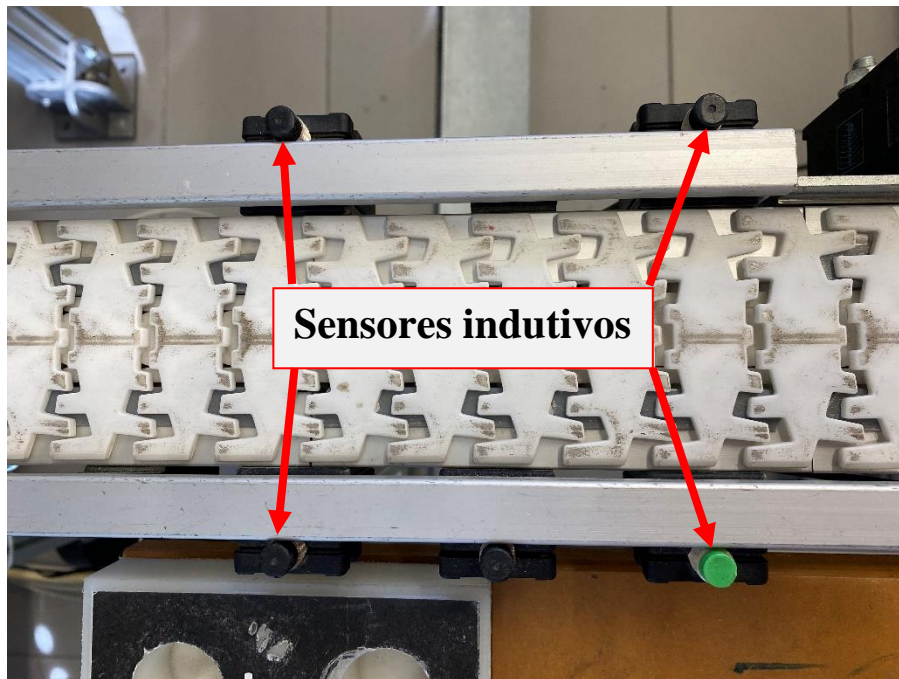


Figura 18 - Representação da localização dos sensores indutivos

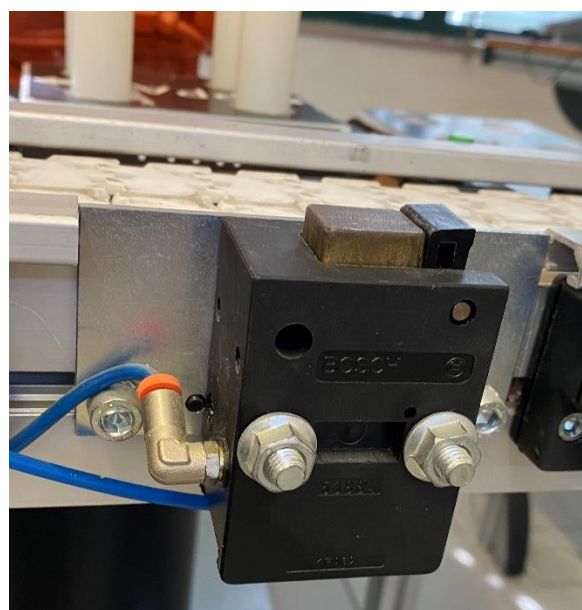


Figura 19 - Representação da localização dos *Stoppers*

Na Tabela 1 encontra-se a saída correspondente do PLC para ativar ou desativar cada Stopper.

Tabela 1 - Interface dos *Stoppers*

PLC S7-1200	Transportador - <i>Stoppers</i>
Q0.0	<i>Stopper</i> no setor de fabrico
Q0.1	<i>Stopper</i> no setor de armazém
Q0.2	<i>Stopper</i> no setor auxiliar
Q0.3	<i>Stopper</i> no setor de montagem

3. Proposta de Solução

O projeto é constituído por quatro equipamentos: o autómato, os dois robôs e o transportador, que engloba os vários sensores e atuadores. O autómato é o *master*, pois é ele que ordena a execução de tarefas aos outros equipamentos denominados de *slaves*.

3.1 Descrição do funcionamento

No transportador, temos quatro paletes a circular, duas paletes de matérias-primas (Torno e Fresadora) e duas de produtos acabados (Torno e Fresadora).

Na fase inicial, o transportador inicia o seu movimento e quando uma determinada paleta atravessa a zona dos sensores, o autómato identifica a paleta através do código recolhido, que, após reconhecimento, executa ou não a ordem de paragem.

Se for executada a ordem de paragem, o PLC ativa os *stoppers* e comunica ao robô a identificação da paleta em questão. De seguida, o robô procede ao seu carregamento ou descarregamento conforme a paleta identificada anteriormente.

Por fim, o robô comunica para o autómato o fim de operação e este procede à desativação dos *stoppers*. A paleta inicia o trajeto seguinte e, da mesma forma, se procede no outro setor de paragem.

Para que se verifique a condição de paragem no setor de armazenamento, é necessário que ocorra uma das seguintes situações:

- Paleta de MPT vazia – o robô carrega a paleta com as peças do armazém;
- Paleta de MPF vazia – o robô carrega a paleta com as peças do armazém;
- Paleta de PAT cheia – o robô descarrega a paleta para a rampa PAT;
- Paleta de PAF cheia – o robô descarrega a paleta para a rampa PAF.

Para que se verifique a condição de paragem no setor de fabrico, é necessário que ocorra uma das seguintes situações:

- Paleta de MPT cheia e *buffer* de MPT vazio – o robô descarrega a paleta, colocando-a no *buffer* em espera para maquinação;
- Paleta de MPF cheia e *buffer* de MPF vazio – o robô descarrega a paleta, colocando-a no *buffer* em espera para maquinação;
- Paleta de PAT vazia e *buffer* de PAT cheio – o robô carrega a paleta, com as peças do *buffer* em espera para transporte;
- Paleta de PAF vazia e *buffer* de PAF cheio – o robô carrega a paleta, com as peças do *buffer* em espera para transporte;

- *Buffer* MPT cheio e *buffer* PAT vazio – o robô procede à simulação do processo de maquinação;
- *Buffer* MPF cheio e *buffer* PAF vazio – o robô procede à simulação do processo de maquinação.

Estes processos ocorrem através da comunicação entre saídas e entradas do autômato e dos robôs. Estas interligações estão representadas na tabela 2.

Tabela 2 - Interface Robótica

PLC		IRB 140		IRB 1400		Descrição
Entradas	Saídas	Entradas	Saídas	Entradas	Saídas	
	Q0.6	DI12_9				Descarregar Palete MP torno
	Q0.7	DI12_10				Descarregar Palete MP fresadora
	Q1.0	DI12_11				Carregar Palete PA torno
	Q1.1	DI12_12				Carregar Palete PA fresadora
	Q0.6 e Q0.7	DI12_9 e DI12_10				Transferir Buf Torno
	Q0.7 e Q1.0	DI12_10 e DI12_11				Transferir Buf Fresa
	Q0.4			DI10_12		Carregar palete MP torno
	Q0.5			DI10_13		Carregar palete MP fresadora
	Q0.4 e Q0.5			DI10_12 e DI10_13		Descarregar palete PAT
	Q0.5 e Q13.0			DI10_13 e DI10_16		Descarregar palete PAF
II.2			DO12_9			Paleta MP torno descarregada
II.3			DO12_10			Paleta MP fresadora descarregada
II.4			DO12_11			Paleta PA torno carregada
II.5			DO12_12			Paleta PA fresadora carregada
II.2 e II.3			DO12_5			<i>Buffer</i> PAT Carregado
II.3 e II.4			DO12_6			<i>Buffer</i> PAF Carregado
II.0					DO10_9	Paleta MP torno carregada
II.1					DO10_10	Paleta MP fresadora carregada
II.0 e II.1					DO10_9 e DO10_10	Paleta PAT descarregada
II.1 e I3.0					DO10_10 e DO10_14	Paleta PAF descarregada

3.2 Graficets

Os *Graficets* podem ser consultados no Anexo I.

Na etapa E13 e E17 encontram-se macro etapas que simplificam e tornam mais legíveis o diagrama pelo facto de o *graficet* não ficar tão grande e complexo.

Os mesmos são apenas referidos ao *graficet* principal (*FirstScan* e largada temporizada das paletes), *graficet* ordem de carregar paleta MPT no robô IRB 1400, *graficet* paleta MPT carregada no robô IRB 1400.

3.2.1 Tabelas de Processo

As recetividades correspondem a condições a verificar para se transitar entre etapas. Na tabela 3, podemos visualizar as recetividades externas, que são condições que vêm de fora do autómato, como, por exemplo: sensores.

Tabela 3 - Identificação e descrição das recetividades externas

Recetividades Externas	Entradas	Descrição
Sensor 1	I0.0	Sensores indutivos do setor de armazém
Sensor 2	I0.1	
Sensor 3	I0.2	
Sensor 4	I0.3	
Sensor 1	I0.4	Sensores indutivos do setor de fabrico
Sensor 2	I0.5	
Sensor 3	I0.6	
Sensor 4	I0.7	
Paleta MP torno carregada	I1.0	Entrada ativada quando o robô 1400 acaba de carregar as peças MPT
Paleta MP fresadora carregada	I1.1	Entrada ativada quando o robô 1400 acaba de carregar as peças MPF
Paleta PAT descarregada	I0.0 e I1.1	Entrada ativada quando o robô 1400 acaba de descarregar as peças PAT
Paleta PAF descarregada	I1.1 e I13.0	Entrada ativada quando o robô 1400 acaba de descarregar as peças PAF
Paleta MP torno descarregada	I1.2	Entrada ativada quando o robô 1400 acaba de descarregar as peças MPT
Paleta MP fresadora descarregada	I1.3	Entrada ativada quando o robô 1400 acaba de descarregar as peças MPF
Paleta PA torno carregada	I1.4	Entrada ativada quando o robô 1400 acaba de carregar as peças PAT
Paleta PA fresadora carregada	I1.5	Entrada ativada quando o robô 1400 acaba de carregar as peças PAF

Na tabela 4 corresponde às recetividades internas, que vêm do autómato, como por exemplo os temporizadores.

Tabela 4 - Identificação e descrição das recetividades internas

Recetividades Internas	<i>Bits</i> de memória	Descrição
Paleta MPT	M10.0	Estado da paleta
Paleta MPF	M10.1	Estado da paleta
Paleta PAT	M10.2	Estado da paleta
Paleta PAF	M10.3	Estado da paleta
<i>Buffer</i> MPT	M40.4	Estado do <i>Buffer</i>
<i>Buffer</i> MPF	M40.2	Estado do <i>Buffer</i>
<i>Buffer</i> PAT	M40.1	Estado do <i>Buffer</i>
<i>Buffer</i> PAF	M40.3	Estado do <i>Buffer</i>
t1	DB23	Temporizador 1
t2	DB49	Temporizador 2
t3	DB63	Temporizador 3
t4	DB64	Temporizador 4
t5	DB 48	Temporizador 5
t6	DB 50	Temporizador 6
t7	DB 8	Temporizador 7
t8	DB 1	Temporizador 8
t9	DB 36	Temporizador 9
t10	DB 9	Temporizador 10
t11	DB 28	Temporizador 11

Na tabela 5 visualizamos as ações externas que estão fora do PLC, como por exemplo os *stoppers*.

Tabela 5 - Identificação e descrição das ações externas

Ações Externas	Descrição	Saída
Stopper 1	Stopper que faz parar as paletes no setor de fabrico	Q0.0
Stopper 2		Q0.1
Stopper 3	Stopper que faz parar as paletes no setor de armazém	Q0.2
Stopper 4		Q0.3
Carregar paleta MP torno	Saída que manda o robô IRB 1400 carregar as peças MP torno	Q0.4
Carregar paleta MP fresadora	Saída que manda o robô IRB 1400 carregar as peças MP fresadora	Q0.5
Descarregar paleta MP torno	Saída que manda robô IRB 140 descarregar as peças MP torno	Q0.6
Descarregar paleta MP fresadora	Saída que manda robô IRB 140 descarregar as peças MP fresadora	Q0.7
Carregar paleta PA Torno	Saída que manda o robô IRB 100 carregar as peças PAT	Q1.0
Carregar paleta PA fresadora	Saída que manda o robô IRB 140 carregar as peças PAF	Q1.1
Descarregar paleta PAT	Saída que manda o robô IRB 100 descarregar as peças PAT	Q0.4 e Q0.5
Descarregar paleta PAF	Saída que manda o robô IRB 100 descarregar as peças PAF	Q0.5 e Q13.0

3.2.2 Equações

A partir do *grafcet*, facilmente se obtêm as equações lógicas para cada etapa, que nos permitem escrever o programa em diagrama de contactos e evitar conflitos durante o seu funcionamento.

Etapa 0: M1.0 (bit 1º ciclo)

Etapa 1: M1.0 (DB23)

Etapa 2: DB23

[S (Q0.0, Q0.1, Q0.2, Q0.3); R (M10.0, M10.2, M10.2, M10.3)]

Etapa 3: M1.0 (DB49)

Etapa 4: DB49

[S (M10.4)]

Etapa 5: M10.4(DB63)

Etapa 6: DB63•/M70.0

[R (Q0.2)]

Etapa 7: DB63•/M70.1 (DB64)

Etapa 8: DB64

[S (Q0.2, M70.0)]

Etapa 9: DB63•/M70.2 (DB48)

Etapa 10: DB48

[S (M70.1); R (Q0.2)]

Etapa 11: DB63 (DB50)

Etapa 12: DB50

[S (M70.2)]

Etapa 13: DB63

Etapa14: I0.0•I0.1•I0.2•I0.3•/M50.7•/M50.6•/M50.4•/M50.3•/M50.2•/M50.0•/M20.1•/M10.0•DB8

Etapa 15: DB8(DB1)

[S (M50.1); R (Q0.1, Q0.2, Q0.3)]

Etapa 16: DB1(DB36)

[S (Q0.4, M20.0); R (Q0.5, Q13.0)]

Etapa 17: DB36

[R (Q0.4)]

Etapa 18: DB63

Etapa 19: I1.0•M20.0 (DB9)

Etapa 20: DB9•/M20.4•/M20.6

[S (Q0.0)]

Etapa 21: DB9 (DB28)

[S (Q0.1, Q0.2, Q0.3, M10.0)]

Etapa 22: DB28

[R (M50.1, Q0.3)]

Etapa 23: DB9

[R (M20.0)]

3.2.3 Programa em diagrama de contactos

Neste ponto, vamos explicar alguns aspetos importantes do programa desenvolvido.

Teve de ser criado o bit do 1.º ciclo (*FirstScan*) e o bit de memória escolhido foi o M1.0

Podemos observar isso na figura 20, onde está a Network 2 do programa em diagrama de contactos.

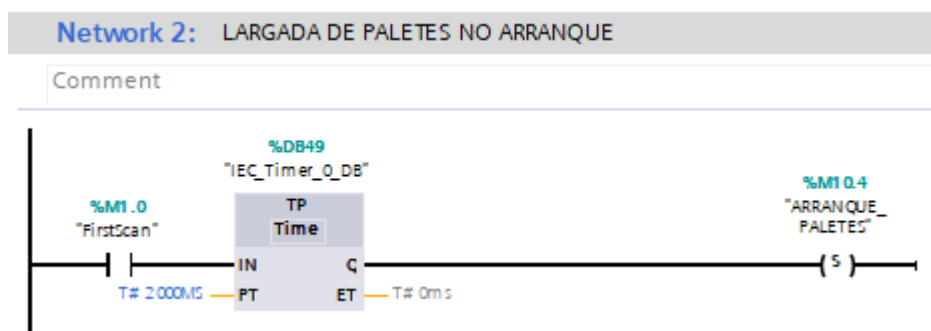


Figura 20 – *FirstScan* na Network 2

No programa utilizámos 2 tipos de temporizadores.

Na figura 21 temos presente um temporizador TON, ou seja, um temporizador com atraso à operação, isto quer dizer que a saída é ativada depois de passar o tempo estipulado, neste caso 2 segundos.

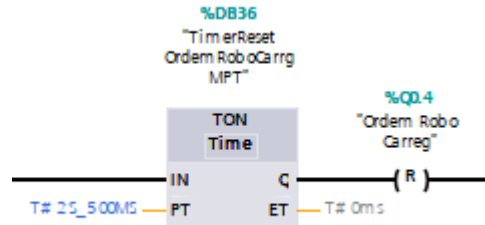


Figura 21 – Temporizador TON

Na figura 22 podemos observar o Diagrama de tempo de pulso do temporizador TON.

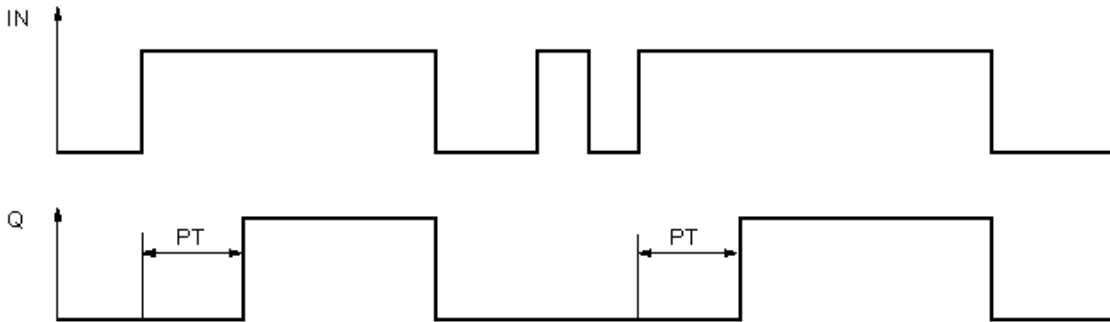


Figura 22 – Pulse Timing Diagrame (Diagrama de tempo de pulso) TON

Na figura 23 temos presente um temporizador TP, ou seja, um temporizador que quando alimentado na sua entrada, a sua saída passa a 1 durante o tempo estipulado na porta PT, independentemente de o sinal de entrada continuar ativo ou não, neste caso o mesmo está programado para ter a saída ativa durante 3 segundos.

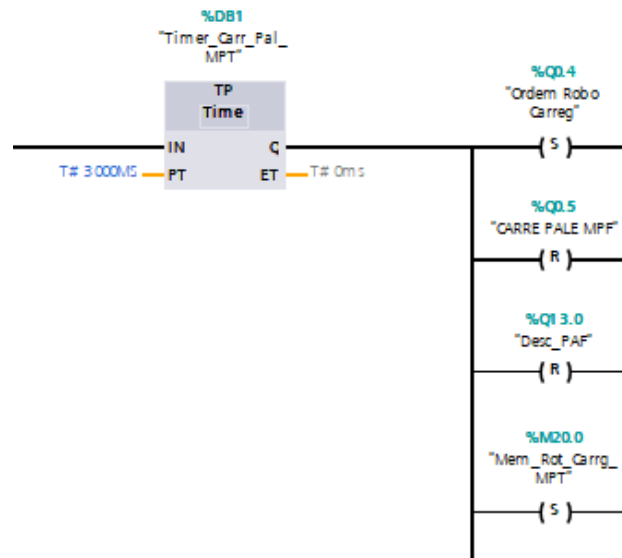


Figura 23 – Temporizador TP

Na figura 24 podemos observar o Diagrama de tempo de pulso do temporizador TP.

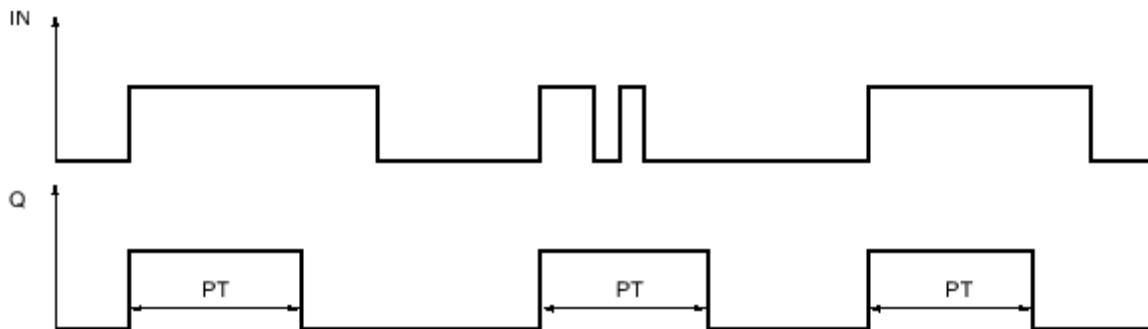


Figura 24 - Pulse Timing Diagrame (Diagrama de tempo de pulso) TP

3.2.4 Melhorias ao programa

Neste ponto do relatório, iremos abordar algumas melhorias que fizemos ao programa

Surgiu-nos um problema em que numa determinada fase do processo de deslocação das paletes que devido aos stoppers estarem a trabalhar todos ao mesmo tempo, as mesmas juntavam-se umas atrás das outras, visto que temos um determinado tempo de espera depois das paletes arrancarem após serem carregadas ou descarregadas pelo robô ao estarem umas atrás das outras iriam passar “em falso” através dos sensores fazendo com que muitas vezes a paleta andasse às voltas no transportador sem nunca ser carregada ou descarregada, tornando o programa pouco eficiente e demorado.

Para solucionar este problema foi criada uma rotina para “largar” as paletes uma a uma de forma temporizada como podemos observar na figura 25. Também foram colocados os *stoppers* a trabalhar individualmente.

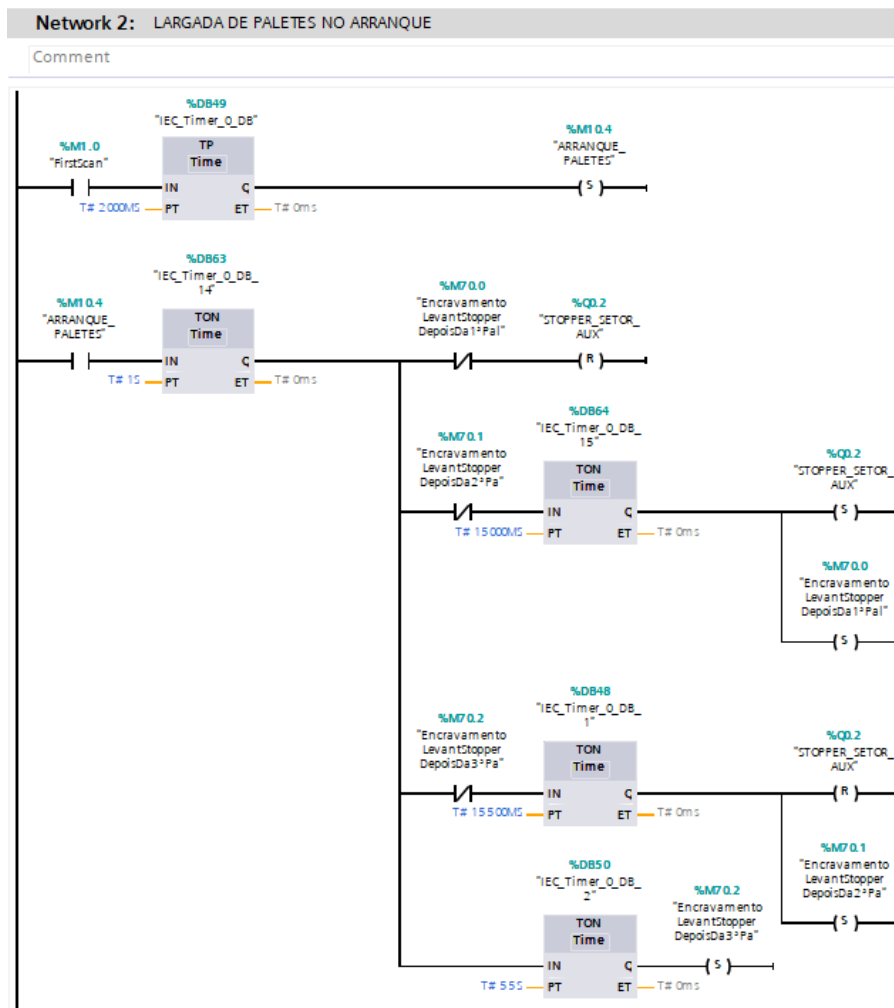


Figura 25 – Network 2: Largada de Paletes no Arranque

Outro problema que nos surgiu foi o facto de que quando o robô retirava todas as peças presentes no armazém, este ficava sem *stock* e o robô ficava constantemente a procurar peças pelas várias posições do armazém. Para solucionar este problema foi adicionado ao programa de carregar as paletes a seguinte parte de código ilustrada na figura 26, para que ao fim de o robô procurar a peça 2 vezes pelo armazém todo, ele pára e surge uma mensagem na consola (figura 27) dizendo que não há peças e é preciso serem repostas, ao fim de o utilizador repor peças no armazém pressiona o “OK” na consola e o robô continua a executar a sua atividade.

```

IF FaltaPecaTorno=8 THEN
  MoveJ pApXArmTorn,v1000,z10,toolESTGV;
  StopMove;
  MgsFaltaPecaTorno:
  TPErase;
  TPWrite "*****";
  TPWrite "*";
  TPWrite "* !!! FALTA PECAS ARMAZEM TORNO !!!! *";
  TPWrite "*";
  TPWrite "*****";
  TPReadFK FaltaTorno, "STOCK REPOSTO?", stEmpty, stEmpty, stEmpty, "Sim", "Não";
  IF FaltaTorno=4 THEN
    FaltaPecaTorno:=0;
    StartMove;
  ENDIF
  IF FaltaTorno=5 THEN
    GOTO MgsFaltaPecaTorno;
  ENDIF

```

Figura 26 - Adição ao programa da rotina Carrega Paleta Torno

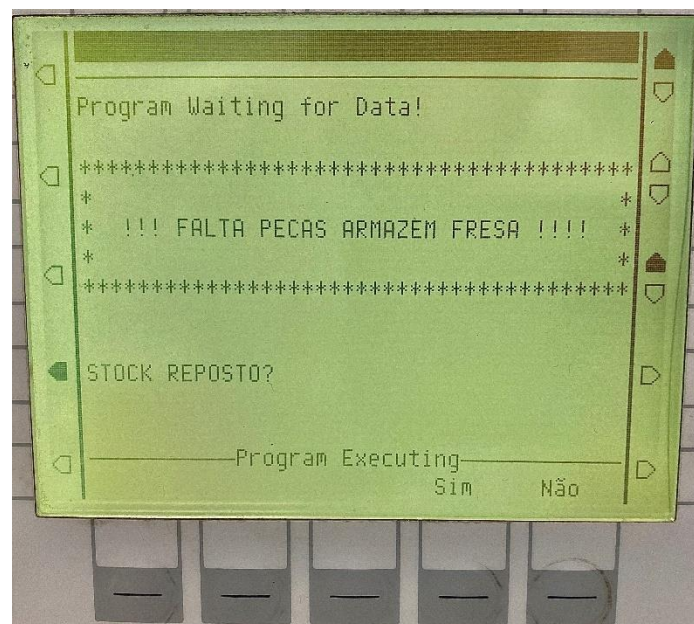


Figura 27 - Mensagem Falta Peças

3.3 Robôs IRB 1400 e IRB 140 da ABB

Para programar o robô do nosso projeto utilizamos o programa RobotStudio da ABB, programa este muito intuitivo e essencial na resolução do projeto pois é possível executar a programação e simulação em 3D.

Após o programa executado no RobotStudio testamos o mesmo em laboratório para verificar o verdadeiro funcionamento do mesmo e executar, portanto, alguns ajustes provenientes de pequenas anomalias.

Além disso para retirar os pontos tem de ser feito em laboratório com o próprio robô utilizando para isso a consola.

3.3.1 Dificuldades

Na fase inicial tivemos algumas dificuldades e perdemos algum tempo ao tirar os pontos das paletes. Em primeiro lugar tiramos um ponto com o robô e os 3 restantes foram *offsets* do primeiro, entretanto se este primeiro ponto não estivesse de esquadria, bem posicionado em termos de inclinação, rotação e os 3 eixos (xyz), os outros 3 sofrem alterações e não ficam bem centrados/posicionados nas paletes.

3.3.2 Solução

A solução para este problema foi executar por tentativa erro até aprimorar a obtenção das coordenadas corretas dos diversos pontos.

3.3.3 Robô IRB 140

Multiplexagem de Entradas

Visto que estavam a faltar entradas para programar no TIA tivemos de fazer a multiplexagem das mesmas, ou seja, fazer combinações entre as mesmas para ter mais entradas.

A multiplexagem implica a junção de várias entradas, no nosso caso de duas, como é possível observar na tabela 6.

Tabela 6 - Multiplexagem das entradas do IRB140

TAREFA	DI_12_9	DI_12_10	DI_12_11	DI_12_12
Descarregar MAT	1	0	0	0
Descarregar MAF	0	1	0	0
Carregar PAT	0	0	1	0
Carregar PAF	0	0	0	1
Transferir <i>Buffer</i> Torno	1	1	0	0
Transferir <i>Buffer</i> Fresa	0	1	1	0

Interface robótica

A tabela 7 apresenta as entradas e saídas do robô IRB 140 e entradas e saídas do autômato. As entradas estão sombreadas a cor verde e as saídas a vermelho.

Tabela 7 - Interface robótica das Entradas e Saídas do robô IRB 140

PLC S7-1200	Robô IRB 140	Descrição
Q0.6	DI12_9	Descarregar Palete MP torno
Q0.7	DI12_10	Descarregar Palete MP fresadora
Q1.0	DI12_11	Carregar Palete PA torno
Q1.1	DI12_12	Carregar Palete PA fresadora
Q0.6 e Q=0.7	DI12_9 e DI12_10	Transferir Buf Torno
Q0.7 e Q=1.0	DI12_10 e DI12_11	Transferir Buf Fresa
I1.2	DO12_9	Palete MP torno descarregada
I1.3	DO12_10	Palete MP fresadora descarregada
I1.4	DO12_11	Palete PA torno carregada
I1.5	DO12_12	Palete PA fresadora carregada
I1.2 e I1.3	DO12_5	<i>Buffer</i> PAT Carregado
I1.3 e I1.4	DO12_6	<i>Buffer</i> PAF Carregado

A tabela 8 apresenta os 4 sensores indutivos que executam a paragem das paletes (I0.4 a I0.7) e as combinações necessárias para detetar a paragem de cada uma das paletes.

Tabela 8 - Interface robótica dos sensores no setor de fabrico

Entradas	Sensores
I0.4	Sensor1 no setor de fabrico
I0.5	Sensor2 no setor de fabrico
I0.6	Sensor3 no setor de fabrico
I0.7	Sensor4 no setor de fabrico
Nome da Palete	Combinações para identificação de cada Palete
MPT	I0.4=1; I0.5=1; I0.6=1; I0.7=1
MPF	I0.4=1; I0.5=0; I0.6=0; I0.7=1
PAT	I0.4=1; I0.5=0; I0.6=1; I0.7=1
PAF	I0.4=0; I0.5=0; I0.6=0; I0.7=1

Programação do Robô IRB 140

Na figura 28, encontra-se, de forma sucinta, o fluxograma do programa principal, evidenciando as diferentes entradas do robô. Estas só serão ativadas quando o autômato enviar o sinal ao robô para que proceda à execução da rotina. Depois de terminar a rotina, o robô volta à posição de descanso (*home*).

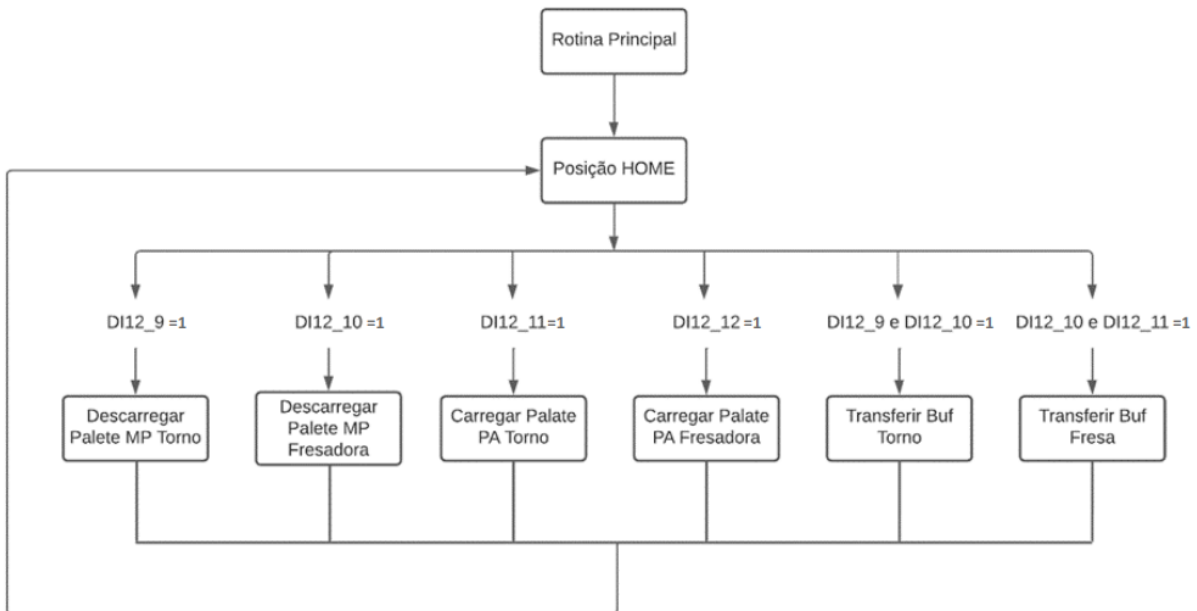


Figura 28 - Fluxograma Programa Principal IRB 140

Programação do Programa Principal RobotStudio:

```
|***** Programa principal *****|
|*****|
PROC main()
  ArmazenTorno:=1;
  ArmazenFresa:=1;
  SingArea\Wrist;
  MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;

  !MoveJ pPaletTorno,v500,z10,toolESTGV;
  rAbrePinca;
  SetDO D012_2,0;
  SetDO D012_1,1;
  IF DI12_9=1 AND DI12_10=0 AND DI12_11=0 AND DI12_12=0 THEN
    TPErase;
    TPWrite "rotina descarrega paleta torno";
    WaitTime 0.7;
    rDescPaletTorno;
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
    SetDO D012_9, 1;
    WaitTime 0.3;
    SetDO D012_9, 0;

  ELSEIF DI12_9=0 AND DI12_10=1 AND DI12_11=0 AND DI12_12=0 THEN
    TPErase;
    TPWrite "rotina descarrega paleta fresa";
    WaitTime 0.7;
    rDescPaletFresa;
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
    SetDO D012_10, 1;
    WaitTime 0.3;
    SetDO D012_10, 0;

  ELSEIF DI12_9=1 AND DI12_10=1 AND DI12_11=0 AND DI12_12=0 THEN
    TPErase;
    TPWrite "rotina transfere bufer torno";
    rTransfBufTorno;
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
    SetDO D012_9, 1;
    SetDO D012_10, 1;
    WaitTime 0.3;
    SetDO D012_9, 0;
    SetDO D012_10, 0;

  ELSEIF DI12_9=0 AND DI12_10=1 AND DI12_11=1 AND DI12_12=0 THEN
    TPErase;
    TPWrite "rotina transfere bufer fresa";
    rTransBufFresa;
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
    SetDO D012_10, 1;
    SetDO D012_11, 1;
    WaitTime 0.3;
    SetDO D012_10, 0;
    SetDO D012_11, 0;

  ELSEIF DI12_9=0 AND DI12_10=0 AND DI12_11=1 AND DI12_12=0 THEN
    TPErase;
    TPWrite "rotina carrega PAT";
    rcarrePalPAT;
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
    SetDO D012_11, 1;
    WaitTime 0.3;
    SetDO D012_11, 0;

  ELSEIF DI12_9=0 AND DI12_10=0 AND DI12_11=0 AND DI12_12=1 THEN
    TPErase;
    TPWrite "rotina carrega PAF";
    rcarrePalPAF;
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
    SetDO D012_12, 1;
    WaitTime 0.3;
    SetDO D012_12, 0;

  ENDIF
  MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;
ENDPROC
```

Figura 29 – Programação do Programa Principal RobotStudio IRB 140

Fluxograma da rotina Descarrega Palete Torno para o *buffer* da matéria-prima (rDescPaletTorno):

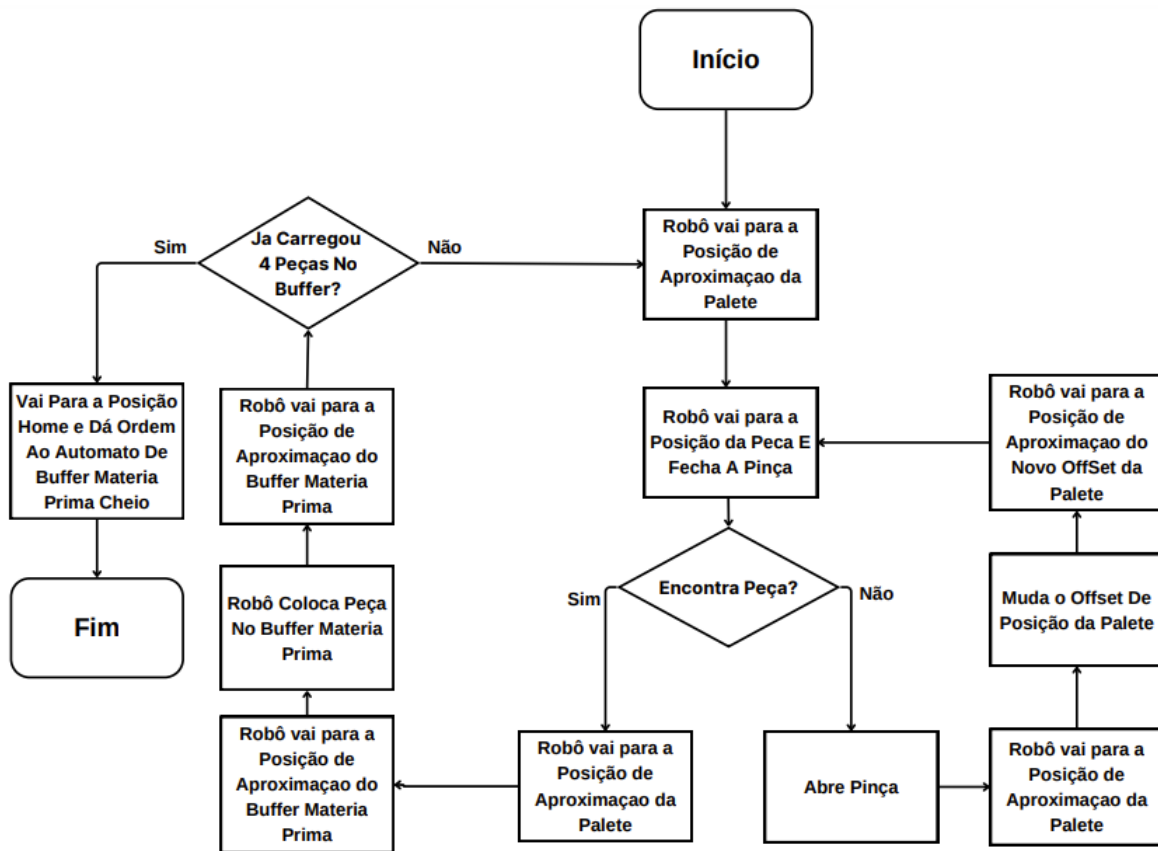


Figura 30 - Fluxograma da rotina Descarrega Palete Torno para o *buffer* da matéria-prima do IRB 140

Programação RobotStudio da rotina Descarrega Palette Torno para o *buffer* da matéria-prima:

```
PROC rDescPaletTorno()
Inicio:
  IF nPecaTorno=1 THEN
    pPegPecPalTor:=pPaletTorno;
  ELSEIF nPecaTorno=2 THEN
    pPegPecPalTor:=Offs(pPaletTorno,80,0,0);
  ELSEIF nPecaTorno=3 THEN
    pPegPecPalTor:=Offs(pPaletTorno,80,-88,0);
  ELSEIF nPecaTorno=4 THEN
    pPegPecPalTor:=Offs(pPaletTorno,0,-88,0);
  ENDIF
  pApxPalTor:=Offs(pPegPecPalTor,0,0,100);
  MoveJ pApxPalTor,v1000,z10,toolESTGV;
  MoveL pPegPecPalTor,v500,fine,toolESTGV;
  rFechaPinca;
  IF DI12_8=1 THEN
    IF nPecaTorno=1 THEN
      pDeixPecBufTor:=Offs(pPegPecBufTor,-90,-80,0);
    ELSEIF nPecaTorno=2 THEN
      pDeixPecBufTor:=Offs(pPegPecBufTor,-88,2,0);
    ELSEIF nPecaTorno=3 THEN
      pDeixPecBufTor:=pPegPecBufTor;
    ELSEIF nPecaTorno=4 THEN
      pDeixPecBufTor:=Offs(pPegPecBufTor,0,-81,0);
    ENDIF
    pApxBufTorno:=Offs(pDeixPecBufTor,0,0,100);
    MoveJ pApxPalTor,v1000,z10,toolESTGV;
    MoveJ pDesvio,v1000,z10,toolESTGV;
    MoveJ pApxBufTorno,v1000,z10,toolESTGV;
    MoveL pDeixPecBufTor,v500,fine,toolESTGV;
    rAbrePinca;
    MoveJ pApxBufTorno,v1000,z10,toolESTGV;
    nPecaTorno:=nPecaTorno+1;
  ELSEIF DI12_8=0 THEN
    rAbrePinca;
    MoveJ pApxPalTor,v1000,z10,toolESTGV;
    nPecaTorno:=nPecaTorno+1;
  ENDIF
  IF nPecaTorno<5 THEN
    GOTO Inicio;
  ELSE
    nPecaTorno:=1;
  ENDIF
ENDPROC
```

Figura 31 – Programação da rotina Descarrega Palette Torno para o *buffer* da matéria-prima do IRB 140

Fluxograma da Rotina Transfere Peças do *buffer* matéria-prima para o *buffer* de produto acabado(rTransfBufTorno):

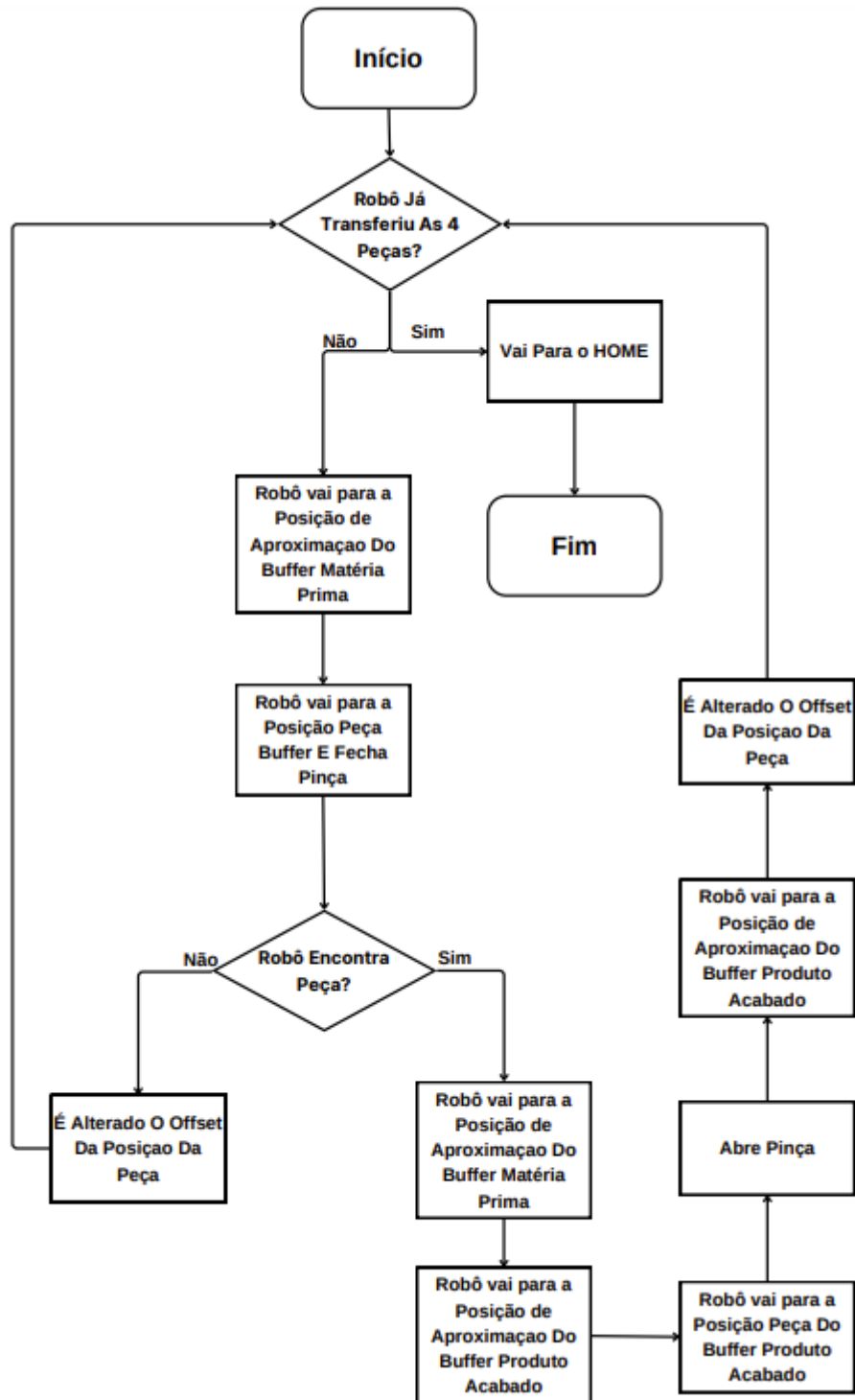


Figura 32 - Fluxograma da Rotina Transfere Peças do *buffer* matéria-prima para o *buffer* de produto acabado do IRB 140

Programação RobotStudio da Rotina Transfere Peças do *buffer* matéria-prima para o *buffer* de produto acabado:

```
PROC rTransfBufTorno()  
Inicio:  
  IF nPecaTorno=1 THEN  
    pPegPecPalTor:=Offs(pPegPecBufTor,-90,0,0);  
  ELSEIF nPecaTorno=2 THEN  
    pPegPecPalTor:=Offs(pPegPecBufTor,-90,-80,0);  
  ELSEIF nPecaTorno=3 THEN  
    pPegPecPalTor:=pPegPecBufTor;  
  ELSEIF nPecaTorno=4 THEN  
    pPegPecPalTor:=Offs(pPegPecBufTor,0,-83,0);  
  ENDIF  
  pApxPalTor:=Offs(pPegPecPalTor,0,0,100);  
  MoveJ pApxPalTor,v1000,z10,toolESTGV;  
  MoveL pPegPecPalTor,v500,fine,toolESTGV;  
  rFechaPinca;  
  IF DI12_8=1 THEN  
    IF nPecaTorno=1 THEN  
      pDeixPecBufTor:=Offs(pPaletPATorn,-90,0,0);  
    ELSEIF nPecaTorno=2 THEN  
      pDeixPecBufTor:=Offs(pPaletPATorn,-90,-80,0);  
    ELSEIF nPecaTorno=3 THEN  
      pDeixPecBufTor:=pPaletPATorn;  
    ELSEIF nPecaTorno=4 THEN  
      pDeixPecBufTor:=Offs(pPaletPATorn,0,-83,0);  
    ENDIF  
    pApxBufTorno:=Offs(pDeixPecBufTor,0,0,100);  
    MoveJ pApxPalTor,v1000,z10,toolESTGV;  
    MoveJ pApxBufTorno,v1000,z10,toolESTGV;  
    MoveL pDeixPecBufTor,v500,fine,toolESTGV;  
    rAbrePinca;  
    MoveJ pApxBufTorno,v1000,z10,toolESTGV;  
    nPecaTorno:=nPecaTorno+1;  
  ELSEIF DI12_8=0 THEN  
    rAbrePinca;  
    MoveJ pApxPalTor,v1000,z10,toolESTGV;  
    nPecaTorno:=nPecaTorno+1;  
  ENDIF  
  IF nPecaTorno<5 THEN  
    GOTO Inicio;  
  ELSE  
    nPecaTorno:=1;  
  ENDIF  
ENDPROC
```

Figura 33 - Programação da Rotina Transfere Peças do *buffer* matéria-prima para o *buffer* de produto acabado do IRB 140

Fluxograma da rotina carrega peças do *buffer* do produto acabado para a paleta PAT:

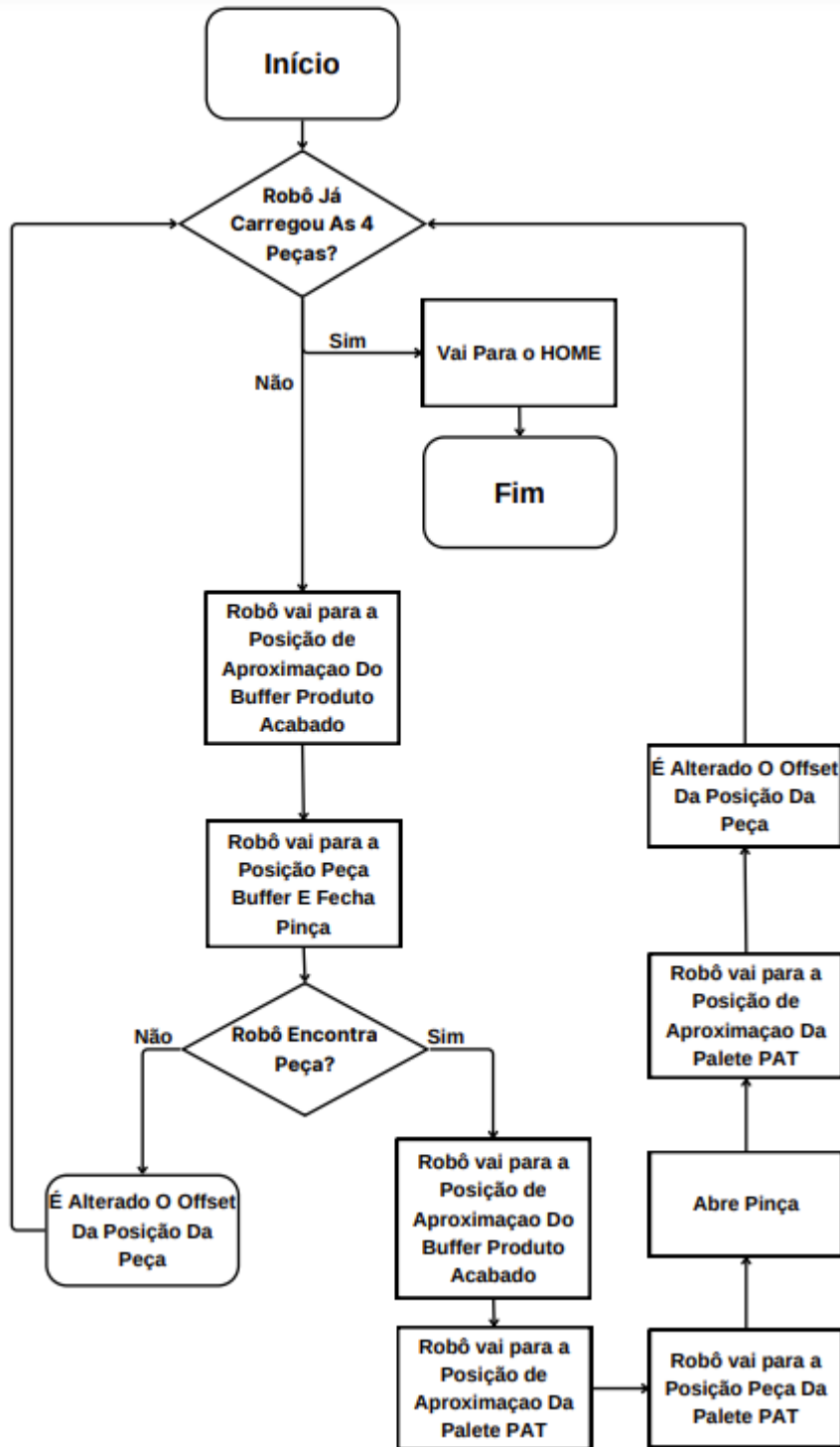


Figura 34 – Fluxograma da rotina carrega peças do *buffer* do produto acabado para a paleta PAT do IRB 140

Programação RobotStudio da rotina carrega peças do *buffer* do produto acabado para a paleta PAT:

```
PROC rcarrePalPAT()

Inicio:
  IF nPecaTorno=1 THEN
    pDeixPecBufTor:=pPaletPATorn;
  ELSEIF nPecaTorno=2 THEN
    pDeixPecBufTor:=Offs(pPaletPATorn,-4,-79,0);
  ELSEIF nPecaTorno=3 THEN
    pDeixPecBufTor:=Offs(pPaletPATorn,-92,0,0);
  ELSEIF nPecaTorno=4 THEN
    pDeixPecBufTor:=Offs(pPaletPATorn,-94,-80,0);
  ENDIF
  pApxBufTorno:=Offs(pDeixPecBufTor,0,0,100);
  MoveJ pApxBufTorno,v1000,z10,toolESTGV;
  MoveL pDeixPecBufTor,v500,fine,toolESTGV;
  rFechaPinca;
  IF DI12_8=1 THEN
    IF nPecaTorno=1 THEN
      pPegPecPalTor:=Offs(pPaletTorno,84,0,3);
    ELSEIF nPecaTorno=2 THEN
      pPegPecPalTor:=Offs(pPaletTorno,81,-88,5);
    ELSEIF nPecaTorno=3 THEN
      pPegPecPalTor:=pPaletTorno;
    ELSEIF nPecaTorno=4 THEN
      pPegPecPalTor:=Offs(pPaletTorno,0,-88,6);
    ENDIF
    pApxPalTor:=Offs(pPegPecPalTor,0,0,100);
    MoveL pApxBufTorno,v1000,z10,toolESTGV;
    MoveJ pDesvio,v1000,z10,toolESTGV;
    MoveJ pApxPalTor,v1000,z10,toolESTGV;
    MoveL pPegPecPalTor,v500,fine,toolESTGV;
    rAbrePinca;
    MoveJ pApxPalTor,v1000,z10,toolESTGV;
    nPecaTorno:=nPecaTorno+1;
  ELSEIF DI12_8=0 THEN
    rAbrePinca;
    MoveJ pApxBufTorno,v1000,z10,toolESTGV;
    nPecaTorno:=nPecaTorno+1;
  ENDIF
  IF nPecaTorno<5 THEN
    GOTO Inicio;
  ELSE
    nPecaTorno:=1;
  ENDIF
ENDPROC
```

Figura 35 - Programação RobotStudio da rotina carrega peças do *buffer* do produto acabado para a paleta PAT do IRB 140

3.3.4 Robô IRB 1400

Multiplexagem de Entradas

Visto que estavam a faltar entradas para programar no TIA tivemos de fazer a multiplexagem das mesmas, ou seja, fazer combinações entre as mesmas para ter mais entradas.

Na tabela 9 podemos observar a multiplexagem das entradas para o Robô IRB 1400.

Tabela 9 - Multiplexagem das entradas do IRB1400

TAREFA	DI10_12	DI10_13	DI10_16
Carrega MPT	0	1	0
Carrega MPF	1	0	0
Descarregar PAT	1	1	0
Descarregar PAF	0	1	1

Interface robótica robô IRB1400

A tabela 10 apresenta a correspondência entre as entradas e saídas do robô 1400 e entradas e saídas do Autómato.

Tabela 10 - Interface robótica do IRB 1400

PLC S7-200 PLC S7-1200	Robô IRB 1400	Descrição
I1.0	DO10_9	Paleta MP torno carregada
I1.1	DO10_10	Paleta MP fresadora carregada
I1.0 e I1.1	DO10_9 e DO10_10	Paleta PAT descarregada
I1.1 e I13.0	DO10_10 e DO10_4	Paleta PAF descarregada
Q0.4	DI10_12	Carregar paleta MP torno
Q0.5	DI10_13	Carregar paleta MP fresadora
Q0.4 e Q0.5	DI10_12 e DI10_13	Descarregar Paleta PAT
Q0.5 e Q13.0	DI10_13 e DI10_16	Descarregar Paleta PAF

A tabela 11 apresenta os 4 sensores indutivos que executam a paragem das paletes (I0.0 a I0.4) e as combinações necessárias para detetar a paragem de cada uma das paletes.

Tabela 11 - Interface robótica dos sensores no setor de Armazém

Entradas	Sensores
I0.0	S1 no setor do armazém
I0.1	S2 no setor do armazém
I0.2	S3 no setor do armazém
I0.3	S4 no setor do armazém
Nome da Pallette	Combinações para identificação de cada Pallette
MPT	I0.0=1; I0.1=1; I0.2=1; I0.3=1
MPF	I0.0=1; I0.1=0; I0.2=0; I0.3=1
PAT	I0.0=1; I0.1=0; I0.2=1; I0.3=1
PAF	I0.0=0; I0.1=0; I0.2=0; I0.3=1

Programação do Robô IRB 1400

Fluxograma Programa Principal:

No programa principal, o robô está à espera de uma informação do Autômato para ativar uma das saídas. Ativando uma entrada, o robô executa a rotina correspondente. Observando o fluxograma da figura 36 existem quatro rotinas, cada uma corresponde a cada palete. Depois de executar a respectiva rotina, o robô volta à posição de repouso

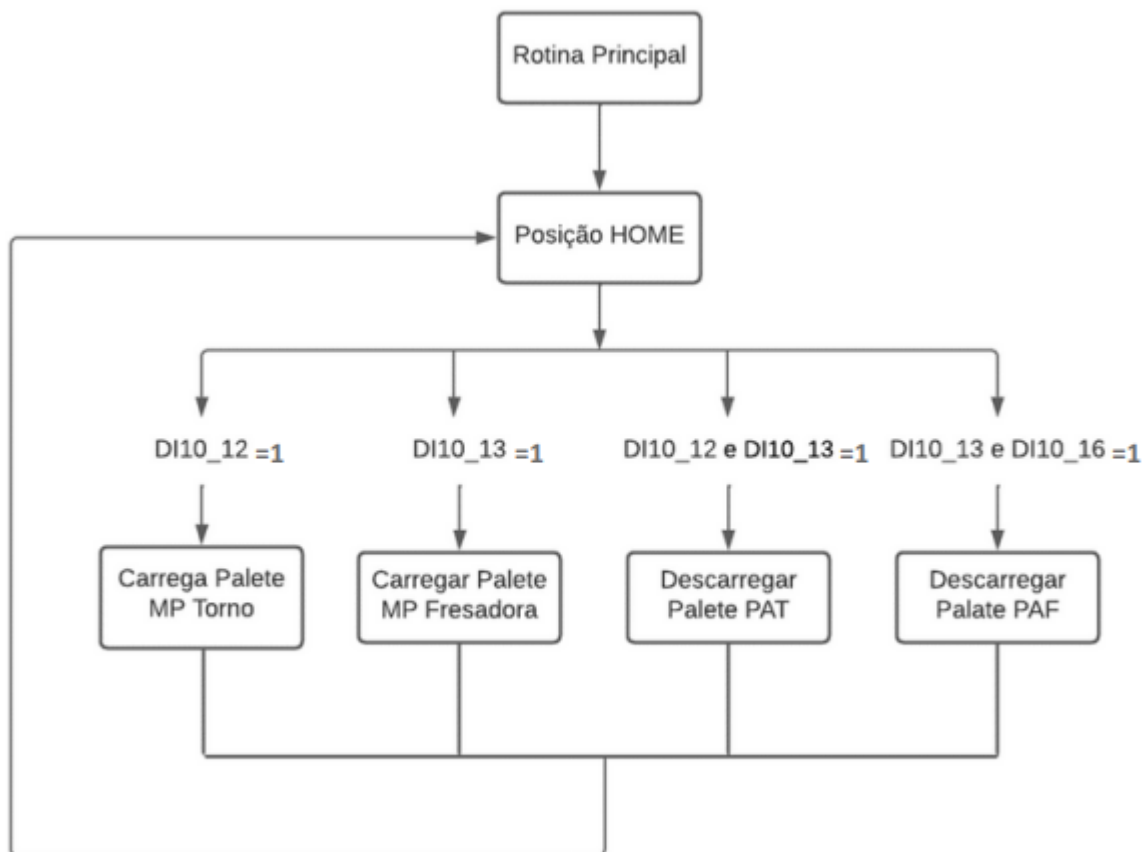


Figura 36 - Fluxograma Programa Principal IRB 1400

Programação do Programa Principal RobotStudio:

```
|*****  
|***** Programa principal *****  
|*****  
PROC main()  
  SingArea\Wrist;  
  MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;  
  rAbrePinca;  
  SetDO D010_3,0;  
  SetDO D010_2,1;  
  IF DI10_12=1 AND DI10_13=0 AND DI10_16=0 THEN  
    TPErase;  
    TPWrite "rotina carrega palete torno";  
    rCarregPalTorn;  
    SetDO D010_9, 1; ! o robo informa o PLC que a palete MPT esta carregada  
    WaitTime 1.5;  
    SetDO D010_9, 0;  
  ENDIF  
  IF DI10_12=0 AND DI10_13=1 AND DI10_16=0 THEN  
    TPErase;  
    TPWrite "rotina carrega palete fresa";  
    rCarregPalFre;  
    SetDO D010_10, 1; ! o robo informa o PLC que a palete MPF esta carregada  
    WaitTime 1.5;  
    SetDO D010_10, 0;  
  ENDIF  
  IF DI10_12=1 AND DI10_13=1 AND DI10_16=0 THEN  
    TPErase;  
    TPWrite "rotina descarrega palete torno";  
    rDescarPalTorn;  
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;  
    SetDO D010_9, 1; ! o robo informa o PLC que a palete PAT esta Descarregada  
    SetDO D010_10, 1;  
    WaitTime 1.5;  
    SetDO D010_9, 0;  
    SetDO D010_10, 0;  
  ENDIF  
  IF DI10_12=0 AND DI10_13=1 AND DI10_16=1 THEN  
    TPErase;  
    TPWrite "rotina descarrega palete fresa";  
    rDescarPalFresa;  
    MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;  
    SetDO D010_10, 1; ! o robo informa o PLC que a palete PAF esta Descarregada  
    SetDO D010_4, 1;  
    WaitTime 1.5;  
    SetDO D010_10, 0;  
    SetDO D010_4, 0;  
  ENDIF  
  
  MoveAbsJ home\NoEOffs,v2000,fine,toolESTGV;  
  
ENDPROC
```

Figura 37 – Programação do Programa Principal RobotStudio IRB 1400

Programação da rotina Regista Paletes Torno (rRegisPalTorno) RobotStudio:

```
PROC rRegisPalTorn()  
  nPaleteTorn:=nPaleteTorn+1;  
  Open "HOME:"\File:="1_LEE_ProjAntMar/Paletes_Torno/Verif_Paletes.txt",file\Append;  
  IF nPaleteTorn=1 THEN  
    Write file,"===== PALETES NO ROBOT IRB-1400 =====";  
    Write file,"Palete      Data      Hora      Tempo Ciclo(seg.)      Ok(1)/NOK(0)";  
  ENDIF  
  Write file,"  \Num:=nPaleteTorn\NoNewLine;  
  Write file,"          \NoNewLine;  
  IF nPaleteTorn=1 THEN  
    Write file,CDate()\NoNewLine;  
  ENDIF  
  IF nPaleteTorn>1 THEN  
    Write file,"          \NoNewLine;  
  ENDIF  
  Write file,"          \NoNewLine;  
  Write file,CTime()\NoNewLine;  
  Write file,"          \NoNewLine;  
  Write file,"          \Num:=TempoDeCiclo\NoNewLine;  
  Write file,"          \Num:=Pecal\NoNewLine;  
  Write file," \Num:=Peca2\NoNewLine;  
  Write file," \Num:=Peca3\NoNewLine;  
  Write file," \Num:=Peca4;  
  Close file;  
ENDPROC
```

Figura 38 - Programação Principal RobotStudio IRB 1400

Fluxograma da rotina Verifica Paleta Torno (rVerifPalTorno):

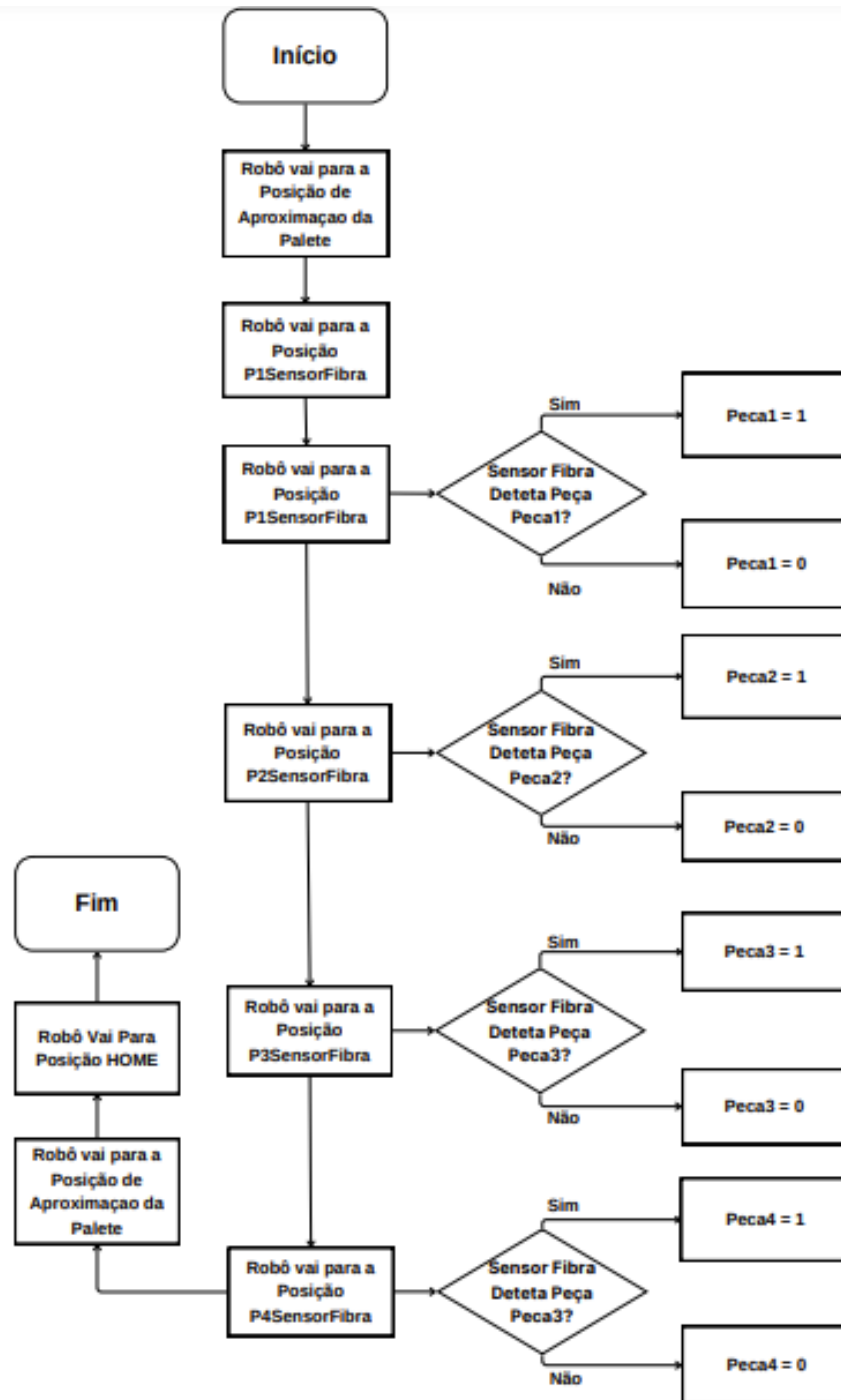


Figura 39 - Fluxograma da rotina Verifica Paleta Torno IRB 1400

Programação RobotStudio da rotina Verifica Paleta Torno:

```
PROC rVerifPalTorn()
  Peca1:=0;
  Peca2:=0;
  Peca3:=0;
  Peca4:=0;
  pApxPalete:=Offs(p1PaletTorn,0,0,170);
  MoveJ pApxPalete,v1000,z10,toolESTGV;
  MoveL P1SensorFibra,v500,fine,toolESTGV;
  WaitTime 0.2;
  IF DI10_7=1 THEN
    Peca1:=1;
  ELSE
    Peca1:=0;
  ENDIF
  MoveL P2SensorFibra,v500,fine,toolESTGV;
  WaitTime 0.2;
  IF DI10_6=1 THEN
    Peca2:=1;
  ELSE
    Peca2:=0;
  ENDIF
  MoveL P3SensorFibra,v500,fine,toolESTGV;
  WaitTime 0.2;
  IF DI10_7=1 THEN
    Peca3:=1;
  ELSE
    Peca3:=0;
  ENDIF
  MoveL P4SensorFibra,v500,fine,toolESTGV;
  WaitTime 0.2;
  IF DI10_6=1 THEN
    Peca4:=1;
  ELSE
    Peca4:=0;
  ENDIF
  pApxPalete:=Offs(P4SensorFibra,0,0,170);
  MoveJ pApxPalete,v1000,z10,toolESTGV;
ENDPROC
```

Figura 40 – Programação da rotina Verifica Paleta Torno IRB 1400

Fluxograma da rotina Descarrega Paleta Torno para a rampa (rDescarPalTorn):

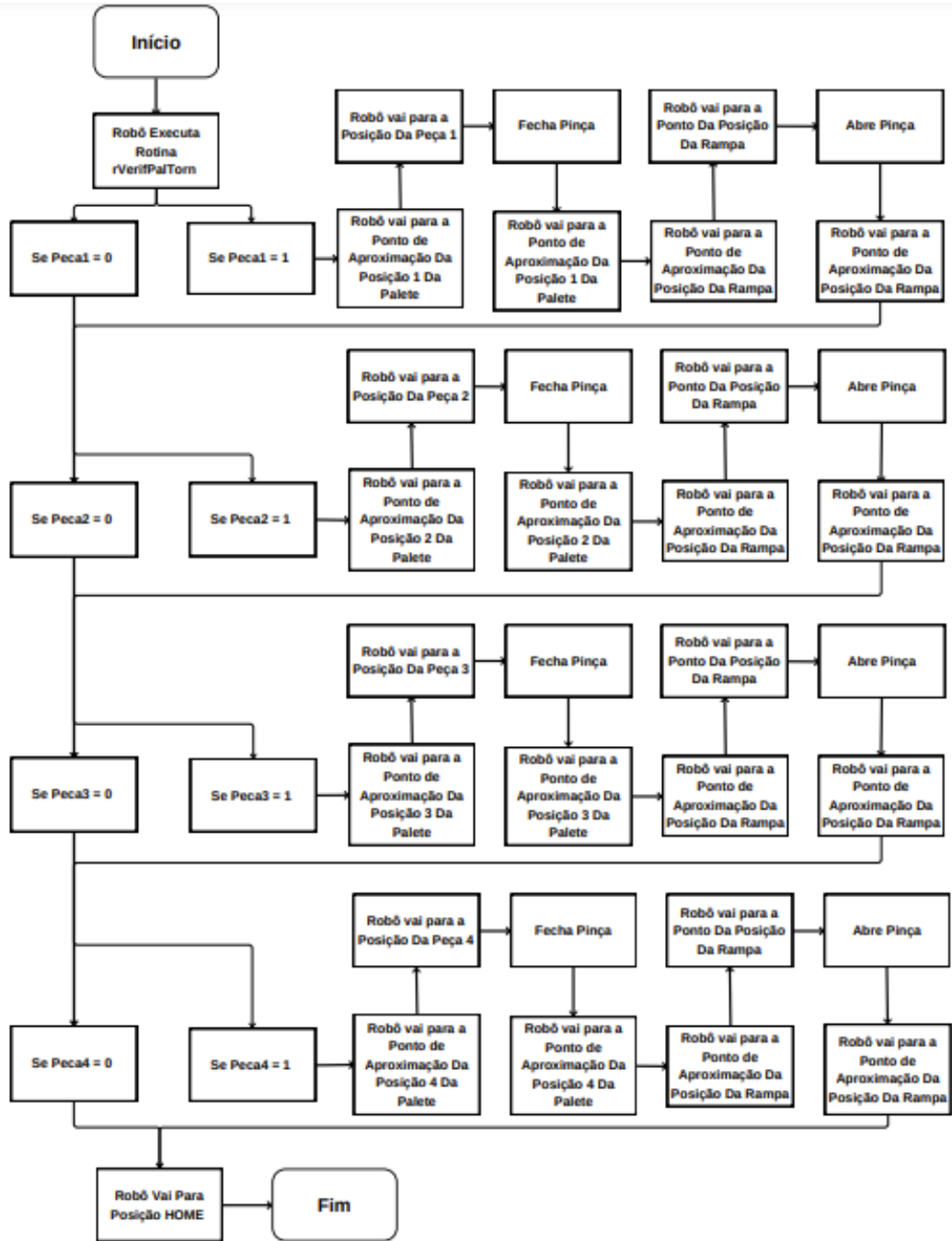


Figura 41 – Fluxograma da rotina Descarrega Paleta Torno para a rampa IRB 1400

Programação RobotStudio da rotina Descarrega Palete Torno para a rampa:

```
PROC rDescarPalTorn()
  rVerifPalTorn;
  IF Peca1=1 THEN
    pDeixaPalete:=p2PaletTorn;
    !MoveL p2PaletTorn,v500,fine,toolESTGV;
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveL pDeixaPalete,v500,fine,toolESTGV;
    rFechaPinca;
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    MoveL pRampaTorn,v500,fine,toolESTGV;
    rAbrePinca;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    Peca1:=0;
  ENDIF
  IF Peca2=1 THEN
    pDeixaPalete:=Offs(p2PaletTorn,0,-88,0);
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveL pDeixaPalete,v500,fine,toolESTGV;
    rFechaPinca;
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    MoveL pRampaTorn,v500,fine,toolESTGV;
    rAbrePinca;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    Peca2:=0;
  ENDIF
  IF Peca3=1 THEN
    pDeixaPalete:=Offs(p2PaletTorn,-80,0,0);
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveL pDeixaPalete,v500,fine,toolESTGV;
    rFechaPinca;
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    MoveL pRampaTorn,v500,fine,toolESTGV;
    rAbrePinca;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    Peca3:=0;
  ENDIF
  IF Peca4=1 THEN
    pDeixaPalete:=Offs(p2PaletTorn,-80,-88,0);
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveL pDeixaPalete,v500,fine,toolESTGV;
    rFechaPinca;
    MoveJ Offs(pDeixaPalete,0,0,100),v1000,z10,toolESTGV;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    MoveL pRampaTorn,v500,fine,toolESTGV;
    rAbrePinca;
    MoveJ Offs(pRampaTorn,0,0,200),v1000,z10,toolESTGV;
    Peca4:=0;
  ENDIF
ENDPROC
```

Figura 42 – Programação da rotina Descarrega Palete Torno IRB 1400

Fluxograma da rotina Carrega Palete Torno (rArmPalTorno):

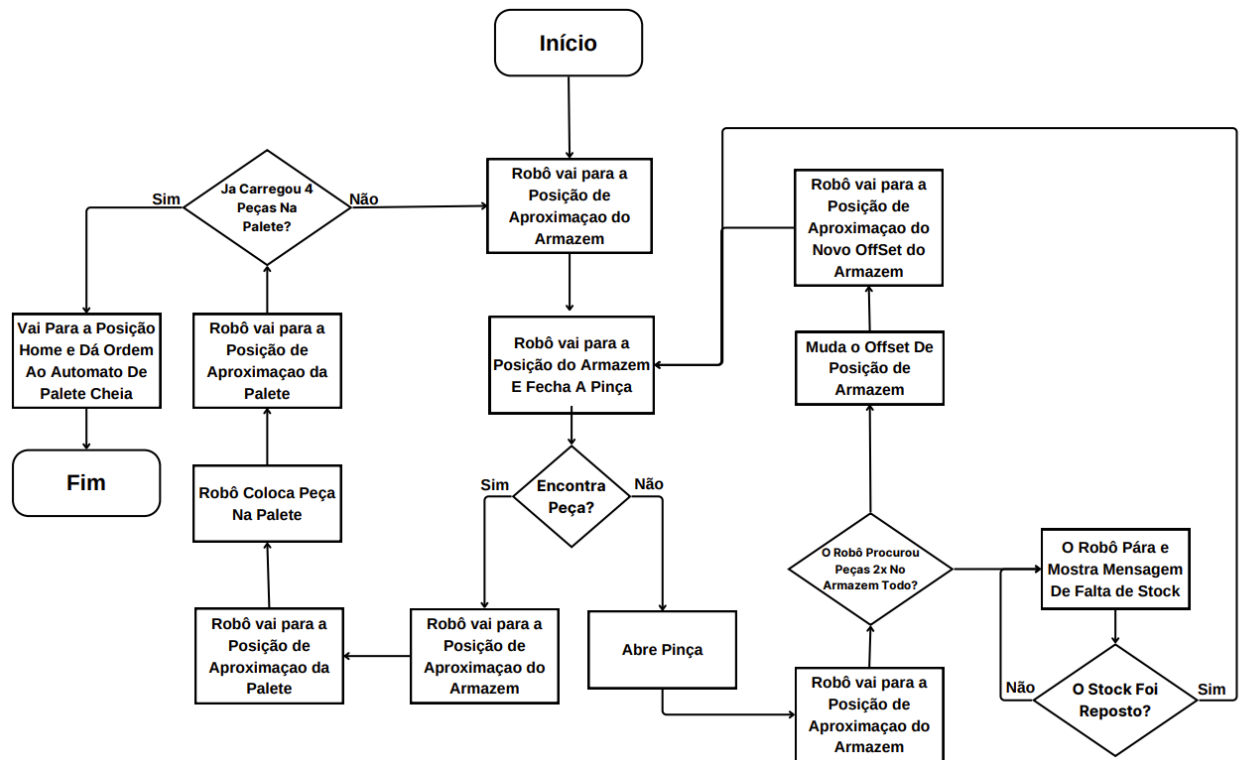


Figura 43 - Fluxograma da rotina Carrega Palete Torno IRB 1400

Programação RobotStudio da rotina Carrega Paleta Torno:

```
PROC rArmPalTorno()  
  ClkReset clock_TempPalete;  
  ClkStart clock_TempPalete;  
  PArmTorn:=pPegPecArmTor;  
Inicio:  
  pApxArmTorn:=Offs(PArmTorn,0,100,0);  
  IF Regressal=1 THEN  
    MoveJ Offs(pPegPecArmTor,0,100,0),v1000,z10,toolESTGV;  
    Regressal:=0;  
  ENDIF  
  IF FaltaPecaTorno=8 THEN  
    MoveJ pApxArmTorn,v1000,z10,toolESTGV;  
    StopMove;  
    MgsFaltaPecaTorno:  
    TPErase;  
    TPWrite "*****";  
    TPWrite "**                *";  
    TPWrite "**  !!! FALTA PECAS ARMAZEM TORNO !!!!  *";  
    TPWrite "**                *";  
    TPWrite "*****";  
    TPreadFK FaltaTorno, "STOCK REPOSTO?", stEmpty, stEmpty, stEmpty, "Sim", "Não";  
    IF FaltaTorno=4 THEN  
      FaltaPecaTorno:=0;  
      StartMove;  
    ENDIF  
    IF FaltaTorno=5 THEN  
      GOTO MgsFaltaPecaTorno;  
    ENDIF  
  ENDIF
```

Figura 44 – Programação da rotina Carrega Paleta Torno IRB 1400

```

MoveJ pApxArmTorn,v1000,z10,toolESTGV;
MoveL PArmTorn,v500,fine,toolESTGV;
rFechaPinca;
IF DI10_11PincaAper=1 THEN
  pApxArmTorn:=PArmTorn;
  pPegPecSaiArTor:=Offs(PArmTorn,0,100,100);
  pLevPecArmTor:=Offs(PArmTorn,0,0,100);
  IF nPeca=1 THEN
    pDeixaPalete:=p1PaletTorn;
  ELSEIF nPeca=2 THEN
    pDeixaPalete:=Offs(p1PaletTorn,0,-88,0);
  ELSEIF nPeca=3 THEN
    pDeixaPalete:=Offs(p1PaletTorn,-80,0,0);
  ELSEIF nPeca=4 THEN
    pDeixaPalete:=Offs(p1PaletTorn,-80,-88,0);
  ENDIF
  pApxPalete:=Offs(pDeixaPalete,0,0,100);
  MoveL PArmTorn,v500,fine,toolESTGV;
  rFechaPinca;
  MoveL pLevPecArmTor,v500,z10,toolESTGV;
  MoveL pPegPecSaiArTor,v1000,z10,toolESTGV;
  MoveJ Offs(pPegPecArmTor,0,100,100),v1000,z10,toolESTGV;
  MoveJ pApxPalete,v1000,z10,toolESTGV;
  MoveL pDeixaPalete,v500,fine,toolESTGV;
  rAbrePinca;
  MoveJ pApxPalete,v500,z10,toolESTGV;
  nPeca:=nPeca+1;
  Regressa1:=1;
ELSEIF DI10_11PincaAper=0 THEN
  rAbrePinca;
  FaltaPecaTorno:=FaltaPecaTorno+1;
  MoveL pApxArmTorn,v500,z10,toolESTGV;
  IF Armazem1=4 THEN
    Armazem1:=1;
    PArmTorn:=pPegPecArmTor;
  ELSE
    PArmTorn:=Offs(PArmTorn,-105,0,0);
    Armazem1:=Armazem1+1;
  ENDIF
ENDIF
IF nPeca<5 THEN
  GOTO Inicio;
ELSE
  ClkStop clock_TempPalete;
  TempoDeCiclo:=ClkRead(clock_TempPalete);
  nPeca:=1;
ENDIF
ENDPROC

```

Figura 45 – Continuação da Programação da rotina Carrega Palete Torno IRB 1400

3.4 Registo Carregamento de Paletes

Após o processo de carregamento o robô IRB 1400 gera um ficheiro (.txt) onde é armazenado o registo do carregamento das paletes(torno/fresa) na parte do armazém, onde este faz o registo da quantidade de paletes carregadas bem como data, hora, tempo de ciclo e a disposição das peças na paleta (0 - s/ peça e 1 - c/ peça).

Na figura 46 podemos verificar um exemplo do mesmo.

```

===== PALETES NO ROBOT IRB-1400 =====
Paleta      Data          Hora          Tempo Ciclo(seg.)  Ok(1)/NOK(0)
  1         2023-06-09   15:38:51     15.26              0 0 1 0
  2         2023-06-09   15:39:17     15.332             1 0 0 1
  3         2023-06-09   15:39:39     15.332             1 0 1 0
===== PALETES NO ROBOT IRB-1400 =====
Paleta      Data          Hora          Tempo Ciclo(seg.)  Ok(1)/NOK(0)
  1         2023-07-09   16:12:05     15.26              0 1 0 0
  2         2023-07-09   16:14:19     15.20              0 1 0 0
  3         2023-07-09   16:16:22     15.30              0 0 1 1
  4         2023-07-09   16:17:05     15.26              0 1 0 0
===== PALETES NO ROBOT IRB-1400 =====
Paleta      Data          Hora          Tempo Ciclo(seg.)  Ok(1)/NOK(0)
  1         2023-08-09   10:12:05     15.20              1 1 1 1
  2         2023-08-09   10:14:19     15.30              0 1 0 0
  3         2023-08-09   10:16:22     15.41              1 1 1 1
  4         2023-08-09   10:17:05     15.26              0 0 0 0

```

Figura 46 – Registo Carregamento das Paletes

3.5 Alarmes

Foi criado um ficheiro (.txt) onde são armazenados os alarmes referentes quer à intrusão como à paragem por parte do operador. Este faz o registo do número do alarme, tipo de alarme (sensor ou operador), data e hora.

Na figura 47 é possível ver um exemplo do mesmo.

```
++++++ REGISTO DE ALARMES DOS SENSORES ++++++
ALARME Nº      TIPO      DATA      HORA
    1      operador  2023-07-17  10:45:26
++++++ REGISTO DE ALARMES DOS SENSORES ++++++
ALARME Nº      TIPO      DATA      HORA
    1      sensor 2  2023-07-18  11:00:14
    2      operador  2023-07-18  11:00:19
    3      sensor 2  2023-07-18  11:12:36
    4      sensor 2  2023-07-18  11:13:52
    5      sensor 2  2023-07-18  11:28:49
    6      sensor 2  2023-07-18  11:40:05
    7      operador  2023-07-18  13:46:42
    8      sensor 2  2023-07-18  13:55:55
    9      sensor 2  2023-07-18  14:20:47
++++++ REGISTO DE ALARMES DOS SENSORES ++++++
ALARME Nº      TIPO      DATA      HORA
    1      sensor 2  2023-07-19  10:00:14
    2      operador  2023-07-19  10:00:19
```

Figura 47 – Alarmes

3.6 Sistema de Segurança

3.6.1 Fluxograma do Sistema de Segurança

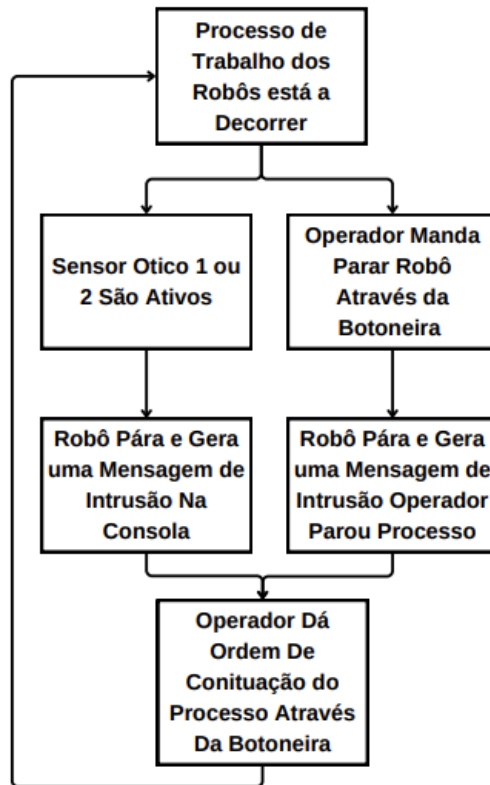


Figura 48 – Fluxograma do Sistema de Segurança

3.6.2 Programação RobotStudio do Sistema de Segurança

```
PROC main()
WAITTIME 0.2;
IF DI10_1=1 THEN          !Verifica se o Sensor óptico 1 (DI10_1) foi ativado
  TPErase;                !Limpa o ecrã da consola e dá a informação de alarme activo
  TPWrite "*****";
  TPWrite "*";
  TPWrite "*      !!!! INTRUSÃO !!!!      *";
  TPWrite "*      !!sensor 1 ativado!!      *";
  TPWrite "*";
  TPWrite "*****";
  rParagem;
  TipoAlarme:=1;
  rRegistaAlarme;        !Faz o registo do alarme no ficheiro de registo
  rArranque;
ELSEIF DI10_2=1 THEN     !Verifica se o Sensor óptico 2 (DI10_2) foi ativado
  TPErase;                !Limpa o ecrã da consola e dá a informação de alarme activo
  TPWrite "*****";
  TPWrite "*";
  TPWrite "*      !!!! INTRUSÃO !!!!      *";
  TPWrite "*      !!sensor 2 ativado!!      *";
  TPWrite "*";
  TPWrite "*****";
  rParagem;
  TipoAlarme:=2;
  rRegistaAlarme;        !Faz o registo do alarme no ficheiro de registo
  rArranque;
ELSEIF DI10_4=1 THEN     !Verifica se a Emergência do operador (DI10_4) foi ativada
  TPErase;                !Limpa o ecrã da consola e dá a informação de alarme activo
  TPWrite "*****";
  TPWrite "*";
  TPWrite "* !!! ROBÔ PARADO PELO OPERADOR !!!! *";
  TPWrite "*";
  TPWrite "*****";
  rParagem;
  TipoAlarme:=3;
  rRegistaAlarme;        !Faz o registo do alarme no ficheiro de registo
  rArranque;
ENDIF
ENDPROC
```

Figura 49 – Programação do Sistema de Segurança

O sistema de controlo e sinalização é fundamental em qualquer sistema de automação, sendo responsável por sinalizar localmente uma ocorrência que possa surgir durante o processo que está a ser executado.

Usámos um programa em *Multitask* em cada robô. Um programa em *Multitask* significa que trabalha em simultâneo com o programa principal.

Na figura 50 temos ilustrado o menu da consola onde é possível ver a *Multitask* ativada.

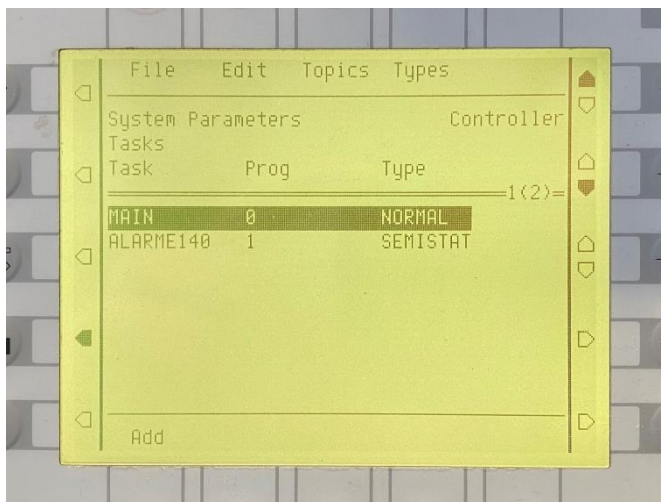


Figura 50 – Menu Visualização Multitask

Em cada setor (fabrico e armazém), existem dois sensores óticos como podemos ver na figura 51 que protegem a área de trabalho de cada robô e, sempre que eles detetarem algo, o robô pára imediatamente o processo.



Figura 51 – Sensores Óticos dos Robôs

O robô só retorna ao programa principal quando receber o OK (botão verde) na consola do operador.

O botão vermelho presente na Botoneira (figura 52) serve para interromper o funcionamento do robô.



Figura 52 - Botoneira para executar a marcha e paragem

Neste sistema de segurança também temos presente um sinalizador luminoso em que acende a luz verde (figura 53) enquanto os robôs estão em funcionamento e acende a luz vermelha (figura 54) quando é interrompido pelos sensores ou é parado manualmente.



Figura 53 - Sinalizador luminoso em marcha c/ lâmpada verde acesa

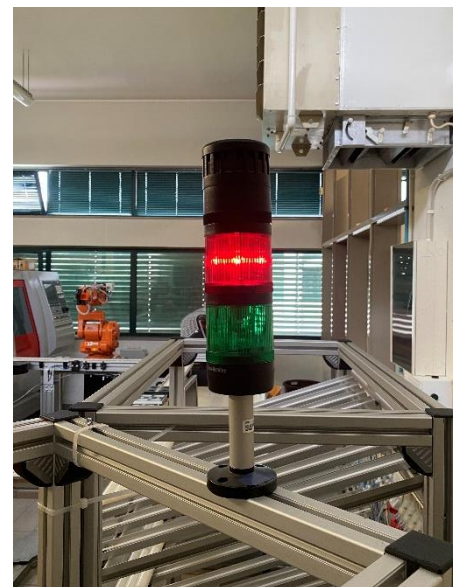


Figura 54 – Sinalizador luminoso em paragem c/ lâmpada vermelha acesa

4. Conclusão e propostas para trabalhos futuros

Neste capítulo é exposto um resumo do trabalho desenvolvido ao longo do projeto e é apresentada uma lista de propostas para trabalhos a desenvolver no futuro.

4.1 Conclusão

Em análise a todo o trabalho desenvolvido conclui-se que todos os objetivos propostos foram conseguidos. O objetivo principal do projeto era a automatização de um processo industrial, controlado por um mestre, o autômato, constituído pelas três *slaves* que são os dois robôs presentes no setor de fabrico e armazém e o transportador.

Depois de vários testes realizados, tanto o autômato como os robôs cumprem as funções para as quais foram projetados, ou seja, o autômato consegue controlar as paletes no transportador e os robôs fazem o carregamento e descarregamento das paletes conforme o pretendido.

Neste projeto, não estava incluído a parte da maquinação, contudo, para tornar o trabalho mais autónomo, decidiu-se realizar a transferência das peças do *buffer* de matérias-primas para o *buffer* de produtos acabados. Este processo simula a parte de uma peça ir à máquina CNC e depois voltar para o *buffer*. Este acréscimo ao projeto faz com que não tenhamos de mexer nas peças, ficando o trabalho totalmente autónomo.

A realização deste projeto vai-nos ser muito útil, pois adquirimos o conhecimento de algumas metodologias para o desenvolvimento de uma CFF. Para além disso, desenvolvemos várias competências, como o sentido de responsabilidade, a capacidade de adaptação a situações imprevistas e o trabalho em equipa.

4.2 Propostas para trabalhos futuros

Mesmo que os objetivos tenham sido cumpridos, existem aspetos que podem completar, de certa forma, o trabalho desenvolvido, tanto a nível da CFF como da interação entre a célula e os operadores.

Uma das principais propostas de desenvolvimento seria a implementação da programação referente à fresagem e tornear das peças nas respetivas máquinas, situação essa que nós anulamos com apenas o transferir peças dos *buffers* da matéria-prima para os *buffers* do produto acabado.

Outra proposta poderia ser a execução de uma programação em que os robôs conseguissem memorizar a presença ou não do número de peças na palete para que não estejam constantemente a verificar a existência de peças nas respetivas paletes para efetuar as descargas.

REFERÊNCIAS

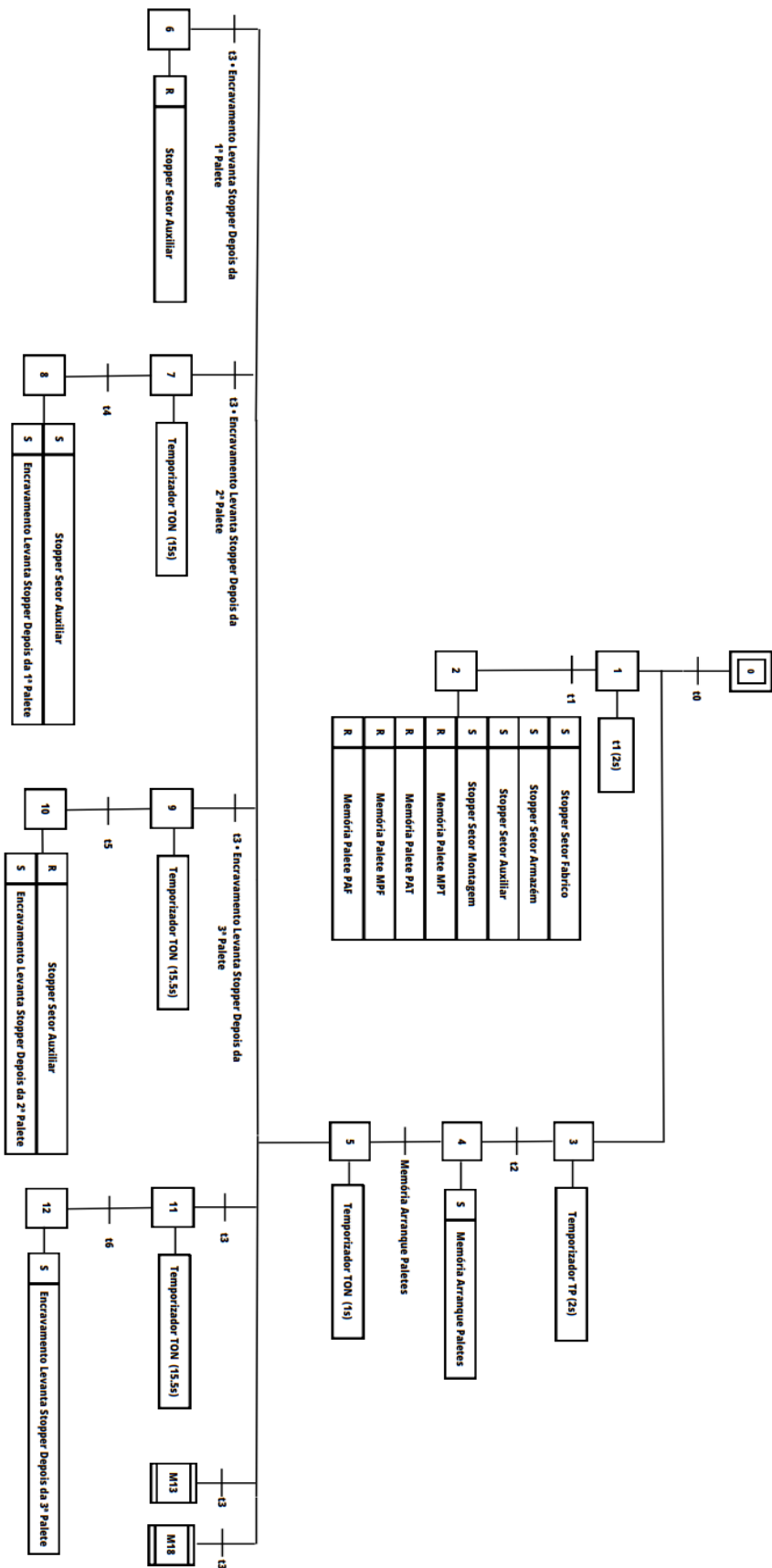
- [1] " PDF Datasheet do Autómato utilizado " [Online] Disponível em: file:///C:/Users/PC/Downloads/6ES72151BG400XB0_datasheet_en.pdf
- [2] A. Ferrolho, Apontamentos das aulas de Automação Industrial, Viseu: DEE-ESTGV-IPV
- [3] A. Ferrolho, Apontamentos das aulas de Robótica Industrial, Viseu: DEE-ESTGV-IPV
- [4] J. Ôlas, Apontamentos das aulas de Robótica Industrial, Viseu: DEE-ESTGV-IPV
- [5] "Sensor Indutivo" [Online] Disponível em: <http://www.tecniar.com.br/noticias/sensores-indutivos-caracteristicas-e-aplicacoes/>
- [6] "RAPID_ReferenceManual" [Online] Disponível em: https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf
- [7] FRANCISCO, António M. S. - Autómatos programáveis. Lisboa: ETEP-Edições Técnicas e Profissionais, 2002. ISBN 972-8480-03-2. Cota [681.5 FRA]
- [8] NOVAIS, José - Programação de autómatos: Método GRAFCET. 2ª ed. Lisboa: Fundação Calouste Gulbenkian, 1994. Cota [681.5 NOV]
- [9] NOVAIS, José - Método sequencial para automatização electropneumática. Lisboa: Fundação Calouste Gulbenkian, 1997. ISBN 972-31-0751-1. Cota [681.5 NOV]
- [10] FIALHO, Arivelto Bustamante - Automação Pneumática: Projectos, dimensionamento e análise de circuitos. São Paulo: Érica, 2003.
- [11] PINTO, J. R. Caldas - Técnicas de Automação.Lisboa: ETEP, 2004. ISBN 972-8480-07-5.
- [12] "Robô ABB IRB1400" [Online] Disponível em: <https://www.eurobots.net/abb-robots-irb1400-m98-p39-pt.html>
- [13] "Robô ABB IRB140" [Online] Disponível em: <https://new.abb.com/products/robotics/industrial-robots/irb-140>

[14] Robotics, K. S. Fu, R. C. Gonzalez, C. S. G. Lee, New York: McGraw-Hill, 1987. ISBN 0-07-022625-3. Cota [681.5 FU ROB]

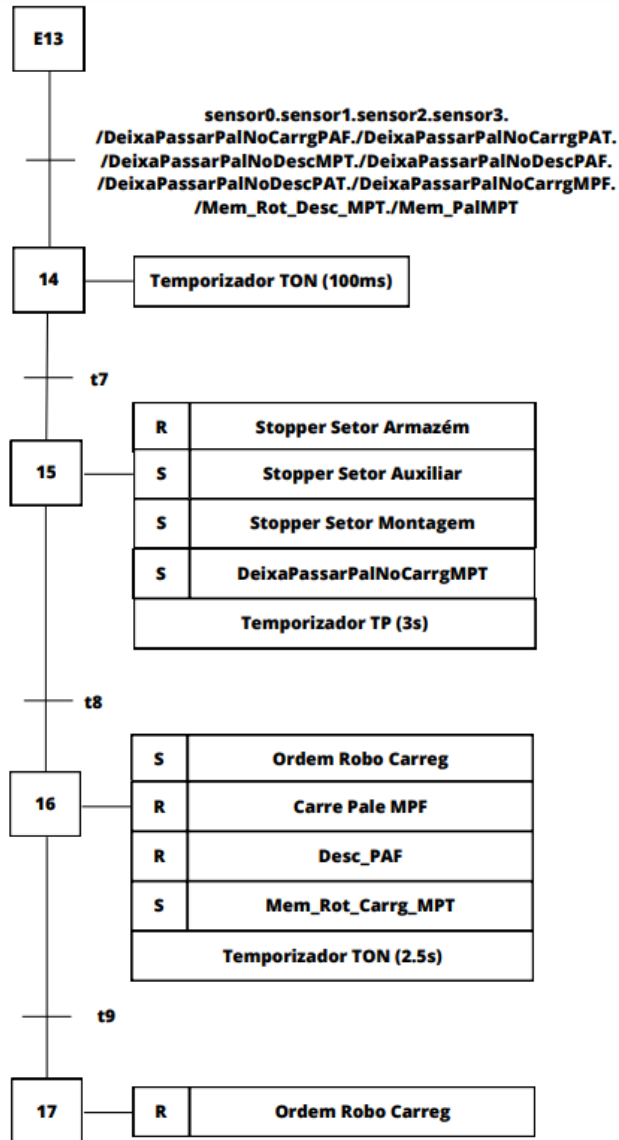
[15] Introduction to Robotics – Mechanics & Control, John J. Craig, Addison - Wesley Publishing Company, 1986. ISBN 0-201-10326-5. Cota [681.5 CRA INT]

[16] Introduction to Robotics, Phillip John McKerrow, Singapore :Addison-Wesley, 1990, ISBN: 0-201-18240-8. Cota [681.5 MCK INT]

ANEXO 1 - GRAFCET PRINCIPAL (FIRSTSCAN E LARGADA TEMPORIZADA DAS PALETES)



ANEXO 2 – GRAFCET ORDEM DE CARREGAR PALETE MPT NO ROBÔ IRB 140



ANEXO 3 - GRAFCET PALETE MPT CARREGADA NO ROBÔ IRB 1400

