



Faculdade de Pindamonhangaba



**Leonardo Lopes Siqueira**

**Mesa CNC Tridimensional XYZ**

**Pindamonhangaba-SP**

**2018**



Faculdade de Pindamonhangaba



**Leonardo Lopes Siqueira**

## **Mesa CNC Tridimensional XYZ**

Monografia apresentado como parte dos requisitos para obtenção do Diploma de Bacharel pelo curso de Engenharia de Controle e Automação da Funvic - Faculdade de Pindamonhangaba.

Orientador: Prof. Me. Marcelo Pinheiro Werneck

**Pindamonhangaba-SP**

**2018**



Faculdade de Pindamonhangaba



**LEONARDO LOPES SIQUEIRA**

Monografia apresentado como parte dos requisitos para obtenção do Diploma de Bacharel pelo curso de Engenharia de Controle e Automação da Funvic - Faculdade de Pindamonhangaba.

Data: \_\_\_\_\_

Resultado: \_\_\_\_\_

**BANCA EXAMINADORA**

*Prof. Me. Marcelo Pinheiro Werneck, FUNVIC – Faculdade de Pindamonhangaba*

**Assinatura:** \_\_\_\_\_

*Prof. Me. Célio Augusto Machado, FUNVIC – Faculdade de Pindamonhangaba*

**Assinatura:** \_\_\_\_\_

*Prof. Dr. Cláudio Augusto Kelly, FUNVIC – Faculdade de Pindamonhangaba*

**Assinatura:** \_\_\_\_\_

*Prof. Me. Daniel Carlos Reis, FUNVIC – Faculdade de Pindamonhangaba (Suplente)*

**Assinatura:** \_\_\_\_\_

Siqueira, Leonardo Lopes

Mesa CNC Tridimensional XYZ / Leonardo Lopes Siqueira / Pindamonhangaba-SP:  
FUNVIC – Fundação Universitária Vida Cristã, 2018.

71f.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação)  
FUNVIC – Fundação Universitária Vida Cristã-SP.

Orientador: Prof. Me. Marcelo Pinheiro Werneck.

1 Introdução. 2 Revisão de Literatura. 3 Funcionamento das Automações Propostas.  
4 Procedimentos. 5 Resultados. 6 Discussão. 7 Conclusões .8 Considerações Finais.

I Mesa CNC Tridimensional XYZ II Leonardo Lopes Siqueira.

Dedico este trabalho a minha família, o alicerce onde pude conquistar muitas oportunidades, aonde sempre encontrei o meu porto seguro. Em especial dedico o meu trabalho a minha irmã (in memoriam), que me inspira todos os dias a amar.

## **AGRADECIMENTOS**

A Deus por conceder o privilégio da vida, pela oportunidade de cursar Engenharia, um aprendizado que levarei para o resto da minha vida.

A minha esposa, por todo o tempo e compreensão na minha dedicação à graduação. Pelo maior presente que poderia me proporcionar, o nosso filho.

Aos meus pais, por seu amor e carinho, em especial a minha mãe, que desde o primeiro momento me incentivou e proporcionou condições para que esse sonho se realizasse.

Aos amigos, pelo apoio, compreensão e tempo dedicado para que os nossos sonhos se realizassem.

Aos professores, a cada um pelo conhecimento compartilhado e dedicação para que o difícil se tornasse fácil.

A todos que de alguma forma contribuíram, para a realização deste trabalho. Para a realização do meu desejo de ser um Engenheiro.

*“A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original.”*

Albert Einstein

## RESUMO

Diante da necessidade proveniente de trabalhos repetitivos manuais, a busca por automação é uma vertente cada vez mais procurada. A produtividade e qualidade é o que impulsiona o mercado a procurar alternativas que maximizem a lucratividade e a imagem das empresas. E para tanto, a automação é um meio muito eficaz e eficiente para chegar a esse ponto. Além da redução do desperdício e falhas que podem ser ocasionadas pelo fator humano. O objetivo deste projeto é desenvolver um sistema automatizado de uma mesa tridimensional capaz de realizar a confecção de peças através das coordenadas XYZ, controlada por um dispositivo CLP responsável pela automação previamente programada através do software TIA® (Totally Integrated Automation) da Siemens™ em um computador utilizando o sistema operacional Windows®. O usuário terá interação com o processo podendo controlar cada etapa através de um IHM. Além do CLP e o IHM, serão utilizados: motor de passo, reles, fontes e um quadro de comando, o projeto contará com programas específicos para processamento dos comandos.

**Palavra-chave:** CLP. IHM. Mesa Tridimensional. Processos.



## **ABSTRACT**

Faced with the need arising from repetitive manual work, the search for automation is an increasingly sought-after aspect. Productivity and quality is what drives the market to look for alternatives that maximize profitability and corporate image. And for that, automation is a very effective and efficient way to get to that point. In addition to the reduction of waste and failures that can be caused by the human factor. The objective of this project is to develop an automated three-dimensional table system capable of manufacturing parts using XYZ coordinates, controlled by a CLP device responsible for pre-programmed automation through Siemens™ TIA® (Totally Integrated Automation) software in one using the Windows® operating system. The user will have interaction with the process, being able to control each step through an HMI. In addition to the PLC and the HMI, there will be used: stepper motor, relays, sources and a control panel, the project will have specific programs for processing the commands.

**Keyword:** PLC. MHI. Control. Three-dimensional Table. Processes.

## LISTA DE FIGURAS

Figura 1 - Inicialização e ciclo de varredura .....	5
Figura 2 - Motor de Passo .....	9
Figura 3 - Motor de passo de relutância com duas faces.....	10
Figura 4 - Sequência de movimento motor de passo RV .....	10
Figura 5 – Motor de passo de ímã permanente de duas faces .....	11
Figura 6 - Sequência de movimento motor de passo IP .....	11
Figura 7 - Motor de passo híbrido .....	12
Figura 8 - Eixo de um motor de passo híbrido .....	12
Figura 9 - Sequência de movimento motor de passo Híbrido .....	13
Figura 10 - IHM (Interface Homem-Máquina) .....	14
Figura 11 - Tela Software TIA® fonte Siemens™ .....	14
Figura 12 - Protocolo de comunicação CLP x IHM.....	15
Figura 13 - Topologia IHM, CLP, Drive e motor de passo.....	15
Figura 14 - Array .....	16
Figura 15 - Array Posi_X .....	16
Figura 16 - Array Posi_Y .....	16
Figura 17 - Array Posi_Z.....	17
Figura 18 - Bloco Tecnológico MC_MoveAbsolute_DB .....	18
Figura 19 - Configuração memória %M20.0 .....	19
Figura 20 - Contato para posição da tabela Posi_X.....	19
Figura 21 - Execução do índice Posi_X .....	20
Figura 22 - Fim da Execução da Tabela Posi_X .....	20
Figura 23 - Execução de movimento .....	21
Figura 24 - Retardo de tempo entre os movimentos.....	21
Figura 25 - CLP S-7 1200 CPU 1214C DC/DC/DC .....	22
Figura 26 - Configuração PTO .....	23
Figura 27 - Configuração PTO eixo X .....	23
Figura 28 - Configuração PTO eixo Y .....	24
Figura 29 - Configuração PTO eixo Z.....	24
Figura 30 - Objetos tecnológicos.....	25
Figura 31 - Configuração Axis_X .....	26
Figura 32 - Configuração Drive Motor eixo X.....	26

Figura 33 - Configuração de Limites eixo X.....	27
Figura 34 - Sensor Reed Switch .....	27
Figura 35 - Configuração Axis_Y .....	28
Figura 36 - Configuração Drive Motor eixo Y .....	28
Figura 37 - Configuração de Limites eixo Y .....	29
Figura 38 - Configuração Axis_Z.....	30
Figura 39 - Configuração Drive Motor eixo Z .....	30
Figura 40 - Configuração de Limites eixo Z .....	31
Figura 41 - Blocos Tecnológicos.....	31
Figura 42 - Bloco tecnológico MC_Power.....	33
Figura 43 - Bloco tecnológico MC_Reset .....	33
Figura 44 - Bloco tecnológico MC_Home .....	34
Figura 45 - Bloco Tecnológico MC_MoveAbsolute.....	35
Figura 46 - Bloco Tecnológico MC_MoveRelative .....	35
Figura 47 - Tela Root Screen.....	36
Figura 48 - Tela para Zerar Matrizes de Posicionamento/Velocidade .....	37
Figura 49 - Tela de Edição dos Valores de Índice, Velocidade e Posição dos Eixos X, Y e Z	38
Figura 50 - Tela de Início do Trabalho.....	39
Figura 51 - Tela Home dos Eixos.....	40
Figura 52 -Tela Movimento Absoluto para o Eixo X.....	42
Figura 53 - Tela Movimento Absoluto para o Eixo Y.....	43
Figura 54 - Tela Movimento Absoluto para o Eixo Z .....	44
Figura 55 - Tela Movimento Relativo para o Eixo X.....	45
Figura 56 - Tela Movimento Relativo para o Eixo Y.....	47
Figura 57 - Tela Movimento Relativo para o Eixo Z .....	48
Figura 58 - Montagem do CLP no Quadro de Comando .....	50
Figura 59 - Fontes de Alimentação IHM e CLP Respectivamente .....	51
Figura 60 - Reles de Interface.....	51
Figura 61 - Instalação dos Drives de Controle dos Motores de Passo .....	52
Figura 62 - Mesa CNC Tridimensional XYZ.....	52
Figura 63 - Quadro de Comando Montado Mesa Tridimensional XYZ .....	53
Figura 64 - Mesa Router Hobby vista de cima.....	54

## **LISTA DE TABELAS**

Tabela 1 - Operadores Lógicos.....	7
Tabela 2 - Tabela com simbologia e exemplo de como montar uma tabela verdade.....	7
Tabela 3 - Definição dos Blocos Tecnológicos.....	32
Tabela 4 - Relação de Materiais Utilizados.....	49

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	1
<b>2 REVISÃO DA LITERATURA</b> .....	2
<b>2.1 Automação</b> .....	2
<b>2.2 Controlador Lógico Programável (CLP)</b> .....	3
2.2.1 PRINCÍPIO DE FUNCIONAMENTO .....	4
2.2.1.1 ENTRADAS.....	5
2.2.1.2 SAÍDAS.....	6
<b>2.3 Álgebra Booleana</b> .....	6
<b>2.4 Tabela Verdade</b> .....	7
<b>2.5 Sistema TIA®</b> .....	8
<b>2.6 Motor de Passo</b> .....	9
2.6.1 RELUTÂNCIA VARIÁVEL .....	9
2.6.2 IMÃ PERMANENTE .....	10
2.6.3 HÍBRIDO .....	11
<b>2.7 Driver Motor de Passo</b> .....	13
<b>2.8 Interface Homem-Máquina</b> .....	13
<b>3 FUNCIONAMENTO DAS AUTOMATIZAÇÕES PROPOSTAS</b> .....	14
<b>3.1 Programa Ladder para Execução dos Movimentos</b> .....	17
<b>3.2 Configuração do CLP, Blocos e Objetos Tecnológicos</b> .....	22
3.2.1 CONFIGURAÇÃO DO CLP PARA UTILIZAÇÃO DAS SAÍDAS RÁPIDAS.....	22
3.2.2 OBJETO DE TECNOLOGIA “AXIS” E BLOCOS DE FUNÇÃO “MOTION CONTROL” .....	24
3.2.3 BLOCOS TECNOLÓGICOS.....	31
3.2.3.1 BLOCO TECNOLÓGICO MC_POWER.....	32
3.2.3.2 BLOCO TECNOLÓGICO MC_RESET.....	33
3.2.3.3 BLOCO TECNOLÓGICO MC_HOME .....	34
3.2.3.4 BLOCO TECNOLÓGICO MC_MOVEABSOLUTE .....	34
3.2.3.5 BLOCO TECNOLÓGICO MC_MOVERELATIVE.....	35
<b>3.3 Configuração da Interface Homem-Máquina (IHM)</b> .....	36
3.3.1 TELA ZERAR MATRIZES X, Y, Z.....	37
3.3.1.2 TELA EDITAR MATRIZ.....	38

3.3.2 TELA EXECUTAR TRABALHO.....	39
3.3.3 TELA HOME DOS EIXO.....	40
3.3.4 TELA MOVIMENTO ABSOLUTO PARA O EIXO X.....	41
3.3.5 TELA MOVIMENTO ABSOLUTO PARA O EIXO Y.....	42
3.3.6 TELA MOVIMENTO ABSOLUTO PARA O EIXO Z.....	43
3.3.7 TELA MOVIMENTO RELATIVO PARA O EIXO X.....	44
3.3.8 TELA MOVIMENTO RELATIVO PARA O EIXO Y.....	46
3.3.9 TELA MOVIMENTO RELATIVO PARA O EIXO Z.....	47
<b>4 PROCEDIMENTOS</b> .....	<b>49</b>
<b>4.1 Materiais Utilizados</b> .....	<b>49</b>
<b>4.2 Montagem</b> .....	<b>50</b>
<b>5 RESULTADOS E DISCUSSÃO</b> .....	<b>53</b>
<b>5.1 Projeto Protótipo Mesa Tridimensional XYZ</b> .....	<b>53</b>
<b>6 CONCLUSÕES</b> .....	<b>55</b>
<b>REFERÊNCIAS</b> .....	<b>56</b>

## 1 INTRODUÇÃO

O Comando Numérico (CN) teve sua primeira aplicação em 1949, no laboratório de Servo Mecanismo do Instituto de Tecnologia de Massachusetts, em associação com a Força Aérea Americana (USAF) e a Parsons Corporations. Muito do que foi desenvolvido na época foi promovido pela USAF, devido a produção de estruturas para os jatos serem em um curto espaço de tempo, constitui-se uma ótima oportunidade para testes de CN. Após a Segunda Guerra Mundial o CN foi muito importante para o avanço tecnológico.

Algumas demandas não eram supridas pelas máquinas convencionais, isso levou a introdução das primeiras máquinas comandadas por CN na indústria, devido a fatores como a tentativa de aumentar o tempo de vida útil do produto e dificuldades nos projetos por causa das constantes mudanças. Em 1970, foram introduzidos os microprocessadores, memórias ROM (Read Only Memory) e computadores dedicados a CN, devido ao avanço da eletrônica. Com essa integração surgiram os Comandos Numéricos Computadorizados (CNC).

Comandos Numéricos Computadorizados são operações que estão contidas em um programa com dados alfa numéricos codificados, que controlam as funções e movimentos de uma máquina sem a intervenção de um operador. O CNC controla a velocidade de avanço, fluido refrigerante, profundidade de corte, dentre outras funções. Podemos definir basicamente a composição do CNC em três partes: unidade de entrada, unidade controladora e unidade de acionamento de potência. Sendo:

- **Unidade de Entrada:** as informações nesta unidade são inseridas através da intervenção humana, podendo ser através de um dispositivo IHM (Interface Homem Máquina), unidades removíveis como cartão de memória, pen drive ou conexão de rede direta com o servidor;
- **Unidade Controladora:** nesta unidade os comandos do código são convertidos em trajetórias de movimentação ou funções que a máquina possui;
- **Unidade de Acionamento de Potência:** é a unidade responsável por receber os sinais emitidos pela unidade controladora e realizar o acionamento dos servomotores e periféricos que estejam dispostos na máquina.

Na atualidade a tecnologia CNC está em aplicação nas diversas áreas da indústria, como robôs e diversos tipos de máquinas, que aumentam a automatização dos processos.

## **2 REVISÃO DA LITERATURA**

Serão apresentados de forma breve, neste capítulo, fundamentos de controle e automação para proporcionar um melhor entendimento da automação da mesa CNC tridimensional XYZ. Serão abordados os fundamentos de funcionalidade do CLP e suas aplicações. Assim como os materiais utilizados e a lógica para desenvolvimento do protótipo da mesa CMC tridimensional XZY.

### **2.1 Automação**

Automação em uma definição simples pode ser descrita como a substituição do trabalho humano ou animal por máquinas. É a operação da máquina ou de um sistema automático sem a intervenção humana. Automático significa ter um mecanismo de atuação própria, com ação requerida em um espaço de tempo que responda a certas condições.

O conceito de automação sugere a ideia da utilização de potência elétrica ou mecânica para acionamento de algum tipo de máquina, que deverá ter algum tipo de inteligência para executar a tarefa de modo eficiente, econômico e com segurança.

Com o avanço da eletrônica, circuito integrado (1960) e microprocessador (1970), foi possível a implantação de inteligência em uma máquina, a um custo razoável. O número de operações complexas automáticas cresceu. Muitos sistemas de automação são possíveis devido aos grandes avanços na eletrônica moderna. A automação na atualidade é fortemente baseada no uso da eletrônica microprocessada que pode fornecer diversos tipos de sistemas eletrônicos programáveis.

Segundo Franchi e Camargo (2009, p.21) “devido ao fato de o computador ser programável, ele proporciona uma grande vantagem em comparação com a lógica por interligação elétrica, utilizada em sistemas de relés e CIs”.

Porém não pode ser descartado o fator humano, pois, ao invés de estar executando a atividade diretamente, este deverá controlar e monitorar a máquina durante o processo.



## 2.2 Controlador Lógico Programável (CLP)

Pode ser definido como Controlador Lógico Programável (CLP) um sistema eletrônico operando digitalmente, projetado para uso em ambiente industrial, que utiliza memória programável para armazenamento interno de instruções orientadas para usuário com finalidade de implementar funções específicas, tais como lógica, sequencial, temporização, contagem e aritmética, para controlar através de entradas e saídas digitais ou analógicas diversas máquinas ou processos (IEC – International Electrotechnical Commission).

Os CLPs são sistemas modulares que podem ter em sua composição básica os seguintes elementos:

- Fonte de alimentação;
- CPU;
- Memória;
- Módulos de entrada e saída;
- Dispositivos de programação;
- Módulos de comunicação.

A tecnologia inserida no CLP só foi possível devido ao advento dos circuitos integrados e do desenvolvimento e evolução da lógica digital. Segundo Franchi e Camargo (2009) dentre inúmeras vantagens do CLP podem ser citadas as seguintes:

- Economia de espaço, pois o seu tamanho é reduzido;
- Podem ser programados sem a necessidade da interrupção do processo de produção;
- Baixo consumo de energia;
- Flexibilidade para expansão do número de entradas e saídas;
- Interface de comunicação com outros CLPs.

Historicamente o CLP foi desenvolvido para melhorar a linha de montagem da empresa General Motors, em 1968, pois a empresa tinha grandes dificuldades quando havia a necessidade de mudar a lógica de controle dos painéis de comando, cada vez que a linha de montagem precisava ser alterada. Essas mudanças oneravam muito o processo além da perda de tempo.

A primeira geração era programada em Assembly, com isso, era necessário conhecer o hardware do equipamento, ou seja, o projeto eletrônico do CLP. A segunda geração surgiu com linguagem de programação de nível médio, devido ao desenvolvimento do “programa monitor”, que transformava em linguagem de máquina o programa inserido pelo usuário. A terceira

geração surgiu com entrada de programação feita através de um teclado, ou um programador portátil que ficava conectado ao mesmo. A quarta geração apresentou uma entrada de comunicação serial, e a programação passou a ser feita através de micro-computadores, essa inovação possibilitou que os programas fossem testados no computador antes de serem transferidos para o CLP. A quinta geração inseriu novos protocolos de comunicação para facilitar a interface entre outros componentes e fabricantes, além de sistemas supervisórios e redes internas de comunicação.

### 2.2.1 PRINCÍPIO DE FUNCIONAMENTO

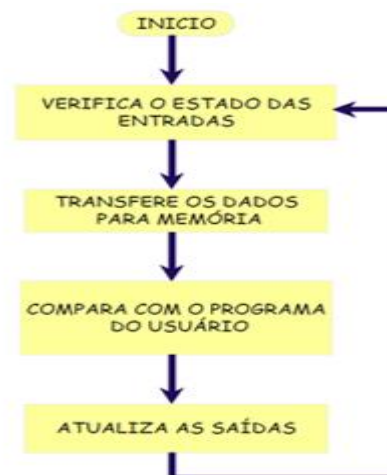
Um CLP pode ter sua funcionalidade dividida em três partes básicas, sendo: as entradas, a unidade central de processamento e as saídas. Durante o funcionamento são realizadas verificações de operações que são denominadas ciclo de varredura. Nesse ciclo são verificados vários itens tais como reconhecimento dos módulos de entrada e saída, estado da memória, dentre outros.

O controlador monitora o *status* do processo em tempo real de uma planta através de um número definido de transdutores, que convertem as grandezas físicas em sinais elétricos, os quais são conectados com as entradas dos CLPs. (Franchi; Camargo, 2009, p.25, grifo do autor).

Esse processo é nomeado de inicialização (Figura 1). Se o hardware está em condições e existir um programa instalado, o programa de inicialização inicia o programa, realiza o ciclo de varredura, que verifica as entradas e saídas, armazena os dados na memória, faz a comparação com o programa e atualiza as saídas caso a imagem esteja diferindo do programa.

Baseado nesse processo Prudente (2007, p.10, grifo do autor) afirma que:

A cada leitura das entradas os dados são armazenados em um registro particular da memória, chamado *imagem do processo das entradas*. O mesmo acontece com os dados de saída: antes de serem colocados nas saídas externas, eles são armazenados em um registro particular da memória chamado *imagem do processo das saídas*[...].



**Figura 1** - Inicialização e ciclo de varredura

Fonte: Internet

### 2.2.1.1 ENTRADAS

As entradas permitem ao CLP receber todas as informações do processo, originados por botoeiras, cantos, relés, sensores, encoders, entre outros dispositivos usados para monitorar o processo e fornecer algum tipo de informação de retorno. As entradas podem ser digitais ou analógicas, bem como internas como externas. Segundo Prudente (2007, p.18, grifo do autor) “a *unidade de entrada* opera sob o sinal de entrada proveniente dos sensores da instalação para torna-la compatível com a CPU do CLP”.

As entradas externas são provenientes do ambiente externo, como por exemplo um sensor, e as internas são provenientes de outros componentes internos do CLP, como por exemplo um contato de um temporizador.

Franchi e Camargo (2009) afirmam que entradas digitais são aquelas de sinal discreto, ou seja, sinais que possuem apenas dois valores, que são denominados de nível alto, representado pelo algarismo 1 e nível baixo representado pelo algarismo 0. Ou seja, um sinal discreto pode ser representado por um interruptor, pois só oferece as opções ligado (nível alto) e desligado (nível baixo).

Entradas analógicas são aquelas de sinal contínuos no tempo e podem assumir qualquer valor entre o mínimo e máximo valor de trabalho da entrada, como por exemplo, um sinal de tacogerador para realizar o controle de rotação de um motor, onde a tensão pode aumentar ou diminuir à medida que a rotação do motor aumenta ou diminui.

### 2.2.1.2 SAÍDAS

As saídas são as interfaces das quais o CLP pode gerenciar a alimentação da carga. Da mesma forma que as entradas, as saídas podem ser digitais, analógicas, internas ou externas. Uma saída externa enviará um sinal elétrico para um componente externo, como por exemplo o controle de um motor. Já as saídas internas pode ser uma bobina de um temporizador interno.

Franchi e Camargo (2009) afirmam que as saídas digitais são semelhantes as entradas digitais, são aquelas que podem oferecer dois valores, nível alto ou baixo, podendo ser ligadas a saída, por exemplo, lâmpadas, solenoides de contadores, solenoide de eletroválvulas ou um dispositivo que só precise ser alimentado com uma tensão nominal ou desligado.

Saídas analógicas são as interfaces que podem variar a continuidade no tempo a tensão ou a corrente sobre a carga, como por exemplo, ser ligada em uma saída analógica, através de uma placa de controle, um motor de corrente contínua que varia a rotação conforme a tensão varia sobre o induzido.

No caso de uma saída analógica podemos adicionar um elemento dentro de uma faixa de valores que corresponde de 0 a 100%. Por exemplo, com uma saída analógica podemos ligar um motor com 40% da sua rotação normal, uma válvula proporcional pode ser aberta 25%. (Franchi; Carmargo, 2009, p.55).

## 2.3 Álgebra Booleana

George Boole foi o matemático que primeiramente desenvolveu uma teoria para tratamento de sistemas binários (Franchi e Camargo, 2009).


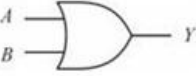
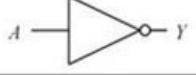


“A álgebra booleana é definida como um conjunto de dois elementos: verdadeiro e falso, ou seja, uma variável representa se uma proposição lógica é falsa ou verdadeira. Por exemplo, uma chave pode estar aberta ou fechada” (Franchi e Camargo, 2009, p.143).

Álgebra de Boole é uma ferramenta matemática que permite descrever uma relação entre as saídas de um circuito lógico e suas entradas (Tabela 1), usando uma equação, chamada de expressão booleana. Na Álgebra de Boole são permitidos apenas dois valores: 0 e 1.

Geralmente os níveis lógicos (0 e 1) representam os níveis de tensão elétrica em um ponto do circuito, sendo 0 considerado o nível lógico baixo (entre 0V e 0,8V) e 1 para considerar o nível lógico alto (entre 2,5V e 5V).

**Tabela 1** - Operadores Lógicos

Fonte: Internet

Operadores	Simbolo	Equação Booleana
E "AND"		$Y = A \cdot B$
OU "OR"		$Y = A + B$
NÃO "NOT"		$Y = \overline{A}$
NÃO E "NAND"		$Y = \overline{A \cdot B}$
NÃO OU "NOR"		$Y = \overline{A + B}$

## 2.4 Tabela Verdade

A tabela verdade (Tabela 2) é uma ferramenta matemática que é muito utilizada no campo de raciocínio lógico (Franchi e Camargo, 2009). O objetivo é verificar a validade lógica de uma argumentação formada por duas ou mais proposições simples. Onde cada proposição simples pode ser verdadeira ou falsa e isso implicará diretamente no valor lógico da proposição composta.

**Tabela 2** - Tabela com simbologia e exemplo de como montar uma tabela verdade

Fonte: Internet

Símbolo	Operação Lógica	Significado	Exemplo
<b>p</b>	.	Proposição 1	p = João é alto.
<b>q</b>	.	Proposição 2	q = Maria é baixa.
~	Negação	não	Se João é alto, "~p" é FALSO.
^	Conjunção	e	$p \wedge q$ = João é alto e Maria é baixa.
v	Disjunção	ou	$p \vee q$ = João é alto ou Maria é baixa.
>	Condicional	se...então	$p > q$ = Se João é alto então Maria é baixa.
<>	Bicondicional	se e somente se	$p < > q$ = João é alto se e somente se Maria é baixa.

## 2.5 Sistema TIA®

TIA® é um software de plataforma integrada para a implementação de soluções de automação. Durante mais de 15 anos, a automação totalmente integrada tem sido o principal suporte para mais de 100.000 produtos de automação (Siemens, 2018). Software pioneiro que inclui uma interface intuitiva que permite uma adaptabilidade mais rápida, não precisando dispendir tempo em demasia em aprender sobre como utilizar o software.

“As funções, as propriedades e as bibliotecas são apresentadas automaticamente consoante a visualização mais intuitiva para cada tarefa pretendida” (Siemens, 2018). Possui uma arquitetura de software avançada baseada em um sistema de navegação simples, garantindo uma ergonomia gráfica, o que maximiza a eficácia e economia do tempo.

O software TIA® foi concebido para os controlares SIMATEC STEP 7, esse por sua vez tem sido considerado um padrão para a programação de controladores nos últimos anos. O TIA® integra em sua plataforma os controladores modulares e controladores SIMATEC baseados em computador, permitindo a configuração, programação, testes e diagnósticos (Siemens, 2018).

A integração possibilita: a. utilização de um conjunto de engenharia para todos os requisitos de automação, com controladores SIMATIC de engenharia fácil e eficaz, desde a micro-automação aos controladores baseados em computador; b. menor tempo de concepção com editores inteligentes de alta performance que agilizam a programação do controlador; c. redução considerável da curva de aprendizagem com configurações de ligação intuitiva com a funcionalidade “arrastar e posicionar” para a configuração do IHM e da rede; d. aumento da qualidade do produto, armazenando os componentes de engenharia modular comprovados numa biblioteca e reutilizando-os facilmente em projetos futuros; e. proteção do investimento através da compatibilidade com versões superiores para futuras inovações do produto; f. proteção do bem-estar dos recursos valiosos através da fácil integração da segurança da máquina. (Siemens, 2018).

Além dos controladores o TIA® permite a programação de dispositivos IHM, incluindo desde painéis básicos até os complexos, incluindo as funcionalidades do sistema SCADA (Siemens, 2018).

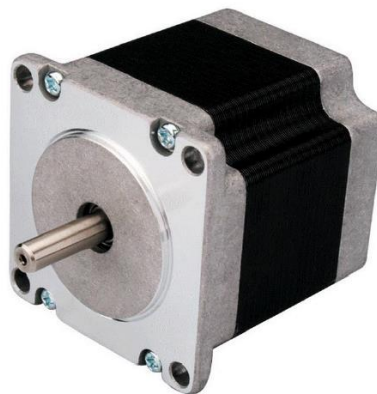
[...] TIA disponibiliza: a. engenharia consistente para toda a largura de banda do IHM – de um painel básico até um sistema SCADA para todas as aplicações IHM; b. proteção do investimento através de funcionalidades facilmente atualizáveis, de painéis SIMATIC IHM orientados para a máquina a uma solução SCADA completa; c. redução do tempo de desenvolvimento do IHM com funcionalidades intuitivas orientadas para a tarefa, desenvolvidas para obter a máxima eficiência da engenharia; d. menor tempo de configuração das comunicações do sistema através do fácil acesso do tipo “arrastar e posicionar” aos dados de todos os controladores. (Siemens, 2018).

## 2.6 Motor de Passo

Motor de passo (Figura 2) é um dispositivo eletromecânico que converte pulsos elétricos em movimentos. O eixo do motor gira de forma discreta a cada pulso aplicado de forma correta, esse giro é geralmente denominado de “passo”. O giro do motor está diretamente ligado ao pulso que recebe, sua velocidade é definida pela frequência com que os pulsos são enviados e o número de voltas é definido pela quantidade de pulsos que recebe. O controle dos pulsos elétricos é realizado através de um dispositivo chamado *driver* (Unesp, 2013)

Pode ser utilizado em projetos onde necessite controlar o ângulo de rotação, velocidade, sincronismo e posição. Devido a sua funcionalidade e características o motor de passo é amplamente utilizado em soluções de automação, como por exemplo routers CNC, máquinas de corte de plasma, controle de válvulas dentre outras.

Os motores de passo podem ser de três tipos: relutância variável, ímã permanente híbrido.

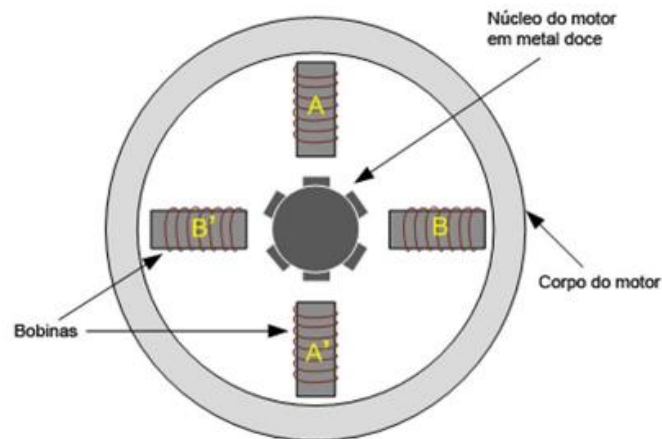


**Figura 2** - Motor de Passo

Fonte: Internet

### 2.6.1 RELUTÂNCIA VARIÁVEL

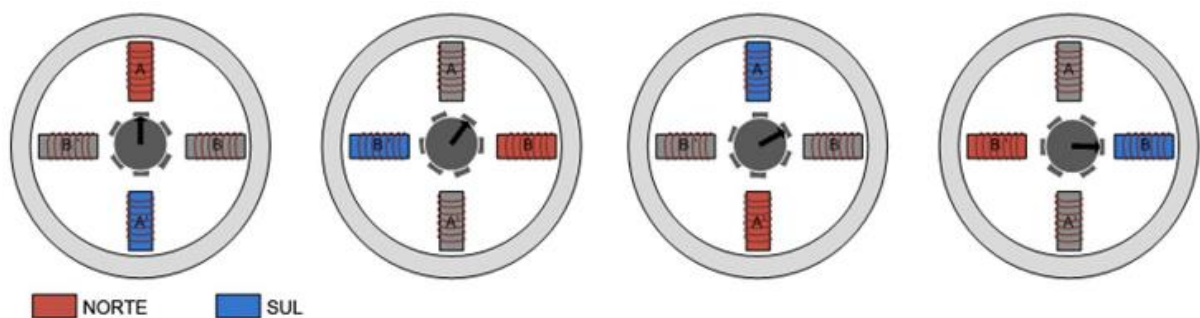
Motores de relutância variável (RV) são aqueles que possuem em sua composição um rotor dentado de ferro doce e um estator bobinado (Figura 3). Por não possuir um campo magnético permanente faz com que tenha torque menor na saída (Unesp, 2013)



**Figura 3** - Motor de passo de relutância com duas faces

Fonte: Unesp, 2013

Os dentes do eixo são atraídos para os polos que estão energizados de um estator e desalinhados em relação a outro, na sequência, o próximo grupo de bobinas é ligado e novamente os dentes do eixo são alinhados para outro estator, desalinhando o anterior, promovendo assim o movimento do motor (Figura 4).



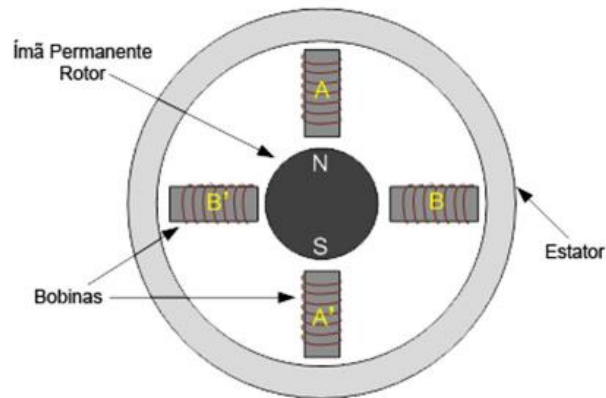
**Figura 4** - Sequência de movimento motor de passo RV

Fonte: Unesp, 2013

### 2.6.2 IMÃ PERMANENTE

Esse tipo de motor de passo ao contrário do RV possui um ímã permanente (IP) em um eixo liso (Figura 5), proporcionando uma mecânica mais simples e economicamente mais barata. Por ter um eixo magnético, ao ser associado ao campo magnético das bobinas proporciona mais torque na saída, porém gera passo ser maior, o que proporciona desvantagem devido a menor precisão do giro (Unesp, 2013).

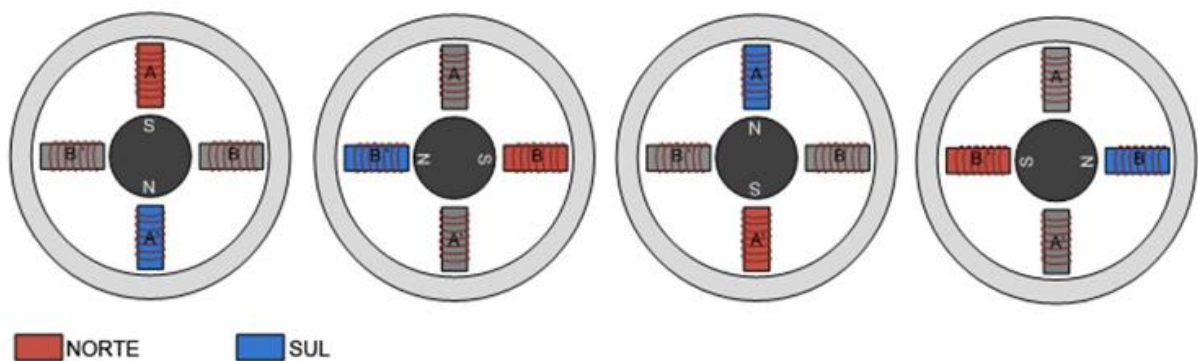




**Figura 5** – Motor de passo de ímã permanente de duas faces

Fonte: Unesp, 2013

O funcionamento é semelhante ao motor de passo RV, quando uma bobina do estator é energizada, o eixo é alinhado com o campo magnético até que o estator seja desligado e o seguinte ligado (Figura 6).

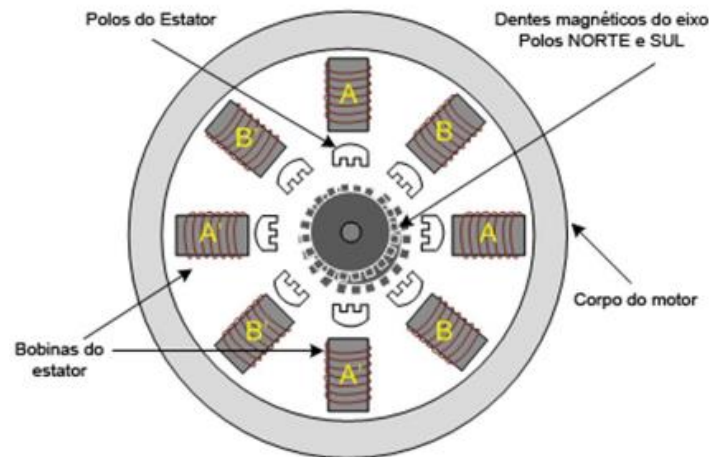


**Figura 6** - Sequência de movimento motor de passo IP

Fonte: Unesp, 2013

### 2.6.3 HÍBRIDO

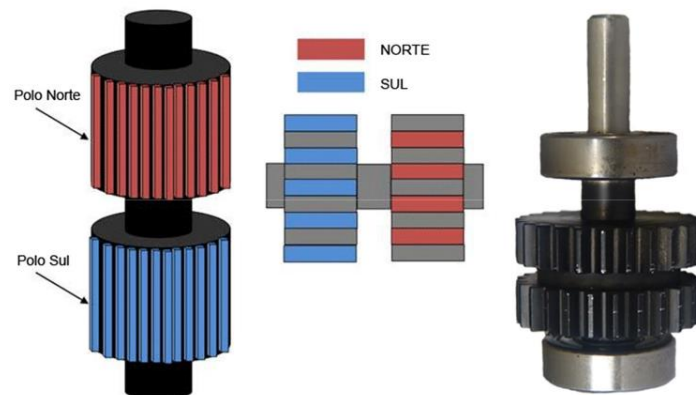
Esse tipo de motor é a mistura do motor de passo de relutância variável com a potência gerada pelo motor de passo de ímã permanente no eixo. Esse motor de passo possui um torque maior associado a precisão nos passos, que podem ter uma variação entre  $3,6^\circ$  e  $0,9^\circ$  contra  $7,5^\circ$  a  $15^\circ$  (Figura 7) em relação ao motor de passo IP (Unesp, 2013).



**Figura 7** - Motor de passo híbrido

Fonte: Unesp, 2013

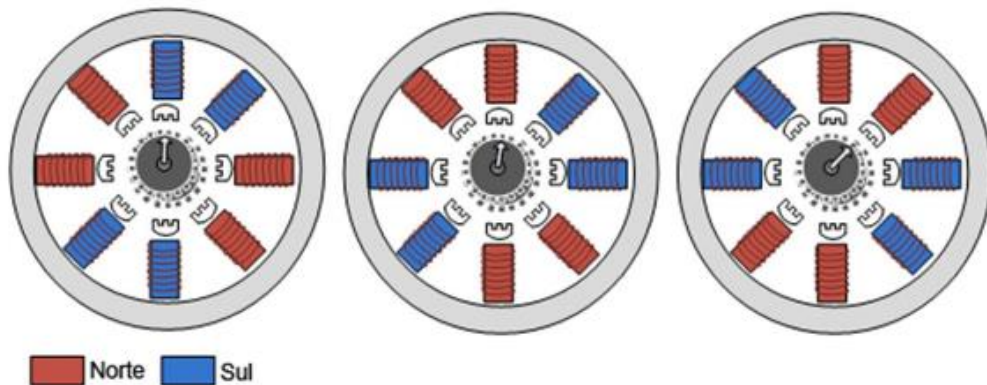
O eixo deste motor tem a sua construção em dois grupos de dentes, um com o polo sul saliente e o outro com o polo norte, os dentes devem ficar alternados conforme pode ser observado na Figura 8.



**Figura 8** - Eixo de um motor de passo híbrido

Fonte: Unesp, 2013

Sua funcionalidade é semelhante aos outros motores de passo, as bobinas devem ser ligadas na sequência para que o eixo possa girar (Figura 9).



**Figura 9** - Sequência de movimento motor de passo Híbrido

Fonte: Unesp, 2013

## 2.7 Driver Motor de Passo

Como a placa microcontrolada (modulo de saída CLP) não fornece corrente suficiente para o acionamento de qualquer motor, é necessário usar um driver ou algum tipo de circuito intermediário. Driver é um dispositivo que atua em conjunto com um controlador gerador de pulso. Este recebe os pulsos elétricos gerados pelo controlador e realiza o chaveamento dos componentes de potência para fornecer a corrente necessária para dar movimento aos motores.

Para se ter uma melhor precisão nos giros que o motor precisa fazer, utiliza-se um driver para que isso aconteça. Ou seja, o driver é a interface que controlará a corrente sobre o motor de passo, garantido assim o giro do eixo no passo que é necessário para o processo.

## 2.8 Interface Homem-Máquina

Em soluções de automação os sistemas em sua maioria são robustos e flexíveis. Quanto maior a flexibilidade do sistema, maior será a interação entre todos os dispositivos, que precisam ser mais inteligentes e eficientes (Gomes, 2018). São os dispositivos os responsáveis por todo o processo. Porém, o fator humano é de muita importância, pois este deverá monitorar de forma presencial ou remota o andamento do processo.

Para que essa interação seja realizada de forma fácil, foi desenvolvida a Interface Homem-Máquina (IHM), que é a apresentação de uma aplicação em uma tela, que interage e facilita a comunicação entre pessoas e máquinas (Figura 10). Para tanto, a IHM deve ser desenvolvida de forma que qualquer usuário compreenda o seu uso e suas funções sem muitas dificuldades.

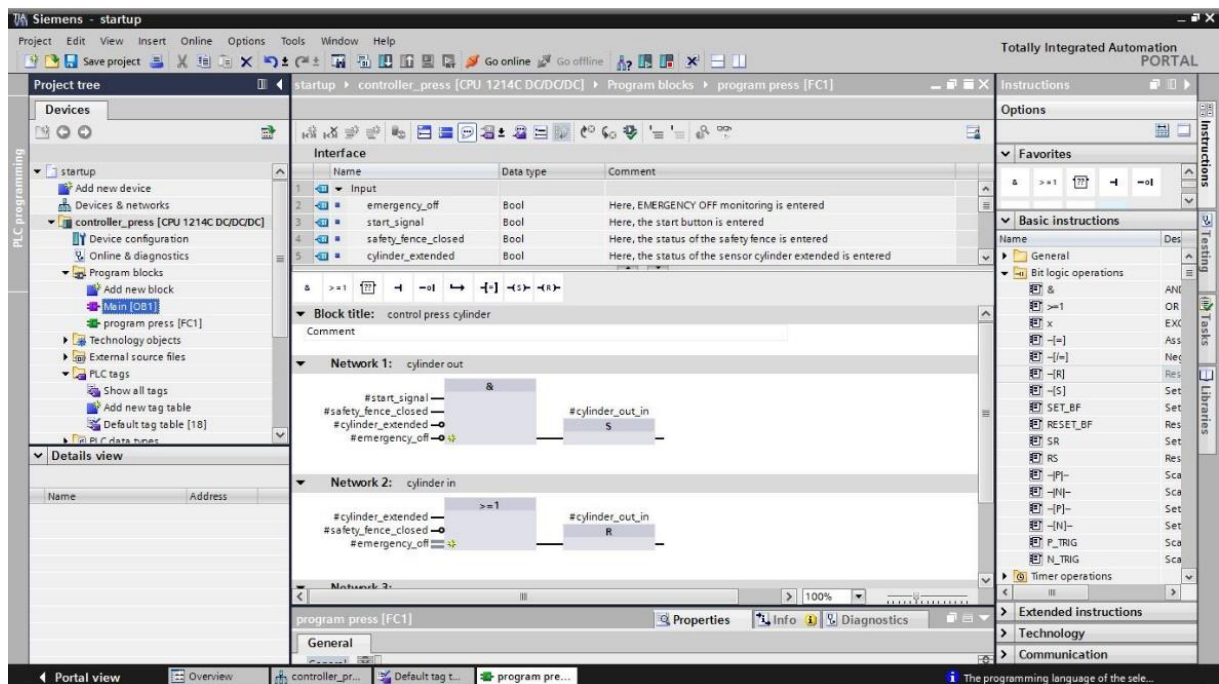


**Figura 10 - IHM (Interface Homem-Máquina)**

Fonte: Siemens, 2018

### 3 FUNCIONAMENTO DAS AUTOMATIZAÇÕES PROPOSTAS

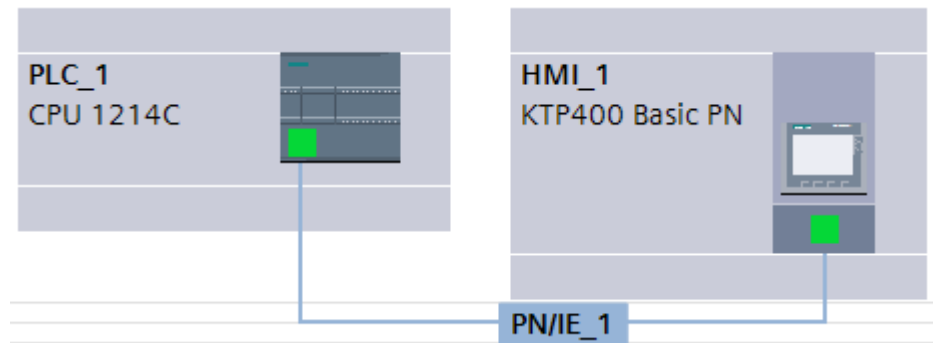
A programação será desenvolvida utilizando a plataforma de software TIA® da Siemens™. Nesta plataforma é possível o desenvolvimento da programação. Na Figura 11, é apresentada uma imagem da tela de edição do programa, para a programação do CLP e IHM.



**Figura 11 - Tela Software TIA®**

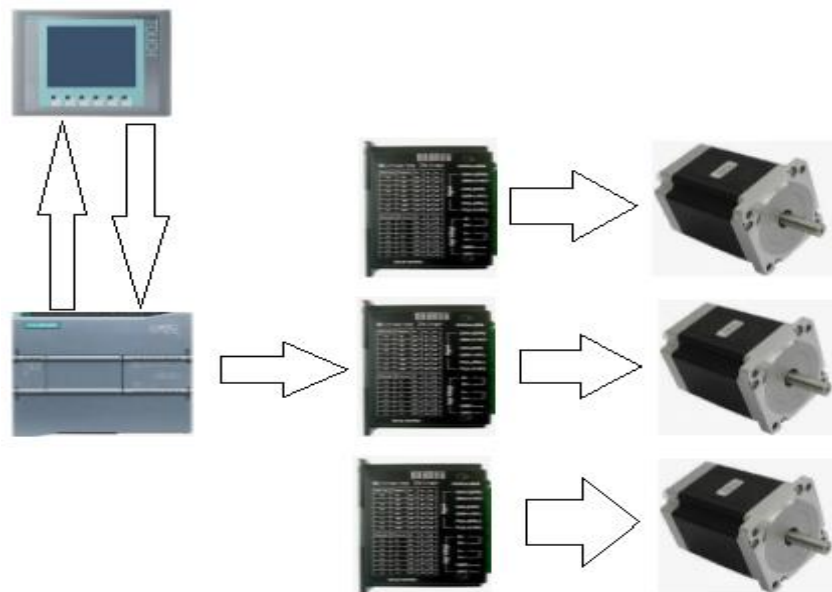
Fonte: Siemens™, 2018

A comunicação do CLP com a IHM é direta com o protocolo de comunicação Profinet, conforme Figura 12. A Figura 13 demonstra a topologia de troca de informação IHM, CLP, Drive e motor de passo.



**Figura 12** - Protocolo de comunicação CLP x IHM

Fonte: Siemens™, 2018

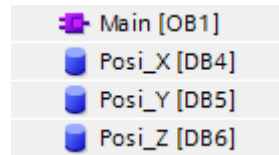


**Figura 13** - Topologia IHM, CLP, Drive e motor de passo

Fonte: Autor

Para executar uma determinada tarefa (trabalho) foi criada uma matriz, também conhecida como Array, determinando a posição, velocidade e orientação para cada objeto tecnológico, como mostrado na Figura 14. Sendo essas Arrays com seus valores de distância,

direção e velocidade editados pela IHM, assim podem ter valores variados conforme a necessidades do usuário.



**Figura 14** – Arrays

Fonte: Autor

Para um melhor entendimento e associação foi pré-definido que para o controle do drive de controle do eixo X foi criado a Array com o nome Posi\_X (Figura 15).

Posi_X		
Name	Data type	
Static		
Posicao_X	Array[0..10] of Real	
Velocidade_X	Array[0..10] of Real	

**Figura 15** - Array Posi\_X

Fonte: Autor

Para o drive de controle do eixo Y também foi criado o Array de controle com o nome Posi\_Y (Figura 16).

Posi_Y		
Name	Data type	
Static		
Posicao_Y	Array[0..10] ...	
Velocidade_Y	Array[0..10] of Real	

**Figura 16** - Array Posi\_Y

Fonte: Autor

Para o drive de controle do eixo Z também foi criado o Array de controle com o nome Posi\_Z (Figura 17).

Posi_Z		
Name		Data type
Static		
Posicao_Z		Array[0..10] ...
Velocidade_Z		Array[0..10] of Real

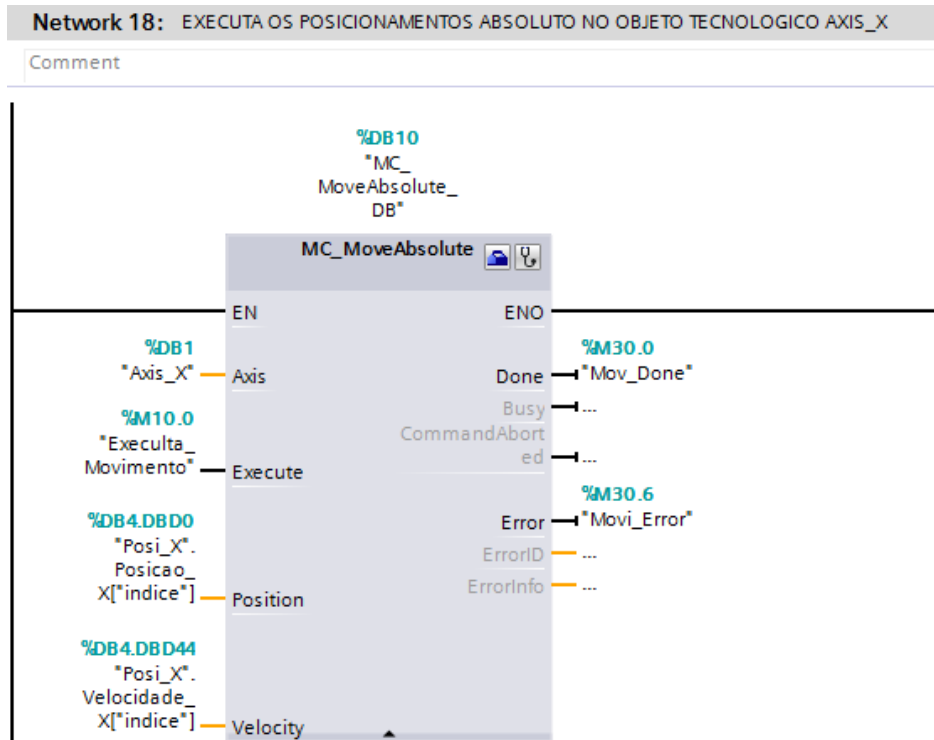
**Figura 17** - Array Posi\_Z

Fonte: Autor

Com essa Array editada via IHM podem ser inseridos quaisquer valores. Para executar uma tarefa ou trabalho foi realizado uma lógica que move cada índice da tabela para o bloco (MC\_MoveAbsolute) e este por sua vez realiza a sequência de pulso na entrada do Drive deslocando o motor para uma posição.

### 3.1 Programa Ladder para Execução dos Movimentos

O bloco tecnológico “MC\_MoveAbsolute\_DB” (Figura 18) com o valor de %m10.0 em 1 (habilitado) executa o valor de posição e orientação na variável %DB4” Posi\_X” Posicao\_X[“índice”] e a velocidade %DB4” Posi\_X” Velocidade \_X[“índice”]. A memória %M30.0 quando setada ou em valor 1 indica se o movimento foi realizado.



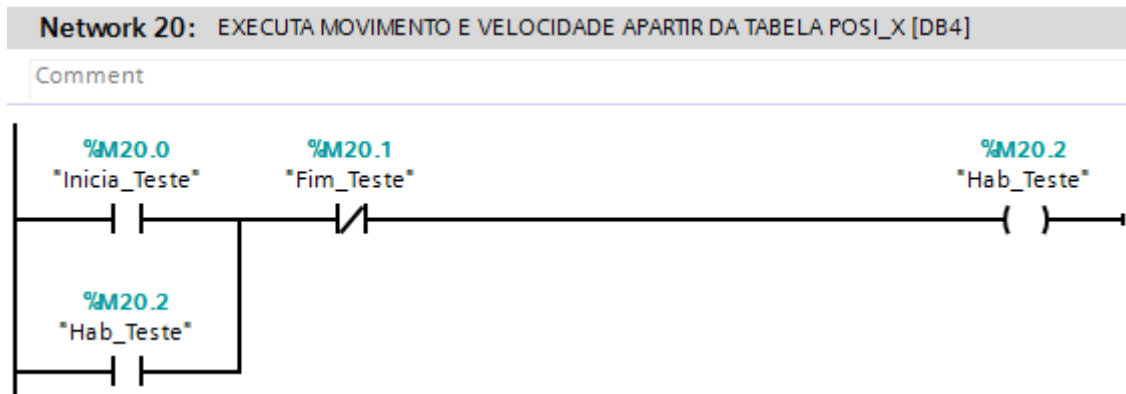
**Figura 18** - Bloco Tecnológico MC\_MoveAbsolute\_DB

Fonte: Siemens™, 2018

Para resetar o bloco e desligar a variável %M30.0 a memória %M10.0 deve ser desligada. Para iniciar um novo movimento a %M10.0 deve ser ligada novamente, na logica isso ocorre conforme o número de índice da Array.

Para o estágio inicial quando o contato da memória %M20.0 tem seu valor setado (Figura 19) via IHM passando a valer 1 e %m20.1 está em 0 (desligada) seta a memória da bobina %m20.2 que continua energizada através do seu contato de selo, dando início a execução do trabalho (movimento/testes).

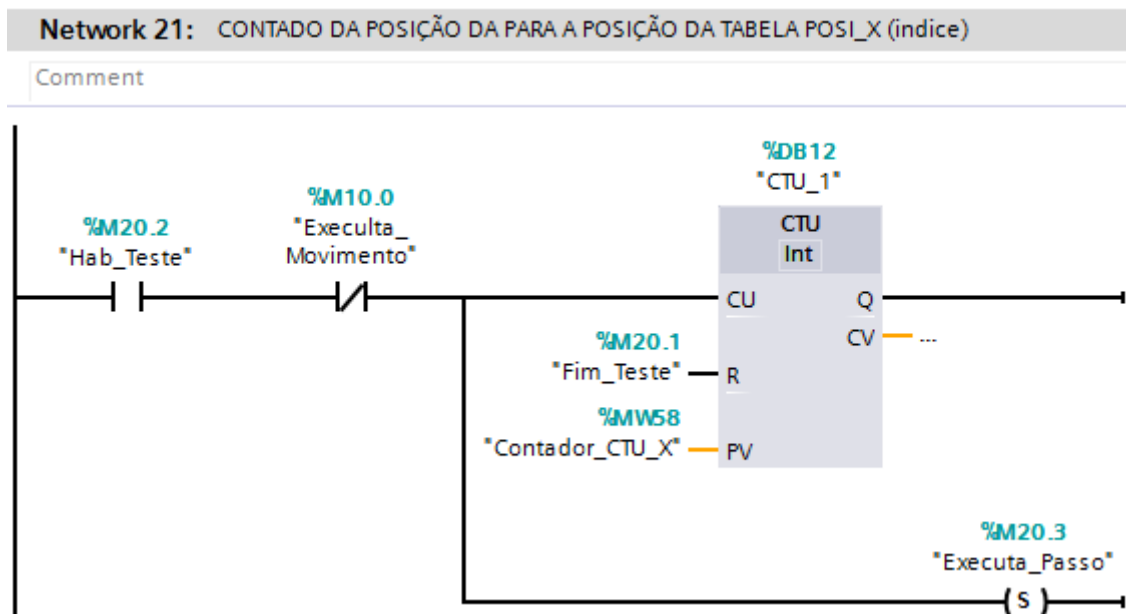




**Figura 19** - Configuração memória %M20.0

Fonte: Siemens™, 2018

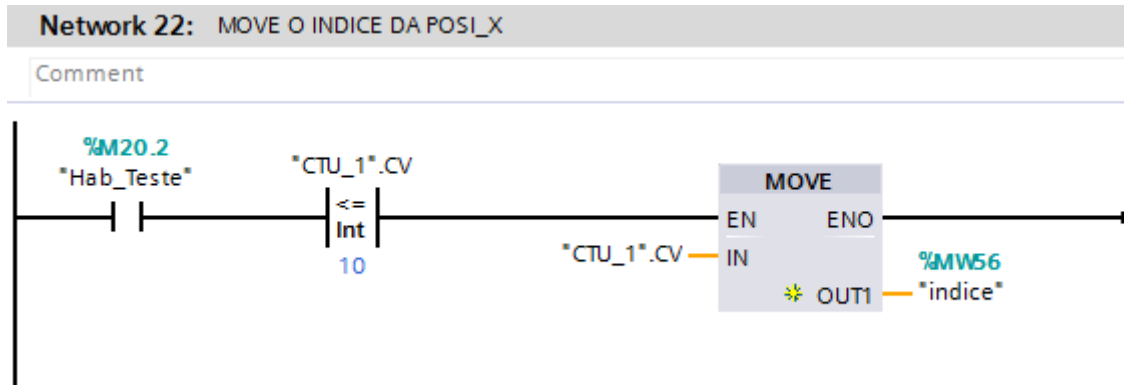
A bobina %M20.2 só será desligada quando o contato da memória %m20.1 for energizado, quando acabar de executar o processo definido pelo número de índice da tabela Posi\_X definida na memória %MW58 (Figura 20).



**Figura 20** - Contato para posição da tabela Posi\_X

Fonte: Siemens™, 2018

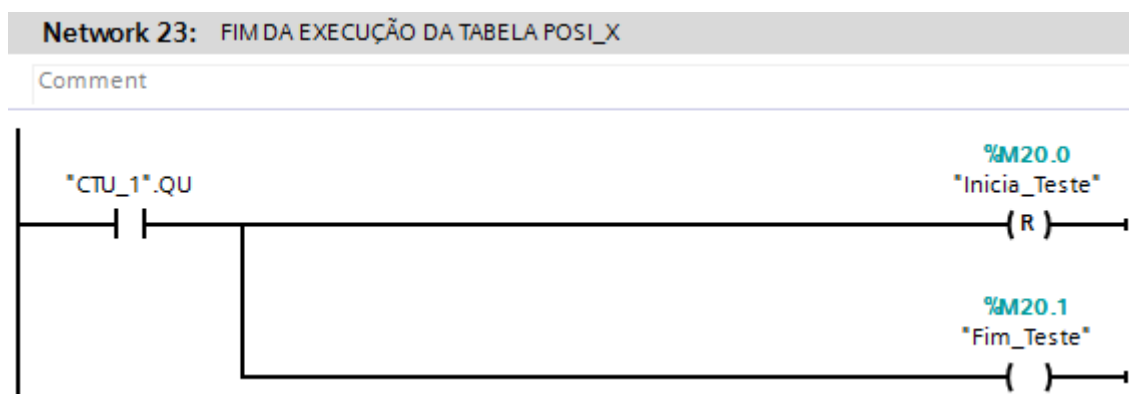
Com o contato da memória %M20.2 energizada e %M10.0 desligada muda o estado do contador que vai ser incrementado a cada variação da memória %M10.0 e também seta o valor %M20.3 que executa o índice da Posi\_X (Figura 21).



**Figura 21** - Execução do índice Posi\_X

Fonte: Siemens™, 2018

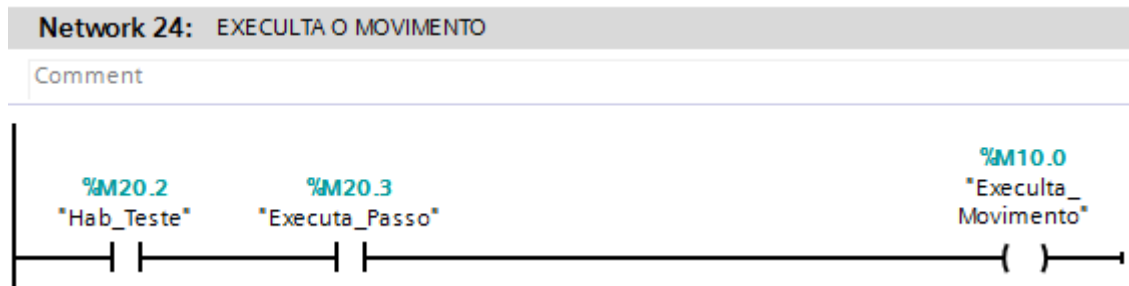
Com o contato da memória %M20.2 setado (energizado/ligado, valendo 1) e com o valor do contador valendo menor ou igual ao valor determinado na %MW58, o bloco “MOVE” move o valor contido no contador através do “CTU\_1”.CV para a variável real %MW56, esta variável contém o valor do contado, para cada valor equivale a um índice da Array que vai sendo incrementado a cada valor contado no contador, e esse valor contido no índice da Array é transferido para o objeto tecnológico “MC\_MoveAbsolute\_DB” que executa o movimento segundo a velocidade posição e orientação setado na Array via IHM (Figura 22).



**Figura 22** - Fim da Execução da Tabela Posi\_X

Fonte: Siemens™, 2018

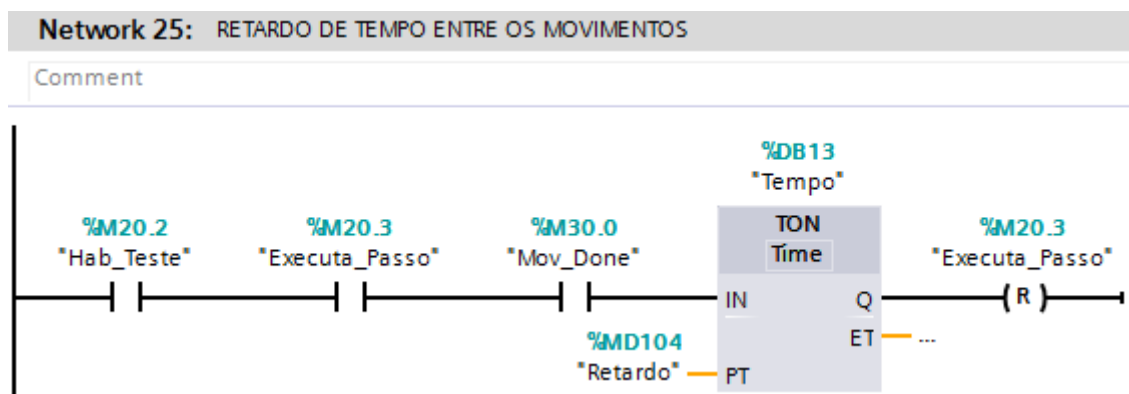
Quando o contador “CTU\_1” estoura o seu valor setando na memória %MW58 habilita o contato “CTU\_1”. QU uma vez energizado reseta as bobinas %M20.0 e %M20.1 dando fim a todo o processo de execução da tabela (Figura 23).



**Figura 23** - Execução de movimento

Fonte: Siemens™, 2018

Com o contato de %M20.2 energizado (setado em 1) e %M20.3 (setado, valendo 1) energiza a memória %M10.0, essa memória por sua vez habilita o bloco MC\_MoveAbsolute a executar o valor do índice da Array (Figura 24).



**Figura 24** - Retardo de tempo entre os movimentos

Fonte: Siemens™, 2018

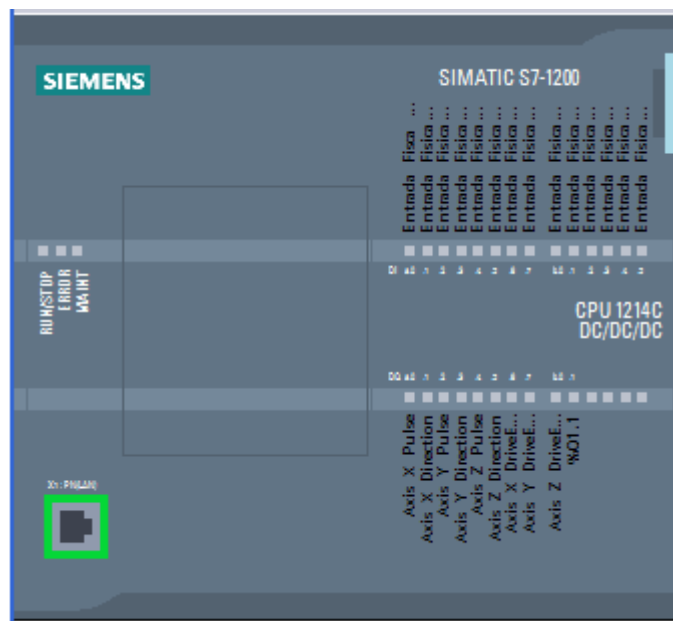
Com os contatos das memórias %M20.2, %M20.3 e %M30.0 energizados o temporizador executa um tempo pré-determinado na memória %MD104. Quando esse valor é atingido, aciona sua saída resetando a memória %M20.3, dando início novamente ao processo, desligando %M10.0 e resetando o bloco “MC\_MoveAbsolute\_DB” e desligando a memória %M30.0 e o processo reinicia novamente até que o valor do contador “CTU\_ 1” seja atingido.

Esta lógica dá um pequeno intervalo entre a execução de um movimento e o próximo, para ajudar a suavizar a parada do motor e arranque do motor evitando o desgaste prematuro do eixo e dispositivos mecânicos.

## 3.2 Configuração do CLP, Blocos e Objetos Tecnológicos

### 3.2.1 CONFIGURAÇÃO DO CLP PARA UTILIZAÇÃO DAS SAÍDAS RÁPIDAS

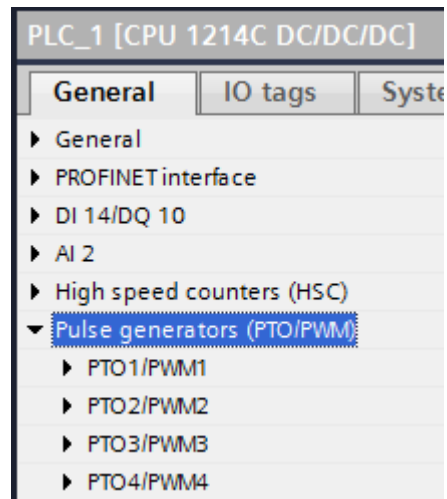
Para o projeto estamos utilizando a CPU siemens s7 1200 1214C DC/DC/DC (Figura 25), com 14 entradas digitais e 10 saídas digitais, das quais 4 são saídas rápidas de 100 kHz e as demais de 30 kHz. Para utilizar as saídas rápidas do CLP deve-se primeiro que ir em suas propriedades e depois em geral para habilitar as saídas de 100 kHz.



**Figura 25** - CLP S-7 1200 CPU 1214C DC/DC/DC

Fonte: Siemens™, 2018

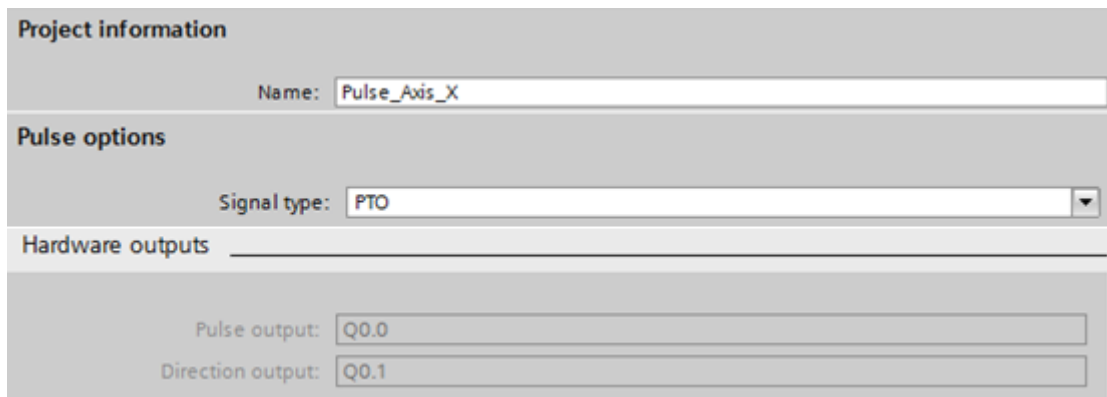
Para utilizar as saídas rápidas e configurar como PTO (saída), o próprio Software determina as saídas pelas quais cada eixo será controlado (Figura 26).



**Figura 26** - Configuração PTO

Fonte: Siemens™, 2018

Para o eixo X foi atribuída a saída Q0.0 para pulso de posicionamento e velocidade e saída Q0.1 para saída de direção horário e antiorário (para frente ou para traz, Figura 27).



**Figura 27** - Configuração PTO eixo X

Fonte: Autor

Para o controle do eixo Y ficou definido as saídas Q0.2, Q0.3. Para o controle de velocidade e posicionamento Q0.2, para o controle da direção Q0.3 (Figura 28).

**Project information**

Name:

**Pulse options**

Signal type:

**Hardware outputs**

Pulse output:

Direction output:

**Figura 28** - Configuração PTO eixo Y

Fonte: Autor

Para controle de velocidade, posicionamento e direção do eixo Z definiu as saídas rápidas de 30 kHz Q0.4 e Q0.5 (Figura 29).

**Project information**

Name:

**Pulse options**

Signal type:

**Hardware outputs**

Pulse output:

Direction output:

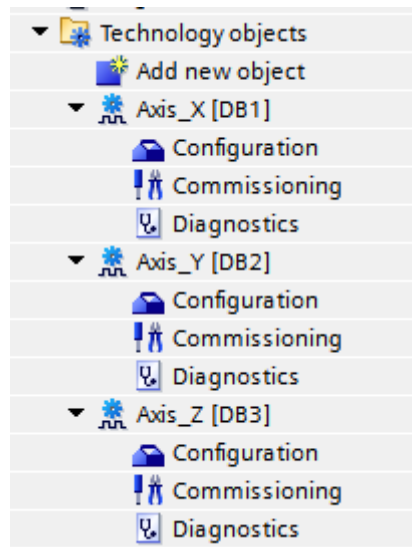
**Figura 29** - Configuração PTO eixo Z

Fonte: Autor

### 3.2.2 OBJETO DE TECNOLOGIA “AXIS” E BLOCOS DE FUNÇÃO “MOTION CONTROL”

O objeto de tecnologia “Axis” é usado para mapear um eixo no controlador, para facilitar o controle do servomotor através da interface de pulso da CPU 1214C S7-1200. O objeto tecnológico “Axis” é controlado por meio de instruções de “Controle de Movimento”. O próximo passo, depois de configurar é determinar as saídas rápidas, devem ser adicionados

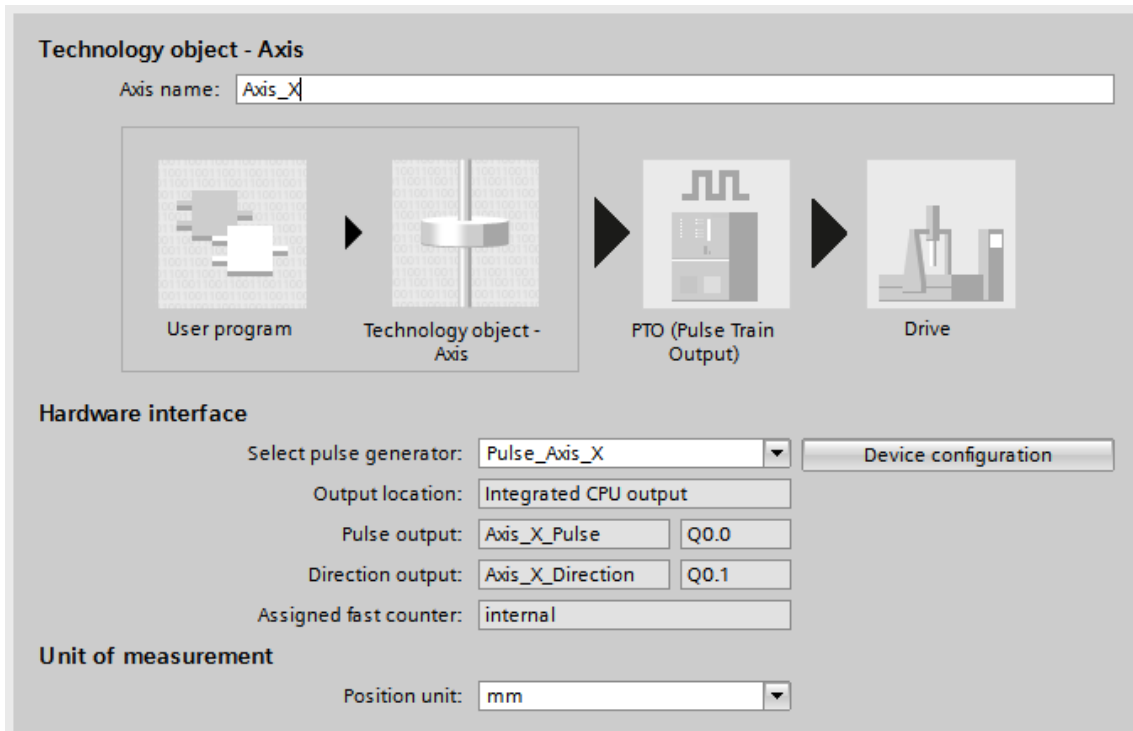
os objetos tecnológicos para o controle de cada eixo. Na Figura 30 foram criados os objetos tecnológicos para cada eixo, nomeados como: Axis\_X para o eixo X, Axis\_Y para o eixo Y e Axis\_Z para o eixo Z.



**Figura 30** - Objetos tecnológicos

Fonte: Autor

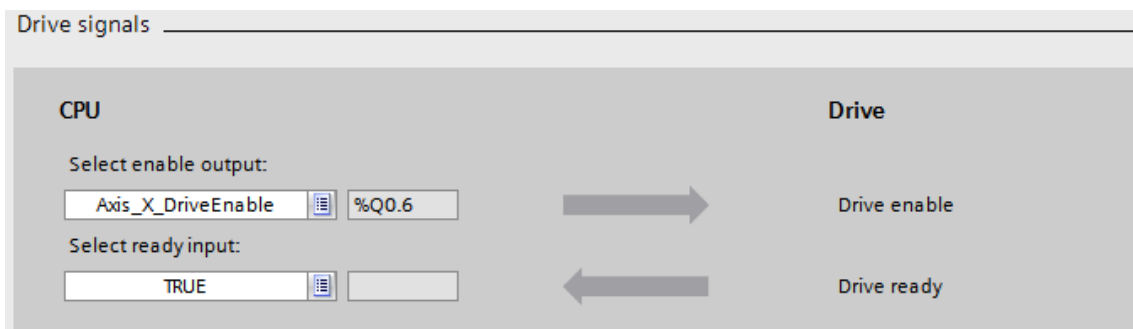
Para determinar as saídas PTO já configuradas na CPU anteriormente foi configurado o objeto tecnológico conforme Figura 31.



**Figura 31** - Configuração Axis\_X

Fonte: Autor

Determinando a saída, habilita-se o drive que controla o motor da coordenada X, conforme Figura 32.

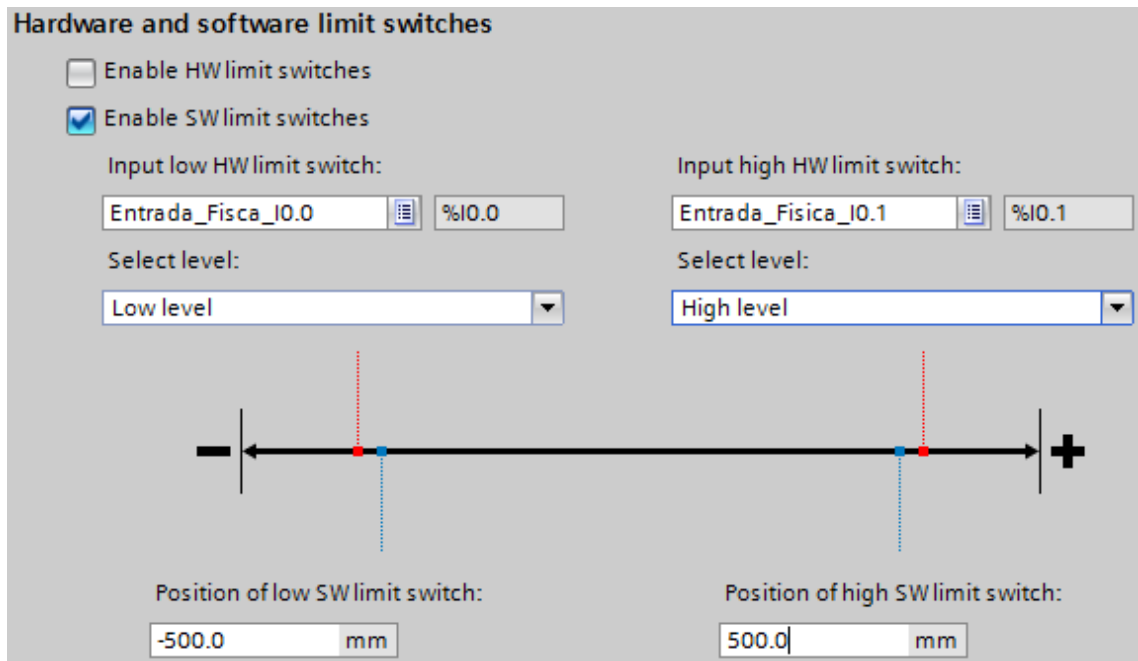


**Figura 32** - Configuração Drive Motor eixo X

Fonte: Autor

A configuração da delimitação do movimento do eixo X físico e lógico, nos limites máximo e mínimo, foi utilizada a entrada física do CLP %I0.0 para mínimo e %I0.1 para máximo (Figura 33). O sensor utilizado para realização da delimitação, foi o tipo Reed Switch (Figura 34).





**Figura 33** - Configuração de Limites eixo X

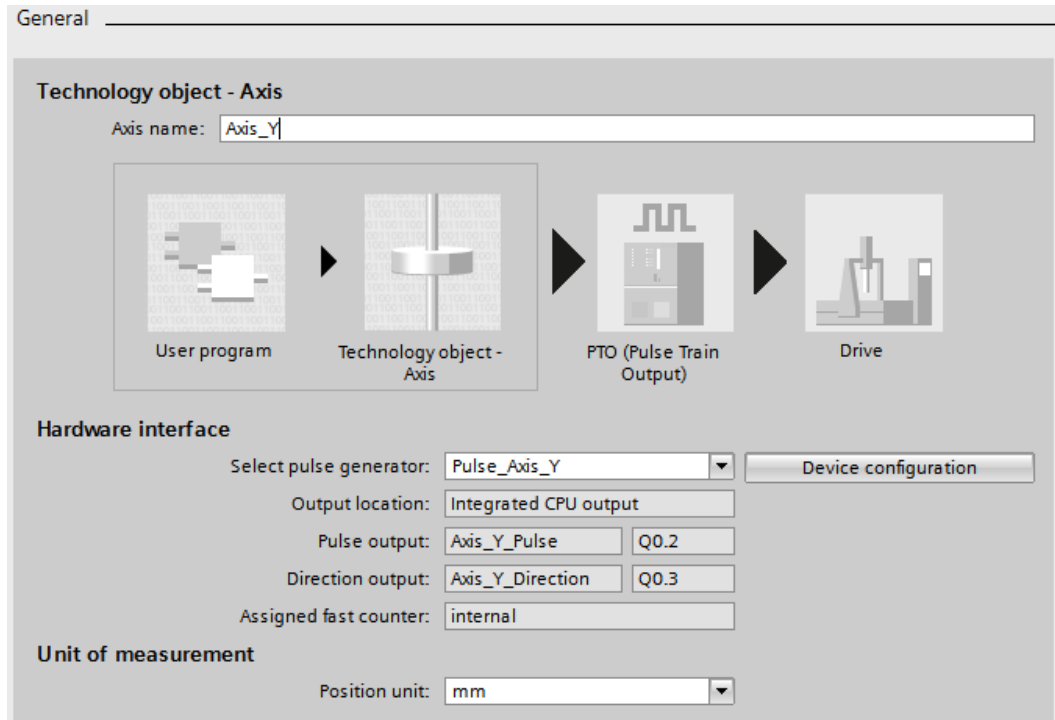
Fonte: Autor



**Figura 34** - Sensor Reed Switch

Fonte: Internet

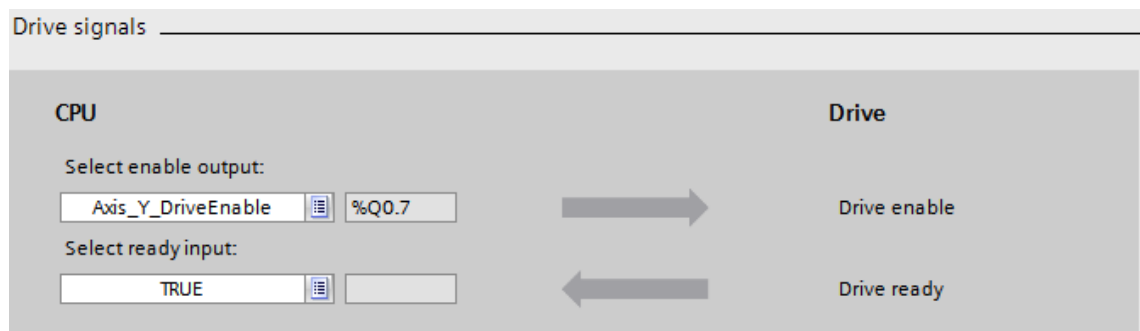
Para determinar as saídas PTO já configuradas na CPU anteriormente para o eixo Y, ficou determinada a saída para habilitar o drive do moto Y a saída Q0.2 para pulso de velocidade e posicionamento e Q0.3 para determina a orientação da direção horário e anti-horário (para frente ou para trás, Figura 35).



**Figura 35** - Configuração Axis\_Y

Fonte: Autor

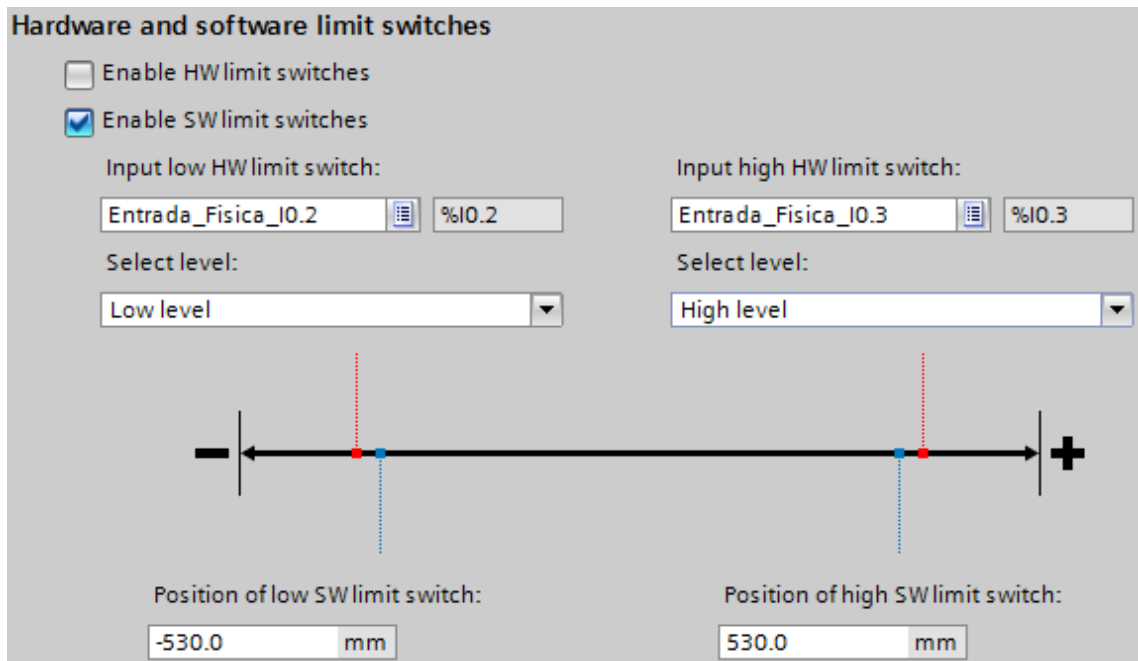
Para habilitar o drive que executa os movimento do eixo Y ficou definido a saída %Q0.7 (Figura 36).



**Figura 36** - Configuração Drive Motor eixo Y

Fonte: Autor

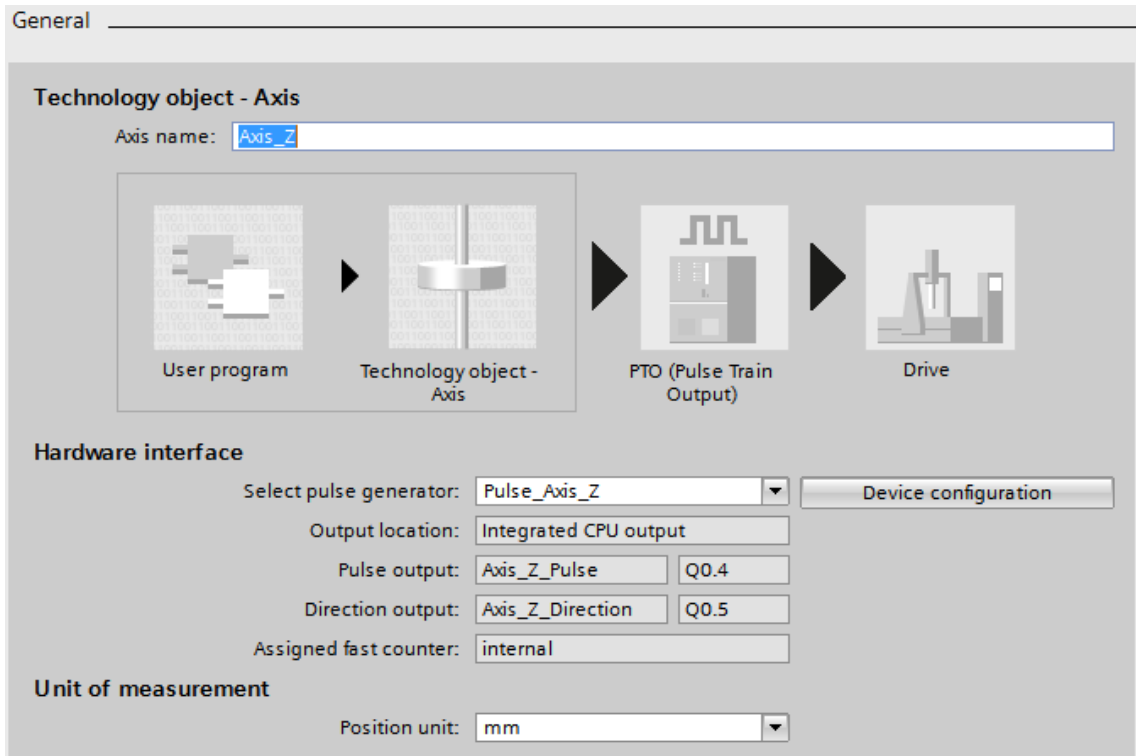
A Configuração dos limites baixo (inferior) e do limite alto (superior) para o deslocamento do eixo Y foi utilizada a configuração física e lógica no CLP, para nível baixo a entrada física %I0.2 que será atuada através de sensores, e para o nível alto %I0.3 (Figura 37). O sensor utilizado para realização da delimitação, foi o tipo Reed Switch (Figura 34).



**Figura 37** - Configuração de Limites eixo Y

Fonte: Autor

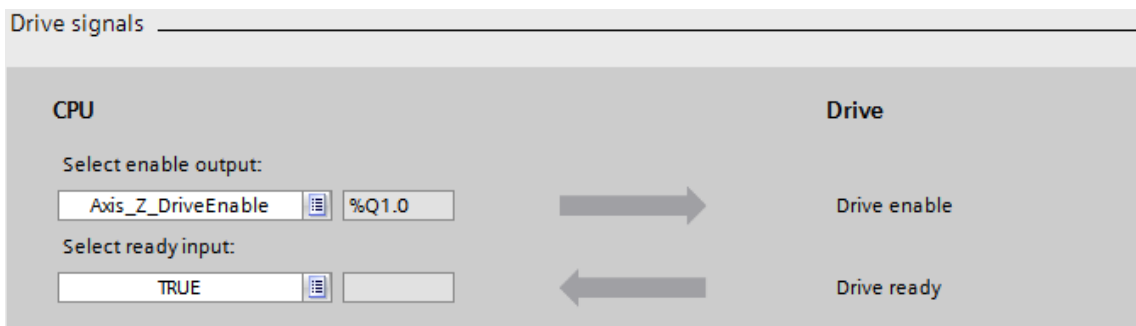
Para determinar as saídas PTO já configuradas na CPU anteriormente para acionamento do drive do eixo Z foram determinadas as modulações com pulso de 30 KHZ, sendo definidas as saídas %Q0.4 para pulso e %Q0.5 para direção (Figura 38).



**Figura 38** - Configuração Axis\_Z

Fonte: Autor

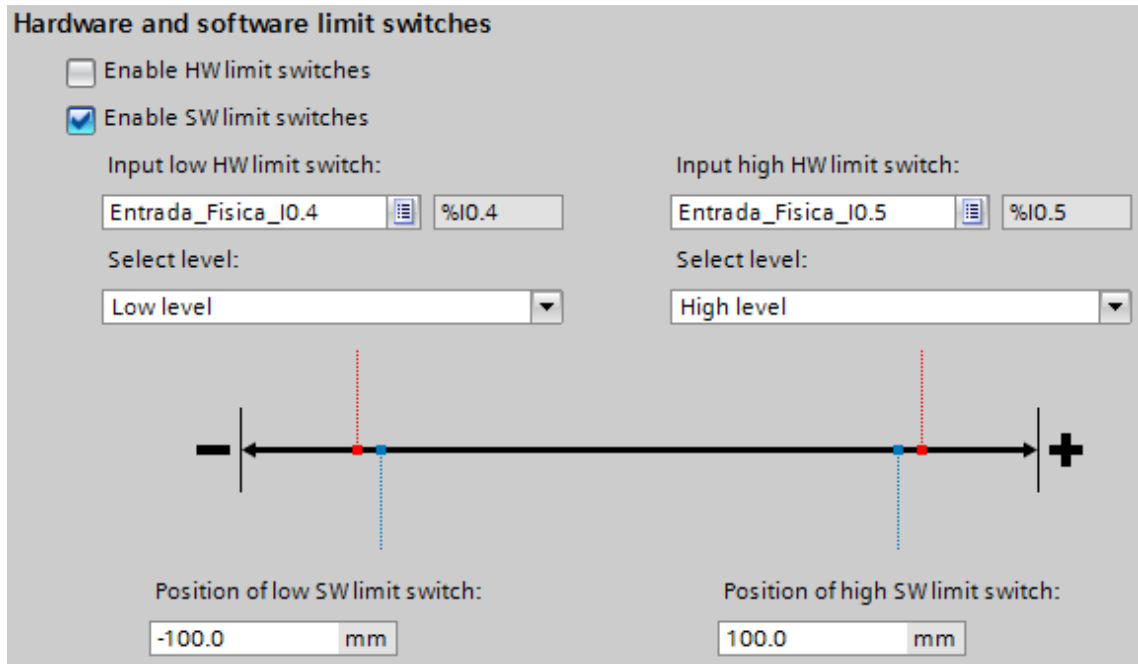
A saída para habilitar o drive de controle do motor Z ficou definido %Q1.0, conforme Figura 39.



**Figura 39** - Configuração Drive Motor eixo Z

Fonte: Autor

A Configuração dos limites baixo (inferior) e do limite alto (superior), para o deslocamento do eixo Z, ficou configurado física e logicamente no CLP, para nível baixo a entrada física %I0.4 que será atuada através sensores, e para o nível alto %I0.5 (Figura 40). O sensor utilizado para realização da delimitação, foi o tipo Reed Switch (Figura 34).



**Figura 40** - Configuração de Limites eixo Z

Fonte: Autor

### 3.2.3 BLOCOS TECNOLÓGICOS

Após criar os objetos tecnológicos para os eixos X, Y, e Z, foram adicionados os blocos tecnológicos que tem por função geral, controlar os drives para cada motor, habilitar e gerar pulso de modulação para velocidade e direção (Figura 41).

Technology	
Name	Version
Counting	V1.1
PID Control	
Motion Control	
S7-1200 Motion Con...	V3.0
MC_Power	V3.0
MC_Reset	V3.0
MC_Home	V3.0
MC_Halt	V3.0
MC_MoveAbsolute	V3.0
MC_MoveRelative	V3.0
MC_MoveVelocity	V3.0
MC_MoveJog	V3.0
MC_CommandTa...	V3.0
MC_ChangeDyna...	V3.0

**Figura 41** - Blocos Tecnológicos

Fonte: Autor

Os blocos tecnológicos para o controle do drive e conseqüentemente dos motores disponíveis constam na Tabela 3

**Tabela 3** - Definição dos Blocos Tecnológicos

Fonte: Siemens™, 2018

No.	Bloco de programa	Função
1.	MC_Power	Ativa ou Desativa o eixo
2.	MC_Reset	Reconhece todos os erros penderentes
3.	MC_Home	Posição inicial do eixo
4.	MC_MoveAbsolute	Posicionamento absoluto do eixo
5.	MC_MoveRelative	Posicionamento realtivo do eixo
6.	MC_Halt	Cancela todos os movimentos, executa “Stop” do eixo
7.	MC_CommandTable	Executa os comandos do eixo como uma sequênciade movimentos (rastreamento de perfil)
8.	MC_MoveVelocity	Move o eixo na velocidade especificada e na direção especificada
9.	MC_MoveJog	Modo de Trabalho

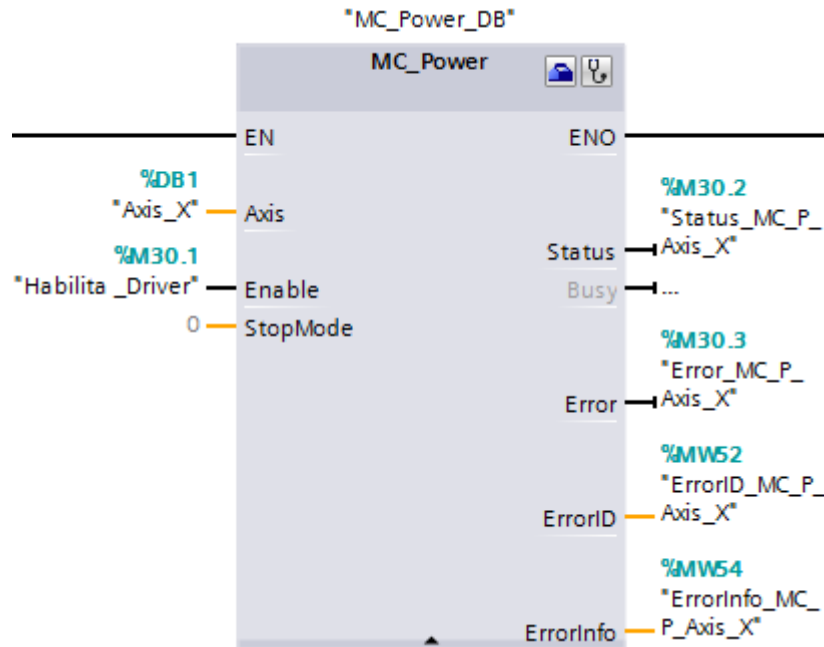
Para o projeto foram utilizados os blocos MC\_Power, MC\_Reset, MC\_MoveAbsolute e MC\_MoveRelative. Estes por sua vez serão melhor detalhados a seguir.

### 3.2.3.1 BLOCO TECNOLÓGICO MC\_POWER

Este bloco é usado para ativar/desativar o drive e conseqüentemente o Eixo do motor (Figura 42). Antes que o eixo possa ser movido, ele deve ser ativado. Quando a memória %M30.1 é “TRUE” (ou vai para o nível alto ou vale 1) é aplicada na entrada “Enable” do bloco “MC\_Power”, a saída %Q0.6 do CLP passa a valer 1 (TRUE) habilitando o drive TB6600 para job (trabalho).

A entrada “StopMode” é utilizada para definir o tipo de parada do eixo, quando estiver em “0” o eixo deve ser desacelerado pela desaceleração em rampa. Quando estiver em “1” deve ser parado imediatamente, ou seja, em modo de emergência.

A saída “Status” do bloco fornece feedback do drive e indica se a unidade está pronta para operação. Erros durante a operação são exibido na saída “Error” para alarme e o identificador de erro correspondente será exibido no Saída “ErrorID”.

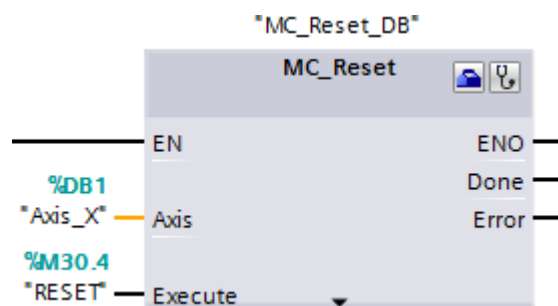


**Figura 42** - Bloco tecnológico MC\_Power

Fonte: Siemens™, 2018

### 3.2.3.2 BLOCO TECNOLÓGICO MC\_RESET

Se um erro reconhecível ocorreu, ele deve ser redefinido por uma borda positiva na entrada “Execute” do bloco “MC\_Reset”, que tem a função de resetar o erro para reiniciar o drive novamente (Figura 43).

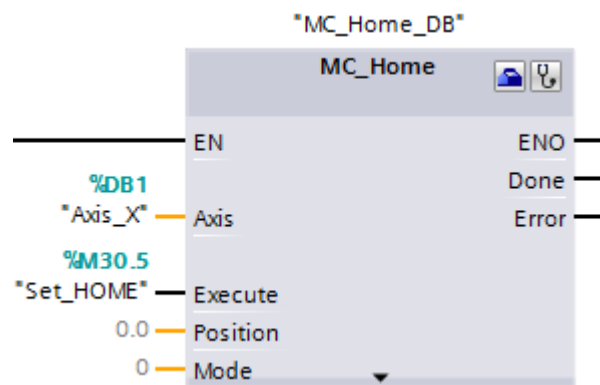


**Figura 43** - Bloco tecnológico MC\_Reset

Fonte: Siemens™, 2018

### 3.2.3.3 BLOCO TECNOLÓGICO MC\_HOME

Antes que o motor possa ser movido para executar uma operação (trabalho ou deslocamento), deve-se determinar sua posição inicial ou posição 0. Isso é possível através do objeto tecnológico MC\_HOME que seta para o CLP sua posição atual como sendo a inicial. Após a execução do trabalho (tarefa/movimento) pode-se retornar à posição inicial para realizar nova tarefa ou trabalho (Figura 44).



**Figura 44** - Bloco tecnológico MC\_Home

Fonte: Siemens™, 2018

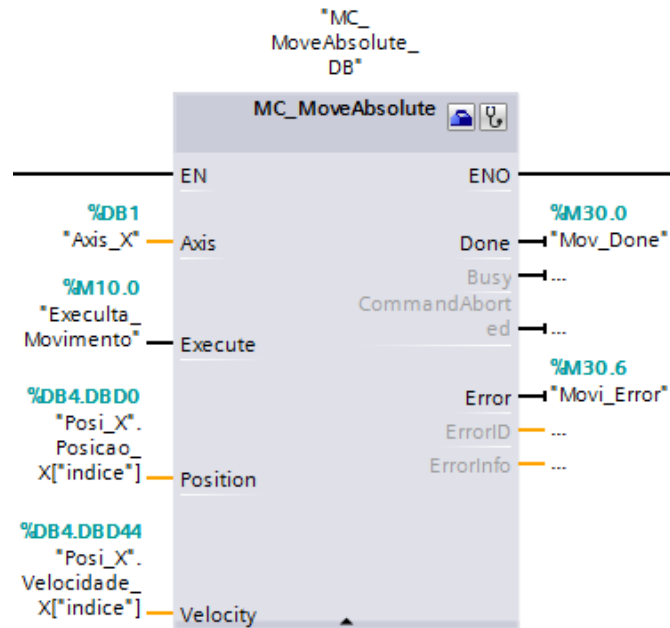
### 3.2.3.4 BLOCO TECNOLÓGICO MC\_MOVEABSOLUTE

Com a ajuda do "MC\_MoveAbsolute", quaisquer posições dentro dos limites mecânicos podem ser aproximadas especificando a posição real em [mm]. Além disso, a velocidade da viagem (ou trajeto/ movimento) tem que ser também definida para sua execução (Figura 45).

Quando o bloco é iniciado por uma borda positiva na sua entrada "EXECUTE", o número de pulsos necessários para atingir a posição alvo é calculado com base na posição atual e a posição alvo. O motor é acelerado com a velocidade especificada durante a viagem (ou trajeto/movimento) na posição desejada.

Sem dúvidas esse bloco, para o projeto é o mais importante, pois é através dele que pode ser determinado ao motor o que deve ser realizado.



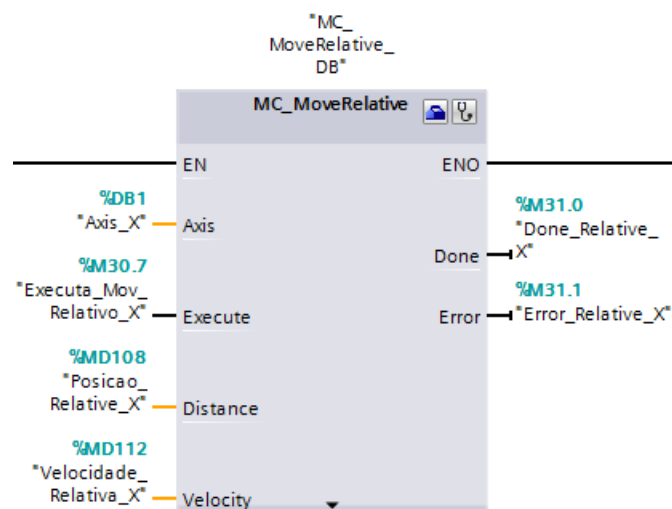


**Figura 45** - Bloco Tecnológico MC\_MoveAbsolute

Fonte: Siemens™, 2018

### 3.2.3.5 BLOCO TECNOLÓGICO MC\_MOVERELATIVE

Além do posicionamento absoluto, um movimento relativo com qualquer distância, direção e velocidade pode ser realizada com o auxílio do bloco “MC\_MoveRelative” (Figura 46). Quando o bloco é iniciado por uma borda positiva na entrada “EXECUTE”, o eixo é movido para a posição e velocidade definhada. Para indicar a direção deve-se usar o sinal (-/+).



**Figura 46** - Bloco Tecnológico MC\_MoveRelative

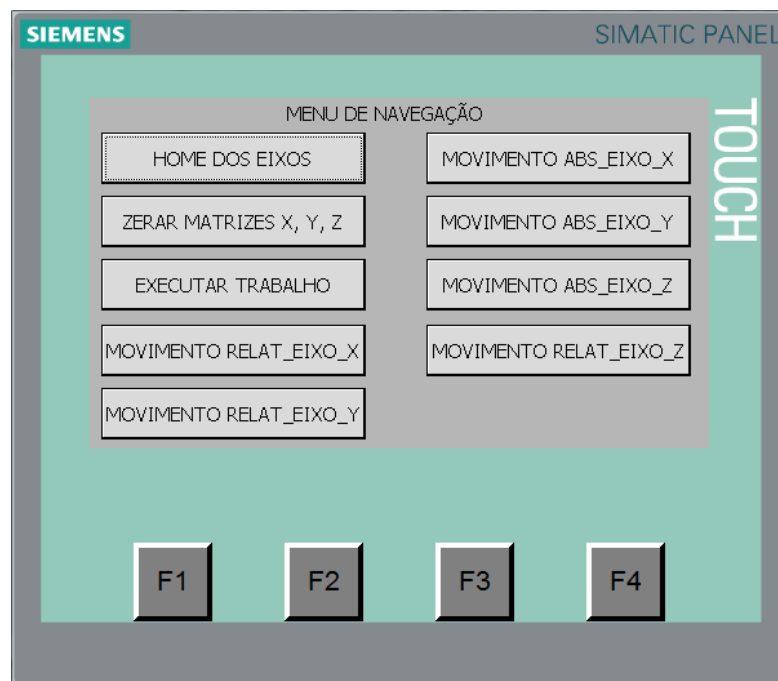
Fonte: Siemens™, 2018

### 3.3 Configuração da Interface Homem-Máquina (IHM)

A tela inicial de navegação (Figura 47), é iniciada na IHM ao ligar, conhecida também como Root Screen, é a primeira tela que aparecera para o operador.

Nela contém 9 botões que tem por função chamar as próximas telas ou screens. Para cada botão da tela inicial é atribuído uma funcionalidade, sendo as chamadas de tela para:

- **HOME DOS EIXO:** posicionamento dos eixos;
- **ZERAR MATRIZES X, Y, Z:** zerar as matrizes;
- **EXECUTAR TRABALHO:** iniciar trabalho;
- **MOVIMENTO RELAT\_X:** execução do movimento relativo do eixo x;
- **MOVIMENTO RELAT\_Y:** execução do movimento relativo do eixo y;
- **MOVIMENTO RELAT\_Z:** execução do movimento relativo do eixo z;
- **MOVIMENTO ABS\_EIXO\_X:** execução do movimento absoluto do eixo x;
- **MOVIMENTO ABS\_EIXO\_Y:** execução do movimento absoluto do eixo y;
- **MOVIMENTO ABS\_EIXO\_Z:** execução do movimento absoluto do eixo z.



**Figura 47** - Tela *Root Screen*

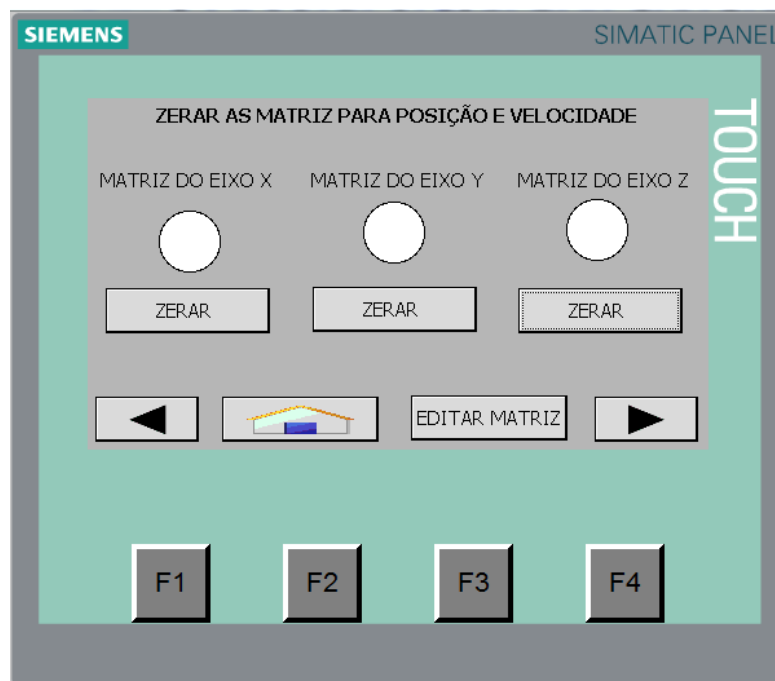
Fonte: Autor

### 3.3.1 TELA ZERAR MATRIZES X, Y, Z

A tela que é acionada ao pressionar o botão “ZERAR MATRIZES X, Y, Z” na Root Screen (Figura 47) é para zera todos os valores das matrizes Posi\_X, Posi\_Y e Posi\_Z e chamar a tela de edição das matrizes (Figura 48). Possui três objetos em círculos para cada matriz, que mudam suas cores de vermelho para verde, onde vermelho indica que as matrizes estão com valores e verde indica que estão zeradas, os objetos em círculos servem para simular uma sinaleira de status do processo, mudam suas cores conforme estabelecido na sua parametrização.

Os botões na tela têm as seguintes funcionalidades:

- **ZERAR:** zera os valores das matrizes
- **Botão com seta direita:** com esse botão é possível avançar para a próxima tela executar trabalho;
- **Botão com seta esquerda:** com esse botão é possível retornar para a tela anterior;
- **Botão com a imagem casa:** é possível retornar a tela inicial da IHM;
- **Botão editar matriz:** chama a tela edição das matrizes.



**Figura 48** - Tela para Zerar Matrizes de Posicionamento/Velocidade

Fonte: Autor

### 3.3.1.2 TELA EDITAR MATRIZ

Essa tela é para a edição da matriz que controla a velocidade, posição e orientação (Figura 49). A caixa de diálogo de campo é utilizada, entre outras funções, para editar o valor de uma variável de entrada ou de saída. Sendo cada componente da tela:

- **Caixa de diálogo campo índice:** escolhe e índice da matriz a ser editado;
- **Caixa de diálogo campo posição:** o valor da posição deslocamento do eixo em milímetros;
- **Caixa de diálogo campo velocidade:** a velocidade para o índice do eixo escolhido;
- **Botão com seta direita:** com esse botão é possível avançar para a próxima tela executar trabalho;
- **Botão com seta esquerda:** com esse botão é possível retornar para a tela anterior;
- **Botão com a imagem casa:** é possível retornar a tela inicial;
- **Botão retorna:** esse botão retorna a tela que a chamou, caso o operador digitou um valor errado ou deseja zerar a matriz novamente, esse botão chama a tela que o chamou.

Após a inserção dos valores nos campos índice, posição e velocidade é necessário salvar, isso deve ser feito utilizando os botões correspondentes para cada caixa de diálogo. Essa tela serve para lembrar o operado de zerar as matrizes antes de editar seus campos.



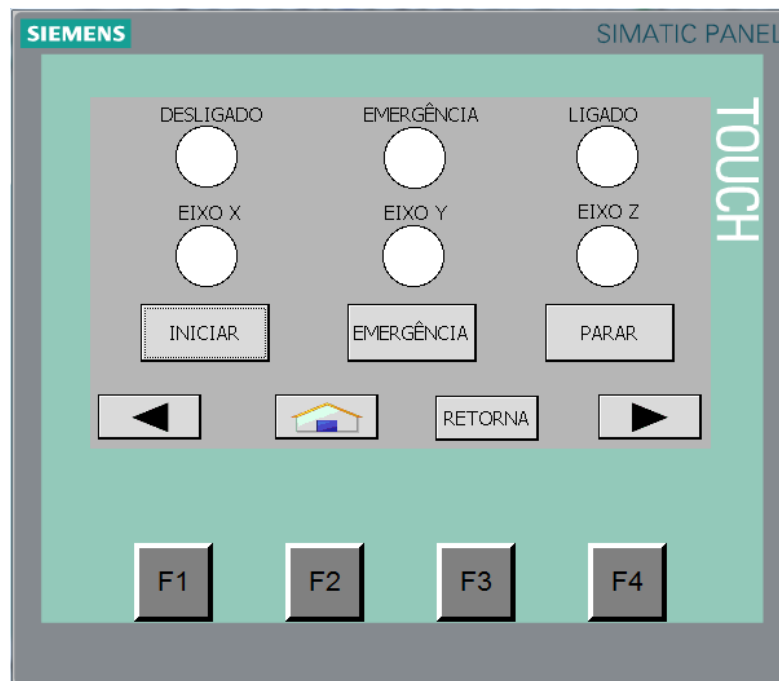
**Figura 49** - Tela de Edição dos Valores de Índice, Velocidade e Posição dos Eixos X, Y e Z

Fonte: Autor

### 3.3.2 TELA EXECUTAR TRABALHO

Tela para execução do trabalho contendo os botões iniciar, parar, acionar emergência e os objetos em círculos que indicam para o operador em qual estágio a mesa está o processo e quais eixos estão em movimento (Figura 50). Sendo as funcionalidades dos botões e os objetos na tela:

- **Botão iniciar:** iniciar o ciclo de operação;
- **Botão emergência:** aciona a parada de emergência;
- **Botão parar:** para o processo;
- **Sinaleiro desligado:** quando verde indica que o processo está parado;
- **Sinaleira ligado:** quando em vermelho indica que o processo foi inicializado;
- **Botão com seta direita:** chama a próxima tela HOME\_EIXO;
- **Botão com seta esquerda:** retorna a tela anterior ZERAR\_MATRIZ;
- **Botão com a casa:** retornar a tela inicial;
- **Botão retorna:** retorna a tela que a chamou.



**Figura 50** - Tela de Início do Trabalho

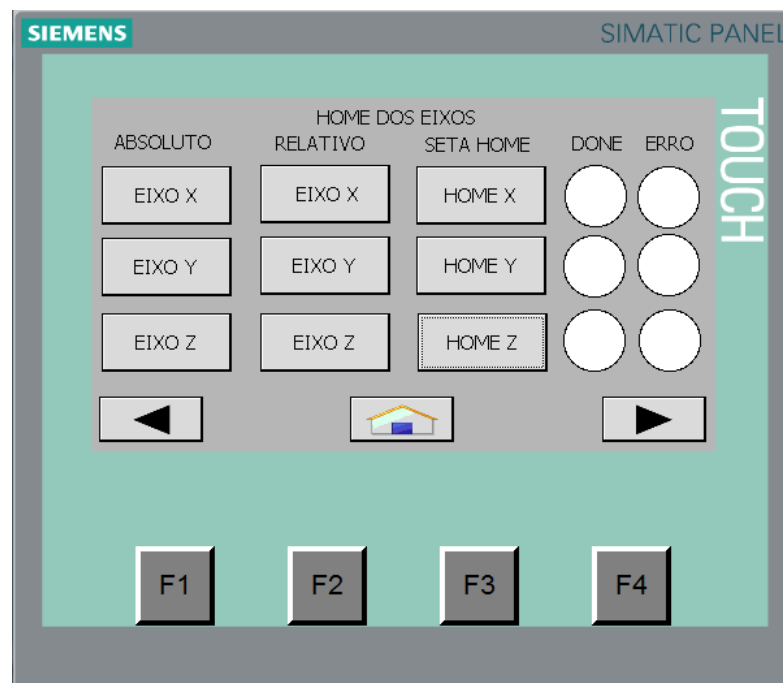
Fonte: Autor

### 3.3.3 TELA HOME DOS EIXO

Antes de iniciar qualquer trabalho o operador deve primeiro realizar o home dos eixos, isso significa que deve primeiro dizer para o controlador qual é a posição inicial dos eixos X, Y e Z (Figura 51).

O operador escolhe qual o eixo e qual tipo de movimento ele vai utilizar para levar o eixo até a posição inicial. A escolha ocorre clicando nos botões desejados que tem por função chamar a tela de movimento relativo e absoluto, posiciona o eixo e depois volta e seta sua posição clicando nos botões HOME para o eixo que posicionou. Sendo cada componente da tela:

- **Os sinaleiros DONE:** em verde indica se o eixo foi setado;
- **Os sinaleiros ERRO:** em vermelho indica que ocorreu uma falha e não foi possível executar o home;
- **Botão com seta direita:** chama a próxima tela MOVIMENTO\_ABS\_EIXO\_X;
- **Botão com seta esquerda:** retorna a tela anterior HOME\_EIXOS;
- **Botão com a imagem casa:** retornar a tela inicial.



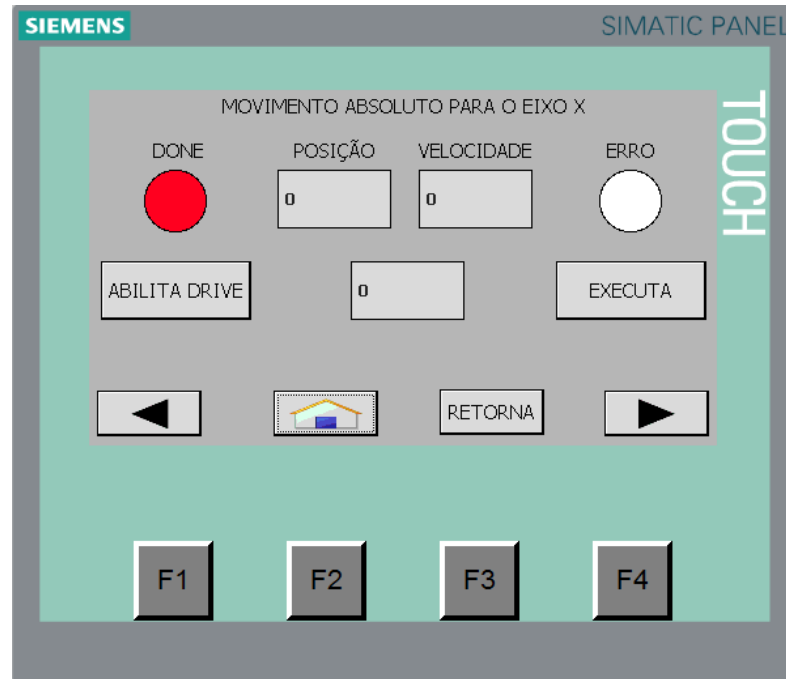
**Figura 51** - Tela Home dos Eixos

Fonte: Autor

### 3.3.4 TELA MOVIMENTO ABSOLUTO PARA O EIXO X

Executa o movimento absoluto para o eixo x para qualquer valor desejado de posição, velocidade e direção, isso é possível através da caixa de diálogo campo posição e velocidade (Figura 52). Sendo cada componente da tela:

- **Botão HABILITA DRIVE:** seta a memória na entrada da porta “EXECUTE” do objeto tecnológico "MC\_MoveAbsolute\_DB" que aciona a saída do controlador habilitando o drive para execução do movimento;
- **Caixa de diálogo campo posição:** move o valor digitado para o primeiro índice da matriz Posi\_X, referente a posição, na entrada “POSITION” do bloco "MC\_MoveAbsolute\_DB" do Axis\_X, responsável por executar o movimento digitado, transferindo em pulso para o drive que aciona o eixo movendo o para a posição;
- **Caixa de diálogo velocidade:** move o valor digitado para o primeiro índice da matriz Posi\_X, referente a velocidade, na entrada “VELOCITY” do bloco "MC\_MoveAbsolute\_DB" do Axis\_X controlando a velocidade do deslocamento do eixo;
- **Botão executa movimento:** seta a memória da entrada “EXECUTE” do bloco "MC\_MoveAbsolute\_DB” que executa o posicionamento, orientação e a velocidade setada nas suas entradas;
- **Botão com seta direita:** chama a próxima tela MOVIMENTO\_ABS\_EIXO\_Y;
- **Botão com seta esquerda:** retorna a tela anterior HOME\_EIXOS;
- **Botão com a imagem casa:** retornar a tela inicial;
- **Botão retorna:** retorna a tela que a chamou.



**Figura 52** -Tela Movimento Absoluto para o Eixo X

Fonte: Autor

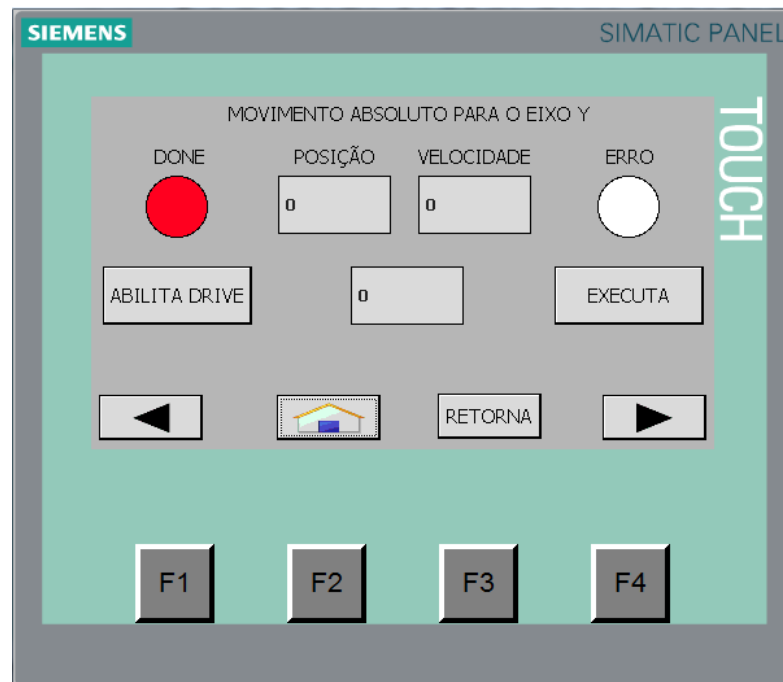
### 3.3.5 TELA MOVIMENTO ABSOLUTO PARA O EIXO Y

Executa o movimento absoluto para o eixo, para qualquer valor desejado de posição, velocidade e direção, isso é possível através da caixa de diálogo campo posição e velocidade (Figura 53). Sendo cada componente da tela:

- **Botão HABILITA DRIVE:** seta a memória na entrada da porta “EXECUTE” do objeto tecnológico "MC\_MoveAbsolute\_DB" do Axis\_Y que aciona a saída do controlador habilitando o drive para execução do movimento;
- **Caixa de diálogo campo posição:** move o valor digitado para o primeiro índice da matriz Posi\_Y, referente a posição, na entrada “POSITION” do bloco "MC\_MoveAbsolute\_DB" Axis\_Y responsável por executar o movimento digitado, transferindo em pulso para o drive que aciona o eixo Y, movendo o para a posição;
- **Caixa de diálogo velocidade:** move o valor digitado para o primeiro índice da matriz Posi\_X, referente a velocidade, na entrada “VELOCITY” do bloco "MC\_MoveAbsolute\_DB, controlando a velocidade do deslocamento do eixo Y;
- **Botão executa movimento:** seta a memória da entrada “EXECUTE” do bloco "MC\_MoveAbsolute\_DB” que executa o posicionamento, orientação e a velocidade setada nas suas entradas;



- **Botão com seta direita:** chama a próxima tela MOVIMENTO\_ABS\_EIXO\_Z;
- **Botão com seta esquerda:** retorna a tela anterior MOVIMENTO\_ABS\_EIXO\_X;
- **Botão com a imagem casa:** retornar a tela inicial;
- **Botão retorna:** retorna a tela que a chamou.



**Figura 53** - Tela Movimento Absoluto para o Eixo Y

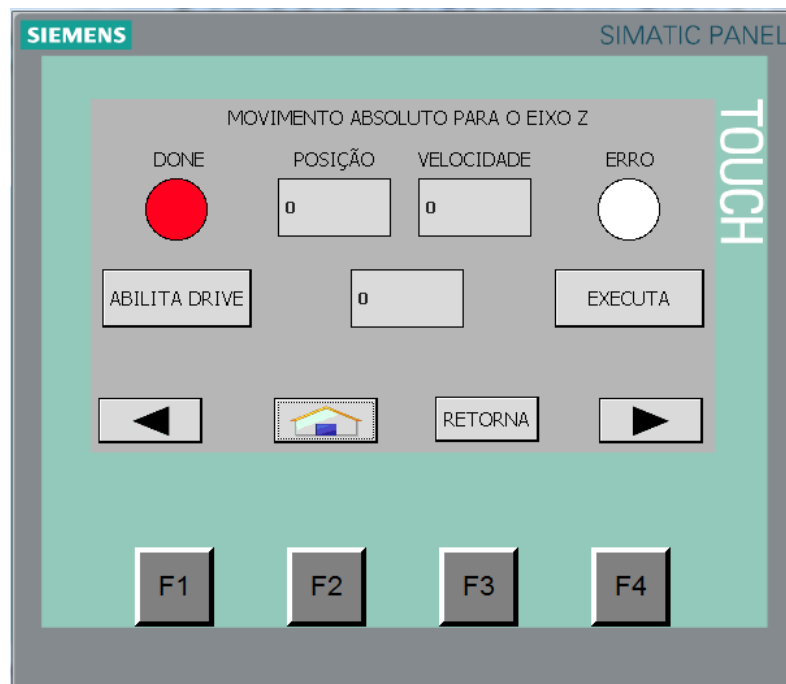
Fonte: Autor

### 3.3.6 TELA MOVIMENTO ABSOLUTO PARA O EIXO Z

Executa o movimento absoluto para o eixo Z, para qualquer valor desejado de posição, velocidade e direção, isso é possível através da caixa de diálogo campo posição e velocidade (Figura 54). Sendo cada componente da tela:

- **Botão HABILITA DRIVE:** seta a memória na entrada da porta “EXECUTE” do objeto tecnológico "MC\_MoveAbsolute\_DB" do Axis\_Z que aciona a saída do controlador habilitando o drive para execução do movimento;
- **Caixa de diálogo campo posição:** move o valor digitado para o primeiro índice da matriz Posi\_Z, referente a posição, na entrada “POSITION” do bloco "MC\_MoveAbsolute\_DB" Axis\_Z responsável por executar o movimento digitado, transferindo em pulso para o drive que aciona o eixo Z, movendo o para a posição;

- **Caixa de diálogo velocidade:** move o valor digitado para o primeiro índice da matriz Posi\_Z, referente a velocidade, na entrada “VELOCITY” do bloco "MC\_MoveAbsolute\_DB, controlando a velocidade do deslocamento do eixo Z;
- **Botão executa movimento:** seta a memória da entrada “EXECUTE” do bloco "MC\_MoveAbsolute\_DB” do Axis\_Z que executa o posicionamento, orientação e a velocidade setada nas suas entradas;
- **Botão com seta direita:** chama a próxima tela MOVIMENTO\_RELAT\_EIXO\_X;
- **Botão com seta esquerda:** retorna a tela anterior MOVIMENTO\_ABS\_EIXO\_Y;
- **Botão com a casa:** retornar a tela inicial.
- **Botão retorna:** retorna a tela que a chamou.



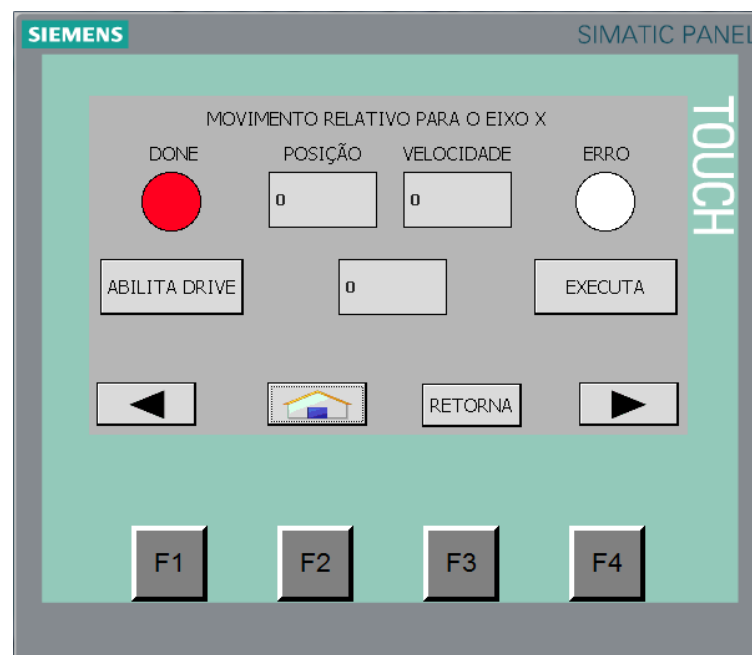
**Figura 54** - Tela Movimento Absoluto para o Eixo Z

Fonte: Autor

### 3.3.7 TELA MOVIMENTO RELATIVO PARA O EIXO X

Executa o movimento relativo para o eixo X, para qualquer valor desejado de posição, velocidade e direção, isso é possível através da caixa de diálogo campo posição e velocidade (Figura 55). Sendo cada componente da tela:

- **Botão HABILITA DRIVE:** seta a memória na entrada da porta “EXECUTE” do objeto tecnológico "MC\_MoveRelative\_DB" do Axis\_X que aciona a saída do controlador habilitando o drive para execução do movimento;
- **Caixa de diálogo campo posição:** move o valor digitado para o primeiro índice da matriz Posi\_X, referente a posição, na entrada “POSITION” do bloco "MC\_MoveRelative\_DB" do Axis\_X responsável por executar o movimento digitado, transferindo em pulso para o drive que aciona o eixo X, movendo o para a posição;
- **Caixa de diálogo velocidade:** move o valor digitado para o primeiro índice da matriz Posi\_X, referente a velocidade, na entrada “VELOCITY” do bloco "MC\_MoveRelative\_DB" do Axis\_X controlando a velocidade do deslocamento do eixo Z;
- **Botão executa movimento:** seta a memória da entrada “EXECUTE” do bloco "MC\_MoveRelative\_DB" do Axis\_X que executa o posicionamento, orientação e a velocidade setada nas suas entradas;
- **Botão com seta direita:** chama a próxima tela MOVIMENTO\_RELAT\_EIXO\_Y;
- **Botão com seta esquerda:** retorna a tela anterior MOVIMENTO\_ABS\_EIXO\_Z;
- **Botão com a imagem casa:** retornar a tela inicial;
- **Botão retorna:** retorna a tela que a chamou.



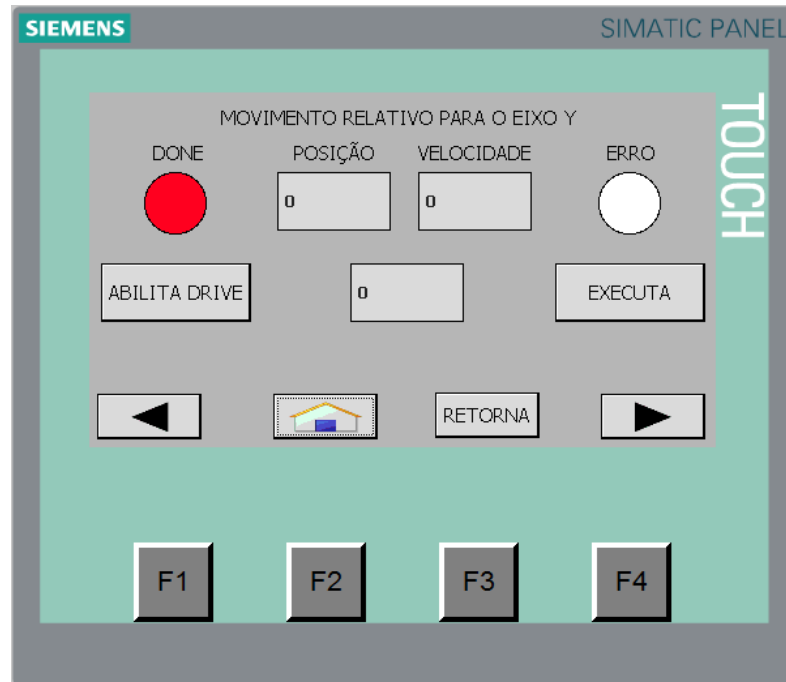
**Figura 55** - Tela Movimento Relativo para o Eixo X

Fonte: Autor

### 3.3.8 TELA MOVIMENTO RELATIVO PARA O EIXO Y

Executa o movimento relativo para o eixo Y, para qualquer valor desejado de posição, velocidade e direção, isso é possível através da caixa de diálogo campo posição e velocidade (Figura 56). Sendo cada componente da tela:

- **Botão HABILITA DRIVE:** seta a memória na entrada da porta “EXECUTE” do objeto tecnológico "MC\_MoveRelative\_DB" do Axis\_Y que aciona a saída do controlador habilitando o drive para execução do movimento;
- **Caixa de diálogo campo posição:** move o valor digitado para o primeiro índice da matriz Posi\_Y, referente a posição, na entrada “POSITION” do bloco "MC\_MoveRelative\_DB" do Axis\_Y responsável por executar o movimento digitado, transferindo em pulso para o drive que aciona o eixo Y, movendo o para a posição;
- **Caixa de diálogo velocidade:** move o valor digitado para o primeiro índice da matriz Posi\_Y, referente a velocidade, na entrada “VELOCITY” do bloco "MC\_MoveRelative\_DB" do Axis\_Y controlando a velocidade do deslocamento do eixo Y;
- **Botão executa movimento:** seta a memória da entrada “EXECUTE” do bloco "MC\_MoveRelative\_DB" do Axis\_Y que executa o posicionamento, orientação e a velocidade setada nas suas entradas;
- **Botão com seta direita:** chama a próxima tela MOVIMENTO\_RELAT\_EIXO\_Z;
- **Botão com seta esquerda:** retorna a tela anterior MOVIMENTO\_ABS\_EIXO\_Y;
- **Botão com a imagem casa:** retornar a tela inicial;
- **Botão retorna:** retorna a tela que a chamou.



**Figura 56** - Tela Movimento Relativo para o Eixo Y

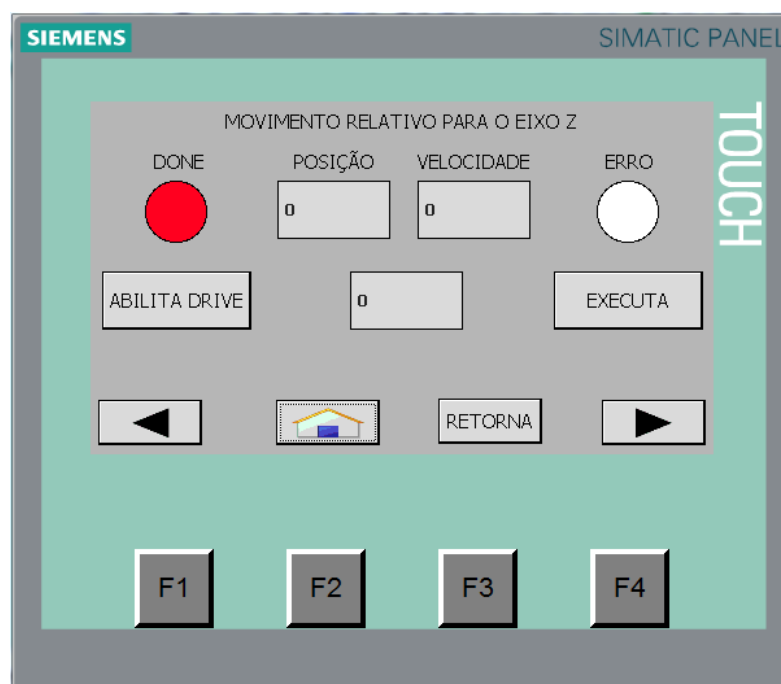
Fonte: Autor

### 3.3.9 TELA MOVIMENTO RELATIVO PARA O EIXO Z

Executa o movimento relativo para o eixo Z, para qualquer valor desejado de posição, velocidade e direção, isso é possível através da caixa de diálogo campo posição e velocidade (Figura 57). Sendo cada componente da tela:

- **Botão HABILITA DRIVE:** seta a memória na entrada da porta “EXECUTE” do objeto tecnológico "MC\_MoveRelative\_DB" do Axis\_Z que aciona a saída do controlador habilitando o drive para execução do movimento;
- **Caixa de diálogo campo posição:** move o valor digitado para o primeiro índice da matriz Posi\_Z, referente a posição, na entrada “POSITION” do bloco "MC\_MoveRelative\_DB" do Axis\_Z responsável por executar o movimento digitado, transferindo em pulso para o drive que aciona o eixo Z, movendo o para a posição;
- **Caixa de diálogo velocidade:** move o valor digitado para o primeiro índice da matriz Posi\_Z, referente a velocidade, na entrada “VELOCITY” do bloco "MC\_MoveRelative\_DB" do Axis\_Z controlando a velocidade do deslocamento do eixo Z;

- **Botão executa movimento:** seta a memória da entrada “EXECUTE” do bloco "MC\_MoveRelative\_DB" do Axis\_Z que executa o posicionamento, orientação e a velocidade setada nas suas entradas;
- **Botão com seta direita:** chama a próxima tela ZERAR MATRIZ;
- **Botão com seta esquerda:** retorna a tela anterior MOVIMENTO\_ABS\_EIXO\_Y;
- **Botão com a imagem casa:** retornar a tela inicial;
- **Botão retorna:** retorna a tela que a chamou.



**Figura 57** - Tela Movimento Relativo para o Eixo Z

Fonte: Autor

## 4 PROCEDIMENTOS

Para a realização da pesquisa as etapas foram planejadas na seguinte ordem: levantamento de literatura para embasamento teórico; estudo dos componentes para desenvolvimento do protótipo; análise do CLP e seu software correspondente; análise da IHM e seu software correspondente; programação do CLP e IHM para interação, comunicação e controle do equipamento que compõe o protótipo.

O material de pesquisa bibliográfica foi baseado em consultas a livros, artigos acadêmicos e sites disponíveis na internet, que pudessem contribuir para o aprimoramento do conhecimento teórico. Baseado nessas pesquisas, foi aplicado o conhecimento no desenvolvimento do protótipo.

Com o desenvolvimento do protótipo é possível observar o movimento do equipamento nos eixos XYZ, movimento através dos motores de passos, o que visa a melhor precisão, evitando perda de material.

### 4.1 Materiais Utilizados

Para o desenvolvimento do protótipo foram pesquisados e adquiridos os materiais que constam na Tabela 4.

**Tabela 4** - Relação de Materiais Utilizados

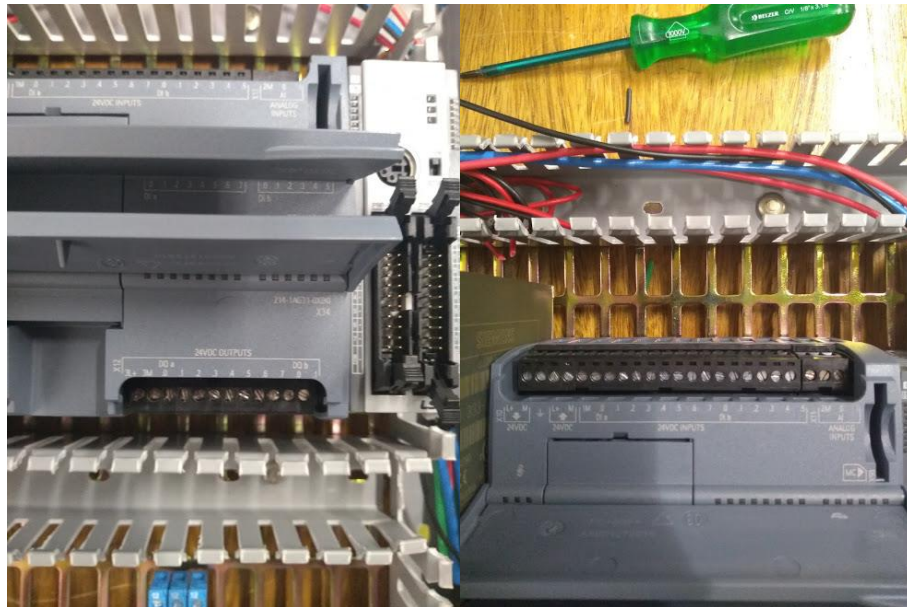
Fonte: Autor

Item	Quantidade	Descrição
1	1	CLP Siemens S7-1200 CPU 1214C
2	1	CLP LS XBM-DN16S
3	1	IHM Siemens KTP400
4	4	DRIVE TB6600
5	3	Motor de Passo
6	39	Rele de Interface Finder 24 DC 1NA 2A
7	1	Fonte AC/DC 127/220 24 DC 5A
8	1	Fonte AC/DC 127/220 24 DC 2 A
9	1	Fonte AC/DC 127/220 24 DC 6A
10	1	Fonte AC/DC 127/220 24 DC 12.5A
11	1	Fonte AC/DC 127/220 5 DC 10A
12	1	Disjuntor unipolar Merlin Gerin B10A k60

13	12	Borne de Conexão Porta Fusível Seccionável com duas entradas e duas saídas
14	40	Borne M10/10.1
15	1	Bornes terra conectados eletricamente ao trilho
16	1	Mesa CNC Router
16	1	Quadro de Comando 500x400x200 MM

## 4.2 Montagem

A montagem do protótipo foi realizada em um quadro de comando com as medidas 500x400x200mm (Figura 58). Começando pela instalação do CLP.



**Figura 58** - Montagem do CLP no Quadro de Comando

Fonte - Autor

Uma vez montado e devidamente fixado, ao CLP e a IHM foram instaladas as fontes de alimentação (Figura 59).

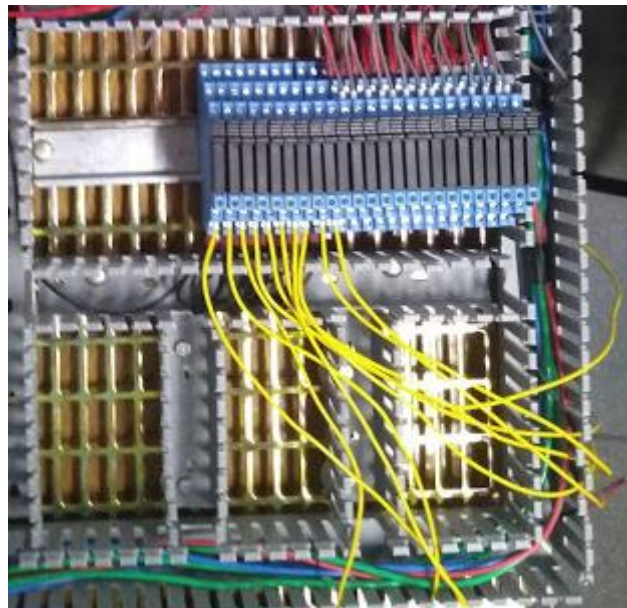




**Figura 59** - Fontes de Alimentação IHM e CLP Respectivamente

Fonte - Autor

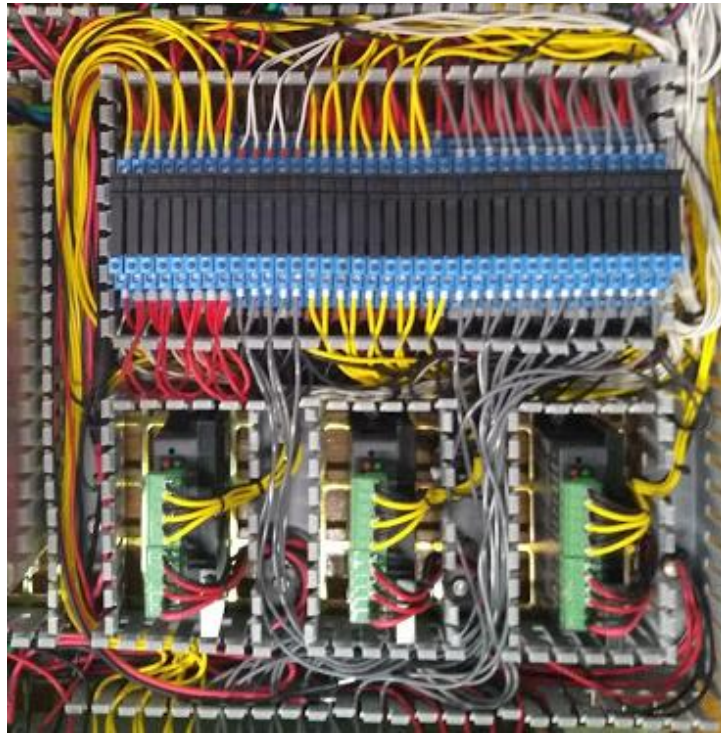
No quadro de comando foram introduzidos os dispositivos reles de interface (Figura 60).



**Figura 60** - Reles de Interface

Fonte - Autor

Após a instalação dos reles de interface, os drives de controle para os motores de passo da mesa tridimensional foram inseridos ao protótipo (Figura 61).



**Figura 61** - Instalação dos Drives de Controle dos Motores de Passo

Fonte - Autor

Para a finalização dos procedimentos de montagem do protótipo, o quadro de comando foi fixado em uma estrutura metálica e a IHM colocada em uma caixa plástica de comando (Figura 62).



**Figura 62** - Mesa CNC Tridimensional XYZ

Fonte - Autor

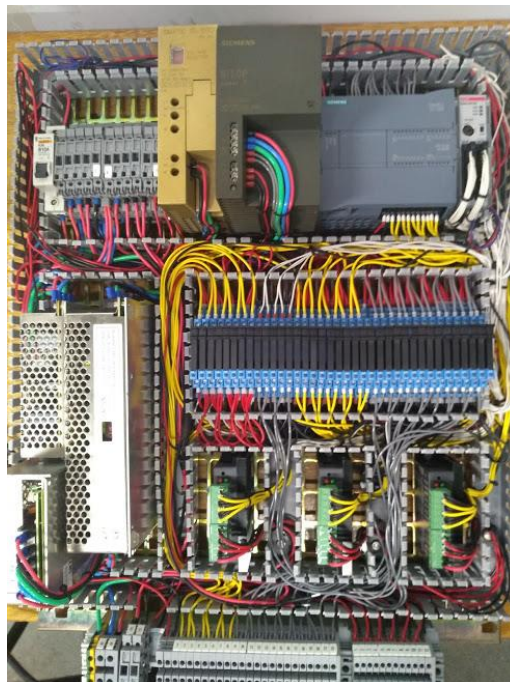
## 5 RESULTADOS E DISCUSSÃO

### 5.1 Projeto Protótipo Mesa Tridimensional XYZ

A finalidade do protótipo é mostrar a aplicabilidade dos conhecimentos adquiridos e desenvolver um projeto em uma escala reduzida de um modelo de mesa tridimensional, voltada a análise de precisão, segurança e redução de custos.

Para isso foi utilizado o CLP, um dispositivo de tecnologia moderna, todo baseado em microprocessadores que utiliza uma memória programável para armazenamento de instruções, automação de fácil flexibilização, que possui controles discretos, que pode executar operações aritméticas, além de funções lógicas, contagens de sequenciamento e temporização, contando ainda com uma IHM para tornar mais intuitiva a relação entre o operador e o processo.

Para o desenvolvimento do protótipo foi utilizada uma mesa Router Hobby, com uma estrutura que simula os movimentos tridimensionais nos eixos X, Y e Z, com os demais itens que compõem a Tabela 4, todos os componentes, exceto a mesa Router Hobby, foram montados em um quadro de comando conforme Figura 63.



**Figura 63** - Quadro de Comando Montado Mesa Tridimensional XYZ

Fonte-Autor



**Figura 64** - Mesa Router Hobby vista de cima

Fonte - Autor

Muitos desafios são encontrados ao passo da montagem da Mesa Tridimensional XYZ (Figura 64), desde a programação em uma plataforma moderna, até a interação com as funcionalidades de lógica dos drives selecionados, porém a que deve ser destacada é a interação entre o CLP e a IHM, pois apesar de serem do mesmo fabricante, a integração através da nova plataforma TIA® da Siemens™ gerou uma nova forma de programar e utilizar a ferramenta de programação além dos componentes físicos.

Com todos os componentes interligados no quadro de comando, a programação tanto no CLP como na IHM, proporcionou a interação possível entre o operador e o processo, que pode ser acompanhado através da tela da IHM, gerando confiabilidade ao processo e a visualização dos movimentos nos eixos X, Y e Z.

O resultado obtido foi muito promissor, pois se consegue realizar a parametrização das matrizes, controlar o índice, a trajetória e a velocidade dos movimentos tridimensionais X, Y e Z. No entanto não pode ser limitada a aplicação dos conhecimentos de automação aqui propostas, é necessário mais estudos e testes em escalas maiores e com movimentos e velocidades mais complexas, tornando cada vez mais real a sua aplicabilidade.

Outro ponto a ser destacado, é a necessidade de um estudo mais completo da plataforma de programação TIA® da empresa Siemens™ para aprimorar cada vez mais a integração do CLP ao IHM, facilitando a intuitividade do processo.

## 6 CONCLUSÕES

O estudo apresentado consegue responder ao questionamento de como pode ser controlado um processo aonde se é necessário precisão, segurança e economia. Precisão nos movimentos dos eixos X, Y e Z através dos drives que realização dos movimentos nos motores de passo. Segurança, pois, o equipamento não fica fadigado ou necessita de descanso entre processos. Finalmente economia, pois não existe o desperdício, uma vez programado para realização do processo, o mesmo respeitará todas as medidas inseridas e movimentos programados.

O projeto de automação proposto cumpre com os objetivos específicos, que se mostra eficiente quanto o controle de velocidade e movimento nos eixos X, Y e Z. Além de contribuir como fonte de pesquisa para trabalhos futuros voltados para controles tridimensionais.

Considerando o que foi apresentado, pode ser concluído que cada vez mais a automação é de fundamental importância em processos produtivos. A sua aplicabilidade é ilimitada, e cada vez mais os profissionais devem estar preparados para conseguir realizar a automação dos processos de forma segura e que utilizem os recursos disponíveis com responsabilidade.

O protótipo apresentado mostra que cada vez mais a precisão é um fator que contribui para a utilização de forma responsável dos recursos disponíveis para os processos de produção, o que permite um melhor aproveitamento de materiais e redução de custos como retrabalho ou perda de material.

## REFERÊNCIAS

ABNT, associação Brasileira de Normas Técnicas. Controladores Lógicos Programáveis. Disponível em: <<http://www.abntdigital.com.br>> Acesso em : 18 jun. 2018.

BRANQUINHO, Marcelo Ayres-Seidl, Ján-Moraes, Leonardo Cardoso de- Branquinho, Thiago Braga-Junior, Járcey de Azevedo. Segurança de Automação Industrial e Scada- Branquinho. 1ª edição, Editora Elsevier. 2014

FRANCHI, Claiton Moro; CAMARGO, Valter Luís Arlindo de. Controladores Lógicos Programáveis-Sistemas Discretos. 2ª edição. São Paulo, SP, Editora Érica. 2009.

GOMES, Helio.O que é IHM. Disponível em <<https://engprocess.com.br/o-que-e-ihm/>>. Acessado em:

PRUDENTE, Francesco. Automação Industrial - PLC, Teoria e Aplicações: Curso Básico. 1ª edição. Rio de Janeiro, RJ, Editora LTC. 2007

Siemens, TIA  
<[https://w5.siemens.com/portugal/web\\_nwa/pt/PortalInternet/negocios/Industry/IA\\_DT/AutomationSystems/Documents/TIA%20Portal.pdf](https://w5.siemens.com/portugal/web_nwa/pt/PortalInternet/negocios/Industry/IA_DT/AutomationSystems/Documents/TIA%20Portal.pdf)> Acessado em: 20 jun. 2018.

Unesp. Motor de Passo. Disponível em  
<<http://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>> Acessado em: 25 jul. 2018

Autorizo cópia total ou parcial desta obra,  
apenas para fins de estudo e pesquisa,  
sendo expressamente vedado qualquer  
tipo de reprodução para fins comerciais  
sem prévia autorização específica do  
autor. Autorizo também a divulgação do  
arquivo no formato PDF no banco de  
monografias da Biblioteca institucional.

Leonardo Lopes Siqueira

Pindamonhangaba, dezembro 2018.