



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

BRUNA SALLES MOREIRA

**AUTOMATIZAÇÃO DE UMA PLATAFORMA DE TRATAMENTO DE
SUPERFÍCIES**

Campina Grande, Paraíba
Novembro de 2017

BRUNA SALLES MOREIRA

AUTOMATIZAÇÃO DE UMA PLATAFORMA DE TRATAMENTO DE SUPERFÍCIES

*Trabalho de Conclusão de Curso submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande como
parte dos requisitos necessários para a obtenção do
grau de Bacharel em Ciências no Domínio da
Engenharia Elétrica.*

Área de Concentração: Instrumentação Eletrônica e Controle

Orientador:

Professor Jaidilson Jó da Silva, D. Sc.

Campina Grande, Paraíba
Novembro de 2017

BRUNA SALLES MOREIRA

AUTOMATIZAÇÃO DE UMA PLATAFORMA DE TRATAMENTO DE SUPERFÍCIES

*Trabalho de Conclusão de Curso submetido à Unidade
Acadêmica de Engenharia Elétrica da Universidade Federal de
Campina Grande como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no Domínio da
Engenharia Elétrica.*

Área de Concentração: Instrumentação Eletrônica e Controle

Aprovado em ____ / ____ / _____

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professor Jaidilson Jó da Silva, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

Dedico este trabalho à minha mãe, Livia Coelho Salles, à minha irmã, Isadora, à minha avó, Lêda, e a todos os colegas que de forma direta ou indireta me ajudaram nessa conquista. Obrigada!

AGRADECIMENTOS

À Deus, dedico o meu agradecimento maior, pela dádiva da vida e por iluminar minha caminhada.

Homenageio aqui a minha mãe, Livia Salles, por sempre ter sido meu porto seguro nas tempestades mais severas, minha eterna companheira e melhor amiga. Ao meu pai, André Moreira, pelo amor e incentivo.

A minha avó Lêda Salles, na qual espelhei-me em sua espiritualidade e sabedoria.

Ao Institut National des Sciences Appliquées Lyon pelo ambiente criativo e amigável que proporciona. Aos professores Didier Noterman e Mady Guillemot pela oportunidade e paciência com todas as dificuldades que tive com a língua francesa.

Como também agradeço ao professor Jaidilson, pela orientação ao longo do curso, por toda a paciência, dedicação e ensinamentos.

A José Antônio Neto, amigo e companheiro, agradeço por toda a compreensão, incentivo e carinho. Aos amigos da graduação Clara, Sara e Rodolfo, que fizeram esta jornada muita mais alegre e divertida.

E a todos os meus familiares, mestres e demais amigos que participaram dessa trajetória, colaborando para concretização desta etapa em minha vida.

“Ninguém é tão grande que não possa aprender, nem tão pequeno que não possa ensinar.”

(Esopo)

RESUMO

Neste trabalho, foi estudada uma plataforma de tratamento de superfícies sob o ponto de vista da automação. Inicialmente, realizou-se a concepção mecânica do sistema, depois implementou-se a aplicação no Controlador Lógico Programável (CLP), a fim de atender todas as especificações do projeto. Por fim, uma Interface Homem-Máquina (IHM) foi implementada no painel gráfico Magelis com o intuito de permitir o monitoramento e controle do sistema. A comunicação através do protocolo Modbus também foi estudada, a fim de realizar a troca de dados entre o controlador e os atuadores. Este projeto tem como objetivo desenvolver um experimento para que os alunos de engenharia elétrica do *Institut National des Sciences Appliquées de Lyon* (INSA Lyon) se familiarizem com as ferramentas tecnológicas do mercado de automação industrial atual.

Palavras-chave: Automação Industrial, Controlador Lógico Programável, Interface Homem-Máquina.

ABSTRACT

In this work, a surface treatment platform was studied from the point of view of automation. Initially, the mechanical design of the system was performed, then the application was implemented in the Programmable Logic Controller (PLC) to comply with all project specifications. Finally, a human-machine interface (HMI) was implemented in the Magelis display with the purpose of supervise and control the system. As well, Modbus protocol has been studied in order to perform the exchange of data between the controller and the actuators. This project aims to create an experiment for the electrical engineering students of Institut National des Sciences Appliquées (INSA Lyon's) in order to familiarize themselves with the new technological tools of the current automation market.

Keywords: Industrial Automation, Programmable Logic Controller, Human-Machine Interface.

LISTA DE FIGURAS

Figura 2.1: Operações do ciclo automático.	16
Figura 2.2: Estrutura da mensagem Modbus.	20
Figura 2.3: SoMachine Basic página principal.	22
Figura 2.4: Linha de código Ladder, SoMachine Basic.	24
Figura 2.5: Código em Grafcet, no SoMachine Basic.	25
Figura 2.6: Janela Colocar em funcionamento, no SoMachine Basic.	26
Figura 3.1: Modelo da Plataforma de Tratamento de Superfícies.	27
Figura 3.2: Fotografia da plataforma MAP.	28
Figura 3.3: Fotografia do sensor indutivo 10 mm e do sensor de precisão 4mm utilizados na plataforma.	30
Figura 3.4: Ilustração do funcionamento horizontal da plataforma.	30
Figura 3.5: a) Fotografia do arrancador-controlador da série TeSys U da Schneider Electric; b) Fotografia do inversor de frequência ATV312 da Schneider Electric.	31
Figura 3.6: Fotografia da caixa com botão e chave de urgência, da Schneider Electric.	32
Figura 3.7: Representação do ciclo de funcionamento de um sistema de automação.	33
Figura 3.8: CLP modelo M221, da marca Schneider Electric.	34
Figura 3.9: IHM modelo Magelis HMISTU855, da marca Schneider Electric.	35
Figura 3.10: Fotografia da grade de comando da plataforma.	36
Figura 4.1: Bloco Read_Var, no SoMachine Basic.	37
Figura 4.2: Propriedade do bloco de função Write_Var, no SoMachine Basic.	38
Figura 4.3: Divisão do programa de comando do ATV 312.	39
Figura 4.4: Linha de código, em Ladder, para enviar uma palavra de comando.	39
Figura 4.5: Tabela de animação CMD.	40
Figura 4.6: Linha código do POU MSK.	40
Figura 4.7: Grafcet de comando do ATV 312.	41
Figura 4.8: Grafcet mestre.	42
Figura 4.9: Grafcet modo manual.	43
Figura 4.10: Grafcet de inicialização.	44
Figura 4.11: Código, em Ladder, do comando avançar.	45
Figura 4.12: Código em Ladder, de gestão das entradas.	46

Figura 5.1: Configuração material da IHM no Vijeo Designer.....	48
Figura 5.2: Configuração do equipamento, no Vijeo Designer.....	49
Figura 5.3: Tela de início.....	50
Figura 5.4: Tela do modo automático.....	51
Figura 5.5: Tela do modo manual.....	51

SUMARIO

1	INTRODUÇÃO.....	13
	1.1 OBJETIVO GERAL.....	14
	1.2 OBJETIVOS ESPECÍFICOS.....	14
	1.3 ESTRUTURA DO TRABALHO.....	14
2	REVISÃO BIBLIOGRÁFICA.....	15
	2.1 ESPECIFICAÇÕES DO SISTEMA.....	15
	2.1.1 Modos de Funcionamento.....	15
	2.1.2 Parada de Emergência.....	17
	2.2 REDES INDUSTRIAIS.....	17
	2.2.1 Modbus.....	17
	2.3 SOMACHINE BASIC.....	22
	2.3.1 Modos de Operação.....	23
	2.3.2 Tarefas e Modos de Verificação.....	23
	2.3.3 Programação em linguagem Ladder.....	24
	2.3.4 Programação Grafset (SFC).....	24
	2.3.5 Colocar em Execução.....	25
3	PLATAFORMA DE TRATAMENTO DE SUPERFÍCIES.....	27
	3.1 MODELO PROPOSTO.....	27
	3.2 CONCEPÇÃO MECÂNICA.....	28
	3.3 SENSORES.....	28
	3.4 ATUADORES.....	30
	3.5 CONTROLADOR.....	32
	3.6 INTERFACE HOMEM MÁQUINA.....	34
	3.7 GRADE DE COMANDO.....	35
4	DESENVOLVIMENTO DA APLICAÇÃO PARA O CONTROLE DA PLATAFORMA.....	37
	4.1 COMANDO DO MOTOR A PARTIR DO TeSYS U.....	37
	4.2 COMANDO DO MOTOR A PARTIR DO ATV 312.....	38
	4.3 MODOS DE FUNCIONAMENTO.....	41
	4.3.1 Modo Manual.....	42
	4.3.2 Modo Inicialização.....	43
	4.3.3 Modo Automático.....	44
	4.3.4 Gestão das Entradas e Saídas.....	44
	4.4 GESTÃO DOS DISPOSITIVOS DE SEGURANÇA.....	46

5	CONFIGURAÇÃO DA IHM.....	48
	5.1 CONFIGURAÇÃO MATERIAL.....	48
	5.2 COMUNICAÇÃO COM O CLP.....	49
	5.3 IMPLEMENTAÇÃO DA APLICAÇÃO	49
6	CONCLUSÃO	52
	6.1 SUGESTÃO PARA TRABALHOS FUTUROS	52
7	REFERÊNCIAS.....	53

1 INTRODUÇÃO

A necessidade de automação na indústria e nos mais diversos segmentos está associada, entre diversos aspectos, às possibilidades de aumentar a velocidade de processamento das informações, uma vez que as operações estão cada vez mais complexas e variáveis, necessitando de um grande número de controles e mecanismos de regulação para permitir decisões mais ágeis e, portanto, aumentar os níveis de produtividade e eficiência do processo produtivo dentro das premissas da excelência operacional.

A automação permite economias de energia, força de trabalho e matérias-primas, um melhor controle de qualidade e, nas indústrias, permite elevar os níveis de continuidade e de controle global do processo com maior eficiência, maior utilização da planta, aumenta a produtividade e a segurança operacional. Em essência aproxima ao máximo a produção real à capacidade nominal da planta, ao reduzir ao mínimo possível as horas paradas, de manutenção corretiva e a falta de matéria-prima (CASSIOLATO, C., 2011).

Dessa forma, considerando a importância e a expansão da automação no mercado industrial atual, elaborou-se um modelo que simula a automatização de uma plataforma industrial de tratamento de superfície por imersão das peças metálicas em várias soluções químicas.

Esta plataforma é dotada de três sensores indutivos, quatro sensores de fim de curso e um sensor de indutivo de precisão, os quais se comunicam com o controlador lógico programável (modelo M221, da Schneider Electric). Os dados coletados dos sensores pelo CLP são utilizados na determinação da posição atual em se encontra o depósito com as peças que serão submersas nas soluções químicas. No painel tátil, Magelis, da Schneider Electric, implementou-se uma interface de controle e monitoramento que permite ao usuário acompanhar o posicionamento do depósito, bem como escolher entre os modos de funcionamento do sistema.

1.1 OBJETIVO GERAL

O objetivo primordial deste trabalho foi a automatização de uma Plataforma de Tratamento de Superfícies, a fim de criar um experimento para os alunos de engenharia elétrica do INSA Lyon.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho foram:

- Concepção mecânica da plataforma.
- Desenvolver uma aplicação no Controlador Lógico Programável, a fim de automatizar o sistema.
- Desenvolver uma interface gráfica para controle e monitoramento.

1.3 ESTRUTURA DO TRABALHO

Neste capítulo serão tecidos alguns comentários acerca dos objetivos e da estrutura do trabalho. No Capítulo 2, uma breve revisão bibliográfica sobre as especificações do sistema, sobre redes industriais, especificamente o protocolo Modbus, bem como sobre o *software* SoMachine Basic, que será utilizado para programar o CLP.

Em seguida, no Capítulo 3, são descritos os componentes e as características do sistema, o que inclui a Plataforma de Tratamento de Superfícies, os sensores e atuadores, o controlador e a interface de controle e monitoramento.

O Capítulo 4 apresenta o desenvolvimento da aplicação para o controle da plataforma, detalhando como é realizada a comunicação entre o CLP e os atuadores, a gestão das entradas e saídas e os modos de funcionamento.

No Capítulo 5, é apresentada a configuração da IHM utilizada e as diferentes telas criadas para controlar o sistema.

Por fim, algumas considerações são feitas, no Capítulo 6, acerca dos resultados obtidos e possíveis sugestões para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo, estão descritas, resumidamente, algumas informações básicas sobre as especificações do sistema, assim como um breve resumo sobre a rede industrial Modbus e sobre o *software* SoMachine Basic, que são úteis ao entendimento do projeto.

2.1 ESPECIFICAÇÕES DO SISTEMA

Um modelo de uma plataforma industrial de tratamento de superfície foi projetado e seu funcionamento será brevemente descrito.

Um suporte que se move horizontalmente transporta uma estrutura que permite o movimento vertical de um depósito, no qual as peças metálicas estão localizadas. O depósito que contém as partes a serem tratadas é levado para a estação de carregamento, e então, as peças são submetidas a tratamento em diferentes soluções químicas.

O tempo de imersão nos tanques depende da natureza do banho. Normalmente um gotejamento é necessário antes de realizar a mudança de tanque. Em seguida, as peças são levadas para a estação de descarregamento.

Uma Interface Homem-Máquina é implementada em um painel Magelis, da Schneider Electric, permitindo a interação do operador com a instalação. Todos os botões necessários para controlar a instalação são apresentados nessa interface. Exceto, por questões de segurança, um botão físico de Parada de Urgência e uma chave de duas posições que permite o retorno seguro da instalação, após ultrapasse dos limites de fim de curso.

2.1.1 MODOS DE FUNCIONAMENTO

Primeiramente, na IHM, o operador deve escolher entre os dois modos de operação: Manual ou Automático.

Caso selecione-se a opção Manual, quatro botões estarão à disposição: Avançar, Recuar, Descer e Subir. O deslocamento acontecerá durante todo o tempo que o operador pressione o botão de modo que o suporte não atinja os limites dos sensores de fim de curso. Uma luz laranja acende enquanto o sistema estiver nesse modo.

Se o modo Automático for escolhido, uma nova interface surgirá e o botão Início de Ciclo (DCY, *Départ de cycle*, em francês) deve ser acionado para começar a rotina programada. Caso o suporte não esteja na posição inicial, executa-se um ciclo de inicialização e a luz verde piscará. Nessa situação, para iniciar a produção automática, o operador deve pressionar novamente o botão Início de Ciclo. A luz verde acende enquanto o sistema estiver no modo automático.

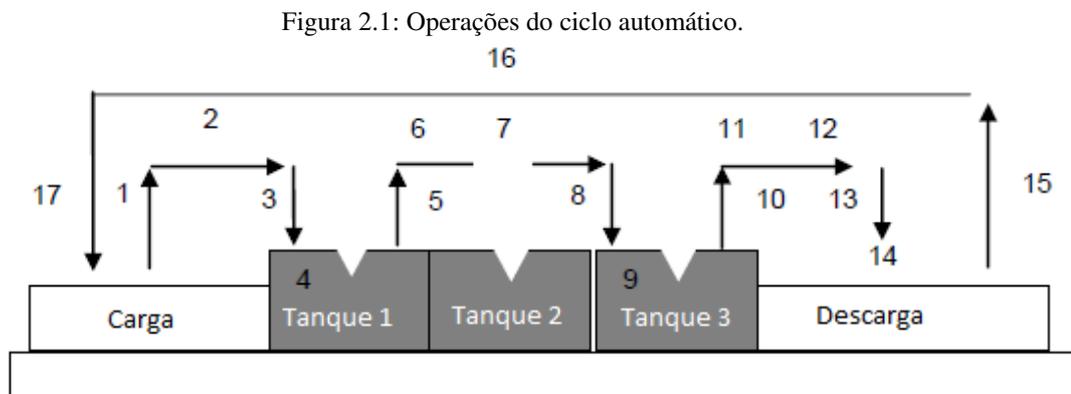
A produção automática é contínua, então, caso deseja-se parar a produção, é preciso pressionar o botão Fim de Ciclo (AFCY, *Arrêt fin de cycle*, em francês). O atual ciclo continua e só parará quando o suporte retornar à posição inicial. Durante esta fase, as luzes verde e vermelha piscam simultaneamente.

2.1.1.1 O CICLO AUTOMÁTICO

Durante a execução do ciclo automático, as seguintes operações são realizadas sequencialmente:

- 4: imersão no Tanque 1 durante 4 s
- 6: gotejamento em cima do tanque 1 durante 3 s
- 9: imersão no Tanque 3 durante 4 s
- 11: gotejamento em cima do tanque 3 durante 3 s
- 14: simulação de operação descarga de 3 s

Pode-se observar na Figura 2.1 onde cada operação ocorre.



2.1.2 PARADA DE EMERGÊNCIA

A segurança de uma instalação requer o uso de um botão de parada de emergência para desligar a energia da instalação.

Um contato auxiliar está ligado a uma entrada do CLP, dessa forma, o controlador é informado em caso do engate da parada de emergência. A gestão da reinicialização da instalação será fornecida pelo controlador uma vez que o botão é desbloqueado. Essa sequência será abordada na seção 4.4.

No caso de uma Parada de Emergência, o programa desativa todas as saídas do sistema e a luz vermelha pisca rapidamente.

2.2 REDES INDUSTRIAIS

A capacidade de comunicação entre dispositivos e o uso de mecanismos padronizados, abertos e transparentes são componentes indispensáveis do conceito de automação. A comunicação vem se expandindo rapidamente no sentido horizontal nos níveis inferiores, assim como no sentido vertical integrando todos os níveis hierárquicos.

As redes industriais são redes de comunicação dedicadas ao ambiente industrial, permitindo a troca de informação entre diferentes equipamentos. As vantagens alcançadas pelo uso das redes industriais como tecnologia são diversas, como: redução de instalações elétricas, redução do tamanho de painel elétrico, modularização de máquinas e equipamentos, diagnóstico local de falhas, diagnóstico real time em supervisão e flexibilidade na ampliação ou modificação (SANTOS, L., 2016).

2.2.1 MODBUS

Modbus é um protocolo de comunicação serial desenvolvido pela Modicon e publicado pela Modicon® em 1979 utilizado para comunicação entre dispositivos mestre-escravo/cliente-servidor.

Este protocolo define uma estrutura de mensagem que os controladores reconhecerão independentemente do tipo de rede sobre as quais eles se comunicam. Ele descreve o processo que um controlador usa para solicitar acesso a outro dispositivo, como ele responderá às solicitações dos outros dispositivos e como os erros serão detectados e relatados. Ele

estabelece um formato comum para o formato e o conteúdo dos campos de mensagem (National Instruments, 2014).

Modbus é um dos protocolos mais utilizados em automação industrial, graças à sua simplicidade e facilidade de implementação, podendo ser utilizado em diversos padrões de meio físico, como: RS-232, RS-485 e Ethernet TCP/IP (Modbus TCP).

A velocidade de comunicação varia em cada um desses padrões, bem como o comprimento máximo da rede e o número máximo de dispositivos conectados.

2.2.1.1 MODELO DE COMUNICAÇÃO

O protocolo Modbus é baseado no modelo de comunicação mestre-escravo, onde apenas o único dispositivo mestre pode inicializar a comunicação e os demais dispositivos escravos respondem enviando os dados solicitados pelo mestre, ou realizam alguma ação solicitada. Os dados transmitidos podem ser digitais ou analógicos, ou seja, é possível enviar valores numéricos como temperatura e pressão ou enviar um *bit* para ligar e desligar um motor (National Instruments, 2014).

Ao utilizar o meio físico Ethernet, o protocolo Modbus opera com o mecanismo de controle de acesso CSMA-CD, que é próprio da rede Ethernet, com mensagens no modelo cliente-servidor.

2.2.1.2 MODOS DE TRANSMISSÃO

O protocolo Modbus pode ser configurado para trabalhar com um dos dois modos de transmissão disponíveis: ASCII (*American Code for Information Interchange*) ou RTU (*Remote Terminal Unit*), os quais definem como os dados serão empacotados na mensagem, estes modos são escolhidos durante a configuração dos parâmetros de comunicação, tais como: *baud rate*, paridade e *stop bits*.

Ao todo o protocolo Modbus possui 256 endereços onde:

- 0 - (zero) é o endereço de *broadcast*, quando o mestre envia uma mensagem para o endereço zero, todos os escravos recebem a mensagem;
- 1 até 247 endereços disponíveis para os escravos;
- 248 até 255 endereços reservados.

2.2.1.3 MODO RTU

No modo RTU, cada mensagem de 8 *bits* contém dois caracteres hexadecimais de 4 *bits*. A principal vantagem desse modo é que sua maior densidade de caracteres permite um melhor processamento de dados do que o modo ASCII para a mesma velocidade de comunicação. Cada mensagem deve ser transmitida em um fluxo contínuo de caracteres.

No modo RTU não existe um caractere específico que indique o início ou o fim de um telegrama. A indicação de quando uma nova mensagem começa ou quando ela termina é feita pela ausência de transmissão de dados na rede, por um tempo mínimo de 3,5 vezes o tempo de transmissão de um *byte* de dados.

Sendo assim, caso um telegrama tenha iniciado após a decorrência desse tempo mínimo, os elementos da rede irão assumir que o primeiro caractere recebido representa o início de um novo telegrama. E da mesma forma, os elementos da rede irão assumir que o telegrama chegou ao fim quando, recebidos os *bytes* do telegrama, este tempo decorra novamente.

Se durante a transmissão de um telegrama o tempo entre os *bytes* for maior que este tempo mínimo, o telegrama será considerado inválido, pois o controlador irá descartar os *bytes* já recebidos e montará um novo telegrama com os *bytes* que estiverem sendo transmitidos. O tempo para transmitir uma palavra varia de 573 μ s para taxas de comunicação acima de 19200 bits/s e 9 ms para a taxa de 1200 bits/s.

O modo de mensagem RTU inclui um método de checagem de erro que é baseado no *Cyclical Redundancy Checking* (CRC). O campo de checagem de erro contém um valor de 16 *bits* com o resultado do cálculo de CRC sobre o conteúdo da mensagem.

2.2.1.4 MODO ASCII

No modo de transmissão ASCII, cada *byte* em uma mensagem é enviado como dois caracteres ASCII. Apesar de gerar mensagens legíveis pela tabela ASCII esse modo consome mais recursos da rede. A principal vantagem dessa modalidade é que permite intervalos de tempo de cerca de um segundo entre os caracteres sem causar erro, se ocorrer um intervalo maior, o dispositivo receptor assume que ocorreu um erro.

O campo de checagem de erros é baseado no método *Longitudinal Redundancy Check* (LRC), em que a verificação ocorre ao final de cada bloco como se fosse transmitido um caractere extra ao final de cada bloco.

Os dispositivos monitoram constantemente a rede para o início de uma mensagem. Quando uma mensagem é iniciada pelo mestre, todos os dispositivos da rede decodificam o campo de endereço para determinar qual escravo deve receber a mensagem. O início de uma mensagem é reconhecido pelo caractere (:) "dois pontos".

2.2.1.5 DESCRIÇÃO DO PROTOCOLO MODBUS

O protocolo Modbus define uma Unidade de Dados de Protocolo simples (PDU) independente das camadas de comunicação subjacentes. O mapeamento do protocolo Modbus em barramentos específicos ou redes pode introduzir alguns campos adicionais na Unidade de Dados de Aplicação (ADU) (Modbus Organization, 2012). A Figura 2.2 apresenta a estrutura completa da mensagem Modbus.

A PDU e seu código formam a base da especificação do protocolo de aplicação Modbus. Essa especificação define o formato da PDU, os diversos conceitos de dados usados pelo protocolo, o uso de códigos de função para acessar esses dados e a implementação dos diversos códigos de função e as restrições específicas de cada um deles.

Figura 2.2: Estrutura da mensagem Modbus.



O campo de endereço contém dois caracteres (ASCII) ou oito *bits* (RTU). Os dispositivos escravos individuais são atribuídos endereços no intervalo de 1 a 247.

O código de função de uma unidade de dados Modbus é codificado em um *byte*. Os códigos válidos estão no intervalo de 1 a 255 decimal (o intervalo de 128 a 255 é reservado e usado para respostas de exceção). O código de função "0" não é válido.

O campo de dados disponibiliza ao escravo alguma informação necessária pelo escravo para completar a ação específica pelo código da função. O dado é formado de *bytes* de caracteres múltiplos (um par de caracteres ASCII no modo ASCII) ou de dois dígitos hexadecimais no modo RTU, na faixa de 00h até FFh. Os dados tipicamente incluem registradores de endereços, contadores de valores e escrita de dados. Se nenhum erro é encontrado, o campo de dados da resposta do escravo retornará do pedido de dados. Se alguns

erros ocorrem, o campo de dado retorna um código de exceção que a aplicação mestre pode usar para determinar a próxima ação a tomar.

Se nenhum erro é encontrado, o campo de dados da resposta contém os dados solicitados. Se ocorrer um erro relacionado com a função Modbus solicitada, o campo de dados contém um código de exceção que a aplicação do servidor pode usar para determinar a próxima ação a ser tomada.

2.2.1.6 PRINCIPAIS CÓDIGOS DE FUNÇÃO

Existem três categorias de códigos de funções Modbus (Modbus Organization, 2012):

- Códigos de Função públicos
- Códigos de Função definidos pelo usuário
- Códigos de Função reservados

Os códigos de função públicos são códigos bem definidos, únicos, validados pela comunidade Modbus.org, documentados e possuem teste de conformidade disponível.

Os principais códigos de função são:

- 0x01: leitura de 1 a 2000 blocos de *bits* do tipo saídas digitais em um dispositivo remoto. A PDU de solicitação especifica o endereço inicial e o número de bobinas.
- 0x02: leitura de 1 a 2000 blocos de *bits* do tipo entradas digitais em um dispositivo remoto. A PDU de solicitação especifica o endereço inicial e o número de entradas.
- 0x05: escrita em uma única saída digital em um dispositivo remoto. A PDU de solicitação especifica o endereço da bobina que será forçado o estado. O estado solicitado é especificado por uma constante no campo *Coil Value*. O valor 0xFF00 envia “1” e 0x00 “0”.

2.3 SOMACHINE BASIC

O SoMachine Basic é uma ferramenta de programação gráfica criada para facilitar a configuração, o desenvolvimento e a colocação em funcionamento de programas para controladores lógicos Modicon M221 (Schneider Electric, 2016).

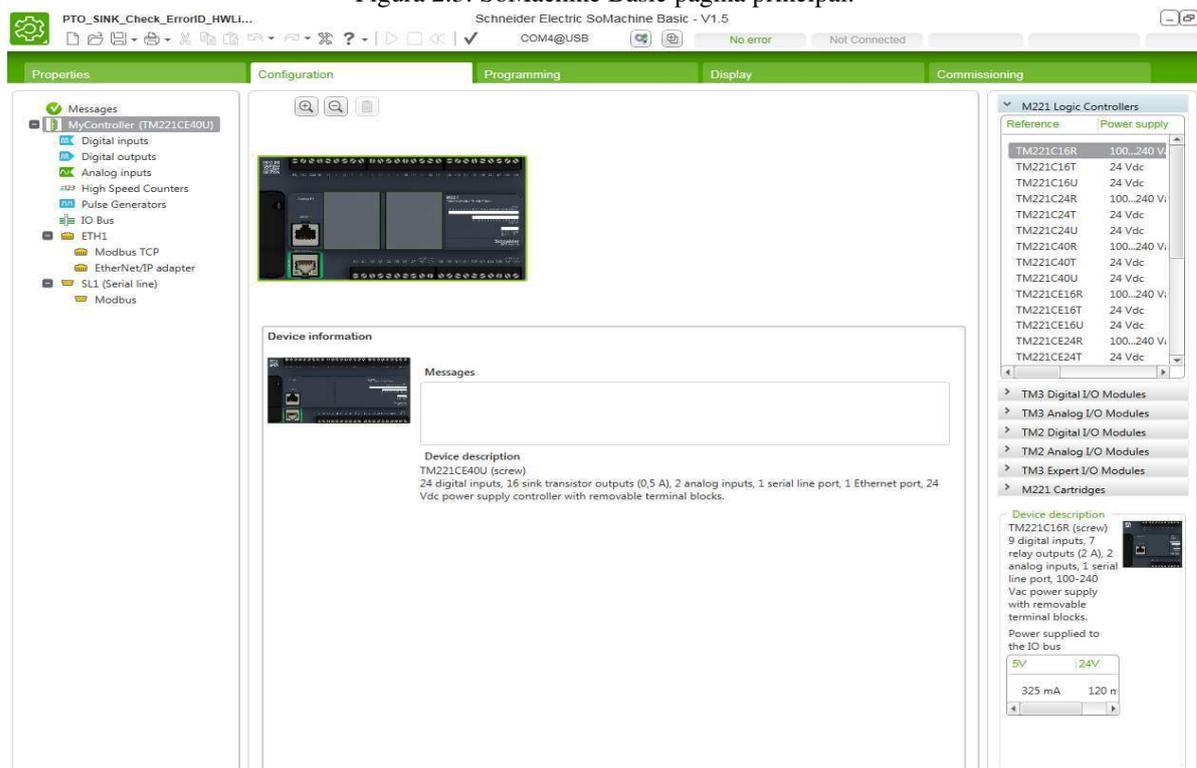
A janela principal está dividida em vários módulos. Cada módulo controla um estágio diferente do ciclo de desenvolvimento e está acessível clicando em uma guia na parte superior da área de módulos: Propriedades, Configuração, Programação, Exibição e Colocar em funcionamento, como apresentado na Figura 2.3.

Cada tela é dividida em 3 zonas: uma árvore de seleção, uma área de edição e um catálogo de referência de produtos organizado por gama.

O programa é organizado em Unidades Organizacionais do Programa (POUs) e seções. POUs são objetos reutilizáveis que contêm uma declaração de variável e um conjunto de instruções utilizadas em um programa. Já as seções consistem em redes (*rungs*) para simplificar a leitura e a navegação dentro do programa.

O SoMachine Basic suporta as seguintes linguagens de programação da IEC-61131-3: Linguagem do Diagrama Ladder, Linguagem da lista de instruções (IL), Grafcet (Lista) e Grafcet (SFC).

Figura 2.3: SoMachine Basic página principal.



2.3.1 MODOS DE OPERAÇÃO

Os modos operacionais proporcionam controle para desenvolver, depurar, monitorar e modificar a aplicação quando o controlador está ou não conectado ao SoMachine Basic. O SoMachine Basic pode operar nos seguintes modos: *off-line*, *on-line* e simulador (Schneider Electric, 2016).

Em modo *off-line*, pode-se configurar SoMachine Basic para que corresponda aos componentes de *hardware* desejados e depois desenvolver a aplicação.

No modo *on-line*, é possível proceder para o *download* da aplicação para o CLP. SoMachine Basic sincroniza a aplicação na memória do PC com a versão armazenada no controlador lógico, permitindo depurar, monitorar e modificar a aplicação.

Em modo de simulador, não é estabelecida nenhuma conexão física com um controlador lógico. Em vez disso, SoMachine Basic simula uma conexão a um controlador lógico e os módulos de expansão para executar e testar o programa.

2.3.2 TAREFAS E MODOS DE VERIFICAÇÃO

SoMachine Basic possui os seguintes modos de busca para a tarefa mestre (Schneider Electric, 2016):

- Modo normal (modo autônomo): uma nova busca começa imediatamente após a busca anterior ter sido concluída.
- Modo periódico (periódico cíclico): uma nova busca começa somente após ter decorrido o tempo de busca configurado da busca anterior. Assim sendo, cada busca tem a mesma duração.

Os POU's estão associados a várias tarefas da aplicação: mestre, periódica e eventos.

- Tarefa mestre: tarefa principal do aplicativo. A tarefa mestre é controlada por uma verificação cíclica contínua (no modo de busca normal) ou especificando o período de busca de 1 a 150 ms (padrão 100 ms) no modo de busca periódica.
- Tarefa periódica: uma sub-rotina de curta duração processada periodicamente. As tarefas periódicas são configuradas especificando-se o período de busca de 1 a 255 ms (padrão 255 ms).

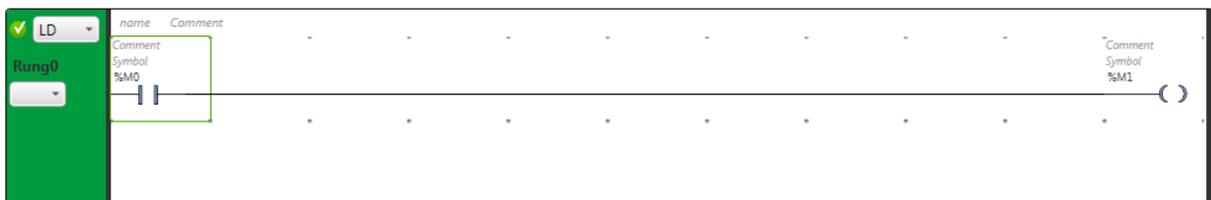
- Tarefa de evento: uma sub-rotina de duração muito curta para reduzir o tempo de resposta do aplicativo.

2.3.3 PROGRAMAÇÃO EM LINGUAGEM LADDER

A linguagem Ladder, também conhecida como lógica de diagrama de contatos, é o sistema de representação que mais se assemelha à tradicional notação de diagramas elétricos. O nome Ladder deve-se à representação da linguagem se parecer com uma escada, do inglês *ladder*, na qual duas barras verticais paralelas são interligadas pela lógica de controle, formando os degraus, do inglês *rungs*, da escada. Os degraus são executados sequencialmente pelo controlador lógico. Portanto, a cada lógica de controle existente no programa da aplicação dá-se o nome de *rung*, a qual é composta por colunas e linhas, conforme apresentado na Figura 2.4.

As instruções em Diagramas Ladder são inseridas arrastando e soltando elementos gráficos da barra de ferramentas que aparece acima do espaço de trabalho de programação em uma célula de grade.

Figura 2.4: Linha de código Ladder, SoMachine Basic.



2.3.4 PROGRAMAÇÃO GRAFCET (SFC)

Grafcet (SFC) é uma linguagem de programação gráfica que descreve uma ordem cronológica de execução de tarefas discretas, conhecidas como etapas. A ordem pela qual as etapas são executadas é determinada pelas transições que conectam as etapas (Schneider Electric, 2016).

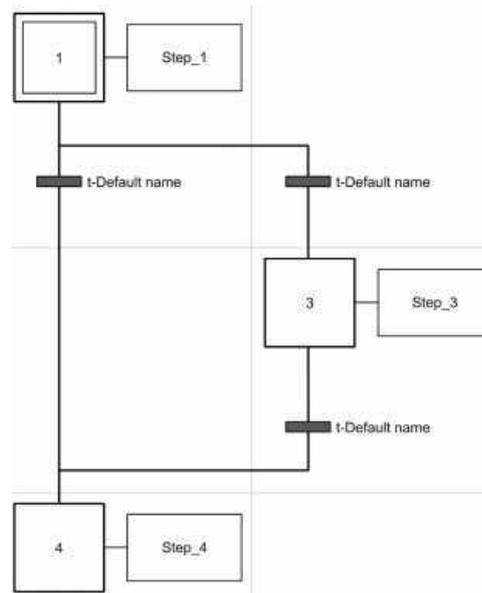
Um Grafcet (SFC) POU possui os seguintes componentes:

- Etapa: executa um conjunto de ações definidas em um ou vários *rungs*. As etapas podem ser:
 - Etapa inicial: executada no início do programa ou após um reinício do controlador. É representada por uma célula com uma borda dupla.

- Etapa regular: etapas executadas condicionalmente após a conclusão da execução da etapa inicial.
- Transição: uma expressão booleana avaliada entre etapas. É a conexão entre duas ou mais etapas. A expressão booleana é definida em um único *rung* escrito em Ladder ou IL.

A Figura 2.5 apresenta um exemplo de um POU desenvolvido em linguagem Grafcet, no SoMachine Basic.

Figura 2.5: Código em Grafcet, no SoMachine Basic.



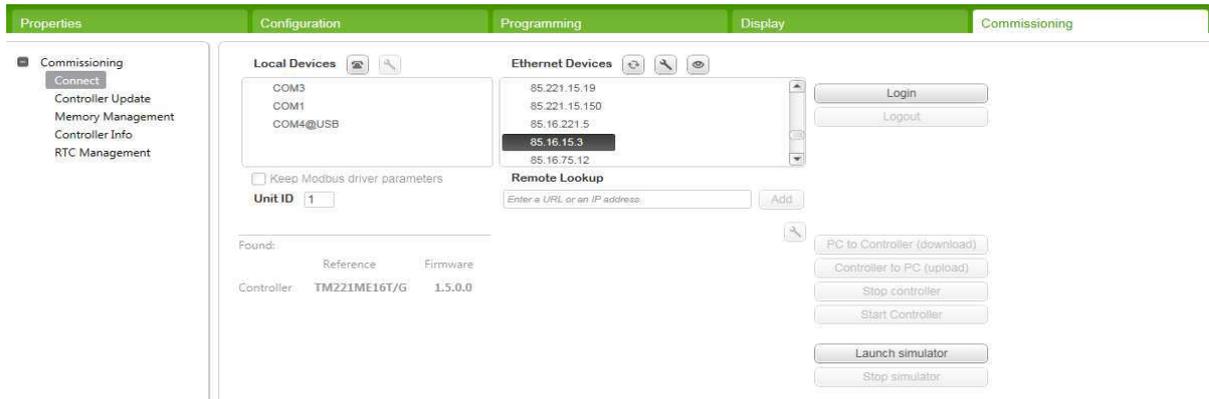
2.3.5 COLOCAR EM EXECUÇÃO

Para colocar em funcionamento a aplicação criada, faz-se necessário realizar os seguintes passos:

1. Conectar-se ao CLP,
2. Baixar a aplicação no CLP (PC para Controlador),
3. Iniciar controlador.

A Figura 2.6 apresenta a janela Colocar em funcionamento, do SoMachine Basic.

Figura 2.6: Janela Colocar em funcionamento, no SoMachine Basic.



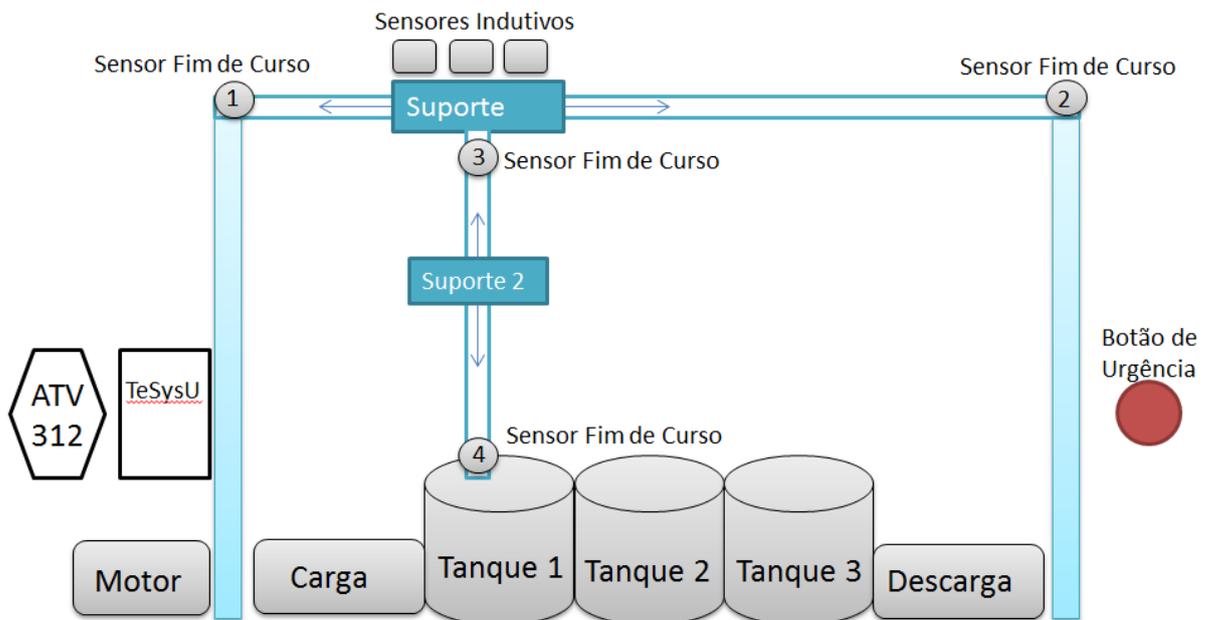
3 PLATAFORMA DE TRATAMENTO DE SUPERFÍCIES

Nas seções a seguir serão abordados aspectos relevantes sobre a plataforma em estudo, i.e., a plataforma de tratamento de superfícies.

3.1 MODELO PROPOSTO

A Figura 3.1 apresenta o modelo proposto para a plataforma de tratamento de superfícies, na qual o motor pode ser controlador tanto pelo inversor de frequência ATV312, quanto pelo módulo arrancador-controlador TeSysU, ambos da Schneider Electric. Posicionaram-se quatro sensores de fim de curso, nas extremidades horizontais e verticais dos movimentos dos suportes. Como também três sensores indutivos acima do suporte com movimento horizontal, para ajustar o posicionamento desse suporte. Um botão de urgência foi adicionado para tratar situações de urgência.

Figura 3.1: Modelo da Plataforma de Tratamento de Superfícies.



3.2 CONCEPÇÃO MECÂNICA

Inspirando-se em um experimento existente no laboratório de Medidas Automatizadas de Produção (MAP), do INSA Lyon, realizou-se a concepção mecânica da Plataforma para Tratamento de Superfícies.

A plataforma utilizada no experimento MAP é ilustrada na Figura 3.2.

Figura 3.2: Fotografia da plataforma MAP.



Dessa forma, na plataforma do experimento tratamento de superfícies, um motor é fixado no canto esquerdo da estrutura maior (amarela) permitindo o movimento horizontal de um suporte, com o auxílio de um sistema de polias. Esse suporte se move acima dos recipientes que contém as soluções químicas, e nele estão afixados sensores que asseguram o posicionamento de um sistema motorizado para subir e descer um depósito, no qual estão dispostas as peças a serem tratadas.

3.3 SENSORES

Foram empregados dois tipos de sensores. São eles o sensor de fim de curso e o sensor indutivo.

Os sensores de fim de curso são sensores que indicam quando um motor ou a estrutura ligada ao seu eixo chegou ao fim do seu campo de movimento. São sensores simples e funcionam de modo similar a uma chave liga/desliga. Neste projeto, utilizaram-se quatro sensores de fim de curso, dois na direção horizontal e dois na direção vertical, a fim de detectar quando a estrutura ligada ao eixo do motor passou do ponto desejável e atingiu a área de isolamento. Esses sensores detectam por meio de contato, dessa forma é preciso que a estrutura toque no sensor para ativá-lo.

Um sensor indutivo é um dispositivo eletrônico capaz de reagir à proximidade de objetos metálicos, pois as características de uma bobina em torno de um núcleo se alteram na presença de materiais ferromagnéticos, variando o campo magnético, o sensor comuta quando a variação é sentida.

Utilizaram-se cinco sensores indutivos com distância operacional de 10 mm. Um para detectar quando a estrutura chega ao ponto mais baixo a fim de realizar carga e descarga das peças, outro para quando a estrutura chega ao ponto mais alto, possibilitando assim o movimento horizontal. E outros três sensores indutivos que com o auxílio de um sensor de precisão, irão posicionar a estrutura horizontalmente.

O sensor de precisão é um sensor indutivo com distância operacional de 4 mm, conferindo maior exatidão no posicionamento da estrutura horizontal.

Os sensores anexados na estrutura horizontal estão posicionados lado a lado, como apresentado na Figura 3.2, e através da combinação binária arranjada, sabe-se a posição atual. Foram posicionados parafusos acima de cada posição (Carga, Posição 1, Posição 2, Posição 3 e Descarga) de forma a conferir uma combinação binária. Por exemplo, na posição Carga, tem-se 0011, ou seja, um parafuso embaixo do sensor de precisão e um embaixo do sensor indutivo que está ao lado do sensor de precisão. Já na Posição 1, tem-se 0111, um parafuso embaixo de dois sensores indutivos presente no meio do suporte e um parafuso embaixo do sensor de precisão. Na Posição 2, tem-se 1111, na posição 3, 1001 e na posição Descarga 1011.

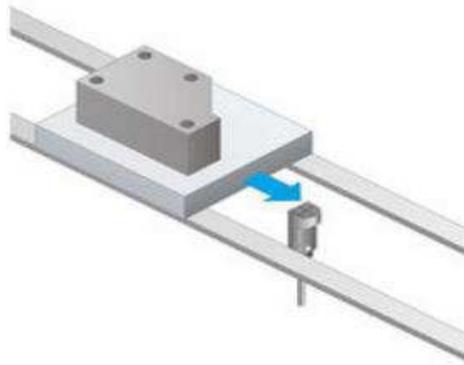
O sensor indutivo de 10 mm descrito e o sensor de precisão podem ser observados na Figura 3.3.

Figura 3.3: Fotografia do sensor indutivo 10 mm e do sensor de precisão 4mm utilizados na plataforma.



A Figura 3.4 ilustra o funcionamento horizontal da plataforma, os sensores estão fixados no suporte que desliza sobre a esteira e no caminho, acima das posições planejadas, estão os parafusos, a fim de indicar a posição atual.

Figura 3.4: Ilustração do funcionamento horizontal da plataforma.



3.4 ATUADORES

Foram empregados dois atuadores a fim de controlar o movimento horizontal. São estes um inversor de frequência e um módulo arrancador controlador.

O arrancador-controlador utilizado neste projeto é o TeSys modelo U (Figura 3.5a), um dispositivo com coordenação total para partidas de motores até 32 A / 15 kW, que assegura as funções de proteção e comando de motores monofásicos ou trifásicos. Constituído por uma base de potência reversora de referência LU2B12BL, da Schneider Electric, de uma unidade de controle que responde às necessidades elementares de proteção das saídas-motor - sobrecarga e curto-circuito, de referência LUCA1XBL, da Schneider Electric, e de um módulo de comunicação acoplável Modbus (Schneider Electric, 2009).

Os inversores de frequência são ideais para o acionamento de motores assíncronos, como o motor utilizado na plataforma para conferir o movimento horizontal, em velocidade variável. Assim, o inversor de frequência escolhido para atuar no sistema foi o modelo ATV312H075M2 da marca Schneider Electric.

O inversor de frequência da série ATV312 (Figura 3.5b) proporciona ao usuário protocolo de comunicação CANopen e Modbus, como também cartões de comunicações opcionais para DeviceNet, Fipio, Modbus TCP e Profibus DP (Schneider Electric, 2011).

O inversor Altivar 312, para potência de 0,75kW (1CV), tensão monofásica de entrada 220V ca (saída trifásica até a tensão de entrada). O dispositivo também apresenta 2 entradas digitais e 3 entradas analógicas (0 – 10V, +/- 10V ou 0 – 20mA), com uma resolução de 0,1Hz para frequências de até 100Hz. Por fim, o inversor possui uma IHM local para configuração das funções e parâmetros de programação do dispositivo (Schneider Electric, 2011).

O motor que move a estrutura horizontalmente é um motor trifásico, que pode ser controlado, pelo TeSys U ou pelo ATV312. Além disso, este motor tem um codificador que informa a posição e o sentido de rotação do seu eixo.

Figura 3.5: a) Fotografia do arrancador-controlador da série TeSys U da Schneider Electric; b) Fotografia do inversor de frequência ATV312 da Schneider Electric.



a)

b)

Para as situações de urgência, foram utilizados dois atuadores: um botão de urgência e uma chave que permite o retorno seguro da instalação, após o ultrapasse dos limites de fim de curso. A Figura 3.6 apresenta a caixa com os dispositivos de urgência utilizados na plataforma.

Figura 3.6: Fotografia da caixa com botão e chave de urgência, da Schneider Electric.



3.5 CONTROLADOR

A National Electrical Manufacturers Association (NEMA) define um Controlador Lógico Programável como um aparelho eletrônico digital que utiliza uma memória programável para o armazenamento interno de instruções para implementações específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, para controlar, através de módulos de entradas e saídas, vários tipos de máquinas ou processos.

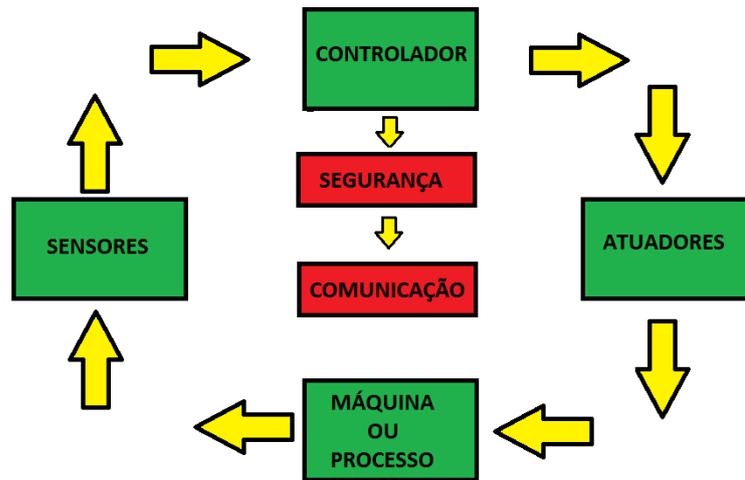
Os CLPs foram desenvolvidos na década de 60, com a finalidade de substituir painéis de relés que eram utilizados nas indústrias automobilísticas para executar controles baseados em lógicas combinacionais e sequenciais.

Atualmente, esses dispositivos são largamente utilizados na indústria no controle microprocessado de processos industriais, por serem facilmente programados via *software* por um operador, ao invés de utilizar relés e bobinas. As linguagens de programação permitem aos usuários se comunicarem com o CLP e definir as tarefas que o mesmo deverá executar. Pela normalização, esses controladores devem ter no mínimo três linguagens de programação: Ladder, Lista de Instruções e Diagrama de Funções.

Portanto, os CLPs são equipamentos microprocessados, projetados para comandar e monitorar máquinas ou processos industriais. Sua operação se baseia na execução de um programa previamente armazenado em sua memória, que tem por função executar o processamento de sinais de entrada e fornecer sinais de saída compatíveis com o acionamento de diversos atuadores.

A Figura 3.7 representa o ciclo de funcionamento de um sistema de automação genérico, onde podemos conferir as funções básicas do CLP.

Figura 3.7: Representação do ciclo de funcionamento de um sistema de automação.



Um CLP geralmente é composto por: fonte, CPU, módulos de entrada, módulos de saídas, módulos de função e módulos de comunicação. Esses são separados e interconectados através de um chassi que fornece proteção, conexão elétrica e suporte mecânico. Desta forma, o CLP pode ser montado na medida exata dos requisitos do sistema a ser controlado.

Esses controladores comunicam-se com diversos tipos de equipamentos e até mesmo com outros controladores. Essa versatilidade é importante, pois é possível a segmentação do controle de um processo entre diversos CLPs, e cada processo pode obter informação do outro através da comunicação entre os controladores.

O CLP utilizado na aquisição de dados dos sensores bem como na atuação sobre o arrancador-controlador e sobre o inversor de frequência foi o modelo TM22140R, da linha M221, do fabricante Schneider Electric, apresentado na Figura 3.8.

O TM22140R é programado por meio do *software* SoMachine Basic versão 1.4, sendo a linguagem escolhida a linguagem de contatos Ladder e a linguagem gráfica Grafcet. Possui 24 entradas discretas, 2 entradas analógicas e 16 saídas discretas, sendo possível a adição de até 7 módulos de expansão. Possui portas de comunicação integradas: USB com conector mini B USB 2.0, Ethernet com conector RJ45 e comunicação serial com conector RJ45 e interface RS232/RS485. Os serviços de comunicação presentes nesta CPU são: Modbus TCP, Ethernet/IP e DHCP cliente. A comunicação com o computador é feita através do protocolo Ethernet (Schneider Electric, 2017).

Figura 3.8: CLP modelo M221, da marca Schneider Electric.



3.6 INTERFACE HOMEM MÁQUINA

A interface operacional, ou seja, o equipamento que garante a interação entre um operador humano e uma máquina ou uma planta industrial, era construída por meio de componentes elétricos e eletromecânicos, sob a forma de painéis sinópticos, muitas vezes complexos e pesados com botões de pressão, interruptores, lâmpadas de sinalização, entre outros (PADHEE, S; SINGH, Y. 2011).

Devido ao aumento da complexidade dos processos industriais, nos últimos anos, esses painéis têm sido progressivamente substituídos por interfaces virtuais muito sofisticadas, usadas para dar apoio a um operador ou a uma equipe de operadores em uma sala de controle. A automação se destina a reduzir ou eliminar completamente a atenção contínua do operador quando o estado do processo está em condições normais. O operador humano age em seguida, como um supervisor. Quando o sistema de supervisão detecta uma anomalia, o operador humano deve agir como um controlador e reagir as condições anormais da planta. Estas instalações são conhecidas como interfaces homem-máquina.

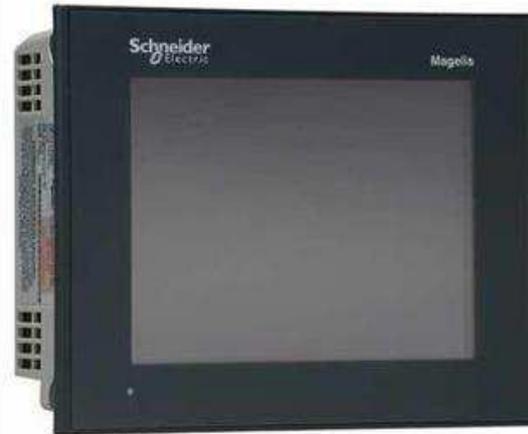
Para oferecer a um usuário uma IHM que lhe permitisse selecionar o modo de operação do sistema bem como observar o estado atual, foi desenvolvida uma interface que é apresentada em um terminal gráfico tátil de referência: Magelis HMISTU855, da Schneider Electric, o qual foi programado no *software* Vijeo Designer, versão 5.1, da Schneider Electric.

Esse *display* gráfico possui uma tela de 14.5 cm, resolução 320 x 240 pixels QCGA, colorido, com tecnologia TFT de 65536 cores e protocolos de comunicação Modbus e

Modbus TCP/IP, com grande variedade de serviços *web*, como a visualização a distância das páginas das aplicações, a partir de um navegador *internet* (Schneider Electric, 2012).

O terminal gráfico utilizado no sistema é ilustrado na Figura 3.9.

Figura 3.9: IHM modelo Magelis HMISTU855, da marca Schneider Electric.



O *software* Vijeo Designer é dedicado à configuração de IHMs Magelis. Possui função de gravação e visualização de vídeos, visualização das imagens em tempo real a partir de uma câmera ligada ao Magelis, codifica/decodifica sequências de gravação de vídeo sob formato MPEG no Magelis ou no formato AVI no Magelis iPC. Como também, simplifica as operações de manutenção, pois a função de acesso remoto *Web Gate* faz com que as aplicações da IHM operem através de um simples navegador de internet utilizando a arquitetura Ethernet (Schneider Electric, 2012).

3.7 GRADE DE COMANDO

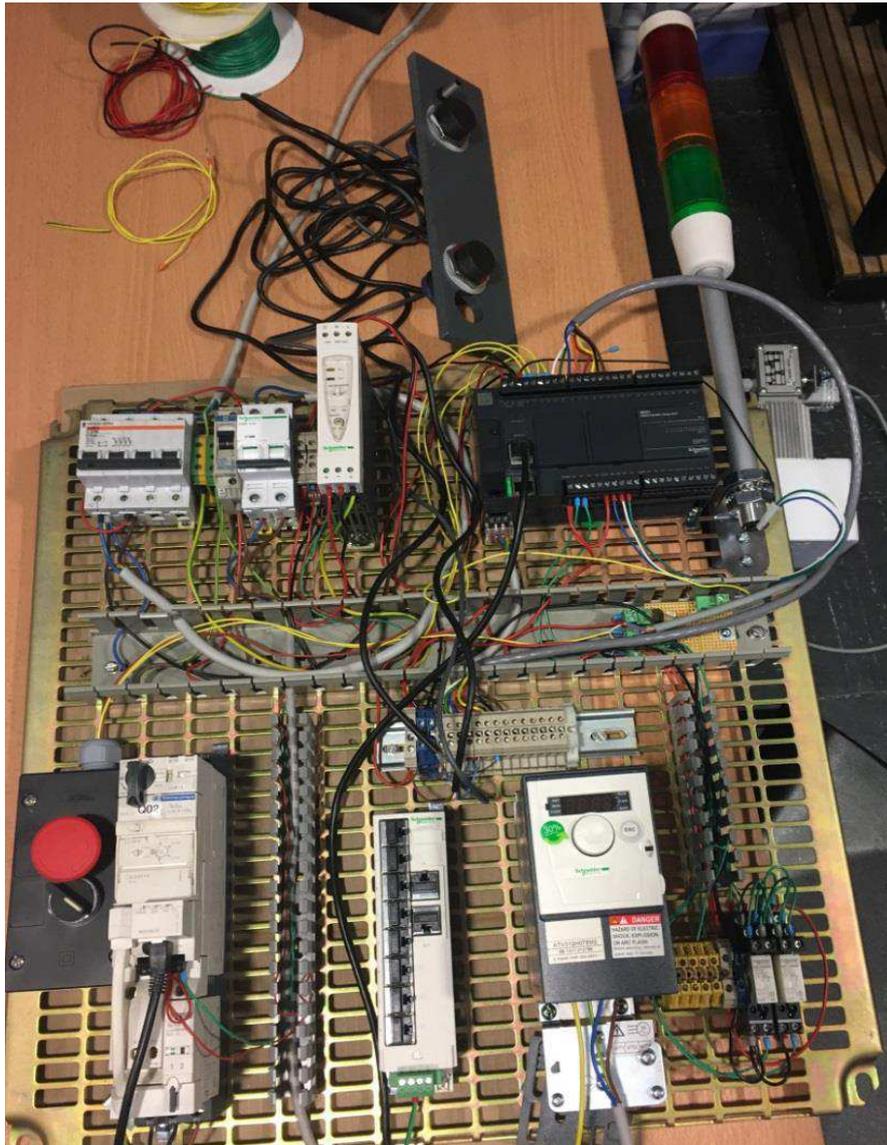
A grade de comando é composta por:

- Três disjuntores: um geral, um para o ATV e um para a alimentação 24V e o CLP;
- Uma alimentação 24V;
- Uma caixa com um Botão de Urgência e uma Chave;
- Um CLP M221, da Schneider Electric;
- Um arrancador-controlador TeSys U, da Schneider Electric;
- Um inversor de frequência ATV 312, da Schneider Electric;
- Um *hub* Modbus;

- Um terminal para entradas e um para saídas;
- Dois relés para controlar o motor subir/descer;
- Um sinalizador com as luzes verde, amarela e vermelha.

A grade de comando é apresentada na Figura 3.10.

Figura 3.10: Fotografia da grade de comando da plataforma.



4 DESENVOLVIMENTO DA APLICAÇÃO PARA O CONTROLE DA PLATAFORMA

A partir de uma nova aplicação desenvolvida no *software* SoMachine Basic versão 1.4, da Schneider Electric, realizou-se a configuração material e a lógica de controle para o atender as especificações propostas no experimento.

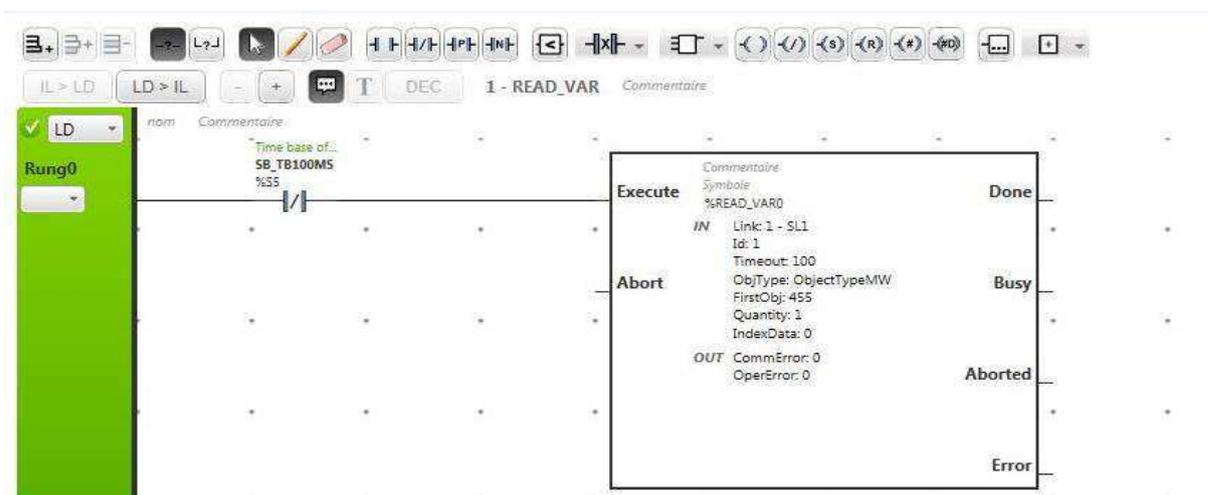
Na seção 4.1, é explicado o controle do motor a partir do TeSys U. A seção 4.2 apresenta o código para o controle do motor através do inversor de frequência ATV 312. Na seção 4.3, apresenta-se os modos de funcionamento do sistema e a gestão de entradas e de saídas. Por fim, na seção 4.4, aborda-se a gestão dos dispositivos de segurança.

4.1 COMANDO DO MOTOR A PARTIR DO TESYS U

Para comandar o motor que permite o movimento horizontal do suporte, podem-se utilizar dois atuadores presentes na plataforma: o arrancador-controlador TeSys U ou o inversor de frequência ATV 312.

SoMachine Basic disponibiliza os blocos de funções: Read_Var e Write_Var, dessa forma, o dispositivo TeSys U é programado de forma simples e rápida, como apresentado na Figura 4.1. Após análise da ficha técnica do dispositivo, os registros 455 e 704 foram utilizados para ler e escrever, respectivamente.

Figura 4.1: Bloco Read_Var, no SoMachine Basic.



Com um duplo clique no bloco de função, acede-se as propriedades do bloco, como apresentado na Figura 4.2. No exemplo abaixo, a primeira linha corresponde ao bloco Write_Var utilizado para comandar o motor.

A janela “Utilisé” está selecionada, pois o bloco de endereço %WRITE_VAR0 foi adicionado ao programa. O “Id” (endereço de identificação) escolhido é 1, pois o TeSys U é escravo número 1. “FirstObj” corresponde ao primeiro registro que será lido ou modificado pelo bloco, dessa forma, o registro 704 será utilizado para comandar o motor. “Quantity” indica o número de registros que serão lidos ou modificados e “IndexData” designa o index da primeira variável da memória formato *memory word* (%MW), que contém o valor do registro lido ou modificado pelo bloco de função. Nesse exemplo, escreve-se no registro 704 o valor da variável da memória %MW1.

Figura 4.2: Propriedade do bloco de função Write_Var, no SoMachine Basic.

Propriétés Write Var											
Utilisé	Adresse	Symbole	Link	Id	Timeout	ObjType	FirstObj	Quantity	IndexData	Commentaire	
<input checked="" type="checkbox"/>	%WRITE_VAR0		1 - SL1	1	100	0 (Mult. reg. - Mbs 16)	704	1	1		
<input type="checkbox"/>	%WRITE_VAR1		1 - SL1	1	100	0 (Mult. reg. - Mbs 16)	0	1	0		

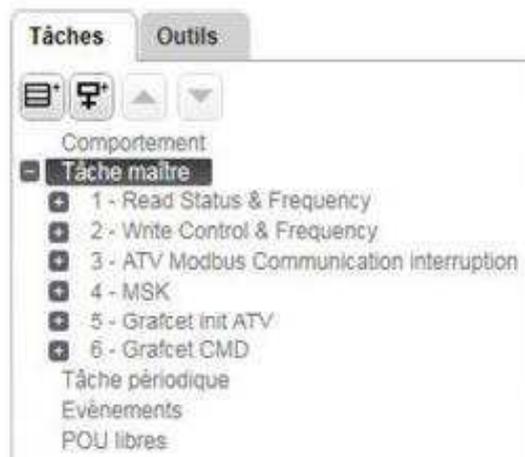
Assim, o comando do arracandor-controlador se faz ao modificar a variável %MW1 que está associada ao registro de comando 704 e em ler o registro de estado por intermediário da variável %MW0, associada ao registro 455.

4.2 COMANDO DO MOTOR A PARTIR DO ATV 312

O controle do motor pelo ATV 312 também é feito através dos blocos Read_Var e Write_Var. Após análise da ficha técnica do dispositivo, os registros 8603 e 8604 foram utilizados para ler o estado e ler a velocidade de referência, respectivamente. Os registros 8601 e 8602 foram utilizados para escrever a palavra de comando e a velocidade de referência, respectivamente.

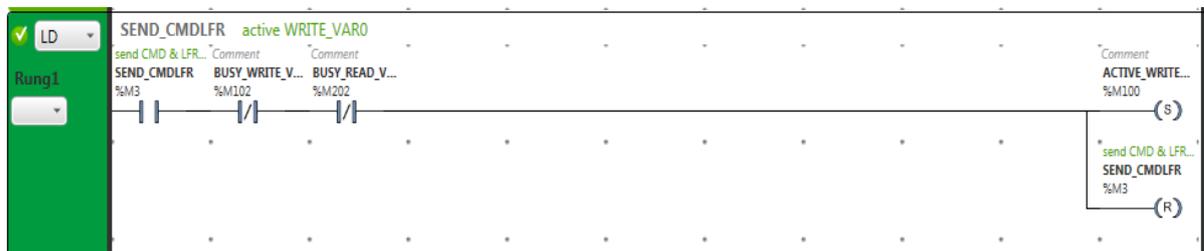
Inspirando-se na biblioteca de exemplos do SoMachine Basic, na qual há um programa proposto para realizar o comando do ATV312, criou-se a aplicação de comando. A Figura 4.3, mostra a divisão do programa em 6 POU's, sendo assim, uma aplicação mais complexa e trabalhosa de implementar do que a do TeSys U, como apresentado na seção anterior.

Figura 4.3: Divisão do programa de comando do ATV 312.



Um trecho do POU *Write Control & Frequency* é apresentado na Figura 4.4. Esta linha de código envia uma palavra de comando correspondente a um determinado estado do inversor de frequência.

Figura 4.4: Linha de código, em Ladder, para enviar uma palavra de comando.



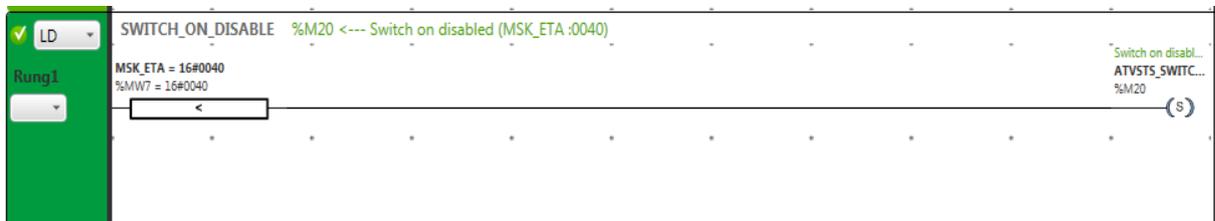
Por exemplo, a figura acima faz com que seja possível colocar o valor "1" no *bit* da memória %M100, este *bit* é conectado à entrada que executa o bloco *Write_Var*. Então sua ativação permitirá a modificação do registo associado ao bloco. Dessa forma todos os comandos são tratados. O ATV 312 pode estar em diferentes estados, dependendo da palavra de controle enviada. A figura 4.5 apresenta a lista de comandos do inversor de frequência presente na Tabela de Animação chamada *CMD*.

Figura 4.5: Tabela de animação CMD.

Used	Trace	Address	Symbol	Value	Force	Comment
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M0	ATVCMD_INIT			Reset and Initialize the ATV to reach the ready mode
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%MW0	ATVCMD_CMD			ATV Mot de commande @ 8601
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%MW1	ATVCMD_LFR			ATV Consigne de vitesse (0.1Hz) (-599.0 Hz ... 599.0 Hz) @8602
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M3	SEND_CMDLFR			send CMD & LFR to the ATV with WRITE_VAR0
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M10	ATVCMD_SHUTDOWN			"Shutdown" (CMD:0006)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M11	ATVCMD_SWITCH_ON			"Switch on"/Contactor command (CMD:0007)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M12	ATVCMD_ENABLE_OPERATION			"Enable operation"/Run command (CMD:000F)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M13	ATVCMD_DISABLE_OPERATION			Halt Stop without leaving the Operation enabled state (CMD:0007)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M14	ATVCMD_DISABLE_VOLTAGE			"Disable voltage"/Authorization to supply AC power (CMD:0000)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M15	ATVCMD_QUICK_STOP			"Quick stop"/Emergency stop (CMD:0002)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M16	ATVCMD_FAULT_RESET			"Fault reset"/Fault acknowledgment active (CMD:0080)

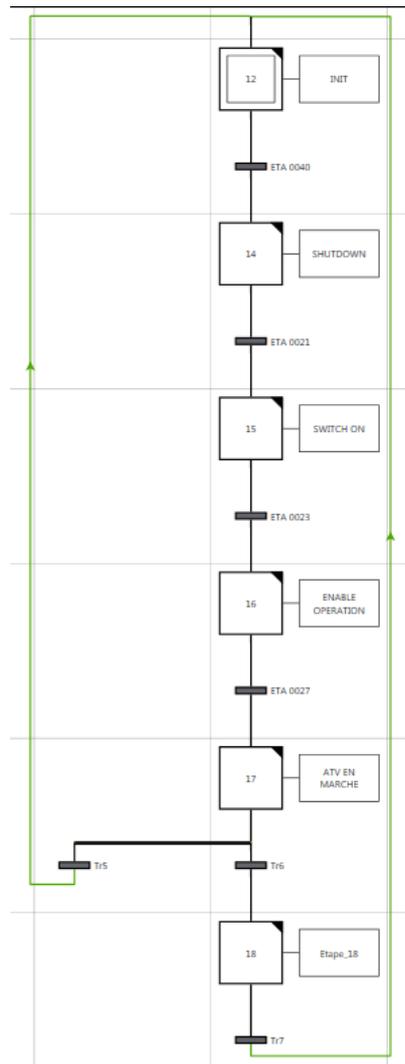
A figura 4.6 é um exemplo de linha de código presente no POU MSK. Assim, quando um comando é enviado, a unidade msk (*mask*) muda de acordo com o novo estado. Um *bit* da memória é definido como 1 ou 0, e isso faz com que a passagem de uma transição no Grafcet Init ATV.

Figura 4.6: Linha código do POU MSK.



O exemplo existente exige que todo o comando seja realizado através de modificações nos *bits* de memória presentes na Tabela de Animação CMD, apresentada na Figura 4.5. Contudo, a fim de simplificar o processo, implementou-se um outro POU: Grafcet CMD, apresentado na Figura 4.7, o qual realiza o comando do Grafcet automaticamente. Dessa forma, é apenas necessário comandar os *bits* de memória %M10 para ligar/desligar e o registro %MW1 para indicar a velocidade aplicada ao motor.

Figura 4.7: Grafcet de comando do ATV 312.

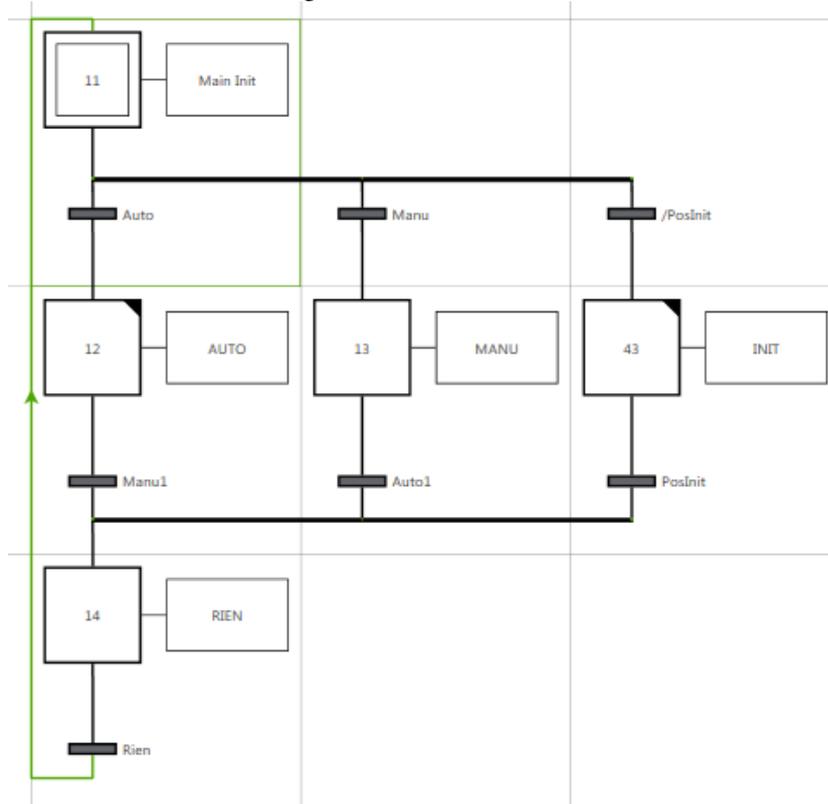


4.3 MODOS DE FUNCIONAMENTO

Na continuação do projeto, utilizou-se o comando do motor a partir do TeSys U.

Para a gestão dos modos de funcionamento do sistema, o Grafcet mestre, apresentado na Figura 4.8, foi implementado e permite a mudança de um modo de funcionamento para outro.

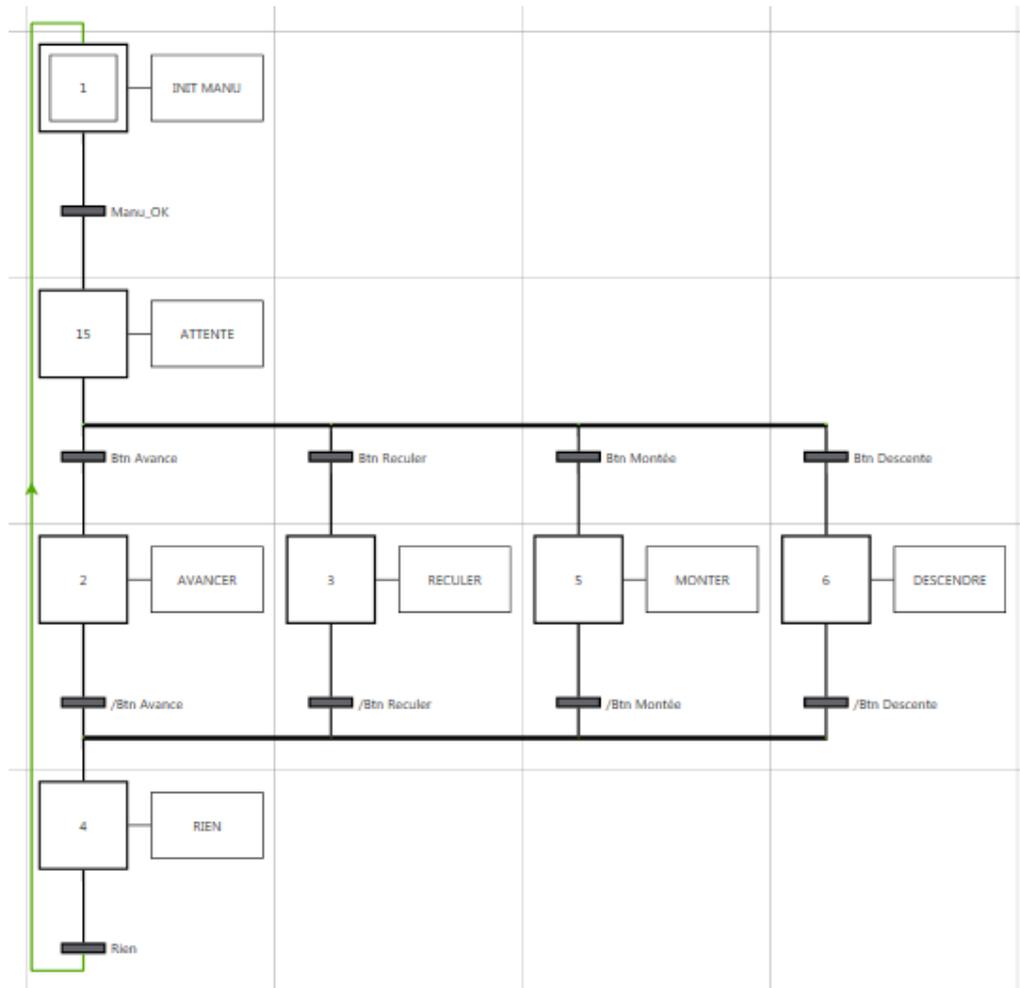
Figura 4.8: Grafcet mestre.



4.3.1 MODO MANUAL

Este modo permite controlar os diferentes movimentos do suporte, tanto horizontal quanto verticalmente. Sendo, então, para cima, para baixo, para frente e para trás. É necessário prestar atenção, pois as condições de transição devem levar em consideração os sensores de fim de curso. A Figura 4.9 apresenta o Grafcet desenvolvido para controlar o modo manual.

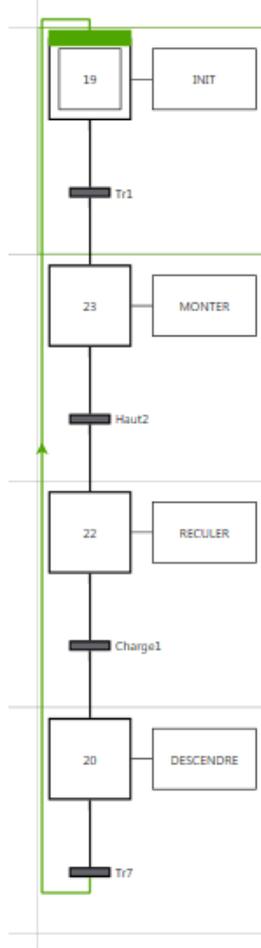
Figura 4.9: Grafcet modo manual.



4.3.2 MODO INICIALIZAÇÃO

No modo automático, se o suporte não está na posição inicial, ao pressionar o botão Início de Ciclo (DCY), realizar-se-á uma sequência de inicialização. Depois de terminar a inicialização, um segundo acionamento do botão irá iniciar o ciclo automático. A Figura 4.10 apresenta o Grafcet desenvolvido para realizar a inicialização.

Figura 4.10: Grafcet de inicialização.



4.3.3 MODO AUTOMÁTICO

A fim de realizar a sequência apresentada na Seção 2.1, um Grafcet Auto foi implementado. Este Grafcet não será aqui apresentado por seu extenso tamanho e por ser apenas uma simples sequência de comandos subir, descer, avançar e recuar, com algumas temporizações.

4.3.4 GESTÃO DAS ENTRADAS E SAÍDAS

O gerenciamento das entradas e saídas é feito em um POU específico e não diretamente nas etapas dos Grafcets. A razão pela qual a gestão é assim realizada, é porque SoMachine Basic destina-se a programação da gama de CLPs de entrada. Estes controladores não são capazes de atualizar as saídas ao mesmo tempo da execução do programa. Assim,

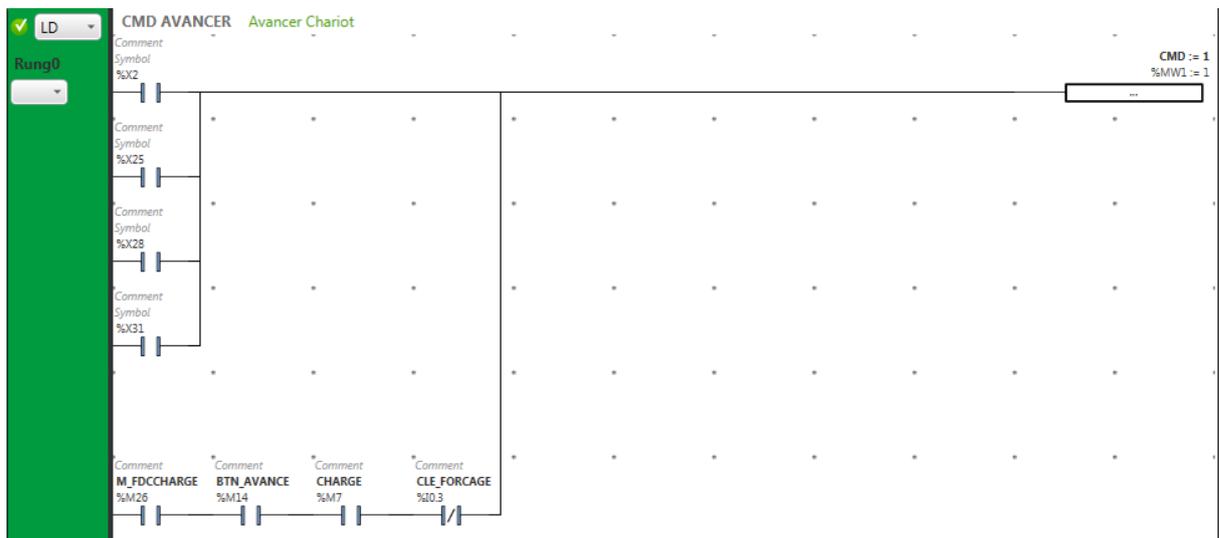
afetar uma saída em diferentes etapas de um Grafcet irá gerar erros ou avisos. Portanto, a gestão de entradas e saídas é realizada em POU's específicos e em linguagem Ladder ou IL.

Nas próximas duas seções é apresentado alguns exemplos de como a gestão das entradas e saídas é implementada.

4.3.4.1 GESTÃO DAS SAÍDAS

A Figura 4.11 apresenta a linha de código do comando avançar. Observa-se que as etapas %X2, %X25, %X28 e %X31 fazem com que %MW1 receba o valor "1", esta palavra da memória está associada ao registro de comando do motor e quando uma dessas etapas for ativada, o motor avançará. Para ocorrer o movimento de avançar, faz-se necessário atenção nas seguintes condições: além das etapas ativas, os sensores de fim de curso não podem estar ativados, o botão Avançar precisa estar pressionado e a chave precisa estar desativada.

Figura 4.11: Código, em Ladder, do comando avançar.



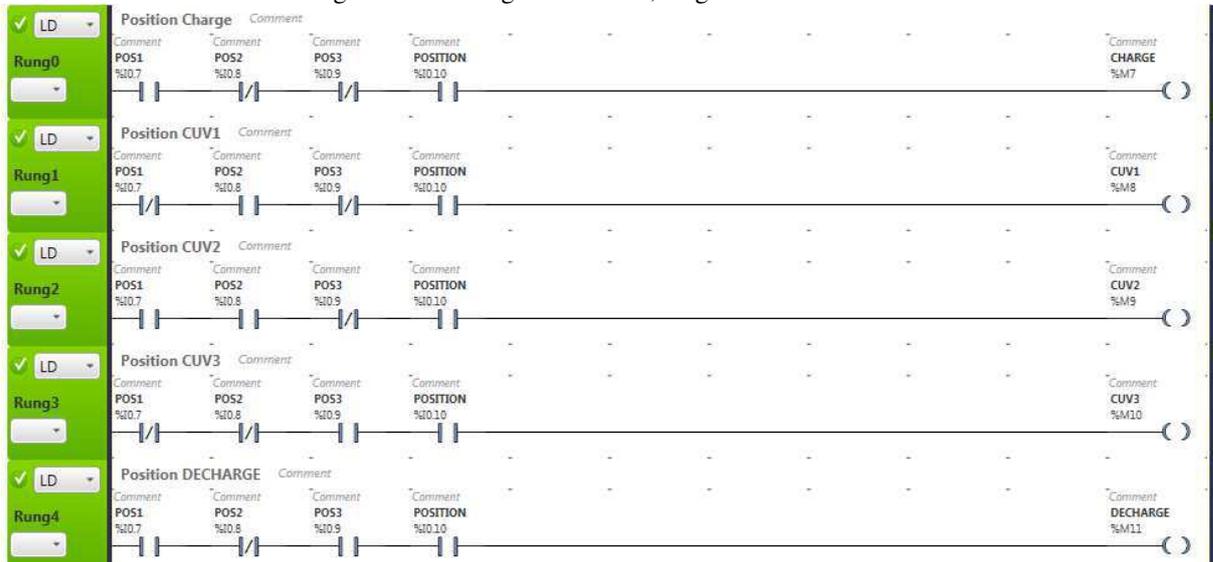
Os movimentos horizontais são controlados por meio do TeSys U que se comunica através da rede Modbus com o CLP. Já o motor que permite Descer e Subir é controlado simplesmente via dois relés.

4.3.4.2 GESTÃO DAS ENTRADAS

Para o POU que gerencia as entradas, as informações provenientes dos sensores de posição são as mais importantes. Portanto, tem-se um *rung* para as 5 posições possíveis em

que pode-se posicionar o suporte. Esta informação será útil para o modo automático, como também para a exibição na IHM. A Figura 4.9 apresenta o código em Ladder desenvolvido para a gestão das entradas.

Figura 4.12: Código em Ladder, de gestão das entradas.



Três sensores de posição são utilizados para a codificar o número do tanque no qual o suporte está situado. Estes sensores são de lógica direta e, portanto, seu estado é posto a 1 quando detectam a presença. Um quarto sensor é utilizado para detectar com precisão quando o suporte está no exatamente acima dos tanques ou das posições de carga e descarga. Ao contrário dos outros três sensores, esse é de lógica inversa, sendo assim, envia 0 quando detecta a presença.

4.4 GESTÃO DOS DISPOSITIVOS DE SEGURANÇA

A gestão da parada de urgência e dos sensores de fim de curso são indispensáveis para a segurança e proteção do material.

A plataforma está equipada com um botão físico de parada de urgência, uma chave e 4 sensores de fim de curso. Estes elementos são elementos de segurança, eles são ligados em lógica inversa, assim, "0" indica uma falha.

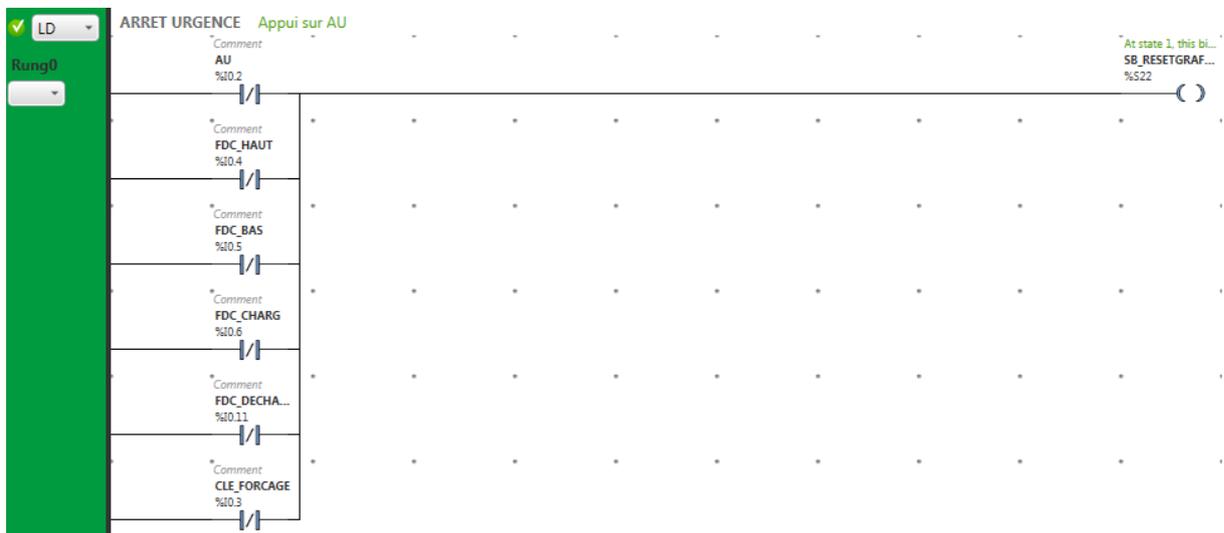
Acionamento do botão Parada de Urgência: nenhum modo de operação ou comando é permitido. Ao desarmar esse botão, permite-se a retomada da função normal e reinicializa-se todos os Graficets.

Detecção de limite pelos sensores de fim de curso: apenas o modo manual é permitido. Para retomar a operação normal, deve-se passar a chave para a posição padrão, e então reposicionar o suporte na posição normal, desabilitando o sensor de fim de curso.

Chave na posição padrão: bloqueia o sistema no estado que ele esteja. A principal função deste equipamento é para desarmar um erro acionado por um sensor de fim de curso. Ao contrário do botão de parada de urgência, o retorno à posição normal não reinicializa o sistema.

A Figura 4.13 apresenta o código em Ladder para a gestão ao apertar-se sobre o botão de urgência.

Figura 4.13: Código, em Ladder, de gestão dos dispositivos de segurança.



O *bit* do sistema %S22 permite interromper todos Grafkets no estado em que se encontram. A única forma de reiniciar esse *bit* da memória é ao colocar a chave em posição normal e ter o suporte em posição normal, sem nenhum sensor de fim de curso ativado.

5 CONFIGURAÇÃO DA IHM

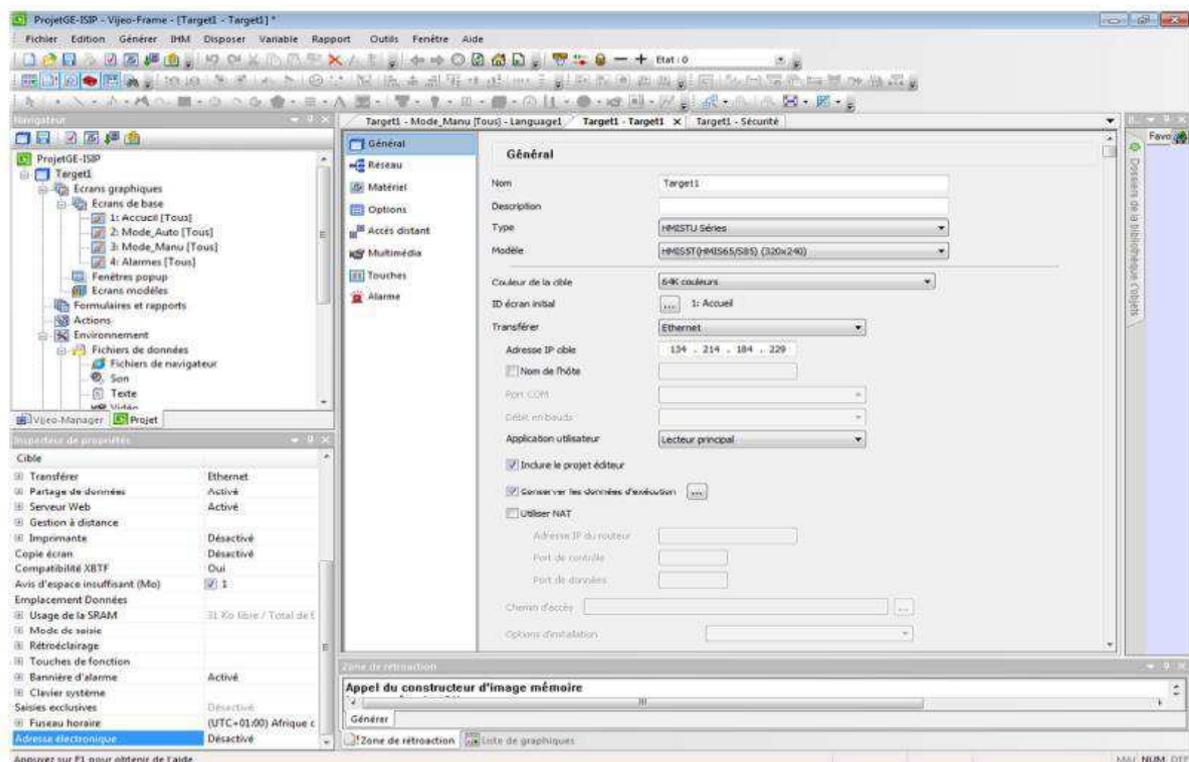
A IHM utilizada modelo Magelis HMISTU855, da Schneider Electric, foi programada por meio do programa Vijeo Designer. Essa ferramenta de configuração é dedicada a toda a gama Magelis da Schneider Electric. Desenvolvido usando tecnologias de mercado padrão, incluindo linguagens HTML/Java para navegação na *web*, JPEG para *screenshots*, MPEG4 para vídeo, Vijeo oferece uma interface intuitiva, com fácil configuração acelerando o processo de criação de aplicativos IHM (Schneider Electric, 2012).

5.1 CONFIGURAÇÃO MATERIAL

Antes de criar as interfaces, é necessário realizar a configuração material do Magelis. Primeiramente, a escolha do tipo e modelo: HMISTU - HMIS5T (HMIS65/S85) e em seguida o IP 134.214.184.229. A primeira transferência para a IHM Magelis deve ser feita através do cabo USB, pois inicialmente nenhum endereço IP é atribuído.

A Figura 5.1 apresenta a janela de configuração geral no Vijeo Designer.

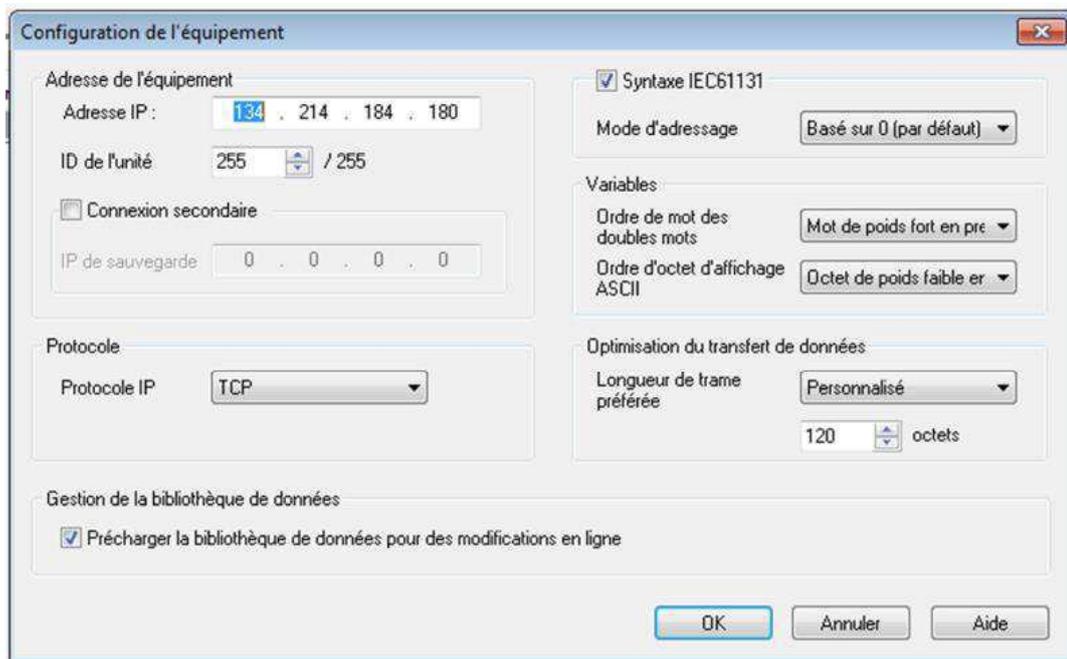
Figura 5.1: Configuração material da IHM no Vijeo Designer.



5.2 COMUNICAÇÃO COM O CLP

A fim de estabelecer a comunicação com o controlador, criou-se na janela Gestão de entradas/saídas um novo equipamento do tipo ModbusTCP/IP02, nomeado EquipementModbus02, sendo esse o CLP utilizado no projeto. A Figura 5.2 apresenta a configuração realizada.

Figura 5.2: Configuração do equipamento, no Vijeo Designer.



5.3 IMPLEMENTAÇÃO DA APLICAÇÃO

Três telas foram desenvolvidas para atender as especificações do projeto, sendo elas: tela inicial, modo automático e modo manual.

A tela inicial Tratamento de superfícies (do francês, Traitement des surfaces), apresentada na Figura 5.3, é a tela de recepção, na qual o usuário pode escolher o modo de funcionamento (Auto e Manu), criar uma nova sequência automática (do francês, Recettes, função não implementada). Cada botão presente nesta tela é conectado a uma variável corresponde no programa do CLP. A hora e a data também estão integradas em todas as telas.

Para os alarmes, criaram-se variáveis conectadas as variáveis do CLP que ativam as situações de alarme: botão de parada de urgência e caso de sensor de fim de curso. Quando um alarme é ativado, uma linha vermelha com a mensagem indicando o tipo de alarme

aparece abaixo da tela atual ao mesmo tempo que a IHM muda automaticamente para o modo manual, a fim de solucionar o problema.

Figura 5.3: Tela de início.

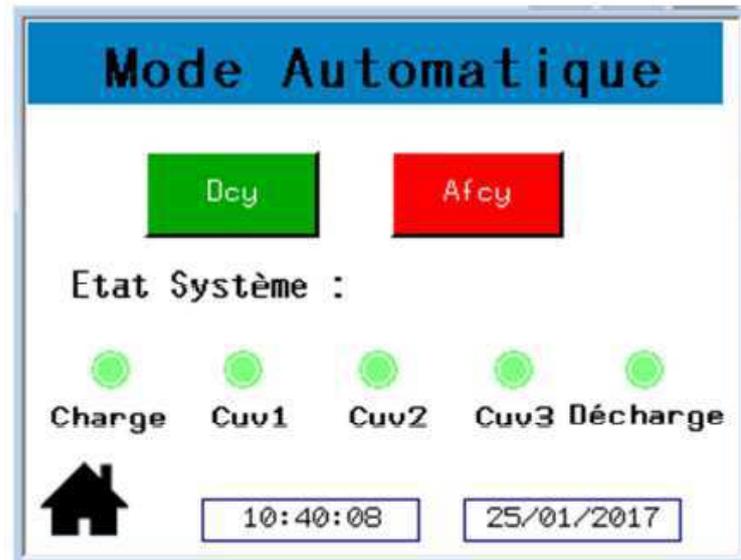


O modo automático, apresentado na Figura 5.4, é composto de três botões:

- Dcy (do francês, *Départ cycle*): Início de ciclo;
- Afcy (do francês, *Arrêt de fin cycle*): Parar o ciclo;
- Home: retornar a tela de início.

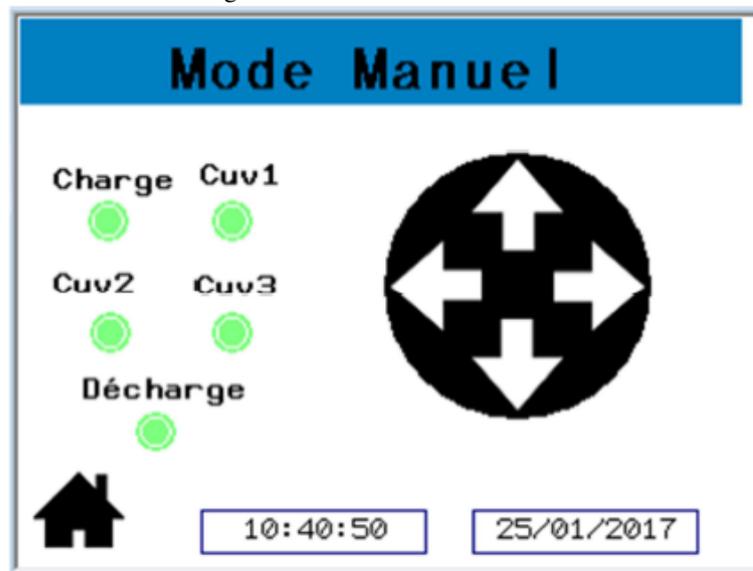
Também pode-se observar os diferentes estados do sistema (do francês, *état do système*): Inicialização, Ciclo contínuo e Parada requisitada. Os LEDs correspondem a posição do suporte horizontal, quando o suporte está em alguma dessas posições o LED fica amarelo, permitindo ao usuário saber a posição sem necessidade de se movimentar até a plataforma.

Figura 5.4: Tela do modo automático.



A tela do modo manual (do francês, Mode Manuel), apresentada na Figura 5.5, possui cinco botões: Avançar, Recuar, Subir, Descer e Home. Os quatro primeiros são apresentados em formato de flecha e permitem o movimento das estruturas, quando pressionado o botão torna-se amarelo. Nessa tela pode-se verificar a posição que a estrutura se encontra através dos LEDs (do francês, charge e decharge, significando as posições de carga e descarga).

Figura 5.5: Tela do modo manual.



6 CONCLUSÃO

A principal contribuição deste trabalho foi a automatização completa da Plataforma de Tratamento de Superfícies, a qual pode ser utilizada pelos alunos de engenharia elétrica do INSA Lyon para realizar o experimento da disciplina Engenharia de Sistemas Integrados de Produção (do francês, Ingénierie des Systèmes Intégrés de Production).

A concepção mecânica avançou, contudo, o material para a construção completa da plataforma não chegou a tempo da realização deste trabalho. Toda a grade de comando foi montada e devidamente configurada. A parte elétrica também foi terminada, restando apenas a montagem final da plataforma.

O comando do motor através do inversor de frequência e do arrancador-controlador foi implementada através do protocolo Modbus, os modos de funcionamento foram elaborados na aplicação do CLP, como também a IHM foi configurada e a comunicação com o controlador estabelecida. Sendo assim, as especificações iniciais do projeto foram atendidas.

6.1 SUGESTÃO PARA TRABALHOS FUTUROS

Como opção para possíveis trabalhos futuros utilizando a plataforma de tratamento de superfícies, sugere-se:

- Adicionar a criação de uma nova sequência automática, na IHM.
- Utilizar um codificador (sensor no motor) para realizar a gestão da posição.
- Comandar o motor via o inversor de frequência ATV312.

7 REFERÊNCIAS

- CASSIOLATO, C., 2011. **Sistemas Inteligentes, redes industriais, conectividade resultados garantidos.** Disponível em: <<http://www.smar.com/brasil/artigo-tecnico/redes-industriais>>. Acesso em: 13 de junho de 2017.
- SANTOS, L., 2016. **As vantagens clássicas para Redes Industriais.** Disponível em: <<http://blog.murrelektronik.com.br/redes-industriais-revendo-vantagens/>> Acesso em: 13 de junho de 2017.
- MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. 2012. **Especificações do protocolo Modbus.** Disponível em: <modbus.org/specs.php>. Acesso em: 18 de junho de 2017.
- Schneider Electric. 2016. **SoMachine Basic Guia de Instruções.** Disponível em: <www.schneider-electric.com> Acesso em: 15 de junho de 2017.
- National Instruments. 2014. **Étude approfondie du protocole Modbus.** Disponível em: <<http://www.ni.com/white-paper/52134/fr/>> Acesso em: 18 de junho de 2017.
- Schneider Electric. 2009. **General Specification for Integrated Motor Starter-Controller below 15 kW, 690 V.** Disponível em: <www.schneider-electric.com> Acesso em: 21 de junho de 2017.
- Schneider Electric. 2011. **Altivar 312 Inversores de frequência para motores assíncronos Manual de instalação e programação.** Disponível em: <www.schneider-electric.com> Acesso em: 21 de junho de 2017.
- Schneider Electric. 2017. **Contrôleurs logiques Modicon M221 – Catalogue.** Disponível em: <www.schneider-electric.com>. Acesso em: 25 de junho de 2017.
- Schneider Electric. 2012. **Magelis HMI STU 655/855 User Manual.** Disponível em: <www.schneider-electric.com>. Acesso em: 25 de junho de 2017.
- Schneider Electric. 2012. **Magelis HMI GTO for Vijeo Designer User Manual.** Disponível em: <www.schneider-electric.com>. Acesso em: 25 de junho de 2017.
- PADHEE, S; SINGH, Y. **Data logging and supervisory control of process using LabVIEW.** Students Technology Symposium (TechSym), Janeiro 2011, pp.329-334.