

UNIVERSIDADE DO ESTADO DO AMAZONAS - UEA
ESCOLA SUPERIOR DE TECNOLOGIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

IGOR PEDROSA SOARES

**AUTOMATIZAÇÃO NO PROCESSO DE RECONFIGURAÇÃO
E AJUSTES LATERAL DE UMA ESTEIRA DE MANUSEIO DE
PEÇAS PARA IMPRESSÃO DE CÓDIGOS VIA NOVOS
PARADIGMAS DE AUTOMAÇÃO**

Manaus

2022

IGOR PEDROSA SOARES

**Automatização no Processo de Reconfiguração e Ajustes Lateral de
uma Esteira de Manuseio de Peças para Impressão de Códigos Via
Novos Paradigmas de Automação**

Versão Original

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Controle e Automação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. José Ruben Sicchar Vilchez

Coorientador: Prof Dr. Raimundo Corrêa de Oliveira

Manaus

2022

Automatização no processo de reconfiguração e ajustes lateral de uma esteira de manuseio de peças para impressão de códigos via novos paradigmas de automação

IGOR PEDROSA SOARES

Trabalho de Conclusão de Curso (TCC) apresentado à Escola Superior de Tecnologia da Universidade do Estado do Amazonas como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Aprovado por:

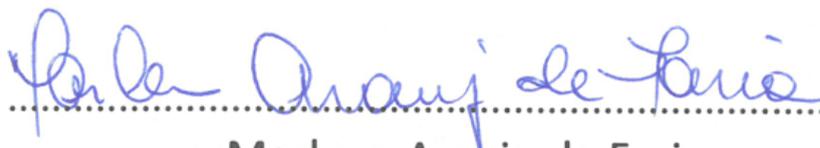
p/



Edson Farias de Oliveira



Renan Landau Paiva de Medeiros



Marlene Araujo de Faria



Igor Pedrosa Soares

À todos ao meu lado,
dedico.

AGRADECIMENTOS

À meus pais, Luiz e Lucinete , que sempre me incentivaram a alcançar meus sonhos e nunca deixaram faltar nada para que eu pudesse alcançá-los. Vocês são o meu exemplo de vida e sem vocês, nada disso seria possível.

Ao mestre Charles Luiz Melo (*in memoriam*) que me fez realmente fazer parte do curso e ao meu irmão Arle Melo (*in memoriam*) com nossas noites de estudo intenso.

A minha amiga e parceira, Juliany , por estar comigo neste período de foco.

Aos meus orientadores, Prof. Dr. José Rubens Sicchar Vilchez e Prof Dr. Raimundo Corrêa de Oliveira, por todo o auxílio e confiança durante a elaboração deste projeto. Obrigado por ter aceitado o convite e me orientado da melhor maneira possível para finalizar este trabalho.

A todos os meus amigos que fiz na faculdade, com ênfase a Carlos Setúbal, Vitor Gadelha, Francisco Arcos e Lucas Batista.

À Digiboard, por fornecerem apoio para execução do projeto.

Aos meus familiares Honoratos com toda sua hospitalidade e carisma.

A Universidade do Estado do Amazonas, por possibilitar a realização deste curso.

A todos que contribuíram direta ou indiretamente para realização deste trabalho.

"Se você não se permitir viver a derrota, não irá vivenciar a vitória."

(Gabriel 'Fallen' Toledo)

RESUMO

A automatização de processos industriais têm ganhado bastante força nas últimas décadas por seus ganhos de produtividade e ascensão da indústria 4.0. Processos de *setup* para produção de novos lotes e inserção de novos produtos são um tanto problemáticos pela demora na adequação e ajuste do ambiente ao produto, processos automáticos sem um planejamento a longo prazo tendem a causar problemas danosos à produtividade sob alta demanda. Sendo assim, este trabalho propõe o desenvolvimento do sistema de automação para uma linha de impressão de códigos de barras totalmente automatizada criando o modelo em *Redes de Petri* baseado nos novos paradigmas de automação, no caso os *Sistemas de Manufatura Reconfiguráveis (SMR)*. Analizando o modelo do fluxo de automação analisou-se o *takt-time* dos processos utilizando *MOST* com o intuito de identificar possíveis gargalos e validar o processo planejado. Após o planejamento e validação é desenvolvido a programação dos *Controladores Lógicos Programáveis (CLPs)* utilizando linguagem Ladder totalmente baseada no modelo em *Redes de Petri*. É realizado também o ajuste de parâmetros nos dispositivos responsáveis pelo movimento focado no controle de posição. O sistema apresentou durante os testes um *takt-time* de 10 segundos utilizando 10% da velocidade máxima do sistema, com rápida adaptação a novos produtos e mudanças de *setup*.

Palavras-chave: Automatização de processos, *Redes de Petri*, SMR, códigos de barras.

ABSTRACT

The automation industrial processes has gained a lot of solidity in recent decades due to its productivity gains and rise up of industry 4.0. *Setup* processes for the production of new batches and insertion of new products are somewhat problematic due to the delay in adapting and adjusting the environment to the product, automatic processes without long-term planning tend to cause harmful problems to productivity under high demand. Therefore, this work proposes the development of an automation system for a fully automated barcode printing line, creating the model in *Petri Nets* based on new automation paradigms, in this case the *Manufacturing Systems Reconfigurable (RMS)*. Analyzing the automation flow model, the *takt-time* of the processes was analyzed using *MOST* in order to identify possible bottlenecks and validate the planned process. After planning and validation, the programming of *Programmable Logic Controllers (PLCs)* is developed using Ladder language totally based on the model in *Petri Nets*. It is also performed the adjustment of parameters in the devices responsible for the movement focused on position control. During the tests, the system presented a *takt-time* of 10 seconds using 10% of the maximum system speed, with rapid adaptation to new products and *setup* changes.

Keywords: Process automation, Petri Nets, RMS, barcode.

LISTA DE ILUSTRAÇÕES

1	Hierarquia de automação segundo IEC 62264-3.	22
2	Exemplo de uma Rede de Petri.	24
3	Tela inicial do PIPE.	25
4	CLP AS300 da marca Delta.	25
5	Dave Emmett em 1968 implementando o primeiro CLP em uma linha da GM.	26
6	Servomotor e um servodriver da marca Delta.	27
7	Tela inicial do ASDASoft.	28
8	Comando MOV escrevendo no registrador D1.	29
9	Tela inicial do software ISPSOft com um programa em Linguagem Ladder.	30
10	Tela inicial do software ISPSOft com um programa em Linguagem Ladder.	31
11	Entrada e pinagem da conexão física do protocolo CANOpen.	32
12	Configuração do CANOpen pelo HWCONFIG.	32
13	Relação de entradas de saídas físicas de um CLP.	33
14	Diagrama de um CLP com saídas a relé	33
15	Arquitetura do funcionamento do protocolo Modbus.	34
16	Exemplo de um leitor de código de barras	37
17	Exemplo de um robô cartesiano da marca WITTMANN BATTENFELD.	38
18	Atuadores lineares pneumáticos.	38
19	Vetosa pneumática em croqui.	39
20	Construção do projeto de automação	40
21	Diagrama da arquitetura do sistema de esteiras	43

22	Modelo do sistema em Redes de Petri dividido em macrolugares	44
23	Detalhamento dos processos do primeiro estágio	45
24	Detalhamento da sub-hierarquia "Setup de máquina".	46
25	Modelagem dos sistema do processo de inserção das PCBs sobre a esteira de forma individual para posterior impressão dos códigos.	47
26	Modelagem dos sistema do manipulador robótico responsável por inserir as PCBs uma por vez.	48
27	Modelagem dos sistema da esteira de impressão dos códigos de barras nas PCBs.	49
28	Modelagem dos sistema de inspeção dos códigos de barras impressos.	50
29	Modelagem em Rede de petri do manipulador responsável pela retirada das placas reprovadas após a inspeção.	51
30	Modelagem em Rede de petri do estágio final do processo de impressão de PCBs.	52
31	Modelagem em Rede de petri do sistema de requisição do manipulador de inserção de placas individuais em simulação.	53
32	Modelagem em Rede de petri do sistema de inspeção dos códigos de barras impressos em simulação com sistema de segurança atuando.	54
33	Modelagem em Rede de petri do sistema de inspeção dos códigos de barras impressos em simulação com sistema de segurança resetado.	55
34	Testes com resoluções de câmera para definição do equipamento de captura de imagem.	61
35	Mini computador NUC responsável pelo gerenciamento das informações e processamentos computacionais.	62
36	Arquitetura das esteiras.	65
37	Arquitetura do sistema de manipuladores e segurança.	66
38	Parametrização de servomotores no ASDASoft.	67
39	Criação de um projeto no ISPSOft.	68

40	Menu de extensões do ISPSOft.	69
41	Menu do HWCONFIG.	70
42	Início do programa em Ladder.	70
43	Estrutura do comando de controle de servomotores.	71
44	Estrutura do comando de controle de servomotores.	71
45	Inicialização do processo de ajuste lateral.	72
46	Bloco de divisão do valor por 2 usando apenas metade do valor para fazer o movimento.	73
47	Sensor mecânico fim de curso instalado nas partes internas e externas das esteiras	74
48	Valor de correção de centralização em relação ao deslocamento do sen- sor fim de curso com o centro.	74
49	Linha de comando para fechamento total das esteiras aferindo a origem do sistema.	75
50	Linha de comando de recebimento do sinal de confirmação da posição de origem do sistema.	75
51	Linha de comando responsável ela abertura da esteira até o valor cal- culado e guardado em D26.	76
52	Ajuste lateral finalizado indicado pela flag do servomotor.	76
53	Relés virtuais utilizados para indicar fim de um processo e início do seguinte.	76
54	Projeto do modelo em CAD.	77
55	Projeto do Modelo montado.	78
56	Painel elétrico de comando das esteiras.	79
57	Painel elétrico de comando dos manipuladores e segurança.	79
58	Ajuste de parâmetros das esteiras via testes.	79
59	Ajuste de parâmetros dos manipuladores via testes.	79

60	Porcentagem de utilização dos servomotores em relação a sua velocidade angular máxima de 6000rpm.	81
61	Desempenho do sistema a partir do processo com menor capacidade de flexibilização, o processo de inserção de PCBs individuais.	82
62	Tempo de setup utilizando os testes em cada placa com velocidades de 50 rpm e 100 rpm.	83
63	Projeção de produtividade do sistema de acordo com a variação no valor do tempo de processo de inserção de placas individuais na esteira por um manipulador robótico.	84
64	Testes de impressão ados códigos de barras.	85

LISTA DE TABELAS

1	Tipo de variável com estado binário.	29
2	Tabela de conversão entre TMU e tempo padrão.	35
3	Sequência de movimentos do processo da figura 23 utilizando método de cronoanálise MOST	56
4	Sequência de movimentos do processo da figura 25 utilizando método de cronoanálise MOST	56
5	Sequência de movimentos do processo da figura 26 utilizando método de cronoanálise MOST	57
6	Sequência de movimentos do processo da figura 27 utilizando método de cronoanálise MOST	57
7	Sequência de movimentos do processo da figura 28 utilizando método de cronoanálise MOST	58
8	Sequência de movimentos do processo da figura 30 utilizando método de cronoanálise MOST	58
9	Sequência de movimentos do processo da figura 29 utilizando método de cronoanálise MOST	59
10	Valores inseridos nos parâmetros P3-00 de cada servodriver.	68
11	Sequência de movimentos do processo comparando o tempo estimado pelo método MOST com o tempo real medido	80
12	Tamanho de PCBs testadas.	83

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CAD	Computer Aided Design
CAN	Controller Area Network
CLP	Controlador Lógico Programável
CPU	Central Processing Unit
GM	General Motors
MOST	Maynard Operation Sequence Technique
NGMS	Next Generation Manufacturing System
PCB	Printed Circuit Board
RAM	Random Access Memory
ROM	Read-Only Memory
RPM	Rotation Per Minute
SMR	Sistemas de Manufatura Reconfiguráveis
SR	Smart Reconfigurables
TMU	Time Measurement Unit

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivos	18
1.1.1	Objetivo Geral	18
1.1.2	Objetivos Específicos	18
1.2	Trabalhos Relacionados	19
1.3	Divisão do Trabalho	20
2	REFERENCIAL TEÓRICO	21
2.1	Automação Industrial	21
2.2	Novos Paradigmas de Automação	22
2.3	Redes de Petri	23
2.3.1	PIPE - Platform Independent Petri Net Editor	24
2.4	CLP - Controlador Lógico Programável	25
2.4.1	Histórico	26
2.4.2	Servomotores	27
2.4.3	Linguagem de Programação Ladder	28
2.4.4	ISPSOft	30
2.4.5	CANOpen	31
2.4.6	Fonte de alimentação 24V	32
2.4.7	Entradas e saídas do CLP	32
2.4.8	Modbus	34
2.5	MOST - Maynard Operation Sequence Technique	34
2.6	Poka Yoke	35

2.7	Takt-Time	35
2.8	Setup de máquina	36
2.9	Impressora laser e gravação	36
2.10	Leitor de código de barras	36
2.11	Esteira Industrial	37
2.12	Robô cartesiano	37
2.13	Atuadores lineares	38
2.14	Ventosas pneumáticas	39
3	MÉTODOS E MATERIAIS	40
3.1	Levantamento de Requisitos	41
3.2	Fluxograma inicial	42
3.3	Modelagem em Redes de Petri	44
3.3.1	Sistema de segurança e emergência	52
3.4	Cronoanálise utilizando MOST	55
3.5	Arquitetura física do sistema e definição dos componentes	59
3.5.1	Tipos de componentes	60
3.5.1.1	Sensores	60
3.5.1.2	Atuadores elétricos	60
3.5.1.3	Comunicação	60
3.5.1.4	Captura de imagem e inspeção de códigos	61
3.5.1.5	Gerenciamento do sistema	61
3.5.1.6	Controlador Lógico Programável	62
3.5.1.7	Atuadores pneumáticos	62
3.5.2	Quantidade de componentes	63
3.5.2.1	Para os Sensores	63

3.5.2.2	Captura de imagem	64
3.5.2.3	Inspeção dos códigos	64
3.5.2.4	Gerenciamento do sistema	64
3.5.2.5	Controlador Lógico Programável	64
3.5.2.6	Atuador pneumático	64
3.5.3	Arquiteturas	64
3.6	Programação do sistema	66
3.6.1	Programação das esteiras	67
4	RESULTADOS E DISCUSSÃO	77
4.1	Valores constantes posição do sistema	79
4.2	Tempo estimado X tempo medido	80
4.3	Fluxo de automação	82
4.4	Modelos de PCB	82
4.5	Takt-time do sistema	83
5	CONSIDERAÇÕES FINAIS	86
	Referências	88
	Apêndice A	92
	Apêndice B	102
	Apêndice C	114
	Apêndice D	126

1 INTRODUÇÃO

Nas últimas décadas, a automação industrial tornou-se o alicerce de indústrias que buscam evoluir seu sistema produtivo (BANGEMANN et al., 2014). Dessa forma, é necessário que o maquinário possua configurações dinâmicas para atender produtos com diversas especificações. Entretanto, em diversos casos, para que isto ocorra de forma acurada, uma ou mais linhas de produção precisam ter seus processos interrompidos, acarretando perdas na produtividade da fábrica.

Alterar um processo e atualizar um fluxo de manufatura em produção requer investimentos e retreinamento de colaboradores para adaptação ao novo sistema. Dessa forma, visar um longo ciclo de vida de equipamentos de manufatura passa a ser uma premissa necessária para uma produção consistente. Além do mais, garantir que um único sistema flexível atue de forma mais eficiente que o processo de *setup* corrente é um desafio que requer a aplicação de novas metodologias (RÖSIÖ; JACKSON, 2009).

Dessa forma, um Sistema de Manufatura Reconfigurável (SMR) é uma opção viável e capaz de prover escalabilidade durante a fabricação de novos produtos, tornando o processo satisfatório e integrando inovação à indústria. Esta metodologia pode ser aplicada em diversos contextos da automação, pois sua principal característica é ser adaptável ao contexto da manufatura, como a falha em máquinas ou mudanças na demanda de mercado.

Assim sendo, visando a automatização de um processo manual, como a fixação de códigos etiquetados em placas de circuito impresso, este trabalho propõe o uso de um SMR automático de ajuste lateral e manuseio integrado à gravação de códigos a laser.

A proposta propõe melhorar o fluxo de produção atual, que é semi-automatizado de fixação de etiquetas de códigos de identificação de placas-mãe de computador a fim de ganhar rapidez, produtividade, flexibilidade, conforto e segurança (para os colaboradores) através do desenvolvimento de um sistema automatizado integrado, baseado

em sistemas de manufatura reconfigurável e paradigmas da automação.

Oferecendo-se um processo moderno, eficiente, robusto, inteligente, automático e versátil, capaz de oferecer não somente um processo de produção célere, de qualidade, eficiente e produtivo, mas também flexível haja vista a necessidade de serem produzidas vários modelos de PCBs(Printed Circuit Boards) de acordo com a demanda planejada.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo deste trabalho é projetar, desenvolver, implementar e validar um sistema de automação de manufatura reconfigurável para inserção de códigos de barras e identificação do produto em placas mãe, que seja capaz de receber todos os tipos e tamanhos das PCBs estimadas.

1.1.2 Objetivos Específicos

Para alcançar o objetivo deste projeto, procura-se alcançar os seguintes objetivos específicos:

- Analisar a viabilidade do projeto quanto ao tempo de processo e takt time do processo atual;
- Identificar os postos ideais para aplicação para reconfiguração de manufatura;
- Projetar a modelagem do fluxo de automação;
- Definir a arquitetura do sistema baseado em seus componentes e sua hierarquia;
- Programar o CLP utilizando linguagem Ladder baseado na modelagem projetada;
- Capturar resultados de tempo e produtividade;

1.2 Trabalhos Relacionados

Esta seção visa apresentar alguns trabalhos relacionados à SMR com os tópicos abordados anteriormente: Sistemas Reconfiguráveis de Manufatura e automação de processos industriais.

Silva (2016) apresenta soluções técnicas em um sistema multiagentes e holon de controle utilizando bancadas que permitem simular postos de trabalho e, por sua vez, podem ser alterados e reconfigurados de acordo com o fluxo desejado. O processo é modelado em Redes de Petri e o desempenho interativo é identificado através da plataforma JADE+WAVE. No caso, da Silva estimula diferentes cenários de reconfiguração dos postos de trabalho, além de novas demandas de produção. Ele confirma agilidade e flexibilidade dos sistemas reconfiguráveis com ótimos ganhos relacionados a novas demandas que necessitam de alteração da planta industrial.

Chora (2021) foca em sistemas Ciber-Físicos propondo arquiteturas de controle baseados em dispositivos de baixo custo (arduíno) e paradigmas de Sistemas Distribuídos de Manufatura com o objetivo de atender a demanda de produtos personalizados em modelos de produção em massa. Os princípios apresentados são aplicados para obter um sistema distribuído, integrável, reconfigurável e descentralizado. A análise conclusiva da utilização dos novos paradigmas de produção salienta as dificuldades obtidas com relação aos recursos utilizados, porém obteve sucesso ao validar sua aplicação em um caso laboratorial, em seu caso, a comunicação entre os dispositivos utilizados foi o ponto chave para validar a solução, complementa ainda afirmando que tal arquitetura pode ser aplicada a diversos tipos de dispositivos Iot para se obter um sistema reconfigurável e escalável.

Andersen, Nielsen e Brunoe (2016) faz uma revisão sobre os pré-requisitos e dificuldades para a realização da manufatura reconfigurável no ambiente fabril, onde são observadas 3 condições principais como requisito para sua implementação, sendo o primeiro e principal a mentalidade para a reutilização de equipamentos, visto que utilizar novos equipamentos de manufatura para novos produtos era menos complicado. O segundo requisito seria a divisão de responsabilidade e integração entre as equipes responsáveis por desenvolvimento e operação devido à grande variabilidade de configuração, e em terceiro, o requisito está relacionado com a escalabilidade e conversibilidade do sistema de produção ao longo do ciclo de vida e a previsão de mudanças e inserções de novos produtos, o autor menciona que há diversos requisitos mais encontrados no

decorrer da implementação do método. Em tal estudo de caso, a empresa focou-se na implementação do primeiro requisito realizando um co-desenvolvimento do projeto entre os desenvolvedores e a produção de forma contínua, o projeto obteve bons resultados de produtividade e melhores ainda relacionados ao desgaste no trabalho manual de colaboradores.

Morgan et al. (2021) promove uma revisão e pesquisa de Sistemas de Manufatura Reconfiguráveis voltado à análise do estado da arte de sistemas distribuídos e descentralizados de controles de máquina e inteligência de máquinas. Ele projeta a próxima geração de máquinas como sendo Smart Reconfigurables (SR), aplicando os conceitos de Indústria 4.0. Tais sistemas terão capacidade de reconfiguração inteligente, isto é, capacidade de saber quando mudar. Estes sistemas causariam um impacto positivo com elevados níveis de eficiência de produção e otimização de fluxo. O trabalho explicita bem as bases dos sistemas reconfiguráveis porém, apenas em fase teórica.

Rösiö et al. (2019) propõe um critério de avaliação para mensurar a capacidade de reconfiguração de sistemas no meio fabril realizando um estudo de caso realizado em modelos reconfiguráveis existentes de indústria automotiva. O sistema de manuseio de peças por veículos auto guiados obteve bons resultados relacionados a modularidade, diagnosticabilidade e integrabilidade, resultados ruins com escalabilidade e divergindo das expectativas, a customização. As regras de avaliação exibem critérios pontuais que podem ser utilizadas como base em novos sistemas visando casos de longo prazo.

1.3 Divisão do Trabalho

O trabalho está organizado da seguinte maneira: após a presente introdução, o Capítulo 2 aborda o Referencial Teórico, o qual aborda todos os assuntos necessários para a implementação do projeto. O Capítulo 3 consiste na implementação do projeto, indicando os materiais e métodos utilizados na elaboração do mesmo. O capítulo 4 apresenta os resultados e discussões obtidos no decorrer do projeto. Por fim, no capítulo 5, apresentam-se as considerações finais e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

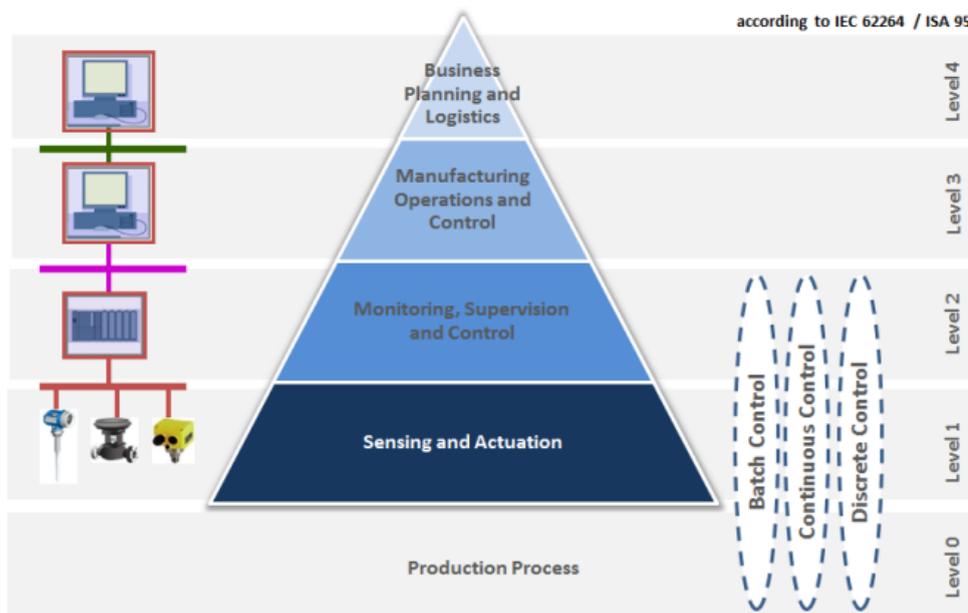
Nesta seção é apresentada a base teórica para construção do protótipo tanto relacionada às partes de hardware e software, quanto às metodologias aplicadas.

2.1 Automação Industrial

Automação é uma palavra advinda do latim *automatus* e significa mover-se por si (CAMPANA; OPLUSTIL, 2011), pode ser explicado como, em uma operação presencial ou remota, um sistema realiza um trabalho sem a necessidade de interferência humana (RIBEIRO, 1999). A automação se dá por uma dinâmica organizada onde seu objetivo principal é o desenvolvimento da humanidade, trabalhar com eficiência é a chave para obter qualidade, consistência e produtividade em quaisquer processos de produção. Automatizar processos é um progresso natural de um sistema fabril em ascensão (FIALHO, 2018).

A automação segue níveis em 5 escalas de uma pirâmide, as camadas vizinhas estão sempre correlacionadas, onde o primeiro nível e mais abrangente é relacionado aos equipamentos de chão de fábrica como sensores ou botões (elementos de entrada) e atuadores (elementos de saída discreta ou analógica) (SOUZA, 2013), O segundo nível está relacionado a itens de controle automatizado como os CLPs, o terceiro nível destina-se a supervisionar os processos realizados e gerar relatórios enviando dados a camada acima, o quarto nível é responsável pelo planejamento de produção e logística, portanto voltada ao gerenciamento da planta, o quinto nível diz respeito ao planejamento estratégico e gerenciamento corporativo, abrangendo todas as tomadas de decisões que afetam a empresa como um todo.

Figura 1: Hierarquia de automação segundo IEC 62264-3.



Fonte: (BANGEMANN et al., 2014)

2.2 Novos Paradigmas de Automação

Os sistemas de manufatura têm como característica três principais objetivos, são eles: custo, qualidade de produto e capacidade de respostas às demandas do mercado (KOREN; GU; GUO, 2018). Processos industriais lidam com uma crescente demanda de novos produtos, o que inviabiliza muitos sistemas implementados de forma mais tradicional trazendo uma necessidade da flexibilização das linhas de produção, tal requisito abriu espaço para novos sistemas de fabricação chamados Next Generation Manufacturing Systems (NGMS). Os Sistemas de Manufatura Reconfiguráveis (SMR) fazem parte do grupo de sistemas NGMS e são capazes de adequar seus software e hardware de acordo com a demanda requerida (BORTOLINI; GALIZIA; MORA, 2018).

SMR são novos paradigmas de manufatura flexível capazes de suprir as demandas variáveis do mercado manufaturando produtos de alta qualidade em quantidades escalonadas (LIRAVIASL et al., 2015). Tal escalonamento é possível por conta de alterações em sua estrutura física juntamente com uma adequação no software que permitem configurações de máquina em um tempo extremamente hábil proporcionando altas taxas de rendimento e aceitação de novos produtos (KOREN et al., 1999). Com

base nisso, podemos estimar longos períodos de utilização dos equipamentos sem se tornarem obsoletos e conseqüentemente menos gastos com novas máquinas.

Um SMR é composto por 5 características base:

- **Modularidade:** A modularidade promove a reutilização de equipamentos do sistema, desde componentes lógicos a físicos, até mesmo a nível de manufatura e peça fabricada;
- **Integrabilidade:** O sistema deve estar preparado para uma rápida integração com o processo atual e preparado para a integração com uma possível nova tecnologia;
- **Convertibilidade:** O sistema está preparado para uma rápida troca entre produtos ou inserção de um novo produto;
- **Diagnóstico:** O sistema é capaz de identificar a origem de problemas ocasionais;
- **Customização:** Ter uma capacidade de flexibilização de alterações de acordo com a necessidade da aplicação.

Utilizando dos requisitos citados, conceitos de indústria 4.0 são aplicados e integrados ao sistema modular e aplicado para suprir a necessidade de integrabilidade das SMR, sabe-se que a indústria 4.0 baseia-se na implementação de tecnologias de comunicação e informação que melhoram a flexibilidade dos sistemas, produtividade, gerenciamento e qualidade, gerando benefícios significativos em diversos setores de uma empresa, sendo assim considerada a quarta revolução industrial ou quarto paradigma de produção industrial (SACOMANO et al., 2018).

2.3 Redes de Petri

Modelos são representações das características do objeto de estudo onde há uma correlação equivalente com o sistema real, de modo que possamos analisar o objeto e suas propriedades obtendo resultados e estimativas sem a necessidade de testes com o modelo real.

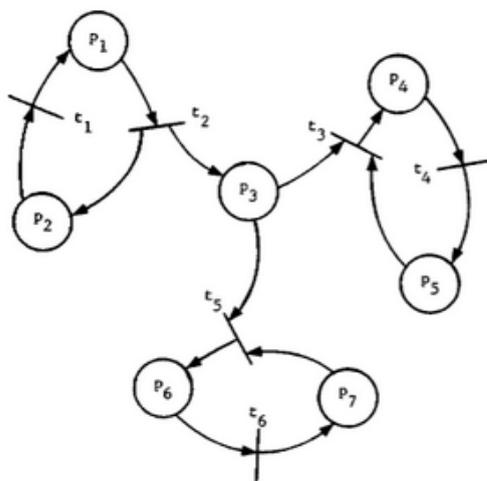
Para compreender o sistema de automação e obter seus requisitos, é necessário construir uma modelagem baseada em como o fluxo de automação deverá se comportar, para isso utilizaremos as Redes de Petri. Formulada na tese de doutorado de

Carl Adam Petri e utilizada apenas focada em atribuir uma solidez própria até os anos 80, onde, desde então tem sido base para diversos trabalhos relacionados a análise de sistemas distribuídos (FOYO; MIRALLES; SILVA, 2011).

Rede de petri é uma técnica formal de modelagem baseada em grafos marcados abstratos voltada a análise de fluxo de informações em um sistema. A principal aplicação das Redes de Petri é obter um modelo de eventos discretos de sistemas. Os gráficos de Redes de Petri possuem dois tipos de nós, onde, os círculos são chamados de lugares e as barras chamadas de transições, os lugares e transições são ligados entre si por arcos (PETERSON, 1977).

O método de modelagem é normalizado pela ISO/IEC 15909-1:2004, permite provar propriedades de protótipos desenvolvidos em diversas áreas como processos de manufatura, software e hardware, mecatrônica, dentre outros. A norma revisada (ISO/IEC 15909-1: 2019) especifica mais detalhadamente campos genéricos de aplicação citadas na norma de 2004 e apresenta novos campos baseados em aplicações mais recentes.

Figura 2: Exemplo de uma Rede de Petri.



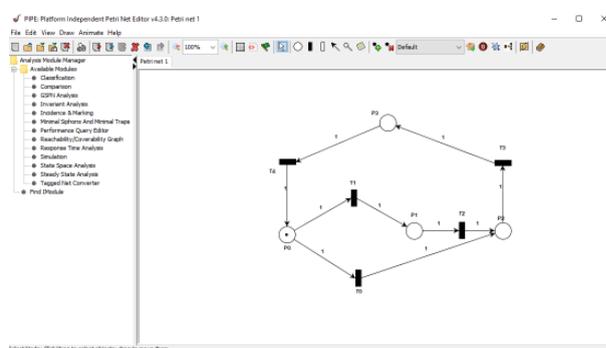
Fonte: (PETERSON, 1977)

2.3.1 PIPE - Platform Independent Petri Net Editor

O PIPE (Platform Independent Petri Net Editor) é uma ferramenta que possui código aberto utilizada para criar e analisar Redes de Petri Estocásticas Generalizadas (GSPNs). É implementado em Java e possui uma interface gráfica simples e de fácil utilização, foi construído no Imperial College London baseado em diferentes projetos

estudantis (BONET et al., 2007).

Figura 3: Tela inicial do PIPE.



Fonte: autor

2.4 CLP - Controlador Lógico Programável

Um dos principais componentes para implementação de projetos de automação em ambientes fabris é o CLP. Segundo a Associação Brasileira de Normas Técnicas (ABNT), os Controladores Lógicos Programáveis (CLPs) são dispositivos eletrônico-digitais com software e hardware voltados para solucionar problemas relacionados à automação industrial (ZANCAN, 2016).

(BOLTON, 2015) define os CLPs como equipamentos que controlam plantas de processo por meio de uma lógica implementada com natureza discreta ou contínua. A NEMA (National Electrical Manufacturers Association) define os CLPs como um aparelho eletrônico-digital programável que utiliza instruções para realizar lógicas complexas e controlar máquinas ou processos por módulos de entradas e saídas.

Figura 4: CLP AS300 da marca Delta.



Fonte: (DELTA, 2022)

Um CLP possui certa semelhança com um computador comum, é composto por

uma CPU (Central Processing Unit), memória RAM e ROM, trabalha com diversos protocolos de comunicação sendo o ModBus e CANopen os mais comuns.

2.4.1 Histórico

Os CLPs surgiram em 1968 desenvolvidos pela General Motors com o intuito de substituir enormes painéis eletromecânicos constituídos por relés e utilizados para o controle lógico de tais sistemas. Os painéis de relé possuíam a característica de difícil alteração em sua lógica de funcionamento necessitando de grandes quantidades de tempo para sua implementação e um elevado gasto monetário (ZANCAN, 2016).

Figura 5: Dave Emmett em 1968 implementando o primeiro CLP em uma linha da GM.



Fonte: (BRODZIK, 2014)

O desenvolvimento deste equipamento trouxe inúmeros benefícios, além da redução de gastos com os enormes painéis de relés, a redução na quantidade de tempo da manutenção nos CLPs reduzindo assim o tempo de ociosidade da máquina, uma programação simples onde a lógica de relés implementada anteriormente é facilmente transposta, fácil extensibilidade necessitando apenas da adição de módulos de acordo com a demanda, robusto e resistente a ambientes hostis e contém recursos lógicos completos (SEGOVIA; THEORIN, 2012).

A Allen-Bradley Company (Rockwell Automation) batizou o termo CLP, assim como “Controladores Lógicos Programáveis”, tornando-se marcas registradas da empresa (BOLTON, 2015).

2.4.2 Servomotores

Servomotores são equipamentos eletromagnéticos semelhantes a uma máquina elétrica convencional em seu estator e seu rotor é composto por ímãs permanentes operando junto a um encoder de forma a controlar suas variáveis por uma malha fechada (MORAES, 2011). Possuem uma ampla usabilidade em equipamentos que necessitam de precisão como manipuladores robóticos e fresas, sua utilização é baseada metodologicamente em um planejamento de caminho e trajetória, controle de rastreamento e controle de motor (WANG et al., 2014).

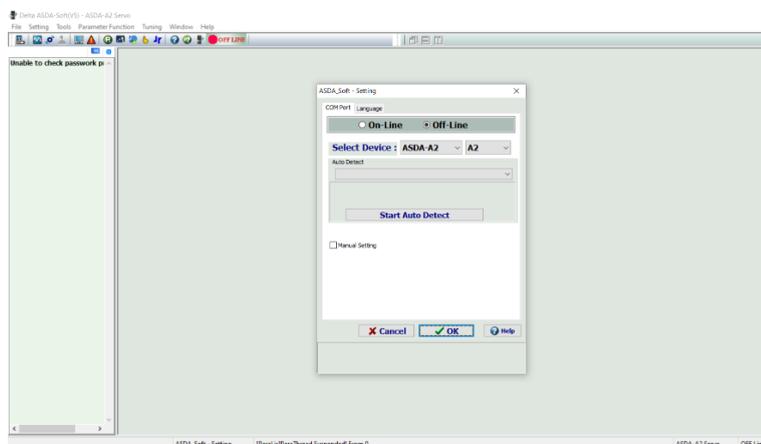
Figura 6: Servomotor e um servodriver da marca Delta.



Fonte: (DELTA INC, 2022)

Os servomotores da marca *Delta Electronics* são parametrizados manualmente ou por um programa chamado *ASDA Soft* onde pode-se salvar um arquivo com os parâmetros no qual se deve inserir nos servodrivers e posteriormente apenas realizar o *upload* para cada servo utilizado.

Figura 7: Tela inicial do ASDASoft.



Fonte: (DELTA INC, 2022)

2.4.3 Linguagem de Programação Ladder

A primeira e a linguagem de programação mais frequentemente utilizada em um CLP é a Linguagem Ladder (MARTINS et al., 2019). O Ladder contém instruções sofisticadas como temporizadores e contadores, mas sua base é simples com contatos e bobinas, o que foi determinante para a aceitação dos projetistas que já realizavam projetos baseados em esquemas de relés (GEORGINI, 2018).

A linguagem realiza um controle de eventos discretos, onde cada saída de linha de comando pode ter dois estados possíveis, "0" ou "1", a entrada das linhas seguem a mesma lógica, as entradas do sistema podem ser de natureza física (botões ou sensores) ou virtual (flags de servomotores ou um dispositivo externo escreve um valor em memórias binárias do tipo "M")(DU; XU; YAMAZAKI, 2010).

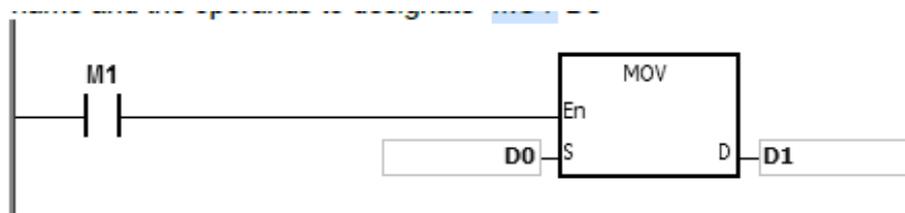
Tabela 1: Tipo de variável com estado binário.

Tipo da variável	Função
X	Relé de entrada
Y	Relé de saída
M	Relé auxiliar
SM	Auxiliar especial
S	relé de passo
T	Temporizador
C	Contador
HC	Contador de 32 bits
D	Registro de dados
SR	Registro de dados especiais
FR	Registro de arquivo
E	Registro de índice

Fonte: (DELTA, 2022)

Um dos comandos que utiliza o tipo de variável "D" é o *MOV*, o comando serve para escrever valores em um registrador de 16 bits ou de 32 bits, no exemplo abaixo escreve-se um valor armazenado em D0 no registrador de 16 bits D1.

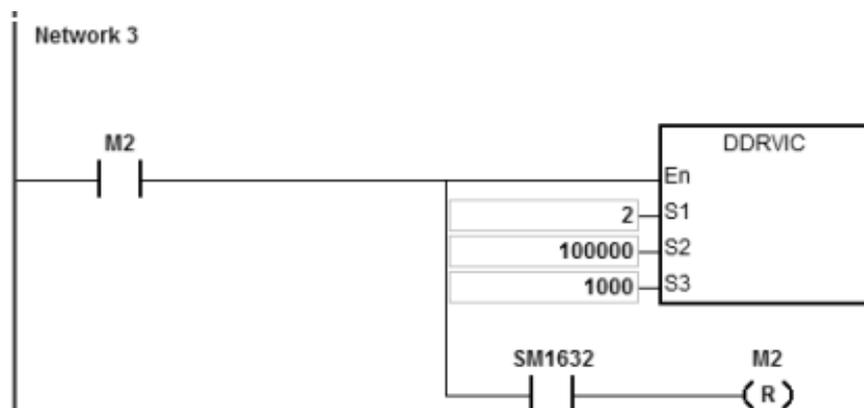
Figura 8: Comando MOV escrevendo no registrador D1.



Fonte: (DELTA, 2022)

Um dos principais comandos utilizados nos programas em Ladder do trabalho será o *DDRVIC*, ele trabalha no incremento de pulsos indicados no parâmetro S2 independente de sua posição atual, o parâmetro S1 indica o número do motor no qual se deseja atuar e o parâmetro S3 indica a velocidade do motor em RPM.

Figura 9: Tela inicial do software ISPSoft com um programa em Linguagem Ladder.



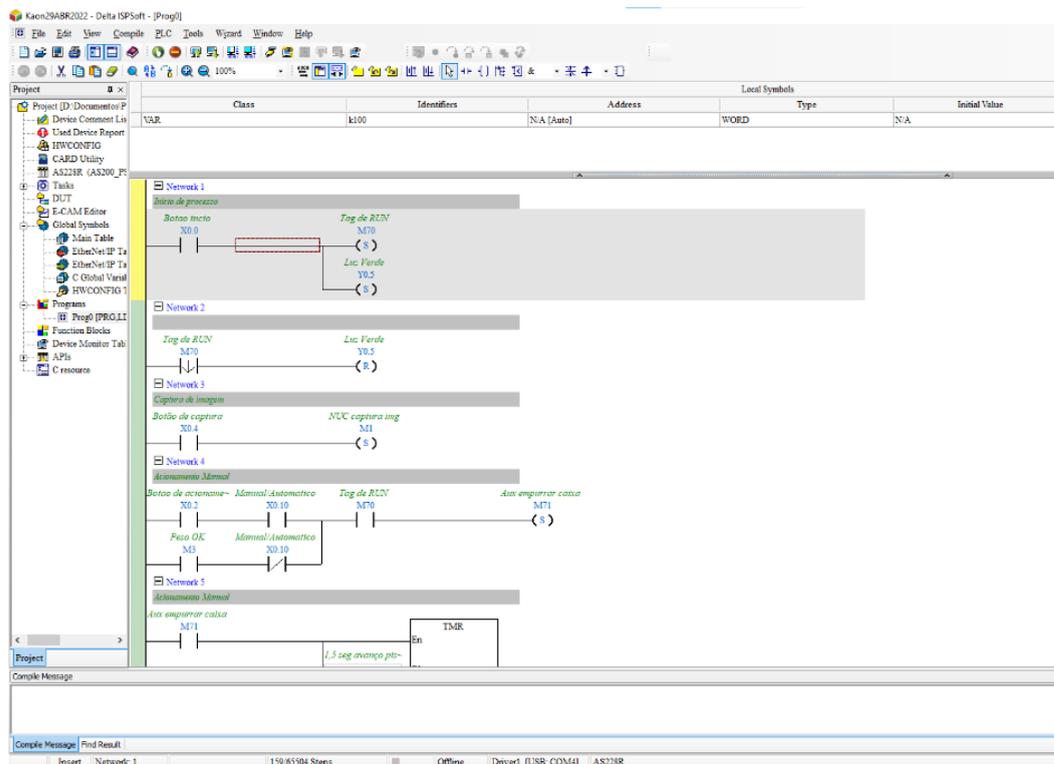
Fonte: (DELTA, 2022)

2.4.4 ISPSoft

Softwares de programação dos CLPs permitem editar e criar programas em Linguagem Ladder por meio de um ambiente gráfico, pode-se controlar todo o sistema por eles e supervisioná-los, os softwares são baseados na norma IEC 61131-3 (GEORGINI, 2018).

Segundo o site oficial da empresa desenvolvedora do software, o ISPSoft é a ferramenta utilizada para o desenvolvimento e edição de programas nos CLPs da marca Delta Electronics e segue o padrão IEC 61131-3 integrando diversas aplicações para desenvolvimento e gerenciamento de projetos (DELTA, 2022).

Figura 10: Tela inicial do software ISPSOft com um programa em Linguagem Ladder.



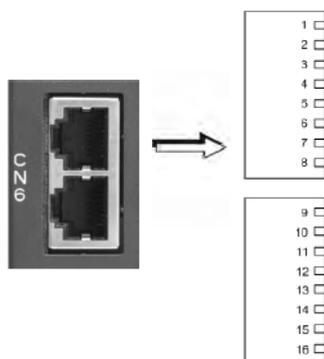
Fonte: Autor (2022)

2.4.5 CANOpen

O protocolo de comunicação CANOpen é baseado em CAN (Controller Area Network), possui transmissão serial e sua estrutura é baseada no modelo ISO/OSI. Foi desenvolvido por diversos grupos interessados em seu uso, tanto fabricantes quanto usuários, foram responsáveis por sua criação (FARSI; RATCLIFF; BARBOSA, 1999). Seu uso é focalizado relacionando a 1ª e 2ª camada de automação atuando como ponte em, por exemplo, controle e comando de um servomotor (CRAVO, 2022).

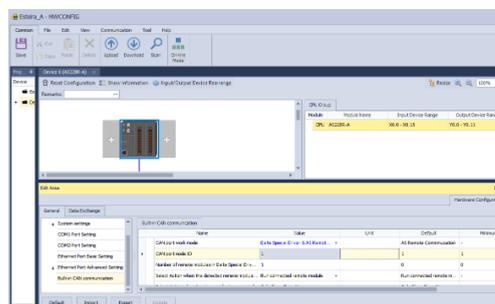
A comunicação do protocolo CANOpen nos servomotores *Delta Electronics* é transmitida por cabos RJ45 inserida na conexão CN6 do servomotor, a parametrização do protocolo CANOpen é realizada na extensão do *ISPSOft* o *HWCONFIG*.

Figura 11: Entrada e pinagem da conexão física do protocolo CANOpen.



Fonte: Autor

Figura 12: Configuração do CANOpen pelo HWCONFIG.



Fonte: Autor

2.4.6 Fonte de alimentação 24V

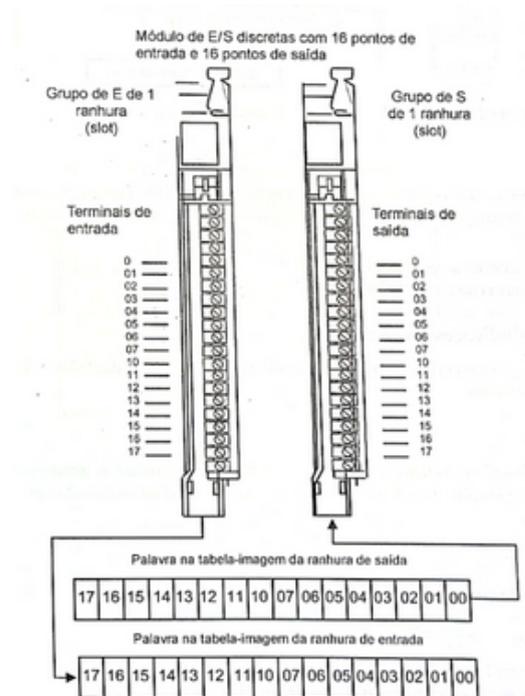
Os CLPs geralmente possuem uma tensão de alimentação de 24Vdc, logo, é necessária uma fonte que transforme 220Vac ou 127Vac na tensão desejada. Tais fontes precisam ser robustas, confiáveis e imunes a níveis elevados de ruídos.

Os sistemas de comando possuem acionamento de 24Vdc normalizado visando a segurança do trabalhador quanto aos riscos elétricos (SILVEIRA, 2018).

2.4.7 Entradas e saídas do CLP

As entradas de um CLP são responsáveis por identificar um sinal advindo de um dispositivo externo inserido no circuito, estes dispositivos são por exemplo chaves, seletoras e sensores.

Figura 13: Relação de entradas de saídas físicas de um CLP.

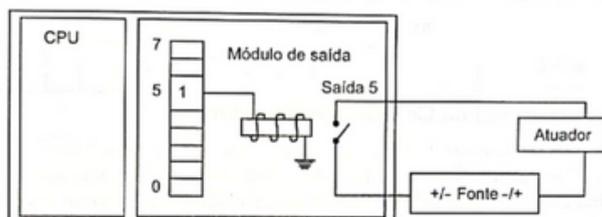


Fonte: (MORAES; CASTRUCCI, 2013)

As saídas do CLP são responsáveis pela partida de motores, acionamento de solenóides, lâmpadas, LEDs ou buzzers.

Uma das tecnologias dos módulos de saídas de um CLP é a saída a relé, onde, quando um solenóide interno é acionado (nível lógico do bit se torna 1), fecha-se o contato no borne de saída do controlador (MORAES; CASTRUCCI, 2013).

Figura 14: Diagrama de um CLP com saídas a relé



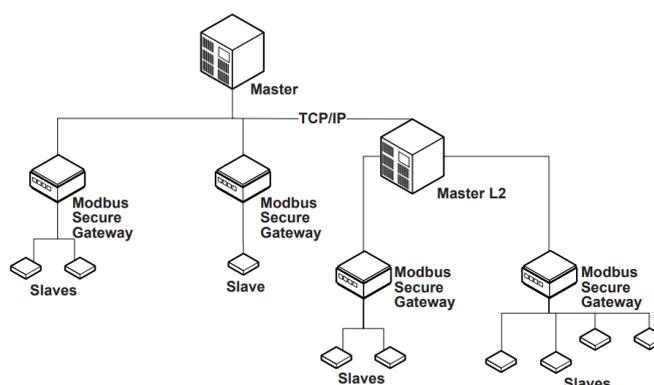
Fonte: (MORAES; CASTRUCCI, 2013)

2.4.8 Modbus

Modbus é um protocolo da camada de aplicação baseado na troca de mensagens cliente/servidor entre dispositivos conectados à rede. Essa comunicação pode ocorrer de forma direta com uma máquina principal e sua réplica ou por meio de *broadcast*, onde a *master* envia informações para as demais (FOVINO et al., 2009).

Cada mensagem permutada via Modbus contém o endereço do dispositivo que irá recebê-la, o comando a ser efetuado e o dado que será utilizado na execução. O Modbus TCP implementa o protocolo TCP em cada transação do Modbus (FOVINO et al., 2009).

Figura 15: Arquitetura do funcionamento do protocolo Modbus.



Fonte: (FOVINO et al., 2009)

2.5 MOST - Maynard Operation Sequence Technique

MOST (*Maynard Operation Sequence Technique*) é uma técnica para medir a duração de atividades em um processo fabril. Esta metodologia surgiu na década de 1970 e foi desenvolvida por H.B Maynard, sendo que um dos principais objetivos é padronizar o tempo de produção, bem como maximizar recursos afim de melhorar a fabricação de itens. É interessante notar que o mesmo tem sido utilizado com efetividade no setor industrial (TUAN et al., 2014).

Existem 3 subcategorias de MOST, estas são:

- Most básico: Aplicação generalizada e não dependente do tempo padrão de produção;

- Mini MOST: Aplicação realizada em ciclos menores;
- Maxi MOST: Aplicação realizada em processos de longa duração.

A metodologia se baseia mapear os movimentos sequenciais de um processo, acredita-se que todas as tarefas são realizadas com a combinação 19 movimentos. Portanto, pode-se, utilizando MOST pode-se calcular a atuação de tempo em valor agregado dos operadores, identificando assim o tempo de ciclo do processo (PUVA-NASVARAN; YAP; YOONG, 2019). O método utiliza o valor de tempo em TMU (Time Measurement Unit), a tabela de auxílio de conversão é mostrada a seguir.

Tabela 2: Tabela de conversão entre TMU e tempo padrão.

1 TMU = 0.00001 hora	1 hora = 100,000 TMU
1 TMU = 0.0006 minuto	1 minuto = 1667 TMU
1 TMU = 0.036 segundo	1 segundo = 27.8 TMU

Fonte: (NAFAL, 2013)

2.6 Poka Yoke

A técnica Poka-Yoke desenvolvida pelo engenheiro japonês Shigeo Shingo com o intuito de alcançar zero defeitos em um processo produtivo e eliminar inspeções de qualidade, a nomenclatura posteriormente definida oficialmente se chamaria *Poka-Yoke*, que significa 'à prova de erros' ((SHIMBUN, 1989)).

Os Poka-Yoke podem ser aplicados tanto para prevenir causas, que consequentemente resultarão em erros e determinar por métodos rápidos e simples a aprovação ou reprovação de um produto (DUDEK-BURLIKOWSKA; SZEWIECZEK, 2009).

2.7 Takt-Time

A palavra 'takt' é de origem alemã e serve para definir um compasso musical e tem sido utilizada no Japão desde anos 30 para definir o ritmo de produção (SHOOK, 1997).

O takt-time é o parâmetro temporal utilizado na produção necessário para atender a demanda. É resultado da razão entre o tempo disponível para a produção e o número

de unidades a serem produzidas. Outra definição pode ser a seguinte: "takt-time é o ritmo de produção necessário para atender a um determinado nível considerado de demanda, dadas as restrições de capacidade da linha ou célula."(ALVAREZ; JR, 2001). Concretamente, o takt-time é a diferença de tempo entre o primeiro produto acabado e o segundo produto acabado, este tempo deve ser menor ou igual ao atendimento de demanda em tempo hábil.

2.8 Setup de máquina

Um setup ocorre quando há uma troca de produção para outra, não necessariamente de um produto para outro. Um lote diferente de um mesmo produto já será um setup para troca do número do lote (FILIPE, 2022). Portanto, setup de máquina são processos onde há uma necessidade de ajuste da máquina para um novo processamento de manufatura. A diminuição do tempo de setup de máquina resulta em uma produção mais enxuta e econômica (FOGLIATTO; FAGUNDES, 2003). Tal requisito pode ser alcançado por soluções organizacionais de baixo custo ou projetos com valor agregado mais alto (REIS; ALVES, 2010).

2.9 Impressora laser e gravação

Impressoras a laser possuem duas naturezas de geração do laser, uma proveniente de um tubo de CO₂ com valor mais baixo e menor velocidade de gravação e o laser a fibra com intensidades de laser até 100 vezes mais forte, logo, realiza gravações mais rápidas, tais impressoras atendem a linhas de produção de até 900m/mim. A tecnologia funcionam num processo de cortar a tinta em uma superfície para realizar marcas permanentes no produto (VIDEOJET, 2022).

2.10 Leitor de código de barras

O leitor de código de barras é um dispositivo capaz de decodificar um conjunto de informações por meio de um feixe de luz, transformando-os em uma sequência binária capaz de ser compreendida por um computador. Esse dado trata-se de uma imagem contendo barras claras e escuras, espaçadas e em sequência.

Figura 16: Exemplo de um leitor de código de barras



Fonte: (INDUSTRIAL, 2022)

2.11 Esteira Industrial

Um esteira industrial trata-se de um componente que realiza o transporte dentro dos sistemas internos de manufatura. Este é utilizado com frequência para o manuseio de matéria-prima, bem como de produtos durante o processo de fabricação. Sua composição física é bastante variada, podendo ser plástico ou metal. Entretanto, a escolha desse tipo material é baseada na finalidade do transporte.

Existem diversas vantagens no uso de esteiras industriais durante o processo de fabricação, entre essas estão:

- **Flexibilidade:** Aplicação em inúmeros seguimentos de manufatura;
- **Segurança:** Seu material é durável e suportam uma quantidade grande de carga;
- **Rapidez:** O transporte dos produtos é realizado de forma ágil e dinâmica.

2.12 Robô cartesiano

Um robô industrial cartesiano realiza movimentos em eixos lineares, sendo fáceis de manipular e configurar. Estes são utilizados principalmente em tarefas repetitivas como descarga de material. Uma das vantagens é a modularidade, de forma que é possível a construção de robôs com dimensões significativas, para que consigam lidar com todas as máquinas nas linhas de produção.

Figura 17: Exemplo de um robô cartesiano da marca WITTMANN BATTENFELD.



Fonte: (INDUSTRY, 2022)

2.13 Atuadores lineares

Os atuadores lineares realizam a conversão dos movimentos de rotação das engrenagens para um movimento linear de forma a alcançar um resultado preciso no posicionamento de cargas. A utilização desse tipo de equipamento, em sua maioria, é por meio de aplicações em máquinas agrícolas, dispositivos médicos e entre outros.

Figura 18: Atuadores lineares pneumáticos.

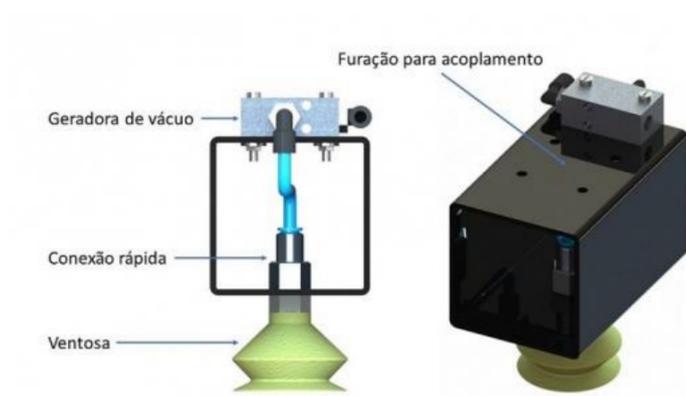


Fonte: autor.

2.14 Ventosas pneumáticas

As ventosas pneumáticas, ou industriais, possuem diversos modelos em sua fabricação, porém, em sua grande maioria, são redondas e de material flexível. Estas, ao serem pressionadas contra o objeto a ser manuseado, criam uma ligação à vácuo e conseguem realizar o transporte do produto.

Figura 19: Vetosa pneumática em croqui.



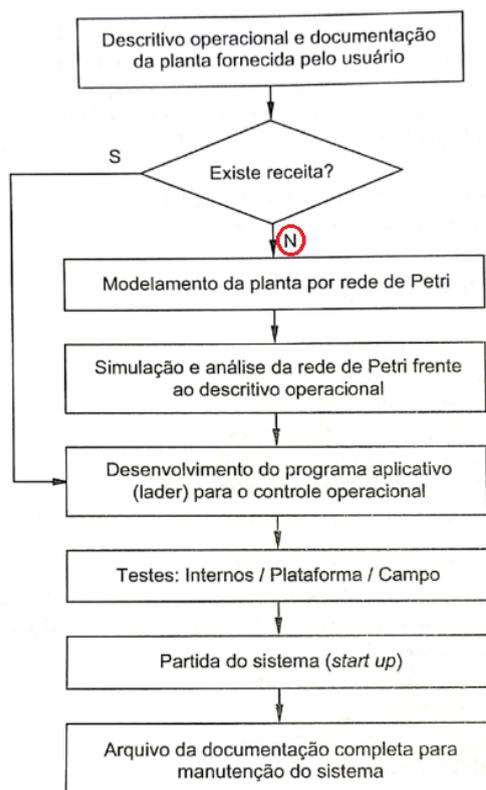
Fonte: (FIEGENBAUM et al., 2016)

3 MÉTODOS E MATERIAIS

Neste capítulo, serão abordados todos os procedimentos para realização do trabalho, assim como os materiais utilizados.

Moraes e Castrucci (2013) descreve passos para construção de um projeto de automação em um diagrama de blocos simples.

Figura 20: Construção do projeto de automação



Fonte: (MORAES; CASTRUCCI, 2013)

Para isto, o capítulo será dividido em seções da seguinte forma:

1. **Levantamento de Requisitos:** aborda os requisitos necessários para o bom fun-

cionamento baseado nas necessidades do usuário e nos requisitos da proposta.

2. **Fluxograma inicial:** são definidos macro processo básicos para o funcionamento do sistema que servirá de base para uma análise mais complexa posteriormente.
3. **Modelagem em Redes de Petri:** são construídos fluxos de automação de funcionamento do sistema modelado utilizando Redes de Petri.
4. **Cronoanálise utilizando MOST:** serão estudados os tempos de ciclo dos postos de trabalho do sistema, identificando o gargalo do processo e validando seu takt-time.
5. **Definição de Equipamentos e arquitetura física do sistema:** aborda os materiais necessários para a realização dos processos e define como os componentes serão dispostos quanto a suas ligações elétricas de comando.
6. **Programação do sistema:** apresenta a programação do sistema utilizando linguagem Ladder embarcada no CLP.

3.1 Levantamento de Requisitos

A primeira etapa do processo foi o levantamento de requisitos do sistema, estes foram descritos como:

- O sistema de esteiras deve ser baseado nas esteiras inseridas atualmente no processo produtivo de manufatura do produto no qual este trabalho se baseia.
- Ser capaz de abranger, em esteiras de correias (o atrito nas placas deve ocorrer apenas em suas extremidades extinguindo qualquer possibilidade de dano em seus sensíveis terminais), placas mãe de diferentes larguras.
- Deve ser modular, de forma onde hajam pelo menos dois módulos, onde o primeiro seja responsável pela impressão dos códigos de barras e o segundo responsável pela leitura dos códigos de barras.
- Permitir a fácil troca de informações com sistemas externos.

- Deverá ser de fácil utilização pelo colaborador, de modo que o mesmo deverá apenas inserir um pacote contendo 50 placas de um mesmo modelo e em seguida apertar o botão de início da operação.
- O sistema deverá ser enclausurado fisicamente conforme regulamentado em norma.
- (ANDERSEN; NIELSEN; BRUNOE, 2016) apresenta uma revisão sobre os pré-requisitos de SMR, onde os principais pontos são a convertibilidade e escalabilidade do sistema, portanto, deve-se garantir, a partir de tais requisitos, um longo ciclo de vida do sistema.
- (MORGAN et al., 2021) projeta que a próxima geração de sistemas de automação serão *Smart Reconfigurables (SR)*, portanto, tais sistemas possuirão o requisito de reconfigurabilidade inteligente.

3.2 Fluxograma inicial

De acordo com os itens citados, será possível construir uma esteira que inicialmente abrigue um pacote que contém 50 placas-mãe virgens, uma esteira responsável pelo processo de impressão das placas unitariamente e a última esteira responsável pela leitura e pela retirada da PCB. Portanto, o sistema será dividido em 3 módulos, um módulo inicial

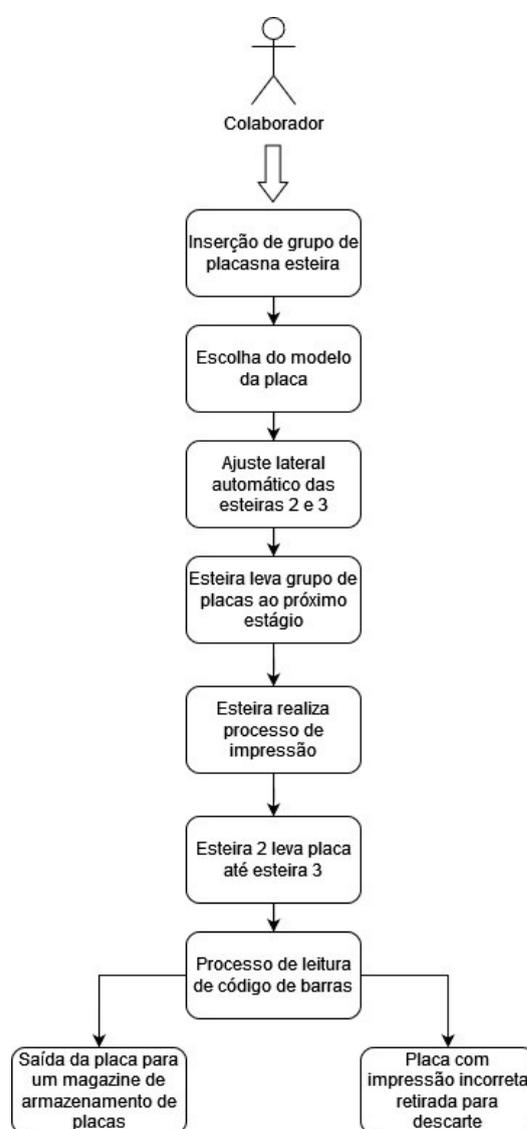
Logo, será necessário projetar pelo menos duas esteiras com ajustes laterais automáticos, pois, o sistema de esteiras precisa garantir uma precisão de $\pm 1mm$ em quaisquer orientações pro conta de duas condições, a primeira é a gravação dos códigos em locais pré-determinados sinalizados na PCB com tinta branca, a segunda é a inserção das placas em magazines de armazenamento onde os slots possuem a dimensão ideal para determinada PCB com erro de $\pm 1mm$ para cada lado:

- **Módulo da esteira de impressão:** Será responsável por levar todos os tipos de PCBs a um local exato para gravação dos códigos de barras.
- **Módulo da esteira de saída:** Será responsável por levar as PCBs com código de barras gravados para uma validação por meio de um leitor de código de barras.

Já o módulo inicial será responsável pela única interação com um colaborador onde o tal deve inserir uma quantidade máxima de 50 PCBs de um único modelo em uma bandeja, apertará um botão para validação da PCB e em seguida o módulo envia o pacote de PCBs ao segundo estágio do processo, cada PCB possui no máximo 200 gramas, logo, estima-se que o módulo inicial carregará um peso de aproximadamente 10 kg levando em consideração apenas o peso das placas.

Baseado nos requisitos apresentados, projetou-se o seguinte fluxograma do sistema:

Figura 21: Diagrama da arquitetura do sistema de esteiras



Fonte: Autor

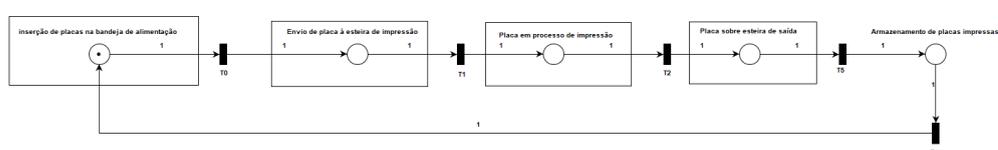
De acordo com o fluxograma construído a partir dos requisitos do sistema, pode-

mos realizar a modelagem de automação do sistema utilizando as Redes de Petri.

3.3 Modelagem em Redes de Petri

O primeiro passo é construir um macro modelo do sistema de esteiras, conforme a figura 22, a modelagem completa e com maior qualidade visual pode ser vista no apêndice A.

Figura 22: Modelo do sistema em Redes de Petri dividido em macrolugares

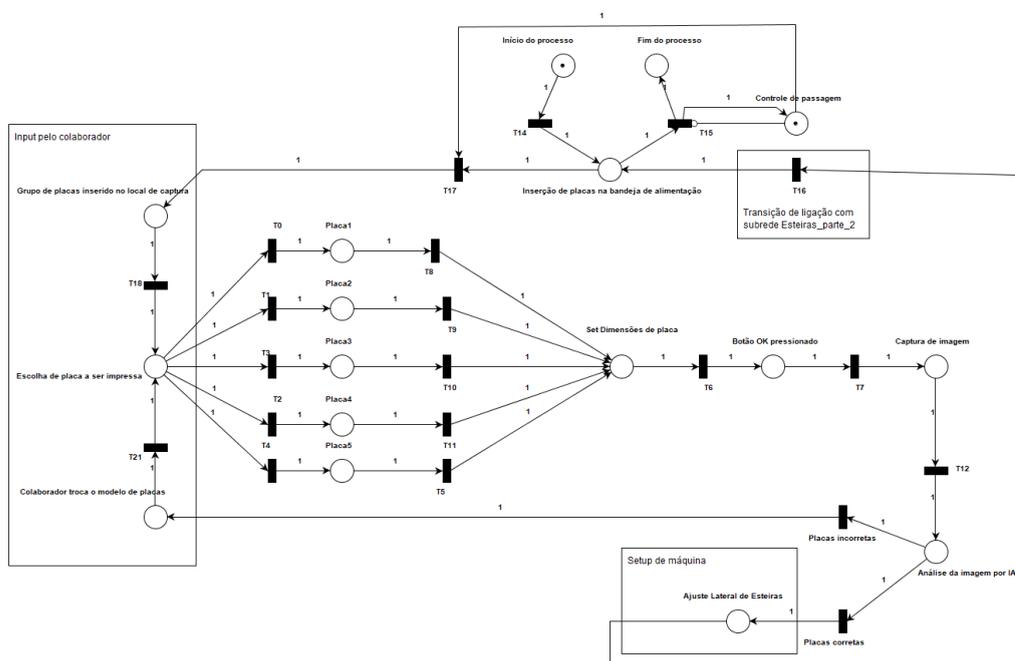


Fonte: Autor

Serão detalhados a seguir, as hierarquias do sistema a fim de identificar os micro-processos no qual o sistema automático deverá realizar.

Extratificando os processos realizados com a interação de um colaborador identificado no macrolugar "Inserção de placas na bandeja de alimentação", temos então o seguinte modelo:

Figura 23: Detalhamento dos processos do primeiro estágio

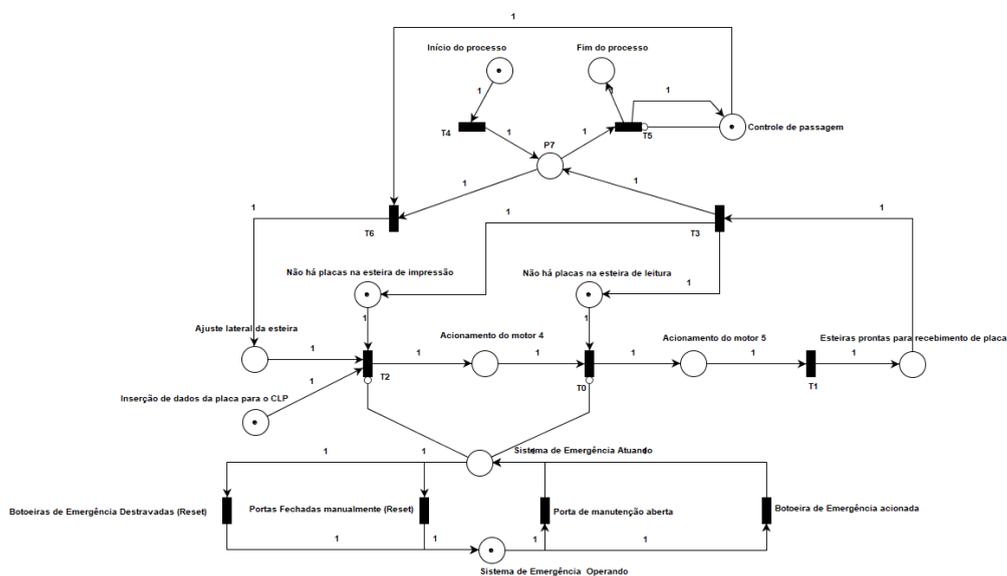


Fonte: Autor

O processo se inicia com a inserção de uma quantidade de PCBs por um colaborador em uma bandeja que abriga PCBs empilhadas com uma quantidade limitada a 50 por vez. Após o posicionamento o colaborador dará início ao processo automático por meio de um botão, uma IA (Inteligência Artificial) realiza a comparação entre o banco de dados interno. Nesta parte será introduzida ao sistema a informação de qual modelo de PCB será inserido no sistema para conseguinte gravação dos códigos. Tal processo é de extrema importância ao trabalho descrito, pois é exatamente onde se inicia a caracterização do sistema como *reconfigurável*, a partir de qual modelo de PCB for escolhido, o CLP recebe a informação do tamanho da PCB de um dispositivo externo via protocolo de comunicação e calcula a abertura ideal para manipulação do produto durante o processo de gravação, portanto, as placas mostradas na figura 23 possuem tamanhos variados, além disso, novos modelos de placas podem ser adicionados ao sistema.

Detalharemos agora uma sub-hierarquia da hierarquia *Inserção de placas na bandeja de alimentação* chamada *Setup de máquina*. O fluxo em Redes de Petri mostra o que ocorrerá para realizar o movimento de ajuste lateral.

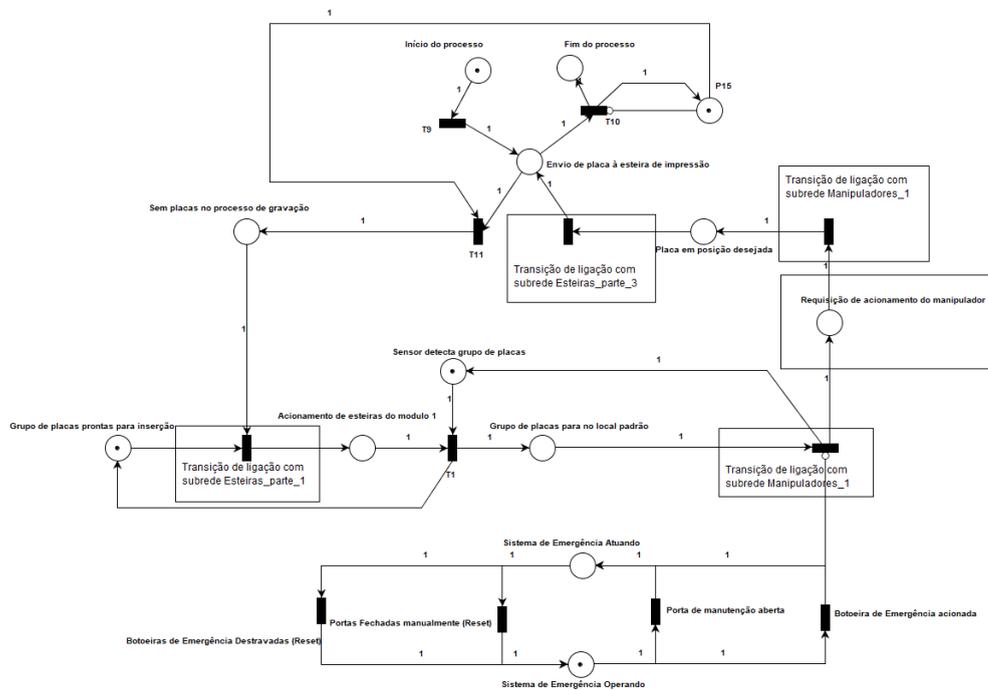
Figura 24: Detalhamento da sub-hierarquia "Setup de máquina".



Fonte: Autor

Para inserção das placas de forma individual e automática na esteira de impressão será necessário um outro mecanismo, portanto, será projetado um manipulador capaz de realizar este trabalho, seus movimentos automáticos são modelados pelas Redes de Petri mostradas a seguir.

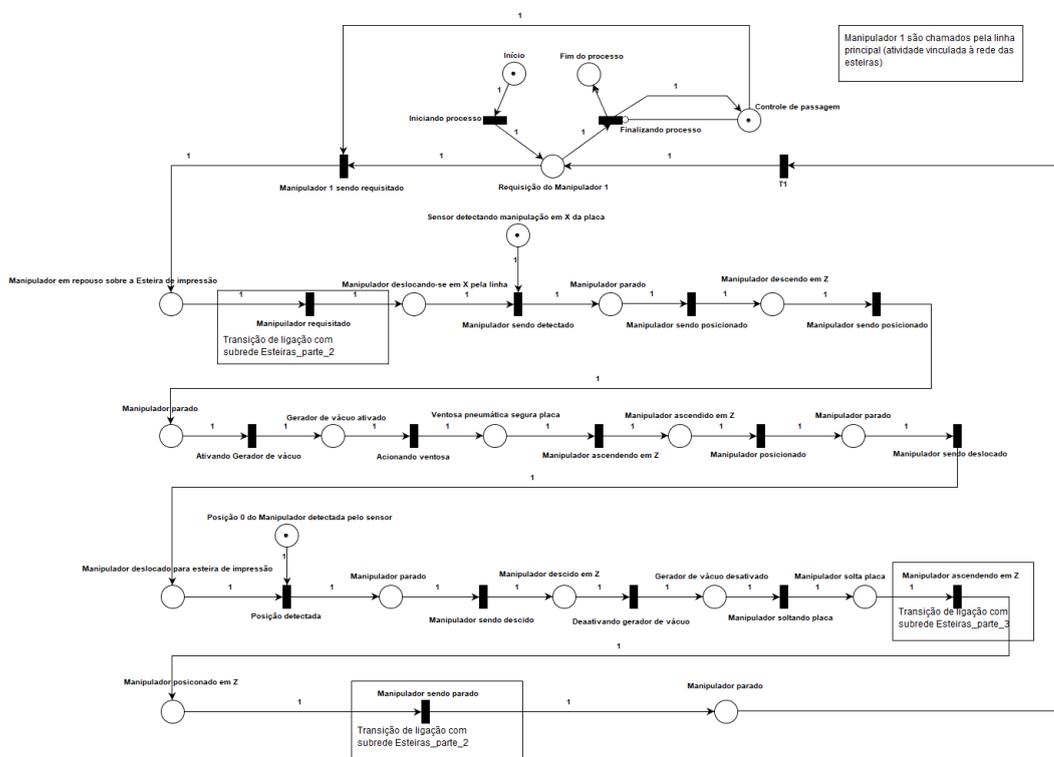
Figura 25: Modelagem dos sistema do processo de inserção das PCBs sobre a esteira de forma individual para posterior impressão dos códigos.



Fonte: Autor

O manipulador será suprimido em uma subhierarquia de modo que seus movimentos para capturar a PCB serão um macrolugar chamado "inserção de PCB unitária por manipulador". Seus micromovimentos posteriormente utilizados para cronoanálise e programação do servomotor são modelados a seguir.

Figura 26: Modelagem dos sistema do manipulador robótico responsável por inserir as PCBs uma por vez.

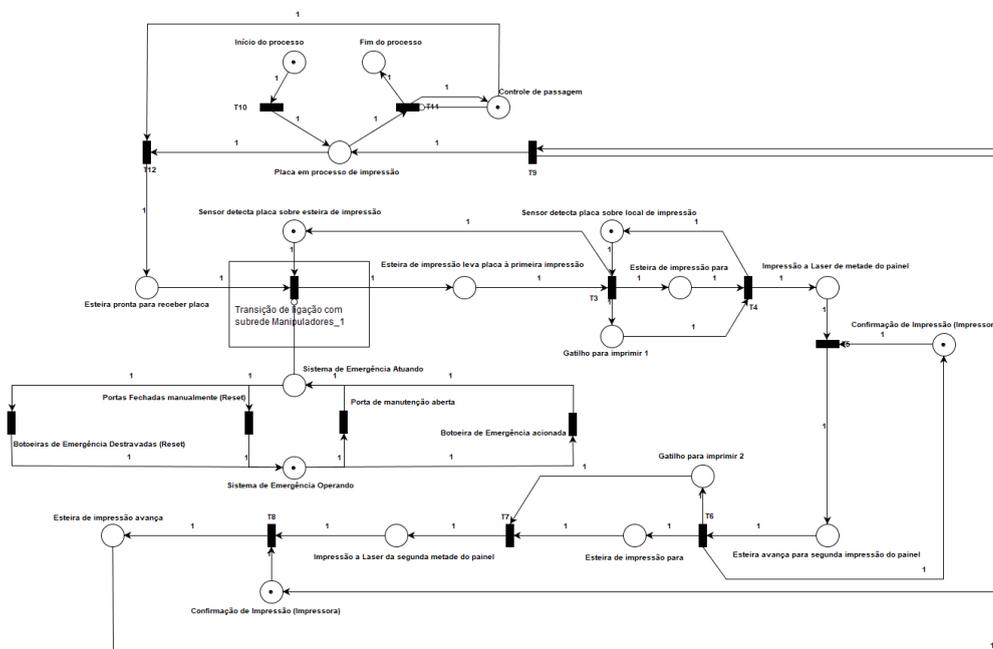


Fonte: Autor

Estima-se que, para movimentos lineares em duas dimensões, um manipulador cartesiano seja capaz de realizar o trabalho de capturar uma PCB de um local e inserí-la de forma precisa em um plano com altura variável. O mecanismo de captura da PCB necessita garantir a integridade da PCB, um mecanismo simples que garante tal requisito são as ventosas, baratas e eficientes, conseguem utilizar o vácuo para carregar o produto com segurança.

Após a inserção da PCB na esteira, inicia-se o processo de impressão de códigos de barras, modelado a seguir:

Figura 27: Modelagem dos sistema da esteira de impressão dos códigos de barras nas PCBs.

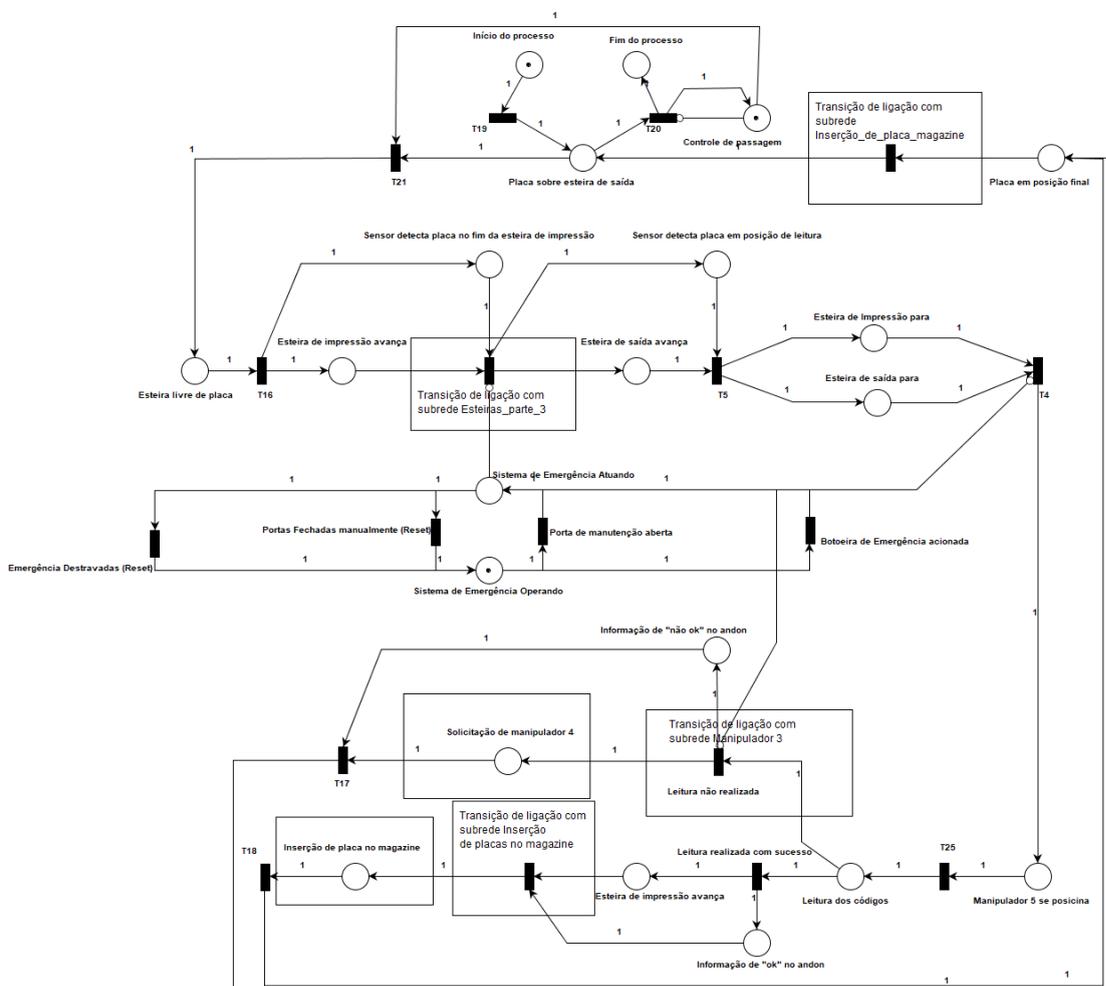


Fonte: Autor

A esteira deverá iniciar seu ciclo no momento em que percebe uma PCB sobre o início do módulo de impressão. Para isso, temos a inserção de um sensor para realizar tal processo, a PCB será levada ao local de impressão de códigos de barras onde haverá um sensor indicando seu local de parada. Uma posição de memória virtual deverá ser alocada no CLP de forma que um dispositivo externo consiga monitorar e perceber o momento correto para iniciar a primeira impressão, após isso, ocorrerá o processo de impressão, em seguida o dispositivo externo enviará a informação de que a impressão foi concluída para o CLP, que por sua vez encaminha a PCB para uma segunda impressão caso houver, o ciclo de impressão dos códigos é repetido e a esteira carregará a PCB ao estágio seguinte.

No estágio a seguir, será realizada uma inspeção nos códigos de barras impressos, modelando o ciclo de automação da inspeção temos:

Figura 28: Modelagem dos sistema de inspeção dos códigos de barras impressos.

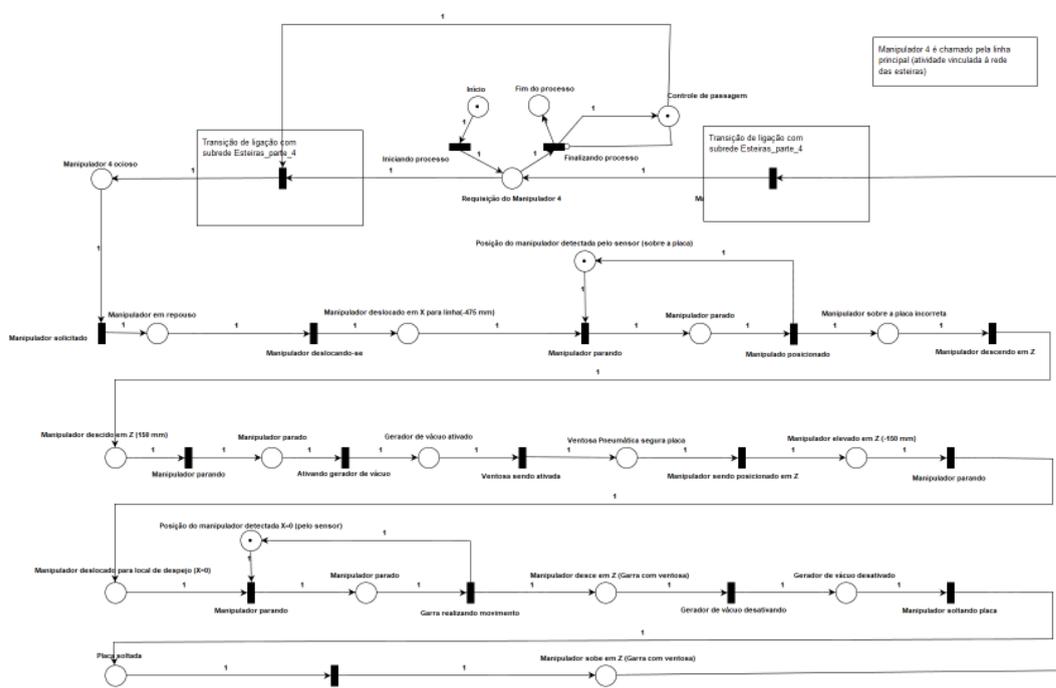


Fonte: Autor

O equipamento responsável pela inspeção dos códigos de barras deve ser o mesmo utilizado para leitura dos códigos durante ou depois do processo, por tal motivo, deverá ser utilizado um leitor de códigos de barras eletrônico padrão do modelo Zebra DS2208 para tal.

Para o caso de, no estágio de inspeção, um código for reprovado, um mecanismo deverá retirar as placas com códigos rejeitados sem que haja interferência no processo. O processo para realizar o trabalho é semelhante ao processo do manipulador responsável por inserir as PCBs uma por vez na esteira, portanto, temos a seguinte subrede modelada:

Figura 29: Modelagem em Rede de petri do manipulador responsável pela retirada das placas reprovadas após a inspeção.

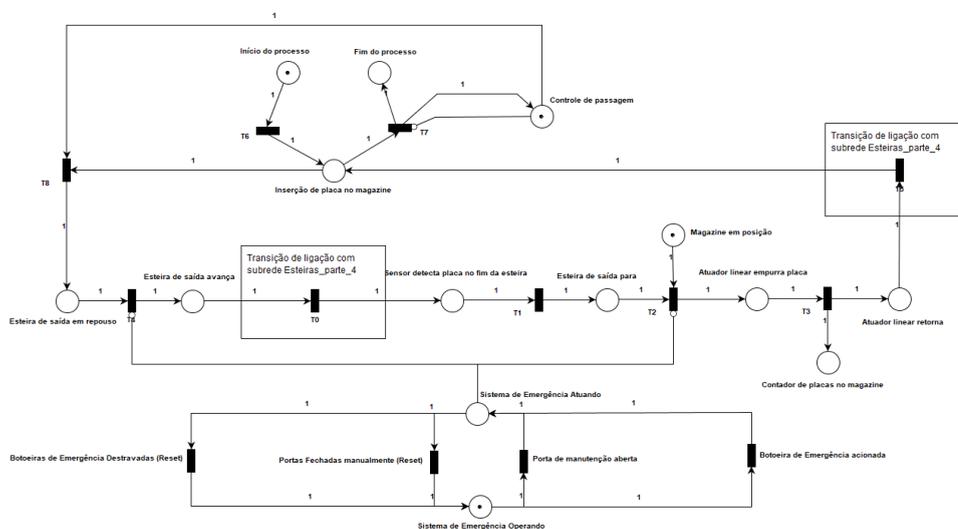


Fonte: Autor

Por se tratar de material de refugo, não há necessidade de um tratamento especial com as PCBs que forem reprovadas na inspeção, com base nesta informação poderá se utilizar de um atuador com apenas duas posições para a orientação vertical, visto que, poderá buscar a PCB em uma certa altura e deixá-la na mesma altura, apenas em uma posição horizontal diferente. Portanto, pode-se reduzir o custo inserindo um atuador linear com natureza de acionamento pneumático.

Caso os códigos sejam aprovados pela inspeção, a PCB segue ao estágio final, onde chegará ao final da esteira e poderá avançar a um estágio seguinte, no caso específico, as PCBs irão para o magazine que será posicionado por um sistema externo, portanto, teremos a continuação do processo modelado da seguinte forma:

Figura 30: Modelagem em Rede de petri do estágio final do processo de impressão de PCBs.

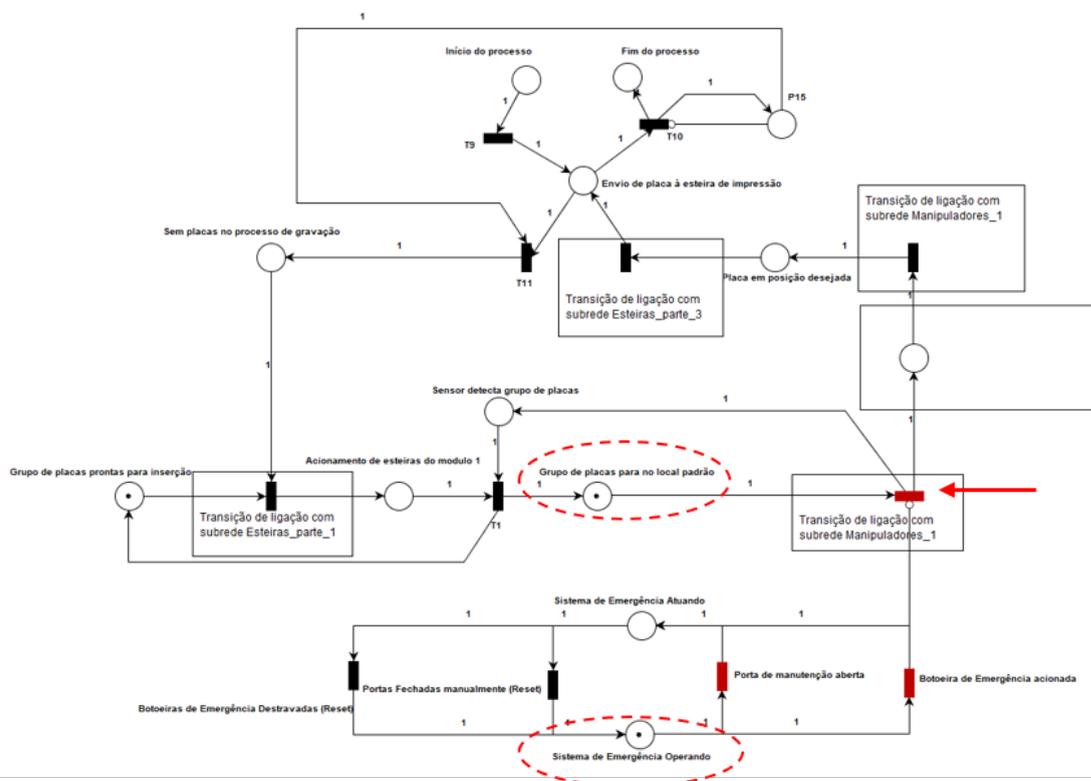


Fonte: Autor

3.3.1 Sistema de segurança e emergência

Conforme NR12, o subitem 12.5.11 estabelece requisitos de segurança ao colaborador em relação a movimentos que possam causar dano de esmagamento, tendo em vista tal requisito, estruturou-se uma estrutura comum na maioria dos subsistemas descritos no qual se trata do sistema de emergência do processo, onde estará atuando e realizando a parada dos movimentos do equipamento em caso de o botão de emergência for pressionado ou caso uma das portas de acesso interno esteja aberta. Na modelagem em redes de petri tal estrutura de controle de passagem de fluxo foi atrelada ao fluxo principal na forma de um inibidor que impede a passagem dos tokens caso esteja acionado, portanto, de forma a exemplificar seu funcionamento temos os seguintes casos utilizando o modelo da figura 28 como referência:

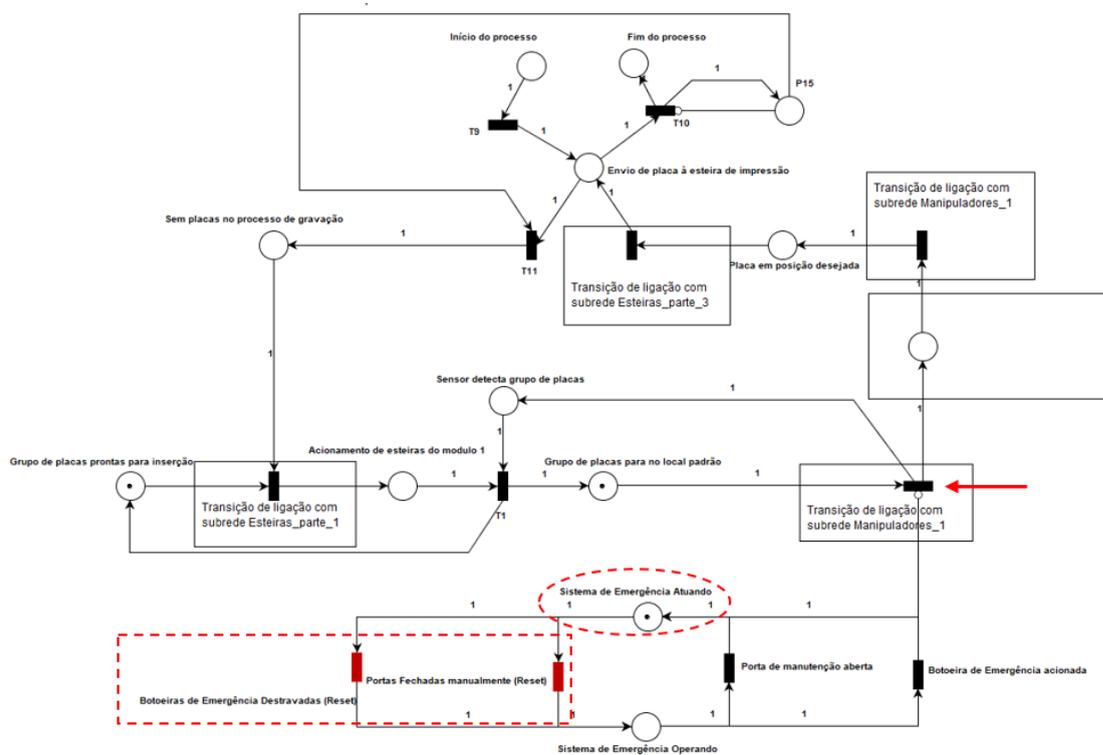
Figura 31: Modelagem em Rede de petri do sistema de requisição do manipulador de inserção de placas individuais em simulação.



Fonte: Autor

Em destaque na figura anterior, temos os dois lugares que podem alternar de estágio em tal momento, o fluxo natural do sistema mais acima está habilitado e ocorrerá normalmente nestas condições, porém, caso ocorra o acionamento do botão de emergência ou uma porta se abra, o sistema de emergência atuará e o movimento será cessado. No caso a seguir, o manipulador robótico não será acionado até que ocorra um reset e a máquina volte ao seu movimento:

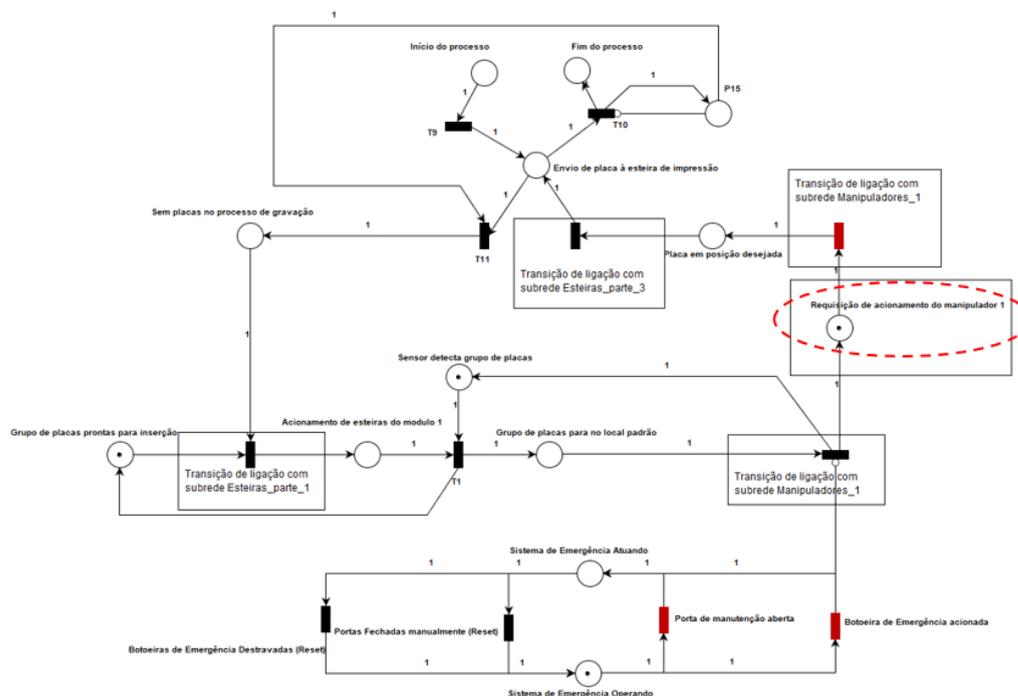
Figura 32: Modelagem em Rede de petri do sistema de inspeção dos códigos de barras impressos em simulação com sistema de segurança atuando.



Fonte: Autor

Portanto, a estrutura utilizada refletirá diretamente no funcionamento dos estágios do processo garantindo que nenhuma estrutura móvel esteja em movimento durante um possível contato com ser humano. Realizando um comando de reset é possível fazer com que o processo retorne normalmente ao seu ciclo de trabalho conforme mostrado na figura a seguir:

Figura 33: Modelagem em Rede de petri do sistema de inspeção dos códigos de barras impressos em simulação com sistema de segurança resetado.



Fonte: Autor

3.4 Cronoanálise utilizando MOST

Tendo modelado o fluxo de automação, pode-se realizar a cronoanálise para identificar os valores de tempo estimados para cada operação e então identificar os gargalos do sistema verificando sua viabilidade e produtividade. O método é adequado visto que simplifica os movimentos sob análise do trabalho realizado em relação a um objeto (NAFAL, 2013). Não é possível comparar o tempo de fluxo do processo modelado com algum fluxo semelhante manual, por esse motivo, foi utilizado o método como validação do tempo máximo por operação, garantindo que um processo não ultrapasse o *takt-time*, tido como o requisito base para realização do projeto.

Além disso, a técnica MOST se aplica na cronoanálise do trabalho manual, portanto, será realizada uma análise dos movimentos robóticos com uma analogia aos movimentos humanos identificando assim o tempo máximo permitido para cada operação. O estudo realizado por (KARIM, 2014) exemplifica a estruturação de uma cronoanálise utilizando essa metodologia. Assim sendo, foi aplicado o MOST ao sistema proposto, obtendo os resultados descritos a seguir.

Para o modelo em redes de petri modelado na 23 têm-se então os seguintes tempos:

Tabela 3: Sequência de movimentos do processo da figura 23 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Operador aperta botão	M1	0,29
IA Lê pilha de placas	T10	2,9
IA aprova resultado e registra	R3	0,87
Ajuste lateral da Esteira de Impressão	I16	4,64
Ajuste lateral da Esteira de Inspeção	I16	4,64

Fonte: Autor

Portanto, conforme o estudo realizado, o *setup* de ajuste lateral realizado para manufatura de todos os modelos é o valor mais alto do primeiro estágio com tempo máximo estimado em 4,64 segundos.

Tabela 4: Sequência de movimentos do processo da figura 25 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Esteira leva Bandeja ao local de inserção	P6	1,74
Solicitação de uso do Manipulador 1	R1	0,29

Fonte: Autor

O tempo estimado para posicionamento da pilha de placas e solicitação do manipulador é de 2,03 segundos.

Tabela 5: Sequência de movimentos do processo da figura 26 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Manipulador se desloca em +X	M3	0,87
Sensor detecta manipulador sobre placa	T1	0,29
Manipulador desce em Z (variável)	P3	0,87
Ventosa Pneumática segura placa	M1	0,29
Manipulador sobe em Z (variável)	P3	0,87
Manipulador se desloca em -X	M3	0,87
Manipulador desce em Z (variável)	P3	0,87
Manipulador solta placa	M1	0,29
Manipulador sobe em Z (variável)	M3	0,87

Fonte: Autor

O tempo estimado para realização do processo de retirada de uma PCB da pilha e inserção na esteira responsável pela impressão dos códigos é de 6,09 segundos.

Tabela 6: Sequência de movimentos do processo da figura 27 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Esteira posiciona a placa p/impressão	M6	1,74
Impressora imprime placa	R42	12,18
Esteira Avança ao próximo estágio	M6	1,74

Fonte: Autor

Conforme análise acima, a impressão dos códigos demorará em até 12,18 segundos.

Tabela 7: Sequência de movimentos do processo da figura 28 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Esteira posiciona a placa leitura	M6	1,74
Leitor de códigos inspeciona a placa	T10	2,9
Impressão aprovada ou reprovada	A1	0,29
Esteira Avança	M6	1,74

Fonte: Autor

Conforme analisado na tabela 7, o maior valor de tempo estimado para uma operação neste estágio será a inspeção dos códigos impressos com 2,9 segundos, caso os códigos forem impressos de forma que na leitura, sejam aprovados, um atuador linear deve apenas empurrar a PCB para um dispositivo de armazenamento externo, de forma que o estado seja executado da seguinte forma:

Tabela 8: Sequência de movimentos do processo da figura 30 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Atuador Avança	M3	0,87
Atuador Recua	M3	0,87

Fonte: Autor

Caso os códigos não sejam lidos ou identificados corretamente o processo baixo irá ocorrer.

Tabela 9: Sequência de movimentos do processo da figura 29 utilizando método de cronoanálise MOST

Estado	MOST(TMU)	Tempo (segundos)
Manipulador se desloca em X	M3	0,87
Manipulador desce em Z	P3	0,87
Ventosa Pneumática segura placa	M1	0,29
Manipulador sobe em Z	P3	0,87
Manipulador desloca-se para origem	M3	0,87
Manipulador desce em Z	M3	0,87
Manipulador solta placa	M1	0,29
Manipulador sobe em Z	M3	0,87

Fonte: Autor

A esteira conseguirá operar normalmente a partir do momento no qual a PCB reprovada for retirada da esteira, portanto, podemos reiniciar o processo normalmente logo após a captura da PCB e transporte vertical, assim, até o estado "Manipulador sobe em Z". Logo, o valor de tempo máximo do processo para retirada do item será de 2,9 segundos.

Conforme os tempos dos processos analisados pelo métodos, pode-se identificar que o gargalo do processo será a impressão dos códigos de barras, o tempo máximo de impressão de 12,18 segundos ainda atende as especificações do cliente, onde temos um Takt-Time de 14,7 segundos. Portanto, deve-se dimensionar componentes que realizem os processos modelados anteriormente onde cada estágio não supere os 12 segundos, logo, deve-se dimensionar o item mais crítico do processo (no caso a impressora) de forma que consiga realizar a gravação de no máximo 6 códigos de barras padrão de 30mm durante o tempo dimensionado.

3.5 Arquitetura física do sistema e definição dos componentes

O processo modelado utilizando Redes de Petri nos auxiliará na programação do sistema e a cronoanálise complementar os requisitos na definição dos componentes utilizados para construção do projeto.

3.5.1 Tipos de componentes

3.5.1.1 Sensores

Para sensoreamento do produto durante o processo, há necessidade de identificar o componente com precisão sem que se altere sua trajetória, serão utilizados para isso sensores fotoelétricos. Para identificar os limites físicos internos e externos das esteiras utilizaremos sensores mecânicos de contato seco por roletes são chaves comutadoras que atuam no modo On/Off sobre um circuito quando uma detecção específica mecânica ocorre no seu elemento (WENDLING, 2010).

3.5.1.2 Atuadores elétricos

Os processos analisados anteriormente nos indicam a necessidade de precisão, tanto nas esteiras quanto nos manipuladores realizando transporte aéreo. Servomotores são elementos ideais por seu ótimo controle de precisão, alto torque e baixa vibração para os sistemas físicos. As PCBs por possuírem apenas 200 gramas, são suscetíveis a ruídos mecânicos apesar de o sistema planejado já garantir sua precisão, os cuidados sobre ruídos de vibrações mecânicas advindos dos motores pode ser aumentado utilizando servomotores dimensionados corretamente.

3.5.1.3 Comunicação

O protocolo CANopen utiliza uma comunicação serial com longas distâncias de transmissão, alta velocidade de transmissão de dados, baixo custo e bastante conhecidos nas aplicações de automação (CAI L. ZHOU; YAN, 2011). Utilizando o protocolo citado pode-se controlar até 32 servomotores ligados em série utilizando um único CLP e ocupando apenas uma saída CAN no controlador.

Visto que teremos uma comunicação externa para troca de informações, um switch intermediará a comunicação dos dispositivos dedicados a atividades computacionais como processamento de imagens e os CLPs por meio de uma ponte física. O computador e o CLP trocam informações via protocolo ModBus. O equipamento também adiciona ao sistema maior capacidade de customização, integrabilidade e modularidade.

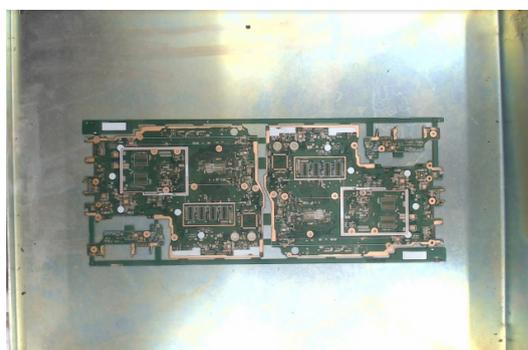
O input de informações sobre qual PCB será manufaturada em qual momento, será

transmitido remotamente. Portanto, será necessária uma tela para visualização de qual modelo de PCB o colaborador deve inserir no processo.

3.5.1.4 Captura de imagem e inspeção de códigos

A imagem capturada no início do processo garantindo que o produto correto seja inserido, deverá ser uma câmera com capacidade de pelo menos 480p para uma detecção dos detalhes da PCB conforme testes realizados.

Figura 34: Testes com resoluções de câmera para definição do equipamento de captura de imagem.



Fonte: Autor

3.5.1.5 Gerenciamento do sistema

O processamento do tipo do modelo que irá ser manufaturado no sistema, a validação via Inteligência Artificial e inspeção dos códigos impressos são realizados por um computador.

Os computadores NUC possuem robustez considerável e capacidade de gerenciamento de diversos periféricos simultâneos podendo ter tal capacidade melhorada dependendo do processador. Irão gerenciar aa informações sobre qual código de barras será impresso em tempo real, conterà uma IA embarcada classificando tipo de PCB a ser manufaturada como forma de Poka-Yoke.

Figura 35: Mini computador NUC reposável pelo gerenciamento das informações e processamentos computacionais.



Fonte: Autor

3.5.1.6 Controlador Lógico Programável

O dispositivo será responsável por comandar os movimentos, atuações e entender os sinais enviados por sensores será o CLP com um algoritmo em Ladder embarcado com controle local, utiliza-se a rede Ethernet apenas para envio de informações de supervisão e recebimento de informação de mudança de modelo de PCB. O Módulo CANopen já existe de forma integrada ao sistema original dos modelos de CLPs com família AS da empresa Delta Electronics, permitindo melhor fluidez e compactibilidade a arquitetura física.

3.5.1.7 Atuadores pneumáticos

Para movimentos lineares onde a distância de atuação forem sempre constantes, atuadores lineares pneumáticos são eficientes e mais baratos em relação a atuadores elétricos. As válvulas solenoides eletropneumática são os componentes que de fato realizam o comando para as atuações e recebem um estímulo incitado pelo CLP, uma válvula consomem apenas uma saída digital do controlador.

Portanto, conforme os requisitos descritos acima, pode-se unir e estruturar a arquitetura e distribuição dos componentes identificando suas quantidades e reunindo informações para construção de um posterior projeto elétrico realizado pela equipe responsável e dimensionamento dos controladores do sistema.

Identificando as quantidades dos itens, temos então:

3.5.2 Quantidade de componentes

3.5.2.1 Para os Sensores

1 Sensor para identificar a pilha de placas no início do processo;

1 Sensor fim de curso do manipulador de inserção das placas na esteira de impressão;

1 Sensor para identificar a chegada da PCB ao início da esteira de impressão;

1 Sensor para identificar a chegada da PCB ao local de impressão dos códigos;

1 Sensor no início do módulo de leitura e inspeção dos códigos de barras;

1 Sensor no final da esteira de leitura dos códigos;

1 Sensor fim de curso do manipulador de retirada de placas reprovadas na inspeção;

1 Sensor na parte interna da esteira de impressão para limitar o movimento e evitar possíveis avarias;

1 Sensor na parte externa da esteira de impressão para limitar o movimento e evitar possíveis avarias;

1 Sensor na parte interna da esteira de inspeção para limitar o movimento e evitar possíveis avarias;

1 Sensor na parte externa da esteira de inspeção para limitar o movimento e evitar possíveis avarias.

Para atuadores elétricos

1 Servomotor para transporte do grupo de placas.

2 Servomotores para o manipulador cartesiano sendo um com freio.

2 servomotores para ajuste lateral das esteiras.

2 servomotores para transporte do produto nas esteiras de impressão e inspeção dos códigos.

Comunicação

1 Switch de conexão cabeada em rede local com de pelo menos 4 portas.

3.5.2.2 Captura de imagem

1 Web cam HD.

3.5.2.3 Inspeção dos códigos

Leitor de códigos de barras laser da marca *Zebra* conforme especificação utilizada no processo.

3.5.2.4 Gerenciamento do sistema

1 computador NUC com Intel Core I3 com 8Gb de memória RAM e 128Gb de armazenamento.

3.5.2.5 Controlador Lógico Programável

2 CLPs com 12 saídas e entradas digitais e protocolo CANopen integrado do modelo Delta AS228R.

2 Fontes DC de 24 Volts.

3.5.2.6 Atuador pneumático

1 atuador pneumático para empurrar a PCB até o local do armazenamento;

1 atuador pneumático para captura de PCB reprovada na inspeção;

2 Válvulas eletropneumáticas solenoides 5/2;

2 geradores de vácuo;

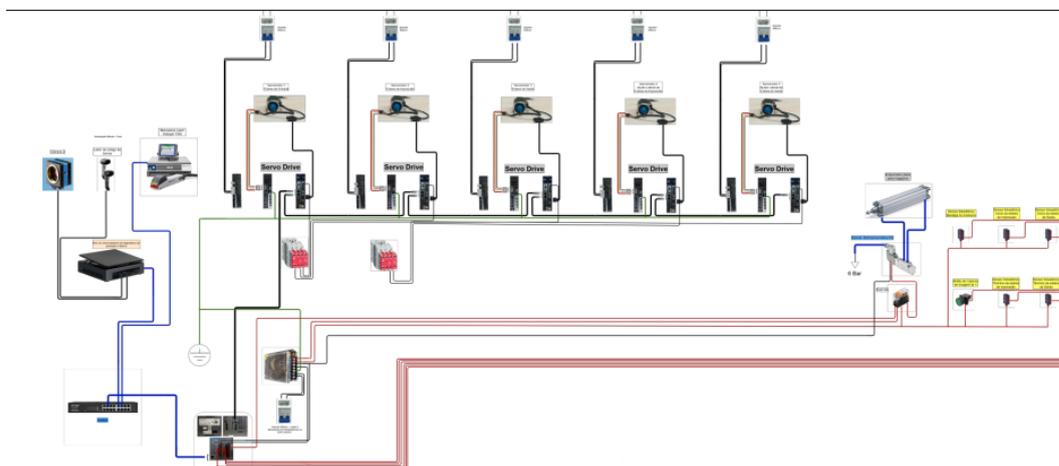
8 ventosas pneumáticas.

3.5.3 Arquiteturas

Realizado o mapeamento dos itens a serem utilizados, pode-se construir uma arquitetura física de toda disposição dos componentes do sistema. Modularmente, o sistema será dividido entre o módulo das esteiras e o módulo dos manipuladores.

A figura a seguir exibe de forma bem visual a distribuição dos componentes das esteiras pelo sistema juntamente com a comunicação dos controladores.

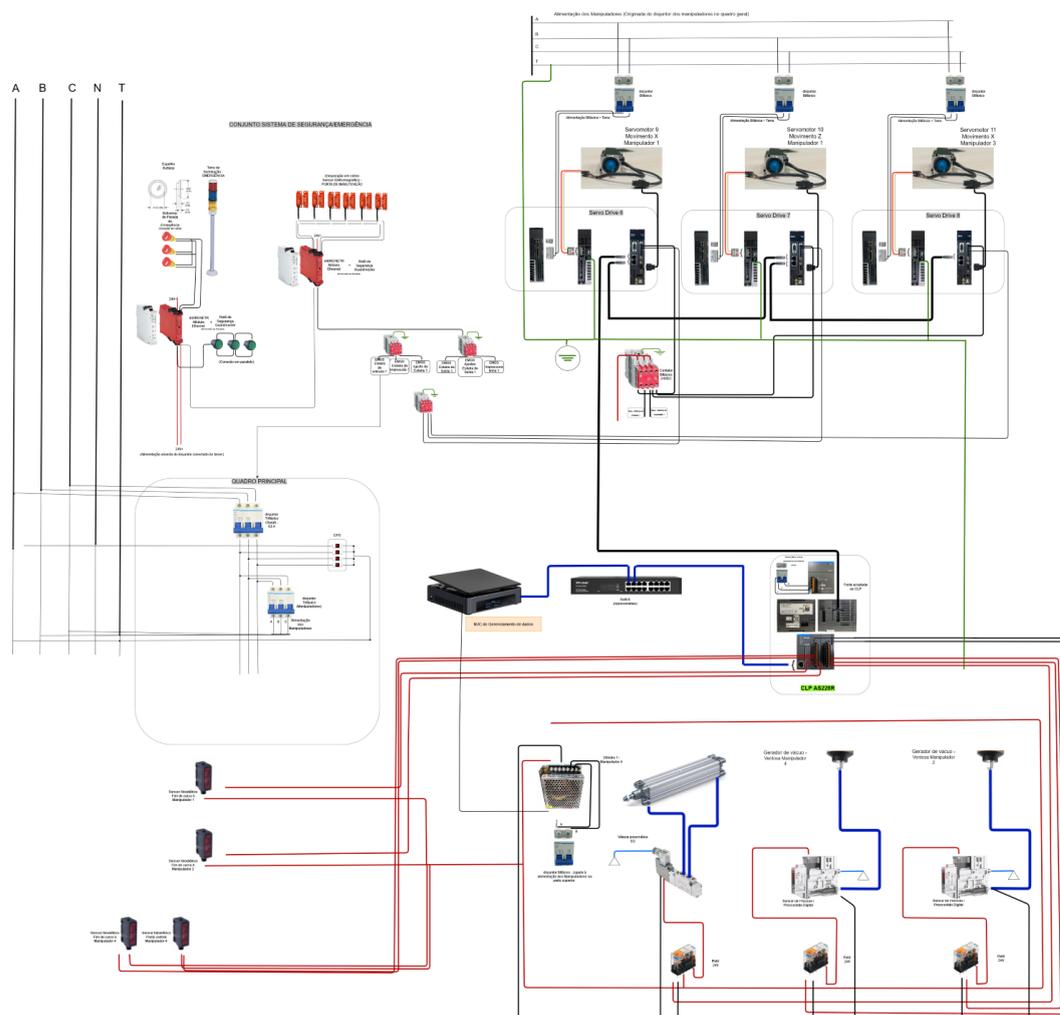
Figura 36: Arquitetura das esteiras.



Fonte: Autor

Já a figura a seguir exibe de forma visual a distribuição dos componentes dos manipuladores. Portanto, os painéis elétricos devem ser montados em locais físicos separados conforme as imagens ilustradas.

Figura 37: Arquitetura do sistema de manipuladores e segurança.



Fonte: Autor

Os componentes "NUC" e o "Switch" estão presentes nas duas imagens, porém são apenas o ponto em comum entre elas, tais componentes possuirão apenas uma unidade de cada e estarão dispostos fora dos painéis elétricos, em local apenas que seja conveniente para as ligações cabeadas.

3.6 Programação do sistema

O sistema irá operar de forma modular, portanto teremos dois programas diferentes embarcados em dois CLPs diferentes, um será responsável por todas as entradas e saídas relacionadas as esteiras e outro será responsável pelas entradas e saídas dos manipuladores. O sistema utiliza o protocolo ModBus para realizar troca de informa-

ções entre os dispositivos, tanto de CLP para CLP quanto dos CLPs para o NUC e vice versa.

3.6.1 Programação das esteiras

Para comandar os elementos motores principais do sistema, é necessário realizar a parametrização dos servodrivens por meio do ASDASoft alterando os parâmetros padrões.

Figura 38: Parametrização de servomotores no ASDASoft.

Code	Value	Unit	Min	Max	Default	Description	
V 1.054							
P2 - 00	KPP	35	rad/s	0	2047	35	Position Loop Gain
P2 - 01	PPR	100	%	10	500	100	Switching Rate of Position Loop Gain
P2 - 02	PFG	50	%	0	100	50	Position Feed Forward Gain
P2 - 03	PFF	5	ms	2	100	5	Smooth Constant of Position Feed Forward Gain
P2 - 04	KVP	500	rad/s	0	8191	500	Speed Loop Gain
P2 - 05	SPR	100	%	10	500	100	Switching Rate of Speed Loop Gain
P2 - 06	KVI	100	rad/s	0	1023	100	Speed Integral Compensation
P2 - 07	KVF	0	%	0	100	0	Speed Feed Forward Gain
P2 - 08	PCTL	36		0	501	0	Special Parameter Write-in
P2 - 09	DRT	2	2ms	0	20	2	D1 Debouncing Time
P2 - 10	D11	0x0001		0x0000	0x015F	0x0101	D11 Functional Planning
P2 - 11	D12	0x0146		0x0000	0x015F	0x0104	D12 Functional Planning
P2 - 12	D13	0x0000		0x0000	0x015F	0x0116	D13 Functional Planning
P2 - 13	D14	0x0117		0x0000	0x015F	0x0117	D14 Functional Planning
P2 - 14	D15	0x0102		0x0000	0x015F	0x0102	D15 Functional Planning
P2 - 15	D16	0x0122		0x0000	0x015F	0x0022	D16 Functional Planning
P2 - 16	D17	0x0123		0x0000	0x015F	0x0023	D17 Functional Planning
P2 - 17	D18	0x0121		0x0000	0x015F	0x0021	D18 Functional Planning
P2 - 18	DO1	0x0101		0x0000	0x013F	0x0101	DO1 Functional Planning
P2 - 19	DO2	0x0103		0x0000	0x013F	0x0103	DO2 Functional Planning
P2 - 20	DO3	0x0109		0x0000	0x013F	0x0109	DO3 Functional Planning
P2 - 21	DO4	0x0105		0x0000	0x013F	0x0105	DO4 Functional Planning
P2 - 22	DO5	0x0007		0x0000	0x013F	0x0007	DO5 Functional Planning
P2 - 23	NCF1	1000	Hz	50	1000	1000	Resonance Suppression (Notch Filter) (1)
P2 - 24	DPH1	0	dB	0	32	0	Resonance Suppression (Notch Filter) Attenuation Rate (1)
P2 - 25	NLP	0,2	0,1ms	0,0	100,0		Depend on Dev(Low-pass Filter of Resonance Suppression)

Fonte: Autor

Na tela da figura 38, os principais parâmetros a serem alterados são os parâmetros P1-01 alterado para 0001 habilitando modo PR (habilitando o controle de posição do servo), o parâmetro P3-01 alterado para 0400 ajustando a comunicação CANOpen para 1 MB/s, o parâmetro P2-10 alterado para 001 para desabilitar a redundância do comando de habilitação de energização via IOs e por fim, alterar o parâmetro P3-00 identificando cada servomotor numericamente (ex: motor da esteira de impressão = 1, motor da esteira de inspeção = 2), como o protocolo CAN funciona de modo serial, o comando será inserido na rede pelo CLP e o motor com parâmetro equivalente, responderá ao comando enviado. Definiu-se então:

Tabela 10: Valores inseridos nos parâmetros P3-00 de cada servodriver.

Servomotor	P3-00
Servo da esteira inicial	0001
Servo do ajuste da esteira de impressão	0002
Servo do ajuste da esteira de inspeção	0003
Servo de transporte de placas para impressão	0004
Servo de transporte de placas para inspeção	0005

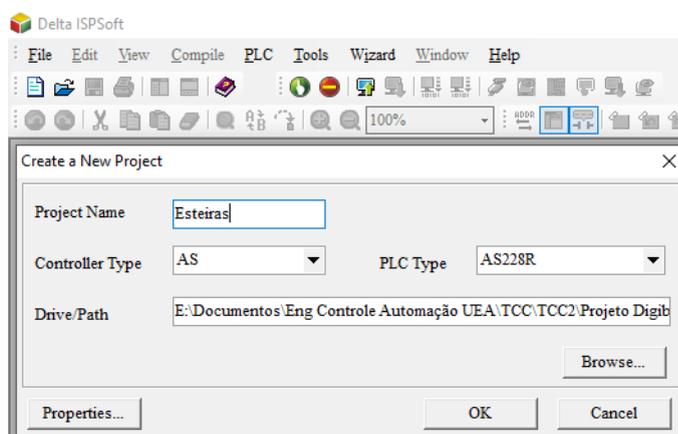
Fonte: Autor

Todos os valores dos parâmetros estão anexados no apêndice C.

Para construção dos programa em linguagem de programação Ladder, será utilizado o software gratuito da empresa *Delta Electronics* e suas extensões encontradas no próprio site.

Para início das operações deve-se instalar o ISPSOft e criar um novo projeto conforme figura a seguir:

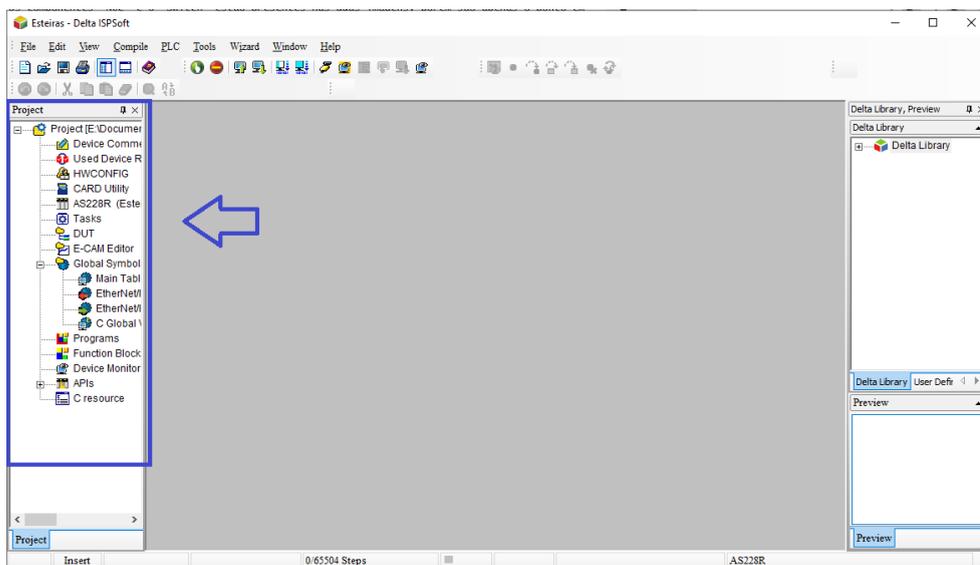
Figura 39: Criação de um projeto no ISPSOft.



Fonte: Autor

Escolhe-se o modelo do CLP utilizado e o local do computador onde o projeto será armazenado. Após a criação do projeto, pode-se notar ao lado direito as extensões que fazem parte do projeto criado.

Figura 40: Menu de extensões do ISPSOft.

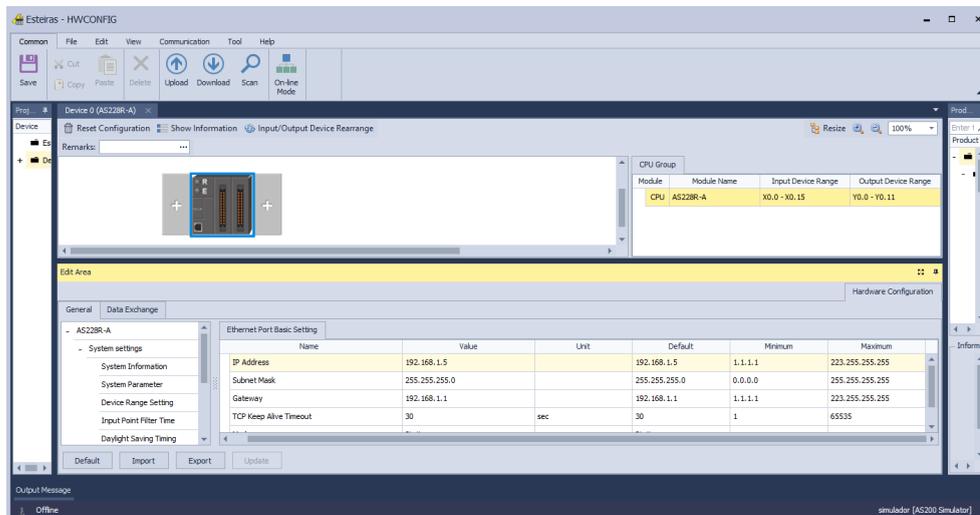


Fonte: Autor

Uzar-se-á apenas a *Programs* e o *HWCONFIG*, no *Programs* deverá ser escrito todo o código em Ladder, enquanto o *HWCONFIG* será responsável por parâmetros de comunicação com os servomotores, taxa de velocidade de comunicação e endereço de IP do CLP.

A extensão *HWCONFIG* tem sua tela de início mostrada a seguir onde serão realizadas as modificações de acordo com o especificado pelo projeto, para o CLP das esteiras usaremos como endereço de IP o 192.168.1.2, deste modo, poderemos enviar ou receber dados do dispositivo desejado apenas a partir de tal parâmetro.

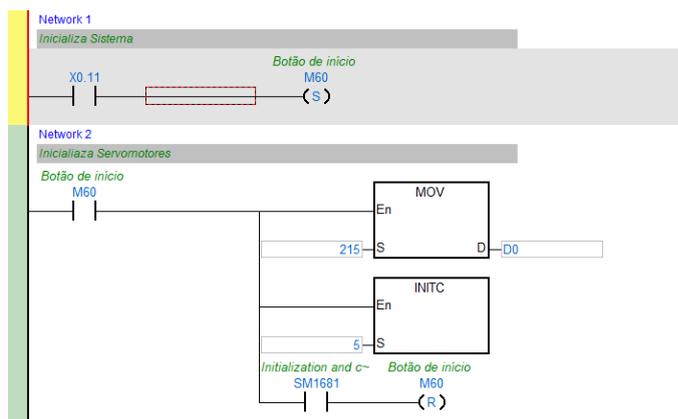
Figura 41: Menu do HWCONFIG.



Fonte: Autor

No *Programs* iniciaremos então, a programação das esteiras conforme o fluxo de automação modelado em Redes de Petri. É necessária a inserção inicial do comando para inicialização dos servomotores via protocolo CANopen chamada *INITC*, no parâmetro *S* do comando *INITC* serão inseridos a quantidade de servomotores que se pretende controlar, no caso das esteiras o CLP controlará 5 servomotores. Um botão pressionado pelo colaborador responsável por inserir as PCBs no processo será o responsável por tal inicialização mostrado na figura a seguir:

Figura 42: Início do programa em Ladder.



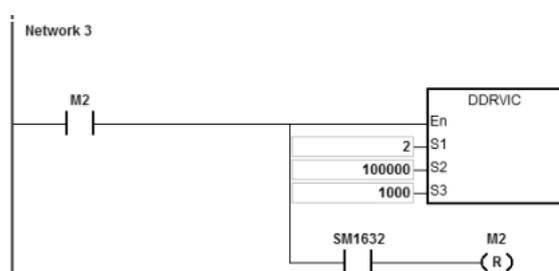
Fonte: Autor

O comando *MOV* escreve uma informação no registrador especificado, no caso, inseriu-se o valor médio do tamanho lateral de uma PCB como forma de evitar um

Bug no programa onde, caso não fosse fornecido um valor para o tamanho da PCB, ocorreria um possível erro onde um valor aleatório não atualizado com variação de -2147483648 a +2147483647 seria escrito no registrador responsável justamente pelo tamanho da abertura das esteiras.

Segundo o manual de programação dos CLPs da família *AS* da marca *Delta Electronics*, a estrutura da linha de comando para controle do servomotor por incremento de pulso é a seguinte:

Figura 43: Estrutura do comando de controle de servomotores.

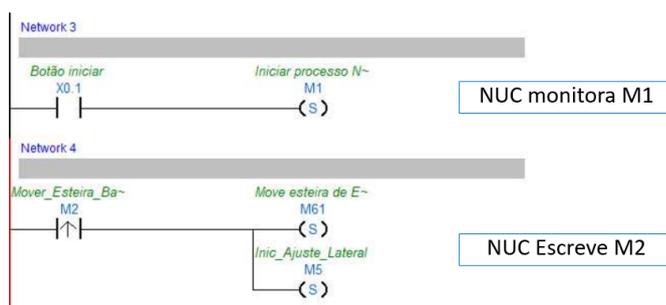


Fonte: (DELTA, 2022)

Onde a *Flag* SM1632 indicará que o servo chegou ao destino desejado de 100000 pulsos por meio de uma troca de estado entrando em nível lógico alto e permanecendo até que receba um comando de *reset*.

Um botão para iniciar o processo deve ser pressionado para início do processo, neste momento o CLP envia o comando para o NUC via Modbus.

Figura 44: Estrutura do comando de controle de servomotores.

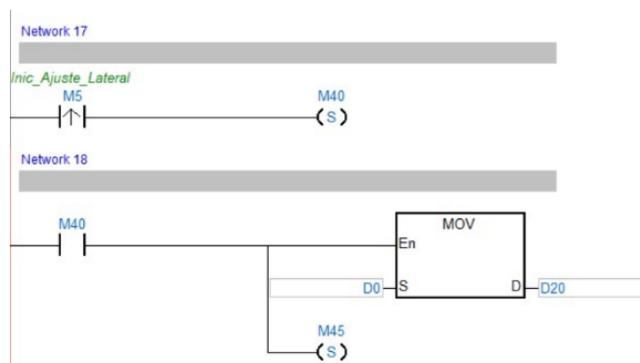


Fonte: Autor

Ao receber o comando do CLP, o NUC escreve o valor do tamanho da PCB em milímetros no registrador D0 que por sua vez, registra o valor em D20 para manter D0

livre para alterações no decorrer do processo. No processo a seguir é realizado uma conversão do tamanho de largura das placas para pulsos de revoluções do servomotor.

Figura 45: Inicialização do processo de ajuste lateral.



Fonte: Autor

Com o valor da largura da PCB que será processada devidamente escrita no registrador D20, divide-se o valor por 2 pois, no sistema mecânico, o fuso de esferas utilizado para realizar a transmissão do movimento possui uma metade com rosca padrão e outra metade com rosca inversa, isto significa que, quando o servomotor girar no sentido horário a esteira se fecha, quando o servomotor girar no sentido anti horário a esteira se abre.

O fator de conversão linear utilizado provém da seguinte estimativa utilizando os parâmetros de resolução de movimento do servomotor P1-44 e P1-45 como padrões (128 e 10 respectivamente):

Passo do fuso = 5mm;

Pulsos para realizar uma volta completa = 100.000 PPU;

Logo,

$$FatorDeConversao = \frac{100000PPU}{5mm}$$

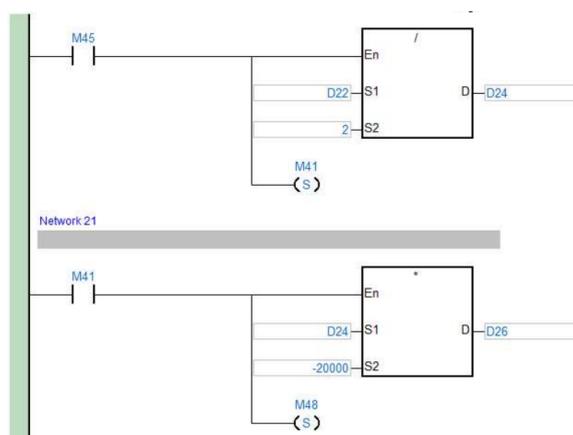
$$FatorDeConversao = 20000 \frac{PPU}{mm}$$

Na figura 64 temos o calculo de conversão de minímetro para pulsos, levando em conta o sistema linear, a equação de primeiro grau é a seguinte:

$$Pulso = -20000 * LarguraPCB$$

Na linha de comando foi realizada a multiplicação entre o valor guardado no registrador D24 e o valor de -20000 (o sinal negativo é inserido devido o comando do sentido de rotação do servomotor), o valor de resultado é guardado no registrador D26 mostrado a seguir.

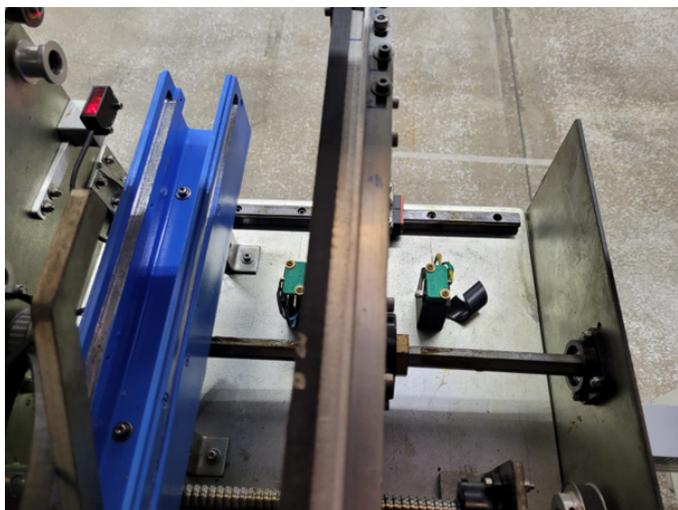
Figura 46: Bloco de divisão do valor por 2 usando apenas metade do valor para fazer o movimento.



Fonte: Autor

Uma adversidade do sistema é sua memória volátil, portanto, quando há um reset no sistema os dados de sua posição absoluta é perdida. Logo, há uma necessidade de criar uma referência, para isso inseriu-se um sensor mecânico fim de curso na parte interna da esteira conforme a figura abaixo.

Figura 47: Sensor mecânico fim de curso instalado nas partes internas e externas das esteiras

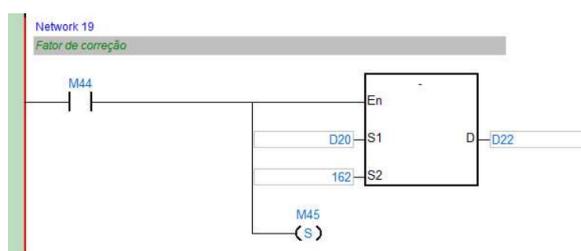


Fonte: Autor

Fecha-se a esteira até o sensor fechar seu contato, com isso temos a referência para então realizar o incremento de pulsos e ajuste lateral.

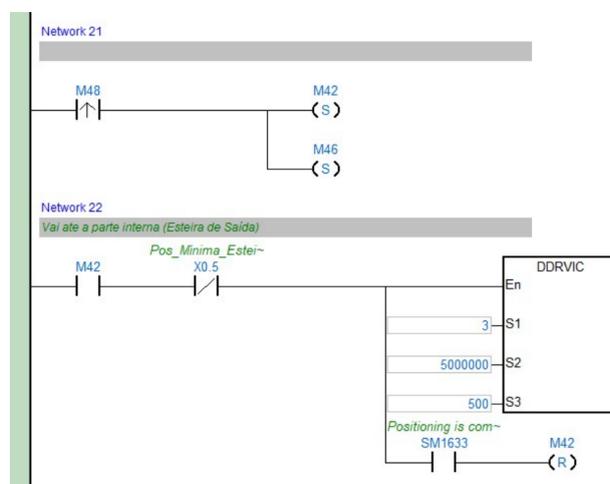
A distância entre o ponto de aferição do sensor e a centralização deve entrar e no cálculo de abertura.

Figura 48: Valor de correção de centralização em relação ao deslocamento do sensor fim de curso com o centro.



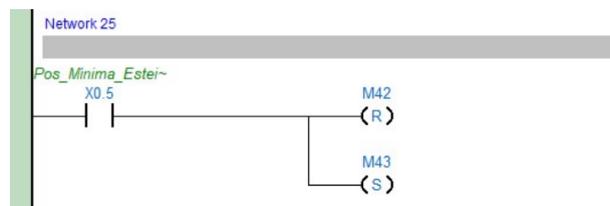
Fonte: Autor

Figura 49: Linha de comando para fechamento total das esteiras aferindo a origem do sistema.



Fonte: Autor

Figura 50: Linha de comando de recebimento do sinal de confirmação da posição de origem do sistema.

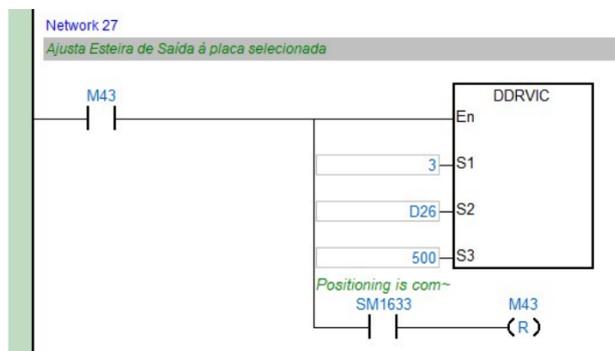


Fonte: Autor

No parâmetro do comando DDRVIC onde insere-se a quantidade de pulsos, pode-se apontar para um registrador que contém um valor, este valor poderá ser alterado por um meio externo, do modo que foi apresentado anteriormente.

Deste modo pode-se configurar de forma automática a abertura ideal da esteira para qualquer largura de PCB com limites de *110mm* a *389mm* e velocidades de *0,1 rpm* a *6000 rpm*.

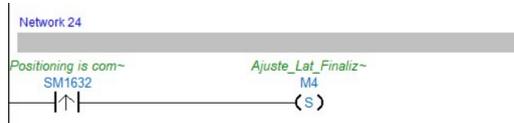
Figura 51: Linha de comando responsável pela abertura da esteira até o valor calculado e guardado em D26.



Fonte: Autor

Ao completar o movimento, A flag do encoder indica que o servomotor chegou ao local desejado, envia a mensagem ao dispositivo externo e continua o processo para o passo seguinte conforme o modelado anteriormente em Redes de Petri.

Figura 52: Ajuste lateral finalizado indicado pela flag do servomotor.



Fonte: Autor

Figura 53: Relés virtuais utilizados para indicar fim de um processo e início do seguinte.



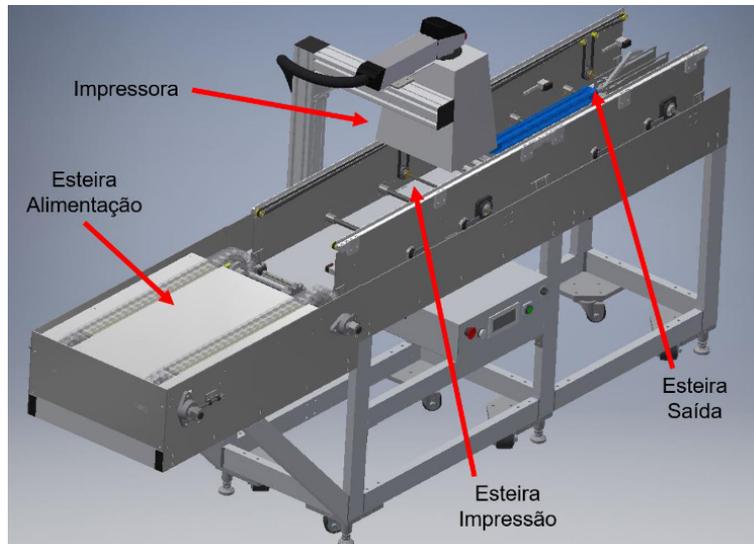
Fonte: Autor

Os programas completos são apresentados no apêndice B.

4 RESULTADOS E DISCUSSÃO

Como visto no capítulo 3, o sistema passou por diversas validações para construir um fluxo capaz de suprir as necessidades de tempo e processo de impressão de códigos de barras em PCBs. A equipe multidisciplinar realizando um trabalho conjunto e alinhado foi capaz de projetar e montar o protótipo de modo que todo o trabalho de automação planejado pudesse ser posto em prática conforme projetado por (CORRÊA et al., 2021) e sua modelagem do sistema proposto esboçado em CAD.

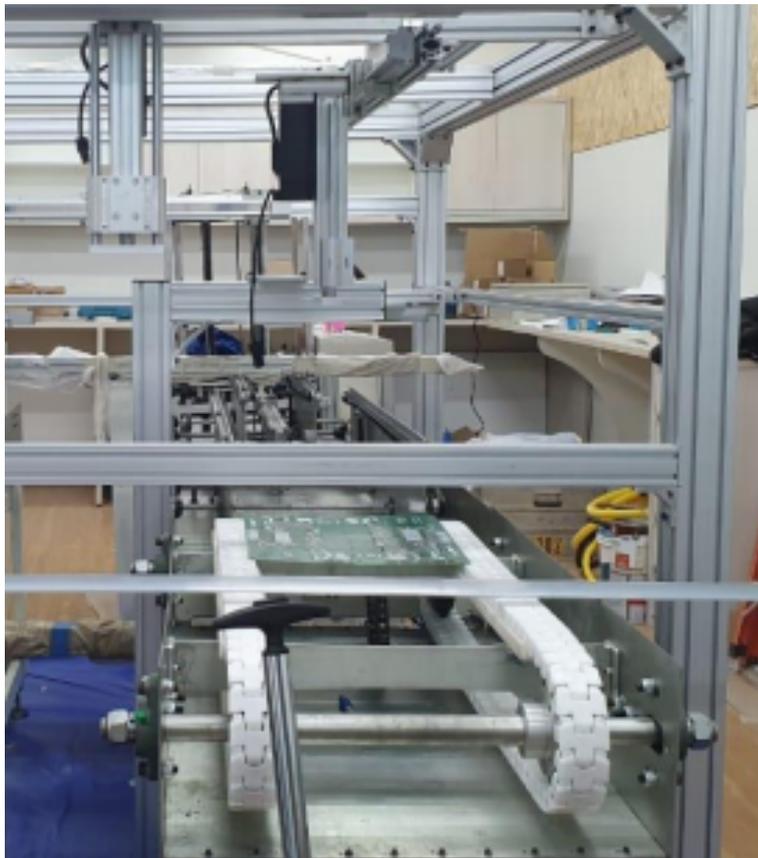
Figura 54: Projeto do modelo em CAD.



Fonte: (CORRÊA et al., 2021)

O modelo real construído com base nos requisitos levantados anteriormente e também onde

Figura 55: Projeto do Modelo montado.



Fonte: Autor

Os controladores foram distribuídos em dois painéis elétricos separados, sendo o painel responsável pelo movimento dos manipuladores também responsável pelo sistema de segurança, portanto, a distribuição dos painéis elétricos nos traz um nível maior de modularidade, melhor refrigeração e ao menos 6 borneiras livres em cada painel.

Figura 56: Painel elétrico de comando das esteiras.



Fonte: Autor

Figura 57: Painel elétrico de comando dos manipuladores e segurança.

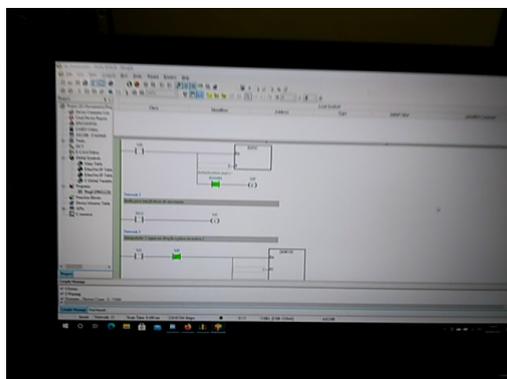


Fonte: Autor

4.1 Valores constantes posição do sistema

Alguns valores constantes relacionados a posição foram encontrados por método empírico. As relações de transmissões de movimento possuem polias com diâmetros diferentes e difíceis de serem medidos tornando o ajuste da quantidade de pulsos mais eficiente para chegar ao resultado, portanto, foram realizados testes com PCBs reais utilizadas no processo até chegarmos às quantidades de pulsos ideais para cada processo que seja constante.

Figura 58: Ajuste de parâmetros das esteiras via testes.



Fonte: Autor

Figura 59: Ajuste de parâmetros dos manipuladores via testes.



Fonte: Autor

Temos então o seguinte:

- -150000 no servomotor que transporta o grupo de PCBs;
- 523663 no servomotor da esteira de impressão;
- 410000 pos impressão;
- 900000 esteira de saída (reset pelo sensor ao final) .

Utilizando tais parâmetros, os pontos desejados foram atingidos com êxito e erro de posição máximo de $\pm 1mm$.

4.2 Tempo estimado X tempo medido

Realizando uma cronoanálise do sistema automático com um teste linear e identificando os macroprocessos, tem-se a comparação da estimativa de tempo de processo utilizando MOST e o tempo real medido dos processos.

Tabela 11: Sequência de movimentos do processo comparando o tempo estimado pelo método MOST com o tempo real medido

Estado	MOST(segundos)	Tempo real(segundos)
Operador aperta botão	0,29	0,5
IA Lê pilha de placas	2,9	1
IA aprova resultado e registra	0,87	1
Ajuste lateral das Esteiras	4,64	17
Esteira de inserção de placas	1,74	4
Manipulador insere placa para esteira	6,09	9
Deslocamento da placa para impressão	1,74	10
Impressão dos códigos	12,18	4
Esteira Avança ao próximo estágio	1,74	10
Leitor de códigos inspeciona a placa	2,9	2
Esteira Avança	1,74	10
Atuador Avança	0,87	2

Fonte: Autor

O ajuste lateral de esteiras foi realizado utilizando a *Placa 4*, na velocidade angular do servomotor utilizou-se valor de 50.0 rpm obtendo 17 segundos como tempo de ajuste, ultrapassando o tempo máximo permitido. Aumentado sua velocidade ao dobro do testado inicialmente ($S3 = 1000 = 100.0 \text{ rpm}$) pode-se realizar o processo de setup para receber uma PCB com largura diferente com metade do tempo obtido no teste e atingindo 1,67% da velocidade máxima do servomotor e valor máximo de tempo de *setup* de 10 segundos a 100 rpm .

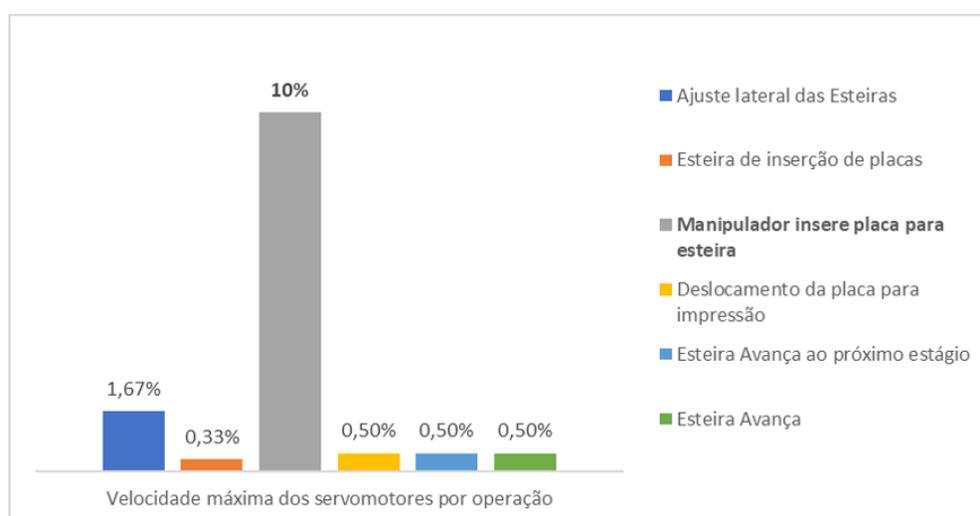
Na esteira de inserção de PCBs utilizou-se o valor de 20.0 rpm , ou seja, 0,33% da velocidade máxima do servomotor.

Nos manipuladores utilizaram-se 600.0 rpm , ou seja, 10% da velocidade máxima do servomotor.

Nas esteiras de impressão e inspeção dos códigos de barras utilizaram-se 30.0 rpm , ou seja, 0,5% da velocidade máxima do servomotor.

De modo que o desempenho do sistema baseado nos servomotores foi medido com as seguintes características:

Figura 60: Porcentagem de utilização dos servomotores em relação a sua velocidade angular máxima de 6000 rpm .

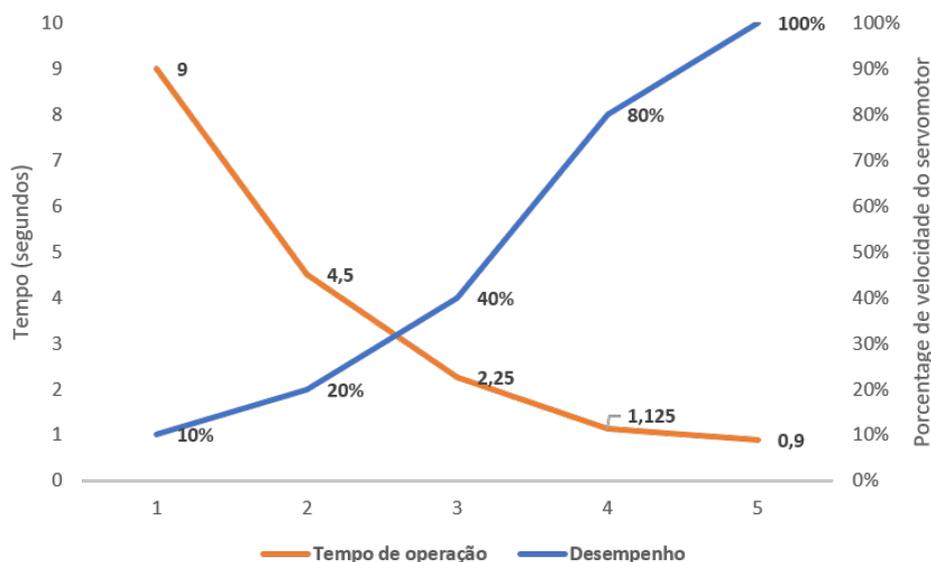


Fonte: Autor

O processo da inserção de PCBs individuais por um manipulador utiliza a maior porcentagem de velocidade do servomotor realizando o processo em 9 segundos a 10% da velocidade máxima, portanto, em relação a customização do sistema e flexibilização na alteração dos parâmetros de velocidade de tempo de operação levando a operação

ao estresse máximo, estima-se atingir o valor 0,9 segundos de operação (utilizando a velocidade de 6000 *rpm*) com suas curvas de desempenho da seguinte forma:

Figura 61: Desempenho do sistema a partir do processo com menor capacidade de flexibilização, o processo de inserção de PCBs individuais.



Fonte: Autor

4.3 Fluxo de automação

No fluxo de automação, houve uma alteração nos microprocessos do fluxo modelado por conta da impossibilidade da aferição de posição absoluta após um reset. Adicionou-se um movimento de fechamento das esteiras para aferição de um ponto de origem.

4.4 Modelos de PCB

O modelos de PCBs testadas possuíam as larguras de seguinte tamanho:

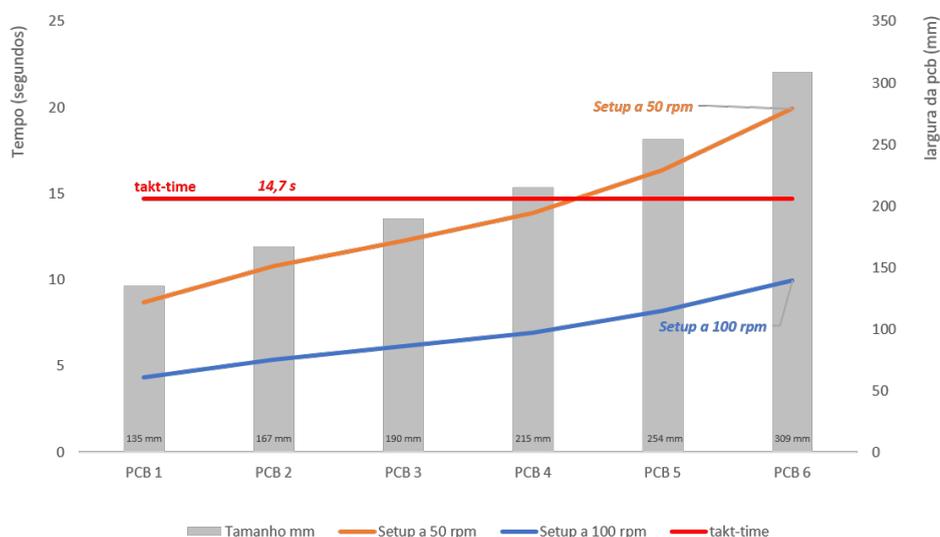
Tabela 12: Tamanho de PCBs testadas.

Modelo	Largura(milímetro)
PCB 1	135
PCB 2	167
PCB 3	190
PCB 4	215
PCB 5	254
PCB 6	309

Fonte: Autor

O desempenho do sistema de ajuste lateral baseado nos testes com 50rpm e 100rpm demonstram desempenhos sincronizados aos estimados atingindo as expectativas de takt-time utilizando apenas 1,67% do recurso máximo entregue pelo sistema.

Figura 62: Tempo de setup utilizando os testes em cada placa com velocidades de 50 rpm e 100 rpm.



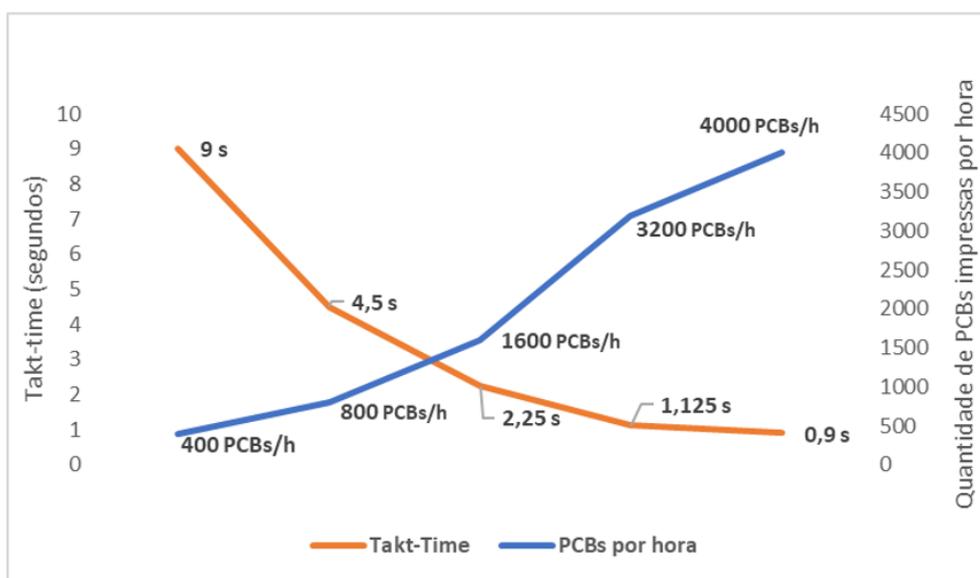
Fonte: Autor

4.5 Takt-time do sistema

De acordo com os tempos reais aferidos, os possíveis gargalos do sistema são os processos de transporte das PCBs com 10 segundos que de acordo com as metodologias apresentadas poderia ser considerado o *takt-time* do sistema, porém, o

tempo de 10 segundos é aferido utilizando apenas 0,5% da capacidade do servomotor. Logo, o takt-time será referenciado de acordo com o processo de maior capacidade de saturação conforme analisado pelo gráfico da figura 60. Baseado em tais dados, a projeção de produtividade do sistema de impressão de códigos de barras possui os seguintes valores:

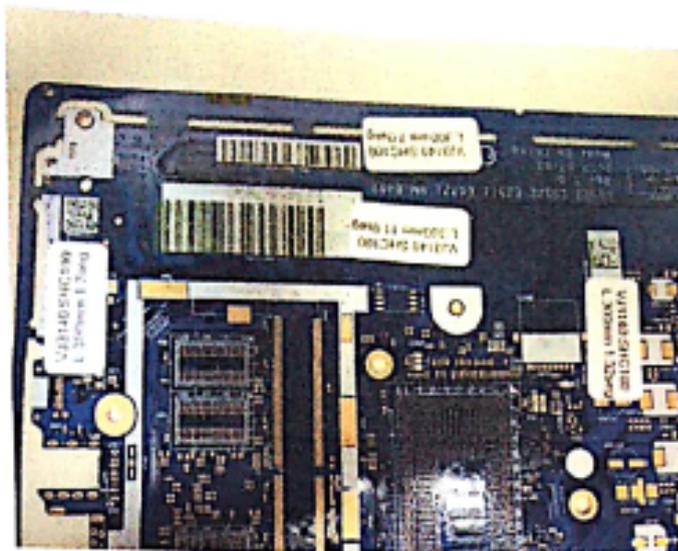
Figura 63: Projeção de produtividade do sistema de acordo com a variação no valor do tempo de processo de inserção de placas individuais na esteira por um manipulador robótico.



Fonte: Autor

O gargalo do sistema estimado no capítulo 3 identificou o processo de impressão como sendo o maior tempo de operação, porém o modelo de impressora dimensionado para o projeto consegue realizar a impressão de 1 código em até 1,3 segundos conforme teste realizado nos laboratórios do fornecedor Videojet Brasil - Datadores Ink Jet e confirmado nos laboratórios da Universidade do Estado do Amazonas.

Figura 64: Testes de impressão ados códigos de barras.



Fonte: Autor

O relatório completo pode ser visto no apêndice D como anexo.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, foi desenvolvido o planejamento e implementação de um protótipo de uma linha de produção completamente automatizada, capaz de imprimir códigos de barras em PCBs e validá-los.

O processo possui repetibilidade garantida por realizar apenas incrementos precisos nas posições das esteiras, logo, um valor de erro de posição não se acumula até o final do processo trazendo consistência e constância ao sistema.

O fluxo de automação foi projetado e utilizado como guia para a programação identificando os microprocessos e gerando uma receita que pôde ser facilmente interpretada e traduzida nos códigos ladder. A atuação humana é mínima para realizar o processo, onde ocorrerá apenas para alimentação inicial de matéria prima e monitoração.

A programação foi baseada nos requisitos do sistema devidamente modelados utilizando as redes de petri com pequenos ajustes no programa para atender os requisitos identificados na modelagem e o fluxo proposto. Os programas funcionam de forma distribuída assíncrona, realizando uma tarefa a partir de uma requisição.

O tempo de processo utilizado como pré requisito para atender a demanda do cliente foi atendido com uma folga de 4 segundos abaixo do máximo utilizando 10% da capacidade dos servomotores sendo assim, capaz de aumentar sua capacidade de produção em até 10 vezes.

A reconfiguração foi aplicada de modo que o sistema atente a escalabilidade de uma posterior aumento de demanda atingindo um valor de tempo abaixo do takt time estimado. A esteira consegue se ajustar e manufaturar qualquer largura de PCB que seja maior que 110 mm e menor que 389 mm com um sistema de setup automático de 10 segundos. Tal processo dá ao sistema alto valor de *Convertibilidade e Inte-*

grabilidade, adicionado ao atendimento dos requisitos impostos utilizando 10% da capacidade do sistema, obtém-se também um maior grau de *Customização*.

O sistema completo é interligado pelo switch e pode ser acessado de um ponto externo para então identificar os a origem dos problemas, porém, os problemas identificados pelo protótipo são apenas de posição, identificados visualmente pelos servomotores.

Portanto, é possível desenvolver o protótipo de um processo de impressão a laser de códigos de barras capaz de abranger todos os tipos de PCBs manufaturados no processo atual adicionado uma margem de pelo menos 25 mm para novos modelos cumprindo os requisitos dos Sistemas de Manufatura Reconfiguráveis. O processo do planejamento do projeto incluindo fluxograma e modelagem em Redes de Petri foram de extrema importância para identificar os pontos necessários de aplicação da reconfiguração e seus requisitos alcançando então o êxito com relação ao onjetivo definido.

Por fim, para dar continuidade à pesquisa e melhorar o projeto apresentado, faz-se a seguir algumas sugestões e considerações para trabalhos futuros.

- a) Construção de modelagem em Redes de Petri temporizadas para acompanhamento supervisorio simplificado do processo;
- b) Utilizar as *flags* já definidas nos servomotores para uma pulverização da informação de diagnóstico diretamente para as camadas mais altas de automação;
- c) Aquisição e configuração de módulos extensores do CLP capazes de armazenar o valor de posição absoluta dos *encoders* nos servomotores mesmo após uma desenergização do sistema;
- d) Inserção de sistema de identificação do colaborador que irá realizar os trabalhos de inserção do grupo de placas no processo;

REFERÊNCIAS

- ALVAREZ, R. d. R.; JR, J. A. V. A. Takt-time: conceitos e contextualização dentro do sistema toyota de produção. **Gestão & Produção**, SciELO Brasil, v. 8, p. 1–18, 2001.
- ANDERSEN, A.-L.; NIELSEN, K.; BRUNOE, T. D. Prerequisites and barriers for the development of reconfigurable manufacturing systems for high speed ramp-up. **Procedia Cirp**, Elsevier, v. 51, p. 7–12, 2016.
- BANGEMANN, T. et al. State of the art in industrial automation. **Industrial cloud-based cyber-physical systems**, Springer, p. 23–47, 2014.
- BOLTON, W. **Programmable logic controllers**. [S.l.]: Newnes, 2015.
- BONET, P. et al. Pipe v2. 5: A petri net tool for performance modelling. In: **Proc. 23rd Latin American Conference on Informatics (CLEI 2007)**. [S.l.: s.n.], 2007.
- BORTOLINI, M.; GALIZIA, F. G.; MORA, C. Reconfigurable manufacturing systems: Literature review and research trend. **Journal of manufacturing systems**, Elsevier, v. 49, p. 93–106, 2018.
- BRODZIK, R. Inside the competition for the first plc. **Control Engineering**, CFE Media LLC, v. 61, n. 9, p. 50–53, 2014.
- CAI L. ZHOU, X. H. H.; YAN, H. Fieldbus canopen design and application. **Beijing University of Aeronautics and Astronautics Press**, Beijing, 2011.
- CAMPANA, G. A.; OPLUSTIL, C. P. Concepts in laboratory medicine automation: literature review. **Jornal Brasileiro de Patologia e Medicina Laboratorial**, SciELO Brasil, v. 47, p. 119–127, 2011.
- CHORA, A. J. V. **Arquitetura para Sistema Distribuído de Manufatura Baseado em Agentes Utilizando Arduinos**. Tese (Doutorado), 2021.
- CORRÊA, R. G. L. et al. Adequação de uma central de impressão à norma regulamentadora 12. Universidade do Estado do Amazonas, 2021.
- CRAVO, E. Protocolo clp: Quais são, quais os principais e suas aplicações. Kalatek Automação, 2022. Disponível em: <<https://blog.kalatec.com.br/protocolo-clp/>>.
- DELTA, E. **Product Introduction**. 2022. Disponível em: <<https://www.deltaww.com/en-us/products/PLC-Programmable-Logic-Controllers/3598>>.
- DELTA INC, E. **Product Introduction**. 2022. Disponível em: <<https://www.deltaww.com/en-US/products/Servo-Systems-AC-Servo-Motors-and-Drives/23>>.

DU, D.; XU, X.; YAMAZAKI, K. A study on the generation of silicon-based hardware plc by means of the direct conversion of the ladder diagram to circuit design language. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 49, n. 5, p. 615–626, 2010.

DUDEK-BURLIKOWSKA, M.; SZEWIECZEK, D. The poka-yoke method as an improving quality tool of operations in the process. **Journal of Achievements in Materials and Manufacturing Engineering**, v. 36, n. 1, p. 95–102, 2009.

FARSI, M.; RATCLIFF, K.; BARBOSA, M. An introduction to canopen. **Computing & Control Engineering Journal**, IET, v. 10, n. 4, p. 161–168, 1999.

FIALHO, A. B. **Automação pneumática–Projetos, dimensionamento e análise de circuitos**. [S.l.]: Saraiva Educação SA, 2018.

FIEGENBAUM, A. et al. Desenvolvimento de uma garra robótica do tipo ventosa para um robô gantry com acionamento pneumático. **Salão do Conhecimento**, 2016.

FILIFE, M. d. A. R. D. **Redução dos tempos de setup aplicada a uma linha de embalagem**. Tese (Doutorado), 2022.

FOGLIATTO, F. S.; FAGUNDES, P. R. M. Troca rápida de ferramentas: proposta metodológica e estudo de caso. **Gestão & Produção**, SciELO Brasil, v. 10, p. 163–181, 2003.

FOVINO, I. N. et al. Design and implementation of a secure modbus protocol. In: SPRINGER. **International conference on critical infrastructure protection**. [S.l.], 2009. p. 83–96.

FOYO, P. M. del; MIRALLES, J. A.; SILVA, J. R. Um verificador formal eficiente para sistemas de tempo real. **X SBAI–Simpósio Brasileiro de Automação Inteligente**, v. 10, p. 1220–1225, 2011.

GEORGINI, J. M. **Automação Aplicada-Descrição e implementação de Sistemas Sequenciais com PLCs**. [S.l.]: Saraiva Educação SA, 2018.

INDUSTRIAL, T. A. **Leitor de Código de Barras**. 2022. Disponível em: <<https://tanca.com.br/automacao-comercial-produto.php?cod=92>>.

INDUSTRY, D. **Porquê escolher um ROBÔ CARTESIANO?** 2022. Disponível em: <<https://guide.directindustry.com/pt/que-robo-industrial-escolher/>>.

KOREN, Y.; GU, X.; GUO, W. Reconfigurable manufacturing systems: Principles, design, and future trends. **Frontiers of Mechanical Engineering**, Springer, v. 13, n. 2, p. 121–136, 2018.

KOREN, Y. et al. Reconfigurable manufacturing systems. **CIRP annals**, Elsevier, v. 48, n. 2, p. 527–540, 1999.

LIRAVIASL, K. K. et al. A framework for modelling reconfigurable manufacturing systems using hybridized discrete-event and agent-based simulation. **IFAC-PapersOnLine**, Elsevier, v. 48, n. 3, p. 1490–1495, 2015.

MARTINS, L. A. et al. Automação de um sistema transportador de rolos tracionados em linguagem ladder. **Revista TechnoEng-ISSN 2178-3586**, v. 1, n. 2, 2019.

MORAES, C. C. d.; CASTRUCCI, P. d. L. **Engenharia de automação industrial**. [S.l.: s.n.], 2013.

MORAES, T. **Acionamento de Servomecanismo**. Monografia (Trabalho de Conclusão de Curso) — Universidade de São Francisco, São Paulo, 2011.

MORGAN, J. et al. Industry 4.0 smart reconfigurable manufacturing machines. **Journal of Manufacturing Systems**, Elsevier, v. 59, p. 481–506, 2021.

NAFAL, K. Most: evolução do sistema de medição do trabalho. **Revista logística**, v. 277, n. 8, p. 38–41, 2013. Disponível em: <https://issuu.com/revista_logistica/docs/revista-logistica-277>.

PETERSON, J. L. Petri nets. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 9, n. 3, p. 223–252, 1977.

PUVANASVARAN, A.; YAP, Y.; YOONG, S. Implementation of maynard operation sequence technique in dry pack operation-a case study. **ARPN Journal of Engineering and Applied Sciences**, v. 14, n. 21, p. 3732–3737, 2019.

REIS, M. E. P.; ALVES, J. M. Um método para o cálculo do benefício econômico e definição da estratégia em trabalhos de redução do tempo de setup. **Gestão & Produção**, SciELO Brasil, v. 17, p. 579–588, 2010.

RIBEIRO, M. A. Automação industrial. **Salvador:[sn]**, 1999.

RÖSIÖ, C. et al. Towards an assessment criterion of reconfigurable manufacturing systems within the automotive industry. **Procedia Manufacturing**, Elsevier, v. 28, p. 76–82, 2019.

RÖSIÖ, C.; JACKSON, M. Enable changeability in manufacturing systems by adopting a life cycle perspective. In: **3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production**. [S.l.: s.n.], 2009. p. 612–621.

SACOMANO, J. B. et al. **Indústria 4.0**. [S.l.]: Editora Blucher, 2018.

SEGOVIA, V. R.; THEORIN, A. History of control history of plc and dcs. **University of Lund**, 2012.

SHIMBUN, N. K. **Poka-yoke: improving product quality by preventing defects**. [S.l.]: Crc Press, 1989.

SHOOK, J. Y. Bringing the toyota production system to. **Becoming lean: Inside stories of US manufacturers**, CRC Press, p. 41, 1997.

SILVA, R. M. d. **Controle de sistemas reconfiguráveis de manufatura**. Tese (Doutorado) — Universidade de São Paulo, 2016.

SILVEIRA, C. B. Fonte 24v: Saiba o que é e como especificar. Citisystems, 2018. Disponível em: <<https://www.citisystems.com.br/fonte-24v/>>.

SOUZA, V. A. **Princípios De Automação Industrial Com Clp**. [S.l.]: Clube de Autores, 2013.

TUAN, S. T. et al. Improvement of workflow and productivity through application of maynard operation sequence technique (most). In: **Proceedings of the 2014 international conference on industrial engineering and operations management**. [S.l.: s.n.], 2014. p. 7–9.

VIDEOJET. **Product Introduction**. 2022. Disponível em: <<https://www.videojet.br.com/br/homepage/products/laser-marking-systems/fiber-lasers.html>>.

WANG, Y. et al. A real-time energy-optimal trajectory generation method for a servomotor system. **IEEE Transactions on Industrial Electronics**, IEEE, v. 62, n. 2, p. 1175–1188, 2014.

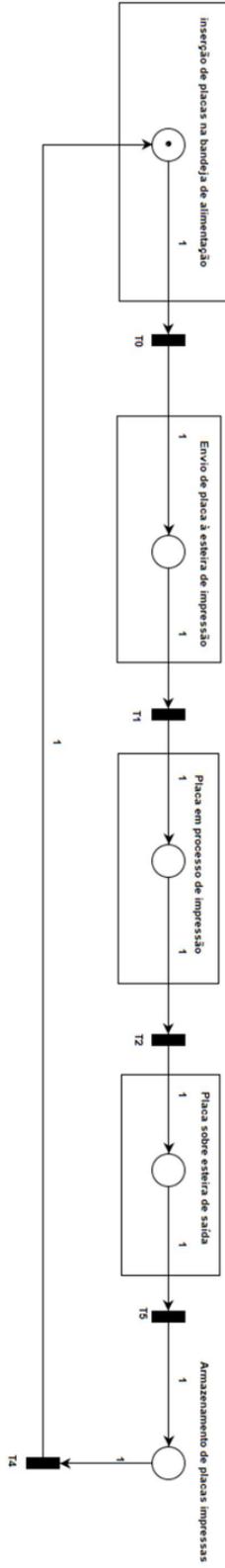
WENDLING, M. Sensores. **Universidade Estadual Paulista. São Paulo**, v. 2010, p. 20, 2010.

ZANCAN, M. D. Controladores programáveis. 2016.

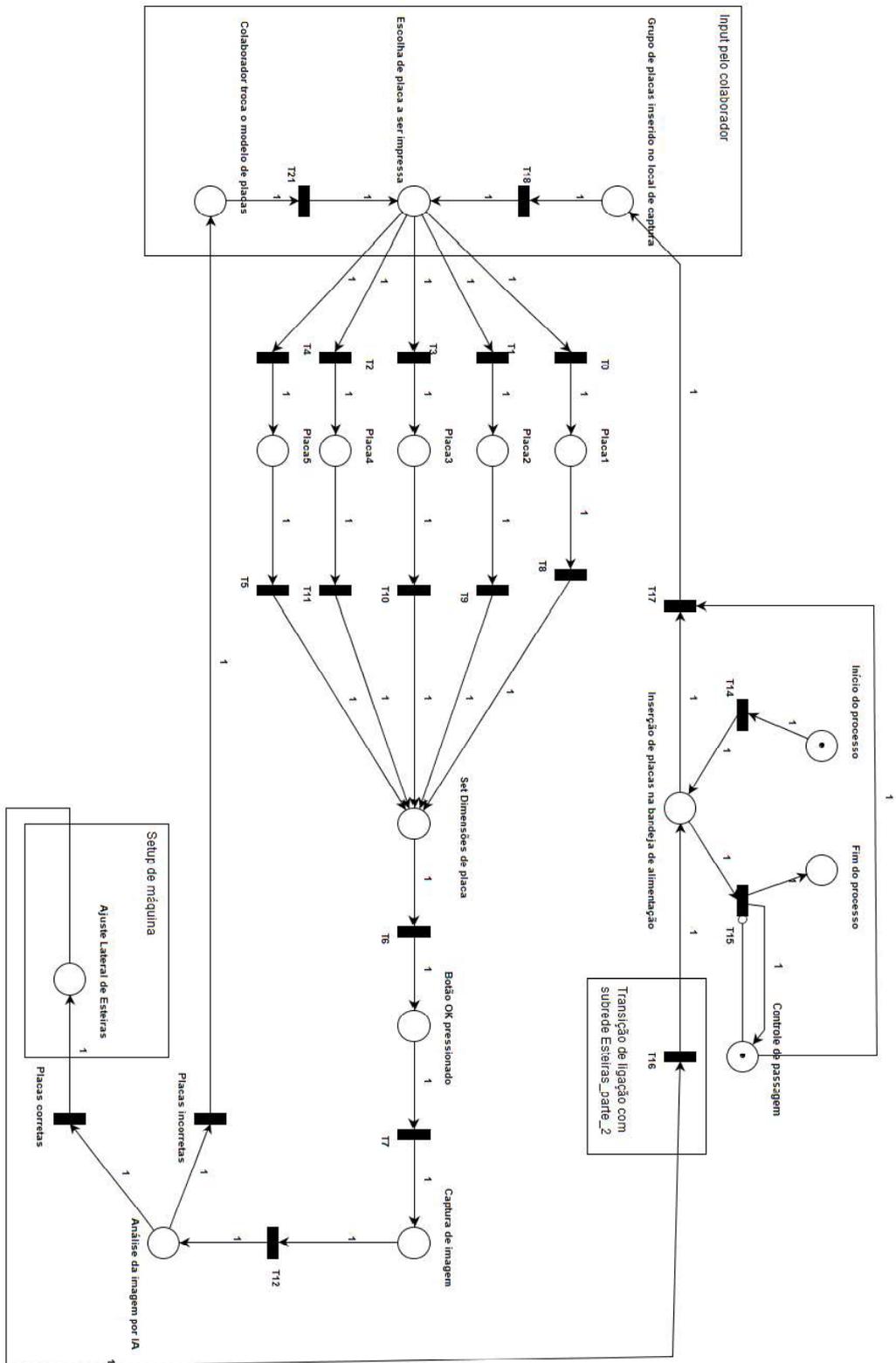
APÊNDICE A

Neste apêndice são apresentados todos os diagramas de modelagem em Redes de Petri construídos para realização do trabalho.

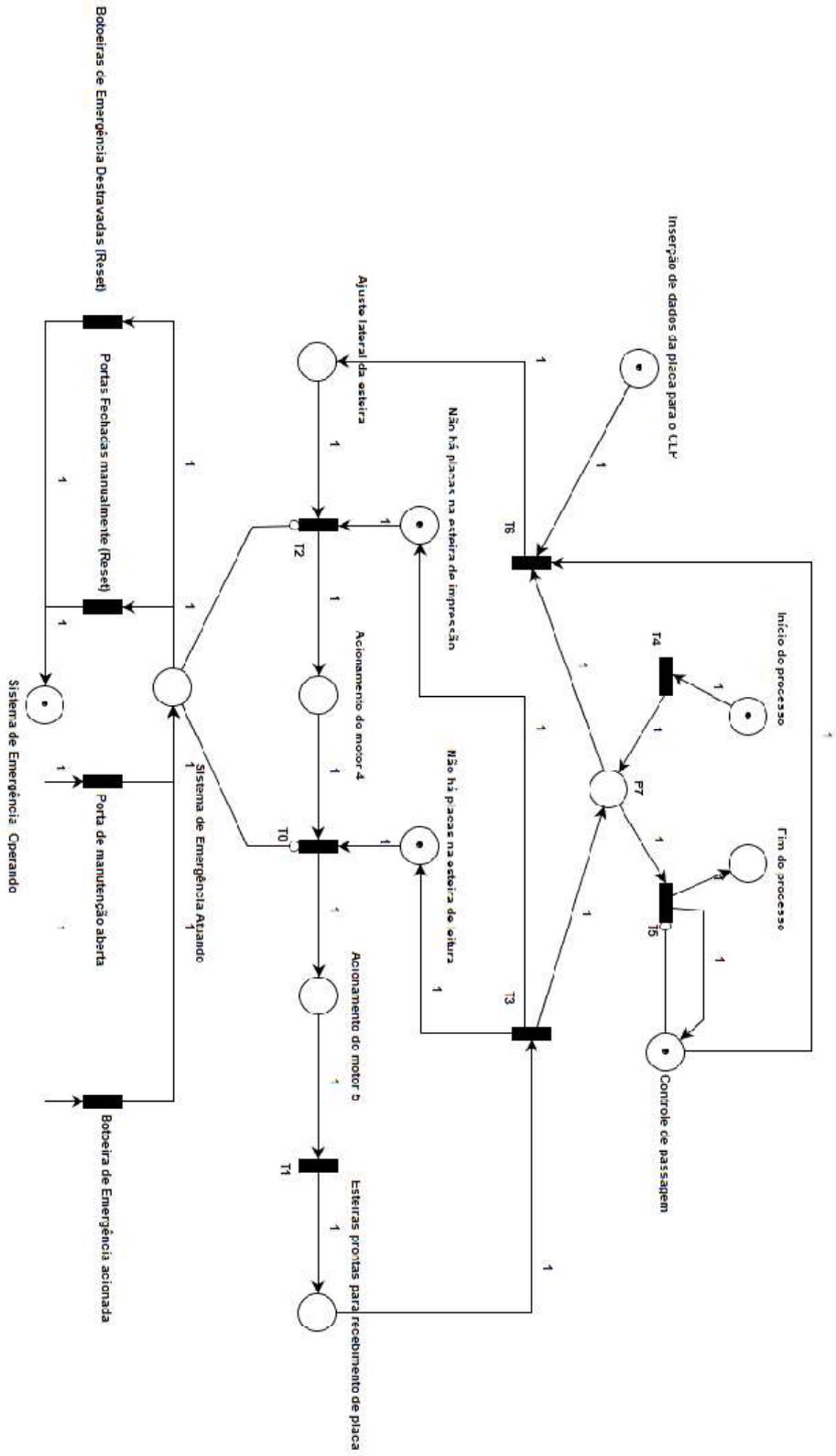
Fluxo Geral



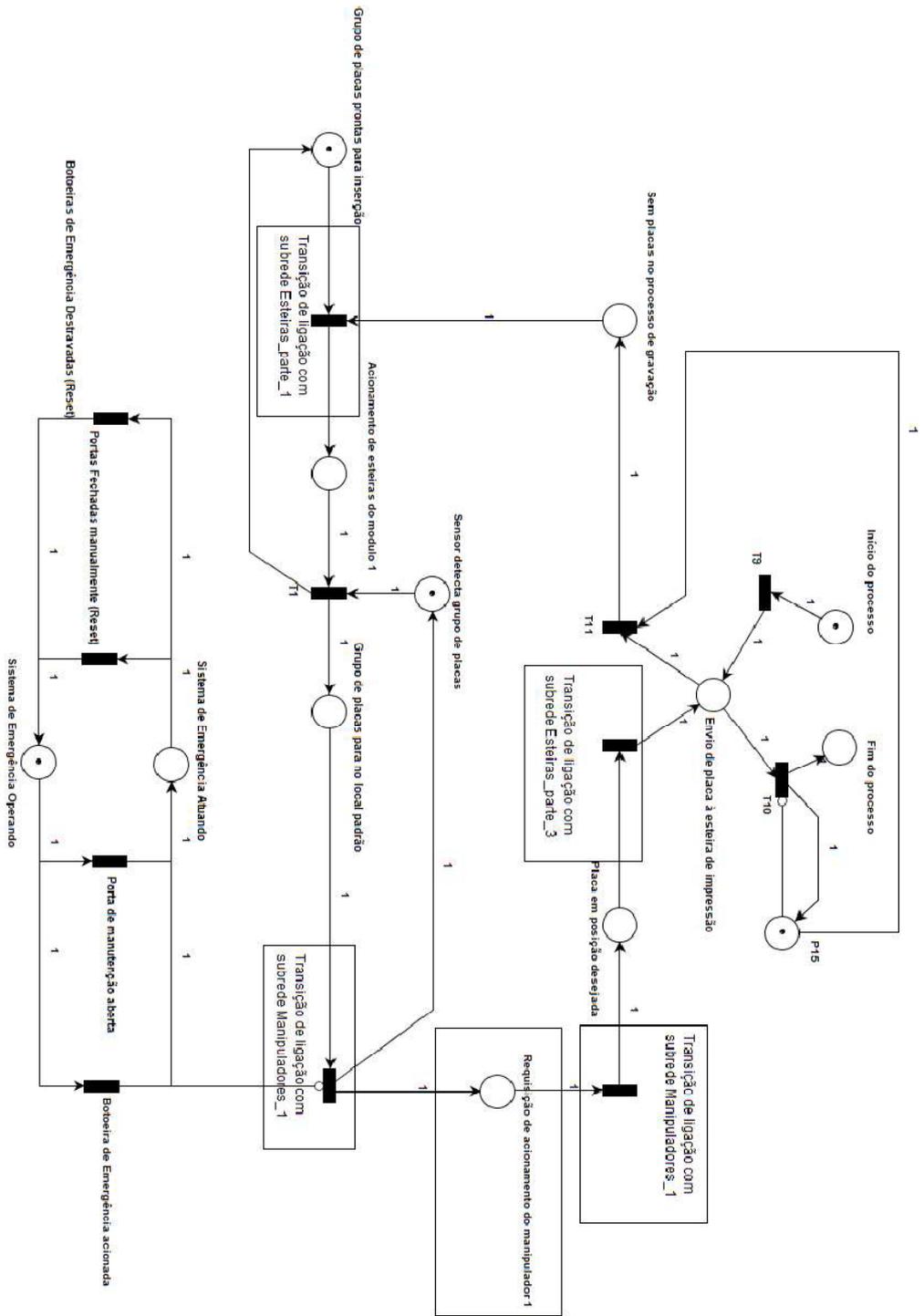
Esteira inicial com inserção das PCBs



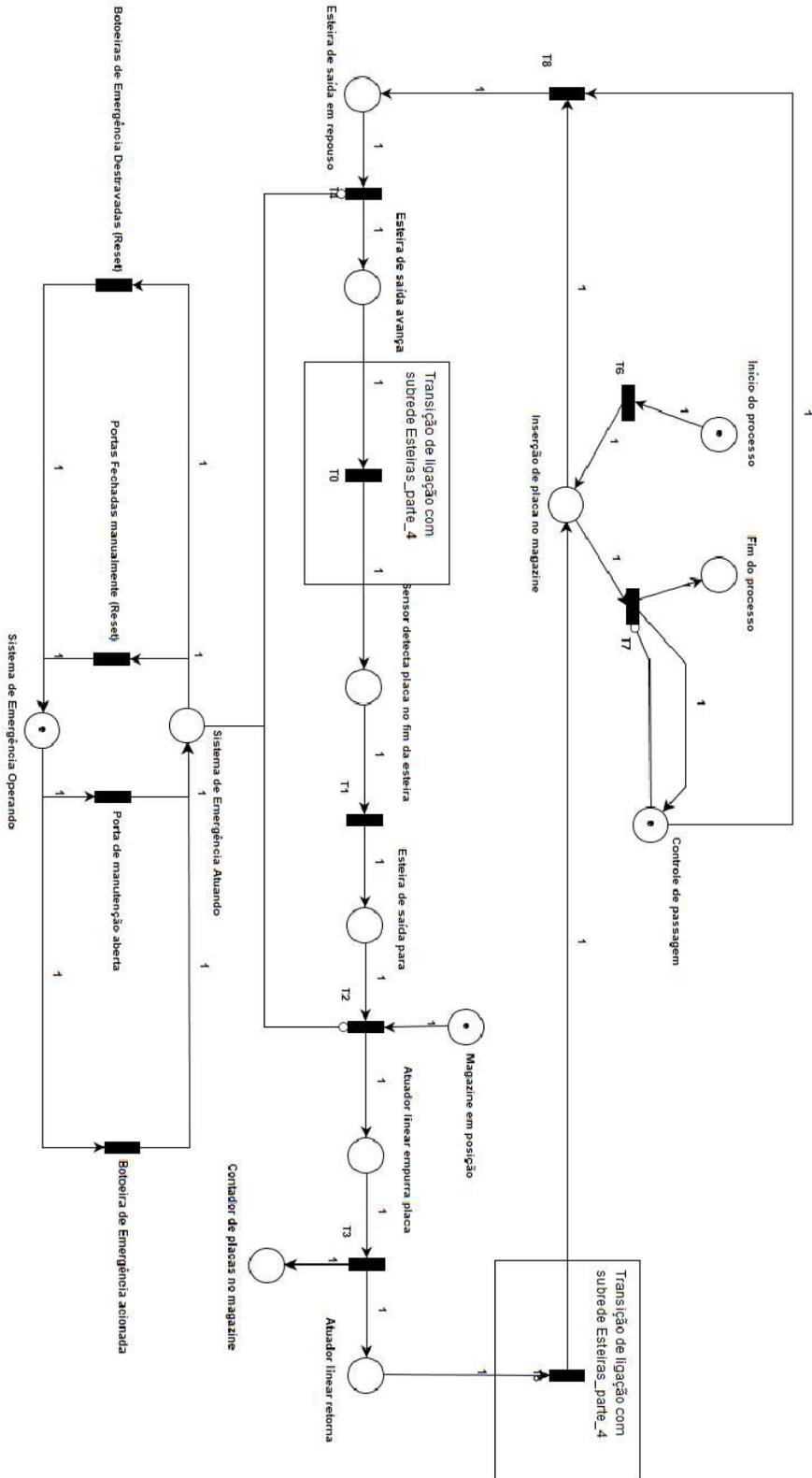
Ajuste lateral, setup automático do sistema



Inserção do grupo de PCBs no processo

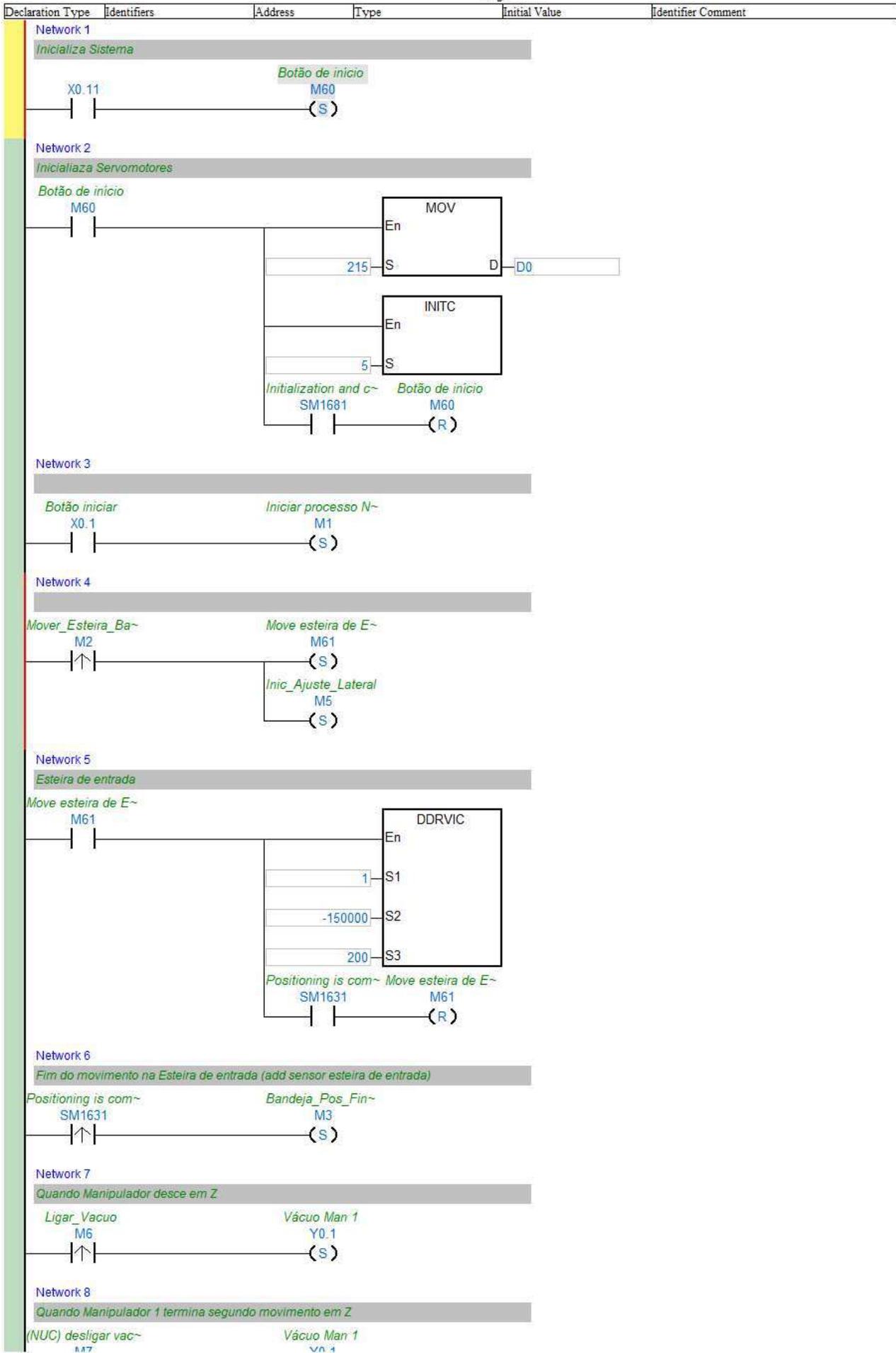


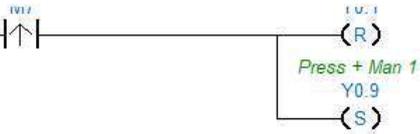
Código aprovado, inserção de PCB no magazine



APÊNDICE B

Neste apêndice estão anexos os códigos em linguagem Ladder embarcados nos CLPs do sistema. A seguir, a programação em Ladder embarcada no CLP responsável pelo comando das esteiras.





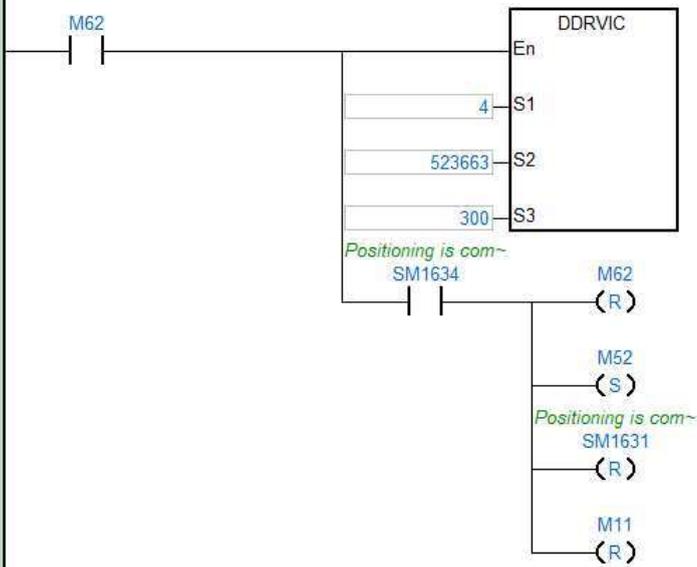
Network 9
Fim do movimento do Manipulador 1



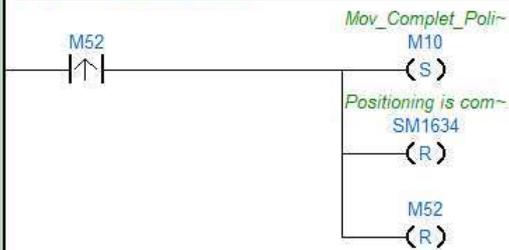
Network 10



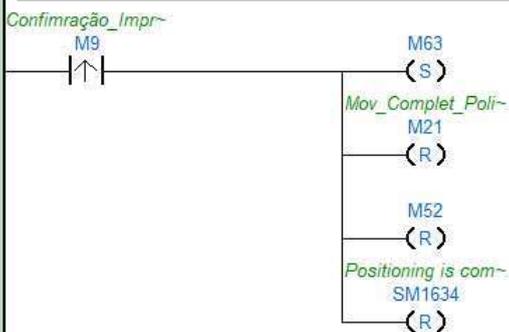
Network 11
Inicio do movimento da esteira de impressão



Network 12
Sinal a enviar para impressora

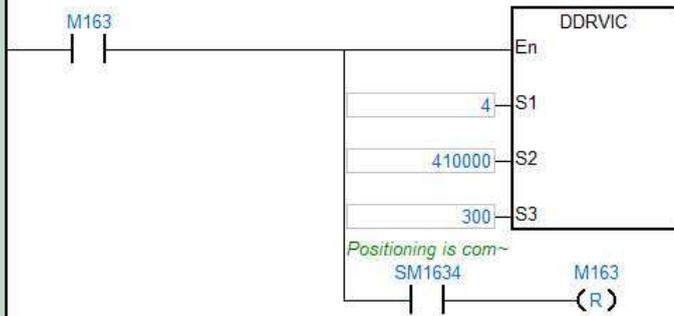


Network 13



Network 14

Completa movimento da esteira de impressão



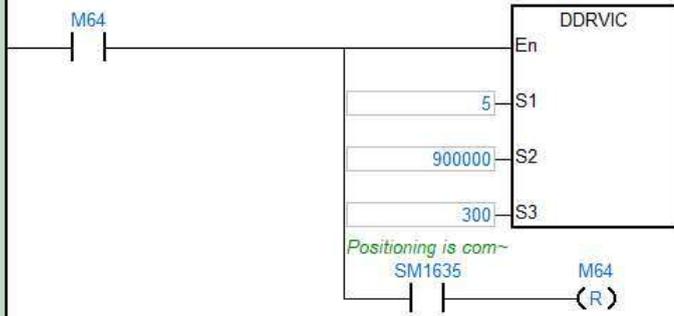
Network 15

Placa detectada no inicio da esteira de saída



Network 16

Movimento da esteira de saída

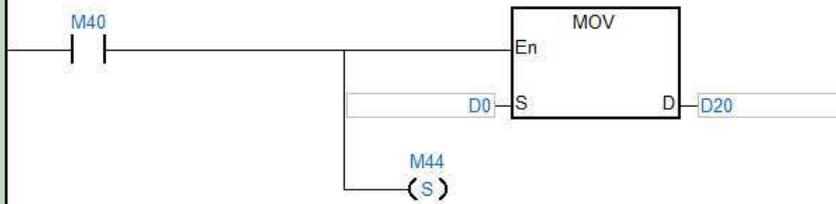


Network 17

Inic_Ajuste_Lateral

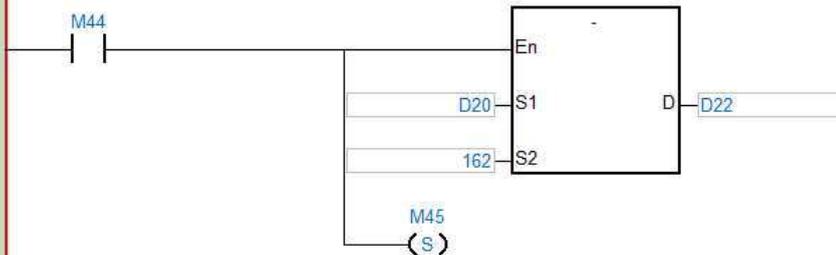


Network 18



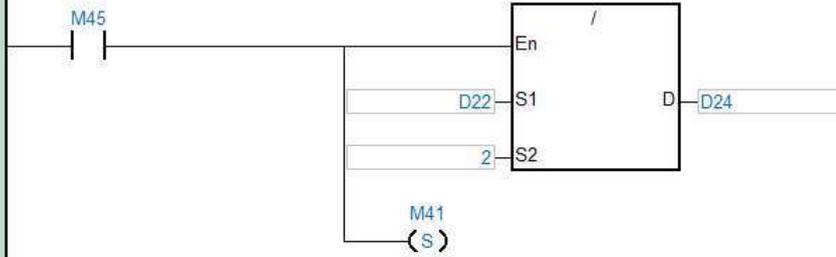
Network 19

Fator de correção

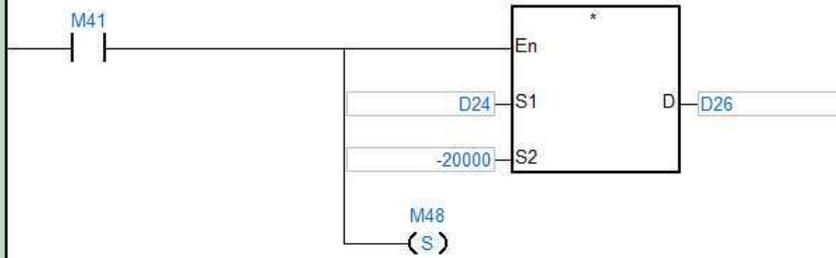


Network 20

Prog0



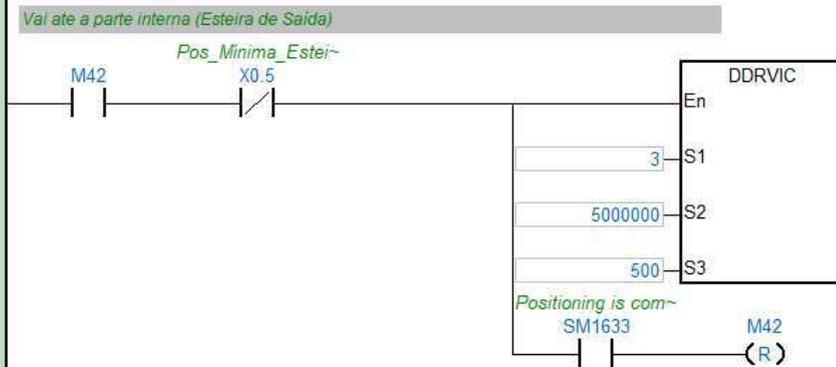
Network 21



Network 22



Network 23



Network 24



Network 25



Network 26

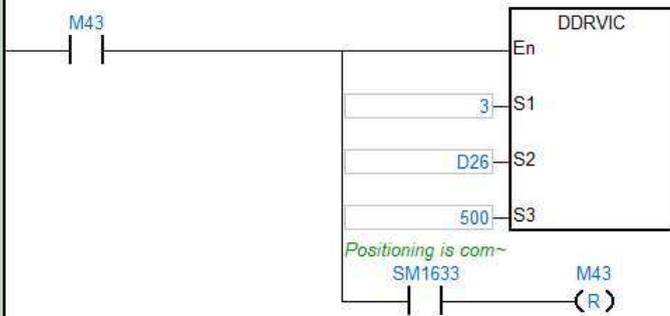


Network 27



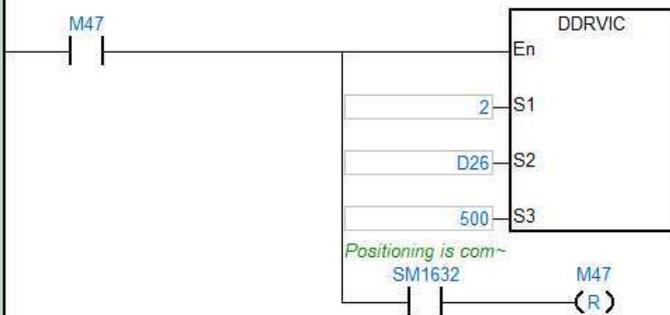
Network 28

Ajusta Esteira de Saída à placa seleccionada



Network 29

Ajusta esteira de Impressão à placa seleccionada



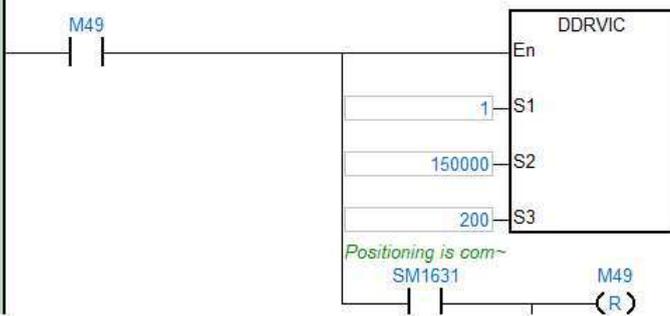
Network 30

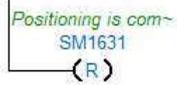
Fim do processo (avisado pelo NUC)



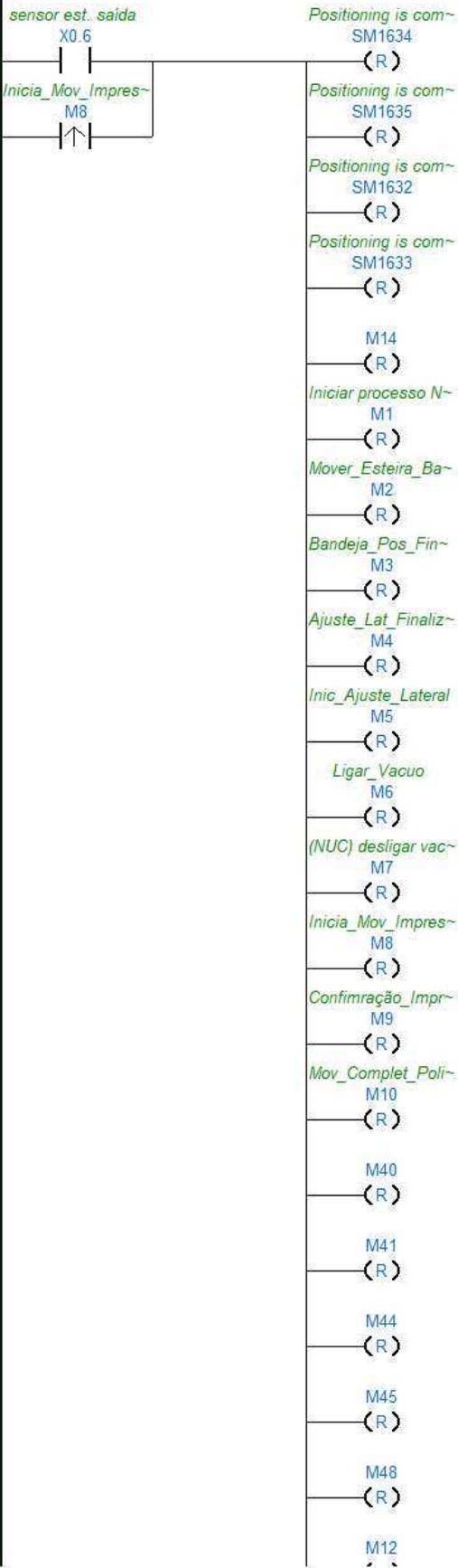
Network 31

Esteira de entrada (retorno)

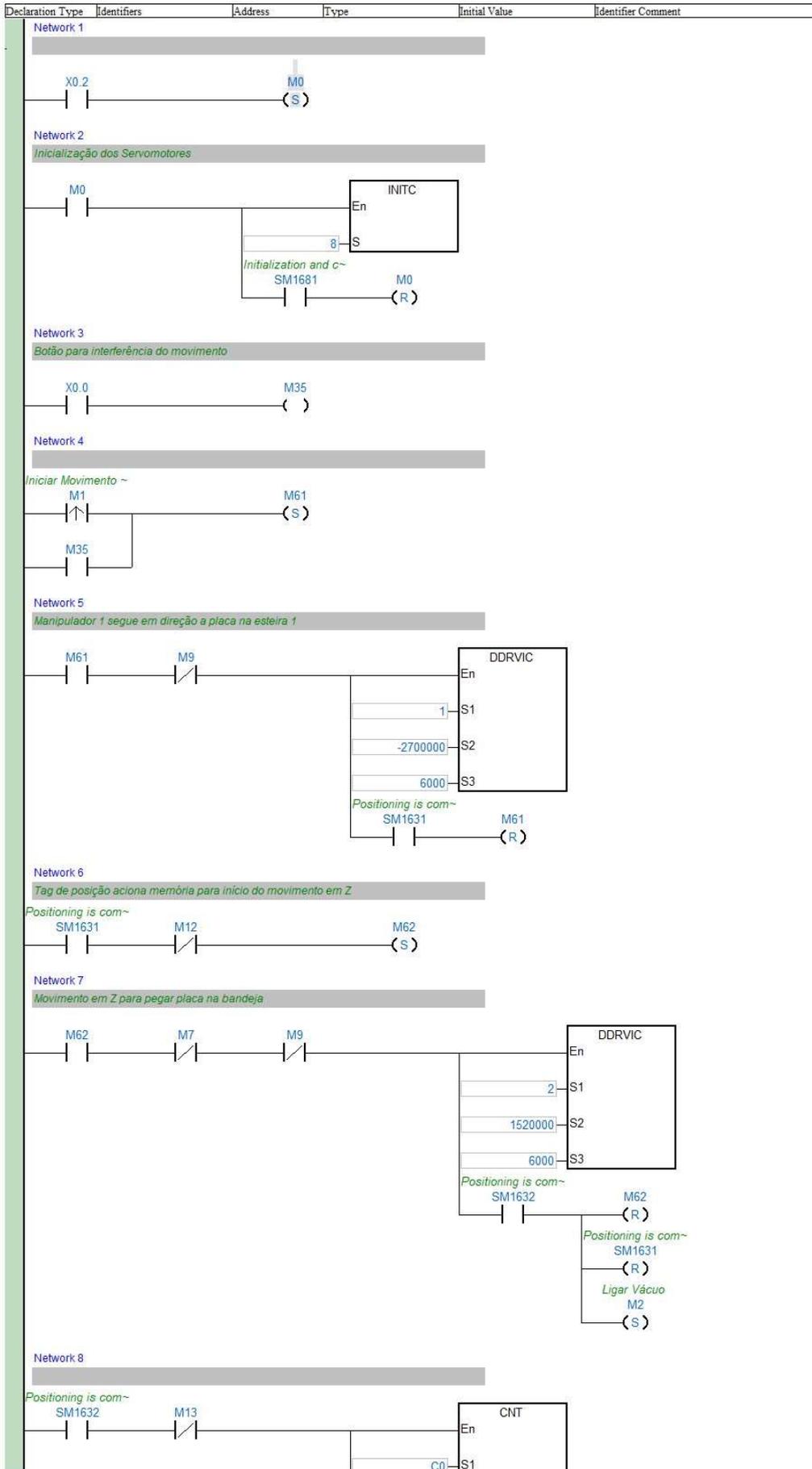




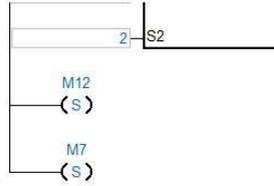
Network 32



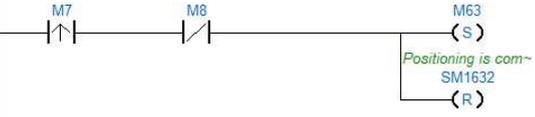
A seguir, a programação em Ladder embarcada no CLP responsável pelo comando dos manipuladores.



Prag0

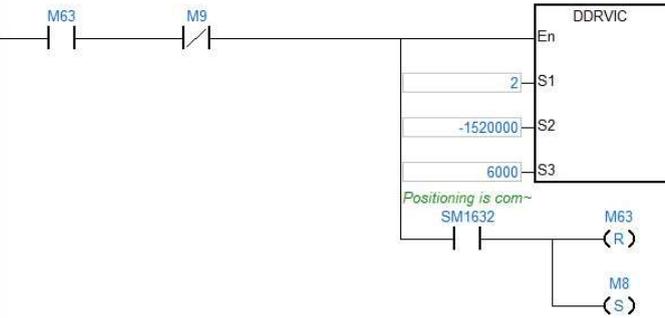


Network 9



Network 10

Movimento em Z subindo com a placa sendo manipulada

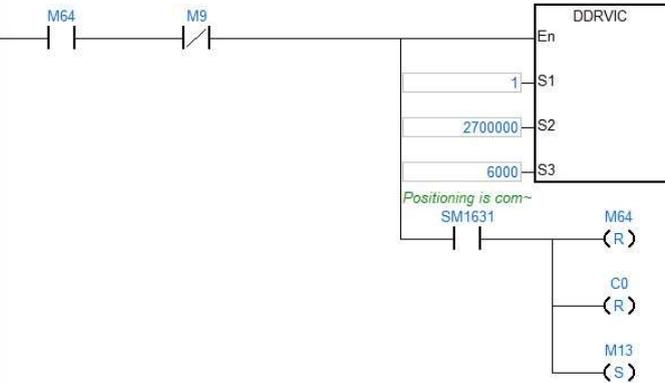


Network 11



Network 12

Movimento em X em direção à esteira de impressão

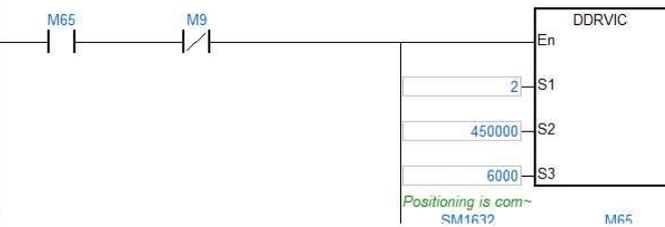


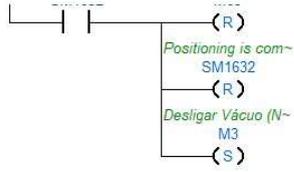
Network 13



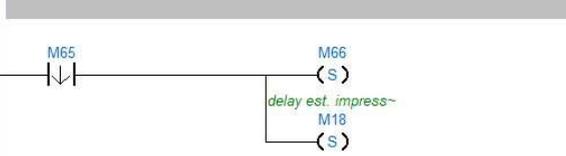
Network 14

Movimento em Z deixando a placa sobre a esteira de impressão





Network 15



Network 16

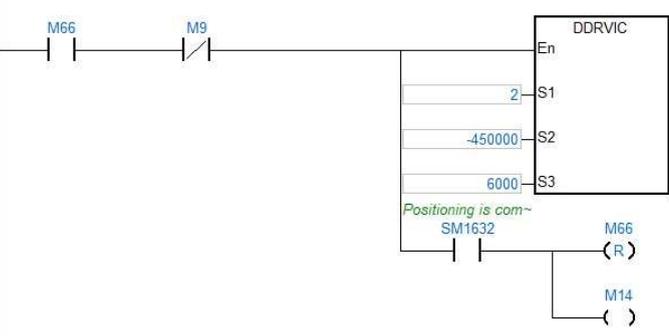


Network 17



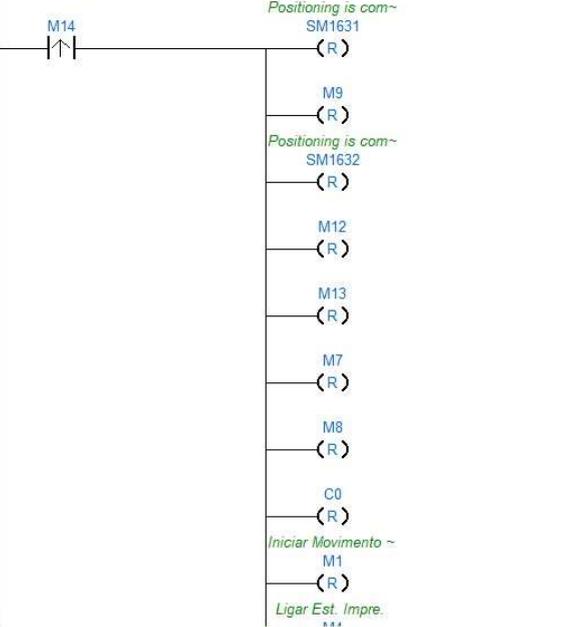
Network 18

Movimento em Z retornando a posição 0

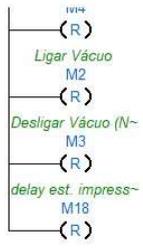


Network 19

Reset de Tags



Prog0



APÊNDICE C

Valores de todos os parâmetros inseridos nos servodrivens relacionados ao controle de posição, resolução, segurança e diversos outros controles possíveis.

ASDA-A2 Servo

Para.	Code	Value	Default	Unit	Description
P0-0	VER	1,054	1,054		Firmware Version
P0-1	ALE	0x0000	0x0000		Alarm Display of the Drive
P0-2	STS	1	1		Drive Status
P0-3	MON	0x0000	0x0000		Analog Monitor Output
P0-4		0x0000	0x0000		Reserved
P0-5		0x0000	0x0000		Reserved
P0-6		0x0000	0x0000		Reserved
P0-7		0x0000	0x0000		Reserved
P0-8	TSON	38	0	Hour	Servo Startup Time
P0-9	CM1	317338	0		Status Monitor Register 1
P0-10	CM2	317338	0		Status Monitor Register 2
P0-11	CM3	317338	0		Status Monitor Register 3
P0-12	CM4	317338	0		Status Monitor Register 4
P0-13	CM5	317338	0		Status Monitor Register 5
P0-14		0	0		Reserved
P0-15		0	0		Reserved
P0-16		0	0		Reserved
P0-17	CM1A	0	0		Display Status Monitor 1
P0-18	CM2A	0	0		Display Status Monitor 2
P0-19	CM3A	0	0		Display Status Monitor 3
P0-20	CM4A	0	0		Display Status Monitor 4
P0-21	CM5A	0	0		Display Status Monitor 5
P0-22		0	0		Reserved
P0-23		0	0		Reserved
P0-24		0	0		Reserved
P0-25	MAP1	0x0000	0x0000		Mapping Parameter #1
P0-26	MAP2	0x0000	0x0000		Mapping Parameter #2
P0-27	MAP3	0x0000	0x0000		Mapping Parameter #3
P0-28	MAP4	0x0000	0x0000		Mapping Parameter #4
P0-29	MAP5	0x0000	0x0000		Mapping Parameter #5
P0-30	MAP6	0x0000	0x0000		Mapping Parameter #6
P0-31	MAP7	0x0000	0x0000		Mapping Parameter #7
P0-32	MAP8	0x0000	0x0000		Mapping Parameter #8
P0-33		0	0		Reserved
P0-34		0	0		Reserved
P0-35	MAP1A	0x0000	0x0000		Target Setting of Mapping Parameter P0-25
P0-36	MAP2A	0x0000	0x0000		Target Setting of Mapping Parameter P0-26
P0-37	MAP3A	0x0000	0x0000		Target Setting of Mapping Parameter P0-27
P0-38	MAP4A	0x0000	0x0000		Target Setting of Mapping Parameter P0-28
P0-39	MAP5A	0x0000	0x0000		Target Setting of Mapping Parameter P0-29
P0-40	MAP6A	0x0000	0x0000		Target Setting of Mapping Parameter P0-30
P0-41	MAP7A	0x0000	0x0000		Target Setting of Mapping Parameter P0-31
P0-42	MAP8A	0x0000	0x0000		Target Setting of Mapping Parameter P0-32
P0-43		0	0		Reserved
P0-44		317338	0		Status Monitor Register (for PC Software)
P0-45		0	0		Status Monitor Register Selection (for PC Software)
P0-46	SVSTS	0x0097	0x0000		Servo Digital Output Status Display

Para.	Code	Value	Default	Unit	Description
P0- 47		0x0000	0x0000		No
P0- 48		0x0000	0x0000		No
P0- 49		0x0000	0x0000		Renew Encoder Absolute Position
P0- 50		0x0000	0x0000		Absolute Coordinate System Status
P0- 51		0	0	rev	Encoder Absolute Position (Multiturn)
P0- 52		0	0	Pulse o	Encoder Absolute Position (Pulse number within Singleturn or PUU)
P0- 53		0x0000	0x0000	ms	General Range Compare Digital Output – Filtering Time
P0- 54		0	0		General Range Compare Digital Output – Lower Limit of 1st Monitoring V
P0- 55		0	0		General Range Compare Digital Output – Upper Limit of 1st Monitoring V
P0- 56		0	0		Reserved
P0- 57		0	0		Reserved
P0- 58		0	0		Reserved
P0- 59		0	0		Reserved
P0- 60		0	0		Reserved
P0- 61		0	0		Reserved
P0- 62		0	0		
P0- 63		0	0		
P1- 0	PTT	0x0002	0x0002		The Type of External Pulse Input
P1- 1	CTL	0x0001	0x0000		Input Setting of Control Mode and Control Command
P1- 2	PSTL	0x0000	0x0000		Speed and Torque Limit Setting
P1- 3	AOUT	0x0000	0x0000		Polarity Setting of Encoder Pulse Output
P1- 4	MON1	100	100	%	MON1 Analog Monitor Output Proportion
P1- 5	MON2	100	100	%	MON2 Analog Monitor Output Proportion
P1- 6	SFLT	0	0	ms	Analog Speed Command (Low-pass Filter)
P1- 7	TFLT	0	0	ms	Analog Torque Command (Low-pass Filter)
P1- 8	PFLT	0	0	10ms	Smooth Constant of Position Command (Low-pass Filter)
P1- 9	SP1	1000	1000	0.1r/mi	Internal Speed Command 1 / Internal Speed Limit 1
P1- 10	SP2	2000	2000	0.1r/mi	Internal Speed Command 2 / Internal Speed Limit 2
P1- 11	SP3	3000	3000	0.1r/mi	Internal Speed Command 3 / Internal Speed Limit 3
P1- 12	TQ1	100	100	%	Internal Torque Command 1 / Internal Torque Limit 1
P1- 13	TQ2	100	100	%	Internal Torque Command 2 / Internal Torque Limit 2
P1- 14	TQ3	100	100	%	Internal Torque Command 3 / Internal Torque Limit 3
P1- 15		0x0000	0x0000		CAPTURE SYNC AXIS – Threshold of Correction
P1- 16		0	0		CAPTURE SYNC AXIS – Offset Compensation
P1- 17		0,000	0,000	ms	Tracking Error Compensation – Additional Time Setting
P1- 18		0,000	0,000	ms	Electronoic Cam (E-Cam) Pulse Phase Compensation – Time Setting
P1- 19		0x0000	0x0000		CAPTURE / COMPARE – Additional Function Settings
P1- 20		0	0		CAPTURE – Masking Range Setting
P1- 21		0	0	Kpps	Electronoic Cam (E-Cam) Pulse Phase Compensation – Min. Frequency
P1- 22		0x0000	0x0000		PR Special Filter Settings
P1- 23		0	0		Compare data offset (Value of P1-23 are remain unchanged)
P1- 24		0	0		COMPARE – Compare data offset(Automatically set P1-24 to 0)
P1- 25	VSF1	1000	1000		Low-frequency Vibration Supression (1)
P1- 26	VSG1	0	0		Low-frequency Vibration Supression Gain (1)
P1- 27	VSF2	1000	1000		Low-frequency Vibration Supression (2)
P1- 28	VSG2	0	0		Low-frequency Vibration Supression Gain (2)
P1- 29	AVSM	0	0		Auto Low-frequency Vibration Supression Setting
P1- 30	VCL	500	500		Low-frequency Vibration Detection
P1- 31		0	0		Reserved
P1- 32	LSTP	0x0000	0x0000		Motor Stop Mode

Para.	Code	Value	Default	Unit	Description
P1- 33		0x0000	0x0000		Reserved
P1- 34	TACC	200	200	ms	Acceleration Constant of S-Curve
P1- 35	TDEC	200	200	ms	Deceleration Constant of S-Curve
P1- 36	TSL	0	0	ms	Accel /Decel Constant of S-curve
P1- 37	GDR	1,0	1,0	1 times	Inertia Ratio and Load Weight Ratio to Servo Motor
P1- 38	ZSPD	10,0	10,0	r/min	Zero Speed Range Setting
P1- 39	SSPD	3000	3000	r/min	Target Speed Detection Level
P1- 40	VCM	3000	Depend	r/min	Maximum Output of Analog Speed Command
P1- 41	TCM	100	100	%	Maximum Output of Analog Torque Command
P1- 42	MBT1	0	0	ms	Enable Delay Time of Brake
P1- 43	MBT2	0	0	ms	Disable Delay Time of Brake
P1- 44	GR1	128	128	pulse	Electronic Gear Ratio (Numerator) (N1)
P1- 45	GR2	10	10	pulse	Electronic Gear Ratio (Denominator) (M)
P1- 46	GR3	2500	2500	pulse	Pulse Number of Encoder Output
P1- 47	SPOK	10	10	r/min	Speed Reached (DO:SP_OK) Range
P1- 48	MCOK	0x0000	0x0000		Operation Selection of Motion Reached (DO:MC_OK)
P1- 49		0	0		Reserved
P1- 50		0	0		Reserved
P1- 51		0	0		Reserved
P1- 52	RES1	40	Depend	Ohm	Regenerative Resistor Value
P1- 53	RES2	40	Depend	Watt	Regenerative Resistor Capacity
P1- 54	PER	12800	12800	pulse	Position Completed Range
P1- 55	MSPD	3000	Depend	r/min	Maximum Speed Limit
P1- 56	OVW	120	120	%	Output Overload Warning Level
P1- 57	CRSHA	0	0	%	Motor Crash Protection (torque percentage)
P1- 58	CRSHT	1	1	ms	Motor Crash Protection Time
P1- 59	MFLT	0,0	0,0	ms	Analog Speed Command
P1- 60		0	0		Reserved
P1- 61		0	0		Reserved
P1- 62	FRCL	0	0	%	Friction Compensation
P1- 63	FRCT	1	1	ms	Friction Compensation Smooth Constant
P1- 64		0x0000	0x0000		Analog Position Command: Activation Control
P1- 65		1	1		Smooth Constant of Analog Position Command
P1- 66	PCM	0,0	0,0	0.1r	Max. Rotation Number of Analog Position Command
P1- 67		0	0		Reserved
P1- 68	PFLT2	4	4	ms	Position Command Moving Average Filter
P1- 69		0	0	Hz	Reserved
P1- 70		0	0	dB	Reserved
P1- 71		0	0		Reserved
P1- 72	FRES	5000	5000	pulse/r	Resolution of Linear Scale for Full-closed Loop Control
P1- 73	FERR	30000	30000	pulse	Error Protection Range for Full-closed Loop Control
P1- 74	FCON	0x0000	0x0000		Full-closed Loop Control of Linear Scale
P1- 75	FELP	100	100	ms	Low-pass Filter Time Constant of Full-closed Loop Control
P1- 76	AMSPD	5500	5500	r/min	Max. Rotation of Encoder Output Setting (OA, OB)
P1- 77		0x0000	0x0000		Reserved
P1- 78		0x0000	0x0000		Reserved
P1- 79		0	0		Reserved
P1- 80		0	0		Reserved
P1- 81		3000	Depend	rpm/10	Max. Speed of 2nd Analog Speed Command
P1- 82		0	0	ms	Filter Switching Time Between P1-40 and P1-81

Para.	Code	Value	Default	Unit	Description
P1-83		0	0	mV	Abnormal Analog Input Voltage Level
P1-84		0	0		
P1-85		0	0		
P1-86		0	0		
P1-87		1	1		
P1-88		2000	2000		
P1-89		0	0		
P1-90		0	0		
P1-91		0	0		
P1-92		0	0		
P1-93		0	0		
P1-94		0	0		
P1-95		0	0		
P1-96		0	0		
P1-97		0	0		
P1-98		0	0		
P2-0	KPP	35	35	rad/s	Position Loop Gain
P2-1	PPR	100	100	%	Switching Rate of Position Loop Gain
P2-2	PFG	50	50	%	Position Feed Forward Gain
P2-3	PFF	5	5	ms	Smooth Constant of Position Feed Forward Gain
P2-4	KVP	500	500	rad/s	Speed Loop Gain
P2-5	SPR	100	100	%	Switching Rate of Speed Loop Gain
P2-6	KVI	100	100	rad/s	Speed Integral Compensation
P2-7	KVF	0	0	%	Speed Feed Forward Gain
P2-8	PCTL	36	0		Special Parameter Write-in
P2-9	DRT	2	2	2ms	DI Debouncing Time
P2-10	DI1	0x0001	0x0101		DI1 Functional Planning
P2-11	DI2	0x0146	0x0104		DI2 Functional Planning
P2-12	DI3	0x0000	0x0116		DI3 Functional Planning
P2-13	DI4	0x0117	0x0117		DI4 Functional Planning
P2-14	DI5	0x0102	0x0102		DI5 Functional Planning
P2-15	DI6	0x0122	0x0022		DI6 Functional Planning
P2-16	DI7	0x0123	0x0023		DI7 Functional Planning
P2-17	DI8	0x0121	0x0021		DI8 Functional Planning
P2-18	DO1	0x0101	0x0101		DO1 Functional Planning
P2-19	DO2	0x0103	0x0103		DO2 Functional Planning
P2-20	DO3	0x0109	0x0109		DO3 Functional Planning
P2-21	DO4	0x0105	0x0105		DO4 Functional Planning
P2-22	DO5	0x0007	0x0007		DO5 Functional Planning
P2-23	NCF1	1000	1000	Hz	Resonance Suppression (Notch Filter) (1)
P2-24	DPH1	0	0	dB	Resonance Suppression (Notch Filter) Attenuation Rate (1)
P2-25	NLP	0,2	Depend	0.1ms	Low-pass Filter of Resonance Suppression
P2-26	DST	0	0		Anti-interference Gain
P2-27	GCC	0x0000	0x0000		Gain Switching and Switching Selection
P2-28	GUT	10	10	10 ms	Gain Switching Time Constant
P2-29	GPE	128000	128000		Gain Switching
P2-30	INH	-5	0		Auxiliary Function
P2-31	AUT1	40	40	Hz	Speed Loop Frequency Response Setting in Auto and Semi-Auto Mode
P2-32	AUT2	0x0000	0x0000		Tuning Mode Selection
P2-33	AUT3	0x0000	0x0000		Semi-Auto Mode Inertia Adjustment

Para.	Code	Value	Default	Unit	Description
P2- 34	SDEV	5000	5000	r/min	The Condition of Overspeed Warning
P2- 35	PDEV	384000	384000	pulse	Condition of Excessive Position Control Deviation Warning
P2- 36	EDI9	0x0100	0x0100		Extended EDI9 Functional Planning
P2- 37	EDI10	0x0100	0x0100		Extended EDI10 Functional Planning
P2- 38	EDI11	0x0100	0x0100		Extended EDI11 Functional Planning
P2- 39	EDI12	0x0100	0x0100		Extended EDI12 Functional Planning
P2- 40	EDI13	0x0100	0x0100		Extended EDI13 Functional Planning
P2- 41	EDI14	0x0100	0x0100		Extended EDI14 Functional Planning
P2- 42		0	0		Reserved
P2- 43	NCF2	1000	1000	Hz	Resonance Suppression (Notch Filter) (2)
P2- 44	DPH2	0	0	dB	Resonance Suppression (Notch Filter) Attenuation Rate (2)
P2- 45	NCF3	1000	1000	Hz	Resonance Suppression (Notch Filter) (3)
P2- 46	DPH3	0	0	dB	Resonance Suppression (Notch Filter) Attenuation Rate (3)
P2- 47	ANCF	1	1		Auto Resonance Suppression Mode Setting
P2- 48	ANCL	100	100		Auto Resonance Suppression Detection Level
P2- 49	SJIT	0x000B	0x000B	sec	Speed Detection Filter
P2- 50	DCLR	0x0000	0x0000		Pulse Clear Mode
P2- 51		0	0		Reserved
P2- 52		100000	100000	PUU	Indexing Coordinates Scale
P2- 53	KPI	0	0	Rad/s	Position Integral Compensation
P2- 54		0	0	Rad/s	The Gain of Synchronous Speed Control
P2- 55		0	0	Rad/s	Integral Compensation to Synchronous Speed
P2- 56		0	0	Rad	Integral Compensation to Synchronous Position
P2- 57		0	0	Hz	The Bandwidth of Synchronous Control
P2- 58		0	0	0.1ms	Low-pass Filter of Synchronous Speed Error
P2- 59		0	0		Reserved
P2- 60	GR4	128	128	pulse	Electronic Gear Ratio (Numerator) (N2)
P2- 61	GR5	128	128	pulse	Electronic Gear Ratio (Numerator) (N3)
P2- 62	GR6	128	128	pulse	Electronic Gear Ratio (Numerator) (N4)
P2- 63		0	0		Reserved
P2- 64		0x0000	0x0000		Reserved
P2- 65	GBIT	0x0008	0x0008		Special-bit Register
P2- 66	GBIT2	0x0000	0x0000		Special-bit Register 2
P2- 67	JSL	1,5	1,5	1 times	The Stable Level of Inertia Estimation
P2- 68		0x0000	0x0000		Switch of Following Error Compensation
P2- 69		0x0000	0x0000		Absolute Encoder Setting
P2- 70		0x0000	0x0000		Read Data Format Selection
P2- 71		0x0000	0x0000		ABS Position Homing
P2- 72		0	0		Reserved
P2- 73		0x0000	0x0000		E-CAM Alignment – Operation Setting
P2- 74		0,000	0,000	ms	E-CAM Alignment – DI Delay Time Compensation
P2- 75		0	0		E-CAM Alignment – Alignment Target Position
P2- 76		0x0000	0x0000		E-CAM Alignment – Source Setting
P2- 77		0x0000	0x0000		E-CAM Master Axis – Pulse Masking Setting
P2- 78		270	270	degree	E-CAM: Area Number #2 + (Polarity is Positive)
P2- 79		360	360		E-CAM: Area Number #2 + (Polarity is Negative)
P2- 80		0	0		Z Phase Source of Homing
P2- 81		0	0		
P2- 82		0	0		
P2- 83		0	0		

Para.	Code	Value	Default	Unit	Description
P2- 84		0	0		
P2- 85		0	0		
P2- 86		0	0		
P2- 87		0	0		
P2- 88		0	0		
P2- 89		0	0		
P2- 90		0	0		
P2- 91		0	0		
P2- 92		0	0		
P2- 93		0	0		
P2- 94		0x0090	0x0090		
P2- 95		0	0		
P2- 96		300000	300000		
P3- 0	ADR	0x0001	0x007F		Communication Address Setting
P3- 1	BRT	0x0400	0x0203	bps	Transmission Speed
P3- 2	PTL	0x0006	0x0006		Communication Protocol
P3- 3	FLT	0x0000	0x0000		Communication Error Disposal
P3- 4	CWD	0	0	sec	Communication Time Out
P3- 5	CMM	0x0000	0x0000		Communication Mechanism
P3- 6	SDI	0x0000	0x0000		Control Switch of Digital Input (DI)
P3- 7	CDT	0	0	0.5ms	Communication Response Delay Time
P3- 8	MNS	0x0000	0x0000		Monitor Mode
P3- 9	SYC	0x5055	0x5055		CANopen Synchronize Setting
P3- 10		0x0000	0x0000		Reserved
P3- 11		0x0000	0x0000		Reserved
P3- 12		0x0000	0x0000		
P3- 13		0x0000	0x0000		
P4- 0	ASH1	0x3130	0x0000		Fault Record (N)
P4- 1	ASH2	0x3130	0x0000		Fault Record (N-1)
P4- 2	ASH3	0x3130	0x0000		Fault Record (N-2)
P4- 3	ASH4	0x3130	0x0000		Fault Record (N-3)
P4- 4	ASH5	0x3130	0x0000		Fault Record (N-4)
P4- 5	JOG	100	20	r/min	JOG Operation
P4- 6	FOT	0x0000	0x0000		Digital Output Register (Readable and Writable)
P4- 7	ITST	0x0005	0x0000		Multi-funciton of Digital Input
P4- 8	PKEY	0x0000	0x0000		Input Status of the Drive Keypad (Read-only)
P4- 9	MOT	0x001B	0x0000		Digital Output Status
P4- 10	CEN	0	0		Adjustment Selection
P4- 11	SOF1	16765	16352		Analog Speed Input Offset Adjustment 1
P4- 12	SOF2	16839	16352		Analog Speed Input Offset Adjustment 2
P4- 13	TOF1	16840	16352		Analog Torque Input Offset Adjustment 1
P4- 14	TOF2	16840	16352		Analog Torque Input Offset Adjustment 2
P4- 15	COF1	16630	16352		Current Detector (V1 phase) offset Adjustment
P4- 16	COF2	16809	16352		Current Detector (V2 phase) Offset Adjustment
P4- 17	COF3	17135	16352		Current Detector (W1 phase) Offset Adjustment
P4- 18	COF4	16844	16352		Current Detector (W2 phase) Offset Adjustment
P4- 19	TIGB	4	2		IGBT NTC Adjustment Deteciton Level (cannot reset)
P4- 20	DOF1	0	0	mV	Offset Adjustment Value of Analog Monitor (Ch1)
P4- 21	DOF2	0	0	mV	Offset Adjustment Value of Analog Monitor (Ch2)
P4- 22	SAO	0	0	mV	Analog Speed Input OFFSET

Para.	Code	Value	Default	Unit	Description
P4-23	TAO	0	0	mV	Analog Torque Input OFFSET
P4-24	LVL	160	160	V	Level of Undervoltage Error
P5-0	MVER	113006	0		Reserved
P5-1		0x0001	0x0000		Reserved
P5-2		0x1000	0x0000		Reserved
P5-3	PDEC	0xEEEF	0xEEEF		Deceleration Time of Auto Protection
P5-4	HMOV	0x0000	0x0000		Homing Mode
P5-5	HSPD1	100,0	100,0	r/min	1st Speed Setting of High Speed Homing
P5-6	HSPD2	20,0	20,0	r/min	2nd Speed Setting of Low Speed Homing
P5-7	PRCM	0	0		Trigger Position Command (PR mode only)
P5-8	SWLP	214748	214748	PUU	Forward Software Limit
P5-9	SWLN	-214748	-214748	PUU	Reverse Software Limit
P5-10	AYSZ	800	0		Data Array - Data Size
P5-11	AYID	0	0		Data Array: Address of Reading/Writing
P5-12	AYD0	0	0		Data Array - Window#1 for Reading/Writing
P5-13	AYD1	0	0		Data Array - Window#2 for Reading/Writing
P5-14		0	0		Reserved
P5-15	PMEM	0x0000	0x0000		PATH#1 ~ PATH#2 No Data Retained Setting
P5-16	AXEN	317338	0	PUU	Axis Position - Motor Encoder
P5-17	AXAU	0	0		Axis Position - Auxiliary Encoder
P5-18	AXPC	0	0		Axis Position - Pulse Command
P5-19	TBS	1,00000	100000		E-Cam Curve Scaling
P5-20	AC0	200	200	ms	Acceleration / Deceleration Time (Number#0)
P5-21	AC1	300	300	ms	Acceleration / Deceleration Time (Number#1)
P5-22	AC2	500	500	ms	Acceleration / Deceleration Time (Number#2)
P5-23	AC3	600	600	ms	Acceleration / Deceleration Time (Number#3)
P5-24	AC4	800	800	ms	Acceleration / Deceleration Time (Number#4)
P5-25	AC5	900	900	ms	Acceleration / Deceleration Time (Number#5)
P5-26	AC6	1000	1000	ms	Acceleration / Deceleration Time (Number#6)
P5-27	AC7	1200	1200	ms	Acceleration / Deceleration Time (Number#7)
P5-28	AC8	1500	1500	ms	Acceleration / Deceleration Time (Number#8)
P5-29	AC9	2000	2000	ms	Acceleration / Deceleration Time (Number#9)
P5-30	AC10	2500	2500	ms	Acceleration / Deceleration Time (Number#10)
P5-31	AC11	3000	3000	ms	Acceleration / Deceleration Time (Number#11)
P5-32	AC12	5000	5000	ms	Acceleration / Deceleration Time (Number#12)
P5-33	AC13	8000	8000	ms	Acceleration / Deceleration Time (Number#13)
P5-34	AC14	50	50	ms	Acceleration / Deceleration Time (Number#14)
P5-35	AC15	30	30	ms	Acceleration / Deceleration Time (Number#15)
P5-36	CAST	0	0		CAPTURE-Start Address of Data Array
P5-37	CAAX	-1	0		CAPTURE-Axis Position CNT
P5-38	CANO	1	1		CAPTURE-The Number of Capturing Times
P5-39	CACT	0x2010	0x2010		CAPTURE-Activate CAP Control
P5-40	DLY0	0	0	ms	Delay Time After Position Completed (Number#0)
P5-41	DLY1	100	100	ms	Delay Time After Position Completed (Number#1)
P5-42	DLY2	200	200	ms	Delay Time After Position Completed (Number#2)
P5-43	DLY3	400	400	ms	Delay Time After Position Completed (Number#3)
P5-44	DLY4	500	500	ms	Delay Time After Position Completed (Number#4)
P5-45	DLY5	800	800	ms	Delay Time After Position Completed (Number#5)
P5-46	DLY6	1000	1000	ms	Delay Time After Position Completed (Number#6)
P5-47	DLY7	1500	1500	ms	Delay Time After Position Completed (Number#7)

Para.	Code	Value	Default	Unit	Description
P5- 48	DLY8	2000	2000	ms	Delay Time After Position Completed (Number#8)
P5- 49	DLY9	2500	2500	ms	Delay Time After Position Completed (Number#9)
P5- 50	DLY10	3000	3000	ms	Delay Time After Position Completed (Number#10)
P5- 51	DLY11	3500	3500	ms	Delay Time After Position Completed (Number#11)
P5- 52	DLY12	4000	4000	ms	Delay Time After Position Completed (Number#12)
P5- 53	DLY13	4500	4500	ms	Delay Time After Position Completed (Number#13)
P5- 54	DLY14	5000	5000	ms	Delay Time After Position Completed (Number#14)
P5- 55	DLY15	5500	5500	ms	Delay Time After Position Completed (Number#15)
P5- 56	CMST	50	50		COMPARE-Start Address of Data Array
P5- 57	CMAX	-2	0		COMPARE-Axis Position
P5- 58	CMNO	1	1		COMPARE-Compare Amount
P5- 59	CMCT	0x0064	0x0064		COMPARE-Activate CMP Control
P5- 60	POV0	20,0	20,0	r/min	Target Speed Setting #0
P5- 61	POV1	50,0	50,0	r/min	Target Speed Setting #1
P5- 62	POV2	100,0	100,0	r/min	Target Speed Setting #2
P5- 63	POV3	200,0	200,0	r/min	Target Speed Setting #3
P5- 64	POV4	300,0	300,0	r/min	Target Speed Setting #4
P5- 65	POV5	500,0	500,0	r/min	Target Speed Setting #5
P5- 66	POV6	600,0	600,0	r/min	Target Speed Setting #6
P5- 67	POV7	800,0	800,0	r/min	Target Speed Setting #7
P5- 68	POV8	1000,0	1000,0	r/min	Target Speed Setting #8
P5- 69	POV9	1300,0	1300,0	r/min	Target Speed Setting #9
P5- 70	POV10	1500,0	1500,0	r/min	Target Speed Setting #10
P5- 71	POV11	1800,0	1800,0	r/min	Target Speed Setting #11
P5- 72	POV12	2000,0	2000,0	r/min	Target Speed Setting #12
P5- 73	POV13	2300,0	2300,0	r/min	Target Speed Setting #13
P5- 74	POV14	2500,0	2500,0	r/min	Target Speed Setting #14
P5- 75	POV15	3000,0	3000,0	r/min	Target Speed Setting #15
P5- 76	CPRS	0	0		CAPTURE - First Position Reset Data
P5- 77	CSAX	0	0		The Position of Synchronous Capture Axis (CAP SYNC AXES)
P5- 78	CSDS	100	100	pulse	The Interval Pulse Number between Each Synchronous Capture Axis
P5- 79	CSDS	0	0	pulse	Error Pulse Number of Synchronous Capture Axis
P5- 80	CSDS	10	10	%	Max. Correction Rate of Synchronous Capture Axis
P5- 81	ECHD	100	100		E-CAM: Start Address of Data Array
P5- 82	ECMN	5	5		E-Cam: E-Cam Area Number N (at least >=5)
P5- 83	ECMM	1	1		E-Cam: Master Gear Ratio Setting M
P5- 84	ECMP	3600	3600		E-Cam: Master Gear Ratio Setting P
P5- 85	ECME	0	0		E-Cam: Number of Area
P5- 86	ECAX	-1	0		E-Cam: Master Axis Position
P5- 87	PLED	0	0		E-Cam: Lead Pulse
P5- 88	ECON	0x0000	0x0000		E-Cam: Activate E-Cam Control
P5- 89	ECRD	0	0		E-Cam: Information of Disengaging Time
P5- 90	CMAP	270	270	deg.	E-Cam: Area No. + The Point of DO ON
P5- 91	CMAN	360	360	deg.	E-Cam: Area No. - The Point of DO OFF
P5- 92	PLED	0	0		E-Cam: Pre-engaged Time of Each Cycle
P5- 93	CSDS	0x0000	0x0000		Motion Control Macro Command: Command Parameter #4
P5- 94	CSDS	0	0		Motion Control Macro Command: Command Parameter #3
P5- 95	CSDS	0	0		Motion Control Macro Command: Command Parameter #2
P5- 96	CSDS	0	0		Motion Control Macro Command: Command Parameter #1
P5- 97	CSDS	0x0000	0x0000		Motion Control Macro Command: Issue Command / Executing Result

Para.	Code	Value	Default	Unit	Description
P5- 98	EVON	0x0000	0x0000		PR# Triggered by Event Rising-Edge
P5- 99	EVOF	0x0000	0x0000		PR# Triggered by Event Falling-Edge
P6- 0	ODEF	0x0000	0x0000		Homing Definition
P6- 1	ODAT	0	0		Origin Definition
P6- 2	PDEF1	0x0000	0x0000		PATH#1 Definition
P6- 3	PDAT1	0	0		PATH#1 Data
P6- 4	PDEF2	0x0000	0x0000		PATH#2 Definition
P6- 5	PDAT2	0	0		PATH#2 Data
P6- 6	PDEF3	0x0000	0x0000		PATH#3 Definition
P6- 7	PDAT3	0	0		PATH#3 Data
P6- 8	PDEF4	0x0000	0x0000		PATH#4 Definition
P6- 9	PDAT4	0	0		PATH#4 Data
P6- 10	PDEF5	0x0000	0x0000		PATH#5 Definition
P6- 11	PDAT5	0	0		PATH#5 Data
P6- 12	PDEF6	0x0000	0x0000		PATH#6 Definition
P6- 13	PDAT6	0	0		PATH#6 Data
P6- 14	PDEF7	0x0000	0x0000		PATH#7 Definition
P6- 15	PDAT7	0	0		PATH#7 Data
P6- 16	PDEF8	0x0000	0x0000		PATH#8 Definition
P6- 17	PDAT8	0	0		PATH#8 Data
P6- 18	PDEF9	0x0000	0x0000		PATH#9 Definition
P6- 19	PDAT9	0	0		PATH#9 Data
P6- 20	PDEF1	0x0000	0x0000		PATH#10 Definition
P6- 21	PDAT1	0	0		PATH#10 Data
P6- 22	PDEF1	0x0000	0x0000		PATH#11 Definition
P6- 23	PDAT1	0	0		PATH#11 Data
P6- 24	PDEF1	0x0000	0x0000		PATH#12 Definition
P6- 25	PDAT1	0	0		PATH#12 Data
P6- 26	PDEF1	0x0000	0x0000		PATH#13 Definition
P6- 27	PDAT1	0	0		PATH#13 Data
P6- 28	PDEF1	0x0000	0x0000		PATH#14 Definition
P6- 29	PDAT1	0	0		PATH#14 Data
P6- 30	PDEF1	0x0000	0x0000		PATH#15 Definition
P6- 31	PDAT1	0	0		PATH#15 Data
P6- 32	PDEF1	0x0000	0x0000		PATH#16 Definition
P6- 33	PDAT1	0	0		PATH#16 Data
P6- 34	PDEF1	0x0000	0x0000		PATH#17 Definition
P6- 35	PDAT1	0	0		PATH#17 Data
P6- 36	PDEF1	0x0000	0x0000		PATH#18 Definition
P6- 37	PDAT1	0	0		PATH#18 Data
P6- 38	PDEF1	0x0000	0x0000		PATH#19 Definition
P6- 39	PDAT1	0	0		PATH#19 Data
P6- 40	PDEF2	0x0000	0x0000		PATH#20 Definition
P6- 41	PDAT2	0	0		PATH#20 Data
P6- 42	PDEF2	0x0000	0x0000		PATH#21 Definition
P6- 43	PDAT2	0	0		PATH#21 Data
P6- 44	PDEF2	0x0000	0x0000		PATH#22 Definition
P6- 45	PDAT2	0	0		PATH#22 Data
P6- 46	PDEF2	0x0000	0x0000		PATH#23 Definition
P6- 47	PDAT2	0	0		PATH#23 Data

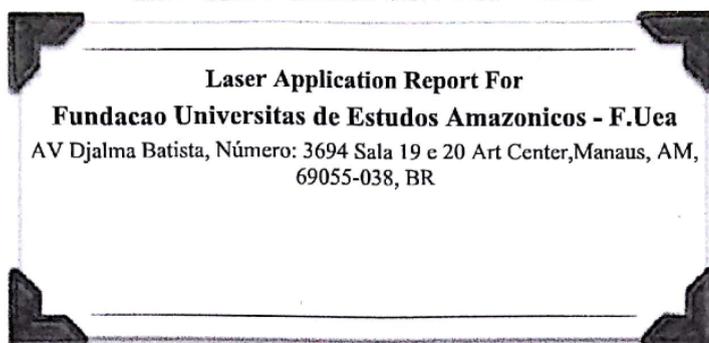
Para.	Code	Value	Default	Unit	Description
P6- 48	PDEF2	0x0000	0x0000		PATH#24 Definition
P6- 49	PDAT2	0	0		PATH#24 Data
P6- 50	PDEF2	0x0000	0x0000		PATH#25 Definition
P6- 51	PDAT2	0	0		PATH#25 Data
P6- 52	PDEF2	0x0000	0x0000		PATH#26 Definition
P6- 53	PDAT2	0	0		PATH#26 Data
P6- 54	PDEF2	0x0000	0x0000		PATH#27 Definition
P6- 55	PDAT2	0	0		PATH#27 Data
P6- 56	PDEF2	0x0000	0x0000		PATH#28 Definition
P6- 57	PDAT2	0	0		PATH#28 Data
P6- 58	PDEF2	0x0000	0x0000		PATH#29 Definition
P6- 59	PDAT2	0	0		PATH#29 Data
P6- 60	PDEF3	0x0000	0x0000		PATH#30 Definition
P6- 61	PDAT3	0	0		PATH#30 Data
P6- 62	PDEF3	0x0000	0x0000		PATH#31 Definition
P6- 63	PDAT3	0	0		PATH#31 Data
P6- 64	PDEF3	0x0000	0x0000		PATH#32 Definition
P6- 65	PDAT3	0	0		PATH#32 Data
P6- 66	PDEF3	0x0000	0x0000		PATH#33 Definition
P6- 67	PDAT3	0	0		PATH#33 Data
P6- 68	PDEF3	0x0000	0x0000		PATH#34 Definition
P6- 69	PDAT3	0	0		PATH#34 Data
P6- 70	PDEF3	0x0000	0x0000		PATH#35 Definition
P6- 71	PDAT3	0	0		PATH#35 Data
P6- 72	PDEF3	0x0000	0x0000		PATH#36 Definition
P6- 73	PDAT3	0	0		PATH#36 Data
P6- 74	PDEF3	0x0000	0x0000		PATH#37 Definition
P6- 75	PDAT3	0	0		PATH#37 Data
P6- 76	PDEF3	0x0000	0x0000		PATH#38 Definition
P6- 77	PDAT3	0	0		PATH#38 Data
P6- 78	PDEF3	0x0000	0x0000		PATH#39 Definition
P6- 79	PDAT3	0	0		PATH#39 Data
P6- 80	PDEF4	0x0000	0x0000		PATH#40 Definition
P6- 81	PDAT4	0	0		PATH#40 Data
P6- 82	PDEF4	0x0000	0x0000		PATH#41 Definition
P6- 83	PDAT4	0	0		PATH#41 Data
P6- 84	PDEF4	0x0000	0x0000		PATH#42 Definition
P6- 85	PDAT4	0	0		PATH#42 Data
P6- 86	PDEF4	0x0000	0x0000		PATH#43 Definition
P6- 87	PDAT4	0	0		PATH#43 Data
P6- 88	PDEF4	0x0000	0x0000		PATH#44 Definition
P6- 89	PDAT4	0	0		PATH#44 Data
P6- 90	PDEF4	0x0000	0x0000		PATH#45 Definition
P6- 91	PDAT4	0	0		PATH#45 Data
P6- 92	PDEF4	0x0000	0x0000		PATH#46 Definition
P6- 93	PDAT4	0	0		PATH#46 Data
P6- 94	PDEF4	0x0000	0x0000		PATH#47 Definition
P6- 95	PDAT4	0	0		PATH#47 Data
P6- 96	PDEF4	0x0000	0x0000		PATH#48 Definition
P6- 97	PDAT4	0	0		PATH#48 Data

Para.	Code	Value	Default	Unit	Description
P6 - 98	PDEF4	0x0000	0x0000		PATH#49 Definition
P6 - 99	PDAT4	0	0		PATH#49 Data
P7 - 0	PDEF5	0x0000	0x0000		PATH#50 Definition
P7 - 1	PDAT5	0	0		PATH#50 Data
P7 - 2	PDEF5	0x0000	0x0000		PATH#51 Definition
P7 - 3	PDAT5	0	0		PATH#51 Data
P7 - 4	PDEF5	0x0000	0x0000		PATH#52 Definition
P7 - 5	PDAT5	0	0		PATH#52 Data
P7 - 6	PDEF5	0x0000	0x0000		PATH#53 Definition
P7 - 7	PDAT5	0	0		PATH#53 Data
P7 - 8	PDEF5	0x0000	0x0000		PATH#54 Definition
P7 - 9	PDAT5	0	0		PATH#54 Data
P7 - 10	PDEF5	0x0000	0x0000		PATH#55 Definition
P7 - 11	PDAT5	0	0		PATH#55 Data
P7 - 12	PDEF5	0x0000	0x0000		PATH#56 Definition
P7 - 13	PDAT5	0	0		PATH#56 Data
P7 - 14	PDEF5	0x0000	0x0000		PATH#57 Definition
P7 - 15	PDAT5	0	0		PATH#57 Data
P7 - 16	PDEF5	0x0000	0x0000		PATH#58 Definition
P7 - 17	PDAT5	0	0		PATH#58 Data
P7 - 18	PDEF5	0x0000	0x0000		PATH#59 Definition
P7 - 19	PDAT5	0	0		PATH#59 Data
P7 - 20	PDEF6	0x0000	0x0000		PATH#60 Definition
P7 - 21	PDAT6	0	0		PATH#60 Data
P7 - 22	PDEF6	0x0000	0x0000		PATH#61 Definition
P7 - 23	PDAT6	0	0		PATH#61 Data
P7 - 24	PDEF6	0x0000	0x0000		PATH#62 Definition
P7 - 25	PDAT6	0	0		PATH#62 Data
P7 - 26	PDEF6	0x0000	0x0000		PATH#63 Definition
P7 - 27	PDAT6	0	0		PATH#63 Data

APÊNDICE D

A seguir, tem-se o relatório enviado pelo fornecedor em relação a testes de qualidade de impressão e velocidade de impressão, garantindo o atendimento ao takt-time do cliente.

Pass



Sales Engineer:

Carlos Silva • 92 98114-0965;

Substrate Detail

Description • PLACA
ELETRONICA

Material • Plastic

Application Request Summary

Marking System • Best Suitable Solid State Solution

Color Change_Laser;Code

Goal of Sample • Quality_Laser;Texture
Change_Laser;Speed_Laser;Code
Size_Laser

Marking Area • 0.0 mm * 0.0 mm

Product Handling • Stationary

Available Mark Time • 0.0 ms

Line Speed • 0.05 fpm

Equipment Result Summary

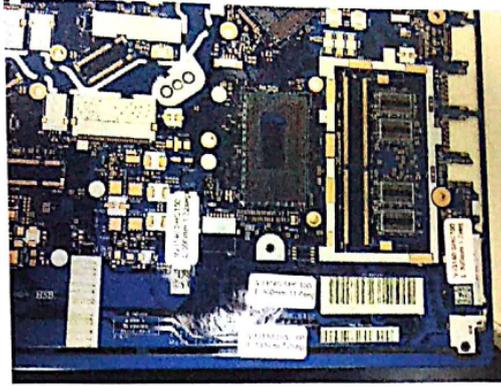
Marking System • LASER.3140

Total Code Size • DIVERSOS

Mark Time • DIVERSOS

Line Speed • Speed Met

Actual Line Speed •



1500 Mittel Blvd • Wood Dale, IL 60191 • USA
Sales • Service • Parts • Installation
1-800-VIDEO-10 • 1-800-843-3610

Pass

Equipment Details

Marking System • LASER.3140
Tube Type • Standard (10.6 µm)
Marking Head • SHC100
Focusing Lens • 300 mm (F300 D48)
Beam Expander • 3.0 X
Total Code Size • DIVERSOS
Mark Time • DIVERSOS
Line Speed • Speed Met

Barcode 1 Details

Barcode Type • Code 128
Barcode Grade • N/A
Barcode Pass/Fail • Pass
Barcode Content • 1234567890
Verified With • Mixi Hand Held

Barcode 2 Details

Barcode Type • QR Code
Barcode Grade •
Barcode Pass/Fail • Pass
Barcode Content • 123456789
Verified With • Mixi Hand Held

Detailed Application Settings

Software • CMark 3.4.0.1 / Smartgraph 10.4
Character Height • DIVERSOS
Line Spacing • DIVERSOS
Font • QRCode Codigo de Barras
Logo Size •
Logo Name •
Jump Speed •
Mark Speed •
Marking Intensity % • 80.0
Calibrated % • 100.0
Pulse Frequency • 20000
Line Width •
Jump Delay •
Mark Delay •
Stroke Delay •
On Delay •
Off Delay •

OCR Details

OCR Type •
OCR Grade •
Verified With •

Printed Message

Código de barras e Codigo 2d (QR code e Datamatrix)

Technician's Notes:

Equipamento utilizado - LASER 3140 SHC100 L:100mm Marcações com tempo de marcação de 1.32 seg ; 1.7 seg ; 7 seg ; 11 seg. Marcações com bom contraste. Leitura dos códigos OK, Testes realizados com leitor HONEYWELL Model - Xenon 1900 ; 3800g e com Smart phone , Aplicativo (I-nigma). OBS: Após marcação necessário remoção de pó que fica sobre a placa. Testes realizados com 10 caracteres numéricos em cada código. Responsável pela amostra - JOSE TAVARES