

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ALISON MADEIRA CUNHA

**CONTROLE E PARAMETRIZAÇÃO DE INVERSOR DE
FREQUÊNCIA VIA ELIPSE SCADA E PROTOCOLO MODBUS**

VIÇOSA
2015

ALISON MADEIRA CUNHA

**CONTROLE E PARAMETRIZAÇÃO DE INVERSOR DE
FREQUÊNCIA VIA ELIPSE SCADA E PROTOCOLO MODBUS**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. B. Sc. Daniel K. D. Villa.

VIÇOSA
2015

ALISON MADEIRA CUNHA

**CONTROLE E PARAMETRIZAÇÃO DE INVERSOR DE
FREQUÊNCIA VIA ELIPSE SCADA E PROTOCOLO MODBUS**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 15 de Julho de 2015.

COMISSÃO EXAMINADORA

Prof. B.Sc. Daniel Khéde Dourado Villa - Orientador
Universidade Federal de Viçosa

Prof. Dr. Alexandre Santos Brandão - Membro
Universidade Federal de Viçosa

Prof. B.Sc. Gerson Ovídio Luz Pedruzi - Membro
Universidade Federal de Viçosa

“As pessoas tiram da vida exatamente o que investiram nela.”

(Joy Adason)

Dedico esta vitória especialmente aos meu pais, Josué e Sirlene, que juntos me criaram e me fizeram ser quem eu sou, dedico, também, aos meus irmãos e familiares.

Agradecimentos

À minha família, pela confiança e motivação e compreensão pela ausência involuntária.

Aos amigos e colegas, pela força e pela vibração com o meu sucesso em relação a esta jornada.

Aos professores e colegas de Curso, pois juntos trilhamos uma etapa importante de nossas vidas.

A todos que, com boa intenção, colaboraram para a realização e finalização deste trabalho e aos que não impediram a finalização deste estudo.

Resumo

Todo procedimento industrial carece ser controlado e monitorado, a criação de sistemas de supervisão e controle é, então, indispensável para o crescimento das empresas, pois reúne todas as variáveis do processo e as apresenta de maneira amigável ao operador. Os inversores de frequência são também importantes na automatização de um processo, porque conseguem controlar e/ou manter constantes o torque e a velocidade de motores, mas para isso os inversores devem, primeiramente, ser parametrizados. Por muitas vezes isto é feito de modo local, entretanto os sistemas supervisórios podem ser utilizados para fazer a parametrização de maneira remota. Para tanto, o padrão serial RS-485 pode ser utilizado como meio de transmissão e o protocolo Modbus para troca de mensagens entre os dispositivos, implementando assim um sistema de comunicação entre inversor e supervisório. Para a criação do sistema supervisório foi utilizado o *software* Elipse SCADA, este sistema além de fazer a parametrização do inversor CFW-08, controla o tipo de acionamento de um motor de indução, podendo se escolher entre partida estrela-triângulo e a partida via CFW-08. O CLP juntamente com contadores tem a função de fazer a escolha do tipo de partida e a implementação do sistema supervisório no protótipo tornou possível a parametrização e o controle do inversor de forma remota.

Abstract

Every industrial procedure needs to be controlled and monitored, supervisory systems are essential for the company growth, because it gathers all process variables and presents in a friendly way to the operator. The AC drivers are also important in automating a process, because they can keep the torque and the speed of the engines constant, but first the inverters must be parameterized, frequently this is done in locally, however the supervisory systems can be used to make the parameter setting remotely. Thus, The RS-485 pattern can be used for data transmission and the Modbus protocol for message exchange between devices, implementing a communication system between inverter and supervisory. For the supervisory system creation was used Elipse SCADA software, this system besides making the CFW-08 inverter parameterization, is capable to choose between star-delta engine starting and the engine connection with the AC driver, the Programmable Logic Controller along with contactors has the function to make the choice of the motor starting. Moreover, the implementation of the supervisory system in a prototype, fulfilled this work objective, making communication with the computer and making it possible to parameterize the inverter and also controls it.

Sumário

1	Introdução.....	14
1.1	Redes Industriais	15
1.1.1	Introdução.....	15
1.1.2	Rede RS-485.....	16
1.2	Inversores de Frequência	17
1.3	Controlador Lógico Programável	18
1.4	Sistemas de Supervisão e Controle.....	19
1.5	Objetivos.....	22
2	Materiais e Métodos	23
2.1	Modbus RTU	23
2.2	CLP WEG Clic-02 20VT-D	25
2.3	Inversor de Frequência CFW-08	26
2.3.1	Aplicação do Protocolo Modbus ao Inversor CFW-08	27
2.3.2	Parametrização do Inversor para Funcionamento na Rede Modbus	29
3	Desenvolvimento do Sistema de Automação.....	31
3.1	Seleção do Tipo de Acionamento.....	31
3.1.1	Acionamento via CLP	33
3.1.2	Acionamento via Inversor de Frequência.....	34
3.1.3	Parametrização do Inversor para Funcionamento do Controle Vetorial	36
3.2	Comunicação Entre os Dispositivos.....	36
3.4	Desenvolvimento do Sistema Supervisório.....	37
3.4.1	Etapas do Desenvolvimento do Supervisório.....	37
3.4.2	Configuração das <i>Tags</i>	42
3.4.3	Leitura e Escrita de Parâmetros	43
3.4.4	Leitura e Escrita de Variáveis Básicas	45
3.4.5	Tela Principal.....	47
3.4.6	Tela Motor	49
3.4.7	Tela Parametrização	50
3.4.8	Tela Alarmes.....	51
3.4.9	Tela Gráficos	52
4	Considerações Finais.....	55
	Referências Bibliográficas	56
	Apêndice A – Configuração do Elipse SCADA na Rede Modbus RTU.....	59
A.1	Configuração do Conversor RS485- USB.....	63
A.2	Configuração das Tags PLC	65

Lista de Figuras

Figura 1 - Níveis da Pirâmide de Automação.....	16
Figura 2 - Gráfico do comprimento do cabo em função da velocidade de comunicação.....	17
Figura 3 - Exemplo de um sistema supervisório. Controle e manutenção de silos.	20
Figura 4 - Tela de Trabalho do Elipse SCADA.....	22
Figura 5 - Modelo Mestre-Escravo.....	24
Figura 6 - Transmissão de dados modo RTU.	24
Figura 7 - Estrutura das mensagens.	25
Figura 8 - CLP da linha Clic02.....	25
Figura 9 - Configurações do Clic02.	26
Figura 10 - Inversor CFW-08 com módulo KRS-485.....	28
Figura 11 - Sistema de automação desenvolvido.	32
Figura 12 - Esquema de conexão dos contadores para partida estrela-triângulo ou via CLP. ...	32
Figura 13 - Contator WEG CW07.....	33
Figura 14 - Módulo de expansão CLW-02/8ER-D.	33
Figura 15 - Diagrama ladder para o sistema de escolha do tipo de partida.	35
Figura 16 - Conexão entre os dispositivos da rede Modbus.....	37
Figura 17 - Conversor RS485-USB STM-8522.	37
Figura 18 - Organização das variáveis.	40
Figura 19 - Condições para ativação de alarmes da <i>tag</i> velocidade.....	40
Figura 20 - Hierarquia entre as telas do sistema supervisório.....	41
Figura 21 - Associação do driver Modbus a uma <i>tag</i> PLC.	42
Figura 22 - Associando uma <i>tag</i> PLC a um objeto <i>setpoint</i>	44
Figura 23 - Leitura do parâmetro P003. (a) Associação da <i>tag</i> PLC à <i>tag</i> expressão. (b) Propriedades do <i>display</i> de corrente.	45
Figura 24 - Tela Principal.....	47
Figura 25 - Criação dos botões de tipo de partida. (a) <i>Tag</i> PLC relacionada às variáveis M01~M03. (b) Valor escrito no botão Estrela-Triângulo.....	47
Figura 26 - Configuração de um objeto de animação. (a) <i>Tag</i> PLC relacionada às variáveis Y01~Y04. (b) Zonas em que o LED “acenderá”.	48
Figura 27 - Tela Motor.	49
Figura 28 - Configuração do botão liga/desliga motor.....	50
Figura 29 - Configuração do botão sentido de giro.....	51
Figura 30 - Tela Parametrização.....	51

Figura 31 - Configuração de um objeto <i>slider</i>	52
Figura 32 - Tela Alarmes.....	52
Figura 33 - Alarmes da <i>tag</i> velocidade.....	53
Figura 34 - Tela Gráficos.....	53
Figura 35 - Propriedades da tendência.	54
Figura 36 - Tutorial de como adicionar o driver Modbus ao Elipse SCADA.....	59
Figura 37 - Configuração do driver Modbus. (a) Três primeiros passos. (b) Quarto passo.	60
Figura 38 - Tipos de operações feitas pelo driver Modbus do Elipse.	61
Figura 39 - Configuração do Modo Modbus e <i>Offset</i>	62
Figura 40 - Mudando o <i>timeout</i>	62
Figura 41 - Configuração da porta serial no Elipse SCADA.	63
Figura 42 - Localização do cabo conversor no Gerenciador de Dispositivos do Windows.	64
Figura 43 - Propriedade de Prolific USB-to-Serial Comm Port.....	64
Figura 44 - Configuração do parâmetros N1, N2, N3, N4 e Scan.....	65
Figura 45 - Configuração de uma <i>tag</i> PLC para acessar o parâmetro P003.....	66

Lista de Tabelas

Tabela 1 - Tipos de Parâmetros do inversor WEG CFW-08.....	18
Tabela 2 - Etapas de planejamento de um sistema supervisorio.	21
Tabela 3 - Endereço das variáveis do CLP.....	27
Tabela 4 - Funções Modbus disponíveis no CFW-08.	28
Tabela 5 - Endereço dos Parâmetros.....	28
Tabela 6 - Endereço das variáveis básicas.....	29
Tabela 7 - Endereço dos bits de estado e comando.	29
Tabela 8 - Ajuste dos parâmetros P308 e P312.	30
Tabela 9 - Ajuste dos Parâmetros para Controle do Inversor.....	30
Tabela 10 - Simbologia utilizada no diagrama ladder.....	34
Tabela 11 - Parâmetros do motor.	36
Tabela 12 - Variáveis do CLP utilizadas no supervisorio.	38
Tabela 13 – Parâmetros e variáveis básicas do inversor utilizados no supervisorio.	38
Tabela 14 - Resoluções utilizadas na comunicação serial.....	39
Tabela 15 - Ciclo de <i>scan</i> definido para cada variável.....	39
Tabela 16 - Endereço das <i>tags</i> utilizadas.....	43
Tabela 17 - Exemplo de telegrama para escrever no parâmetro P100.	45
Tabela 18 - Exemplo de telegrama para ler o parâmetro P003.	45
Tabela 19 - Bits da Variável Básica V03.	46
Tabela 20 - Valores de escrita e leitura da variável básica V03.....	46
Tabela 21 – Situações em que os LEDs do tipo de partida serão ativados.....	48
Tabela 22 - Situações em que os LEDs do tipo de ligação são ativados.	49
Tabela 23 - Detalhamento das funções Modbus.....	66

1 Introdução

Para obter um melhor desempenho, os processos industriais atuais têm uma necessidade de serem controlados para que este controle seja eficiente, é necessário um monitoramento adequado (LOPES, 2009).

A criação de sistemas supervisórios para controlar processos industriais, é essencial para o crescimento das empresas, já que um controle feito de forma competente produz ganhos expressivos de qualidade e produtividade, além de garantir a segurança dos operadores (FERREIRA JÚNIOR, 2012). Existe, também, a necessidade de centralizar informações, para que se tenha o maior número de dados com o menor tempo possível (RESENDE, 2014). Estes sistemas são conhecidos como SCADA (*Supervisory Control and Data Acquisition*). Segundo Pressman (2006 apud RAYSARO, 2012), este é um mecanismo por meio do qual se estabelece um diálogo entre um programa e o ser humano, qual pode-se chamar de Interface Homem-Máquina (IHM).

As formas em que estes sistemas SCADA se comunicam com outros dispositivos, como, por exemplo, um controlador lógico programável (CLP) ou um inversor de frequência, são feitas através da comunicação serial e dos protocolos que efetuarão a conexão e troca de mensagens entre os equipamentos. O protocolo do inversor de frequência e do CLP utilizado neste trabalho é o Modbus RTU. Este é um protocolo bastante difundido, por ser simples, ter especificação aberta e ter grande oferta de produtos por múltiplos fabricantes (GUARESE, 2011). Além disso a utilização deste protocolo reduz o sistema de controle em termos de *hardware*, pois muitos dispositivos podem ser conectados com um condutor, diminuindo a quantidade de cabos, eletrodutos e painéis (ROSÁRIO, 2005 apud PEREIRA, 2011). Com a menor exigência de *hardware*, o tempo de instalação e a necessidade de fontes de energia também são reduzidos.

Nesse trabalho será apresentado um exemplo simplificado de controle de processo. No caso, trata-se do controle e parametrização de um inversor de frequência CFW-08, da empresa WEG, por meio de um computador pessoal usando o *software* Elipse SCADA. A conexão entre inversor/computador será feita através de um cabo de comunicação serial RS485 com um adaptador USB e a troca de informações será dada por meio do protocolo Modbus RTU. Um CLP fará o acionamento de contadores que, dependendo da sequência, fará a partida estrela-

triângulo de um motor de indução ou conectará o inversor de frequência ao motor de indução, que poderá ser monitorado e controlado através de um sistema supervisor.

Em uma empresa, normalmente os inversores ficam no nível mais baixo, distante do operador, então, como os inversores têm a necessidade de serem parametrizados para que funcionem corretamente, é aconselhável a criação de um sistema que faça a parametrização e a operação destes inversores remotamente, assim evita-se a locomoção do operador e agiliza-se o processo. Daí vem a ideia da criação de um sistema SCADA para, parametrizar, monitorar e controlar as ações de um inversor de frequência.

1.1 Redes Industriais

1.1.1 Introdução

Segundo Tovar (1999), as redes industriais são amplamente empregadas no suporte a sistemas controlados por computadores, em funções que vão desde o controle de processos até à automação industrial. Rotineiramente, estes sistemas impõem condições de tempo real à rede de comunicação, ou seja, as mensagens devem ser enviadas e recebidas num tempo menor que um limite imposto.

A revolução da comunicação industrial na tecnologia da automação, segundo Cassiolato (2011), está revelando um enorme potencial na otimização de sistemas de processo e tem feito uma importante contribuição na direção da melhoria no uso de recursos.

Nota-se pela Figura 1 que as redes de comunicação estão presentes em todos os níveis de interação. Com a finalidade de tornar possível a troca de informações entre os computadores, CLPs, relés de proteção, inversores de frequência, controladores de processos, atuadores de válvulas, dentre outros, estas redes devem ser padronizadas para possibilitar uma maior integração de todas essas ferramentas. Vários protocolos foram então criados, cada um com suas características, para que a comunicação fosse padronizada. Os protocolos mais comuns são: MODBUS, PROFIBUS, ETHERNET e FIELDBUS FOUNDATION (VILLA, 2014).

Para César Cassiolato:

A capacidade de comunicação entre dispositivos e o uso de mecanismos padronizados, abertos e transparentes são componentes indispensáveis do conceito de automação de hoje. A comunicação vem se expandindo rapidamente no sentido horizontal nos níveis inferiores (*field level*), assim como no sentido vertical integrando todos os níveis hierárquicos. (CASSIOLATO, 2011)

No processo de produção contemporâneo as informações passam por meio de uma série de níveis corporativos, de tomada de decisões, e estes as retransmitem com os departamentos, para que todas as decisões possam sair da administração da empresa (VILLA, 2014).

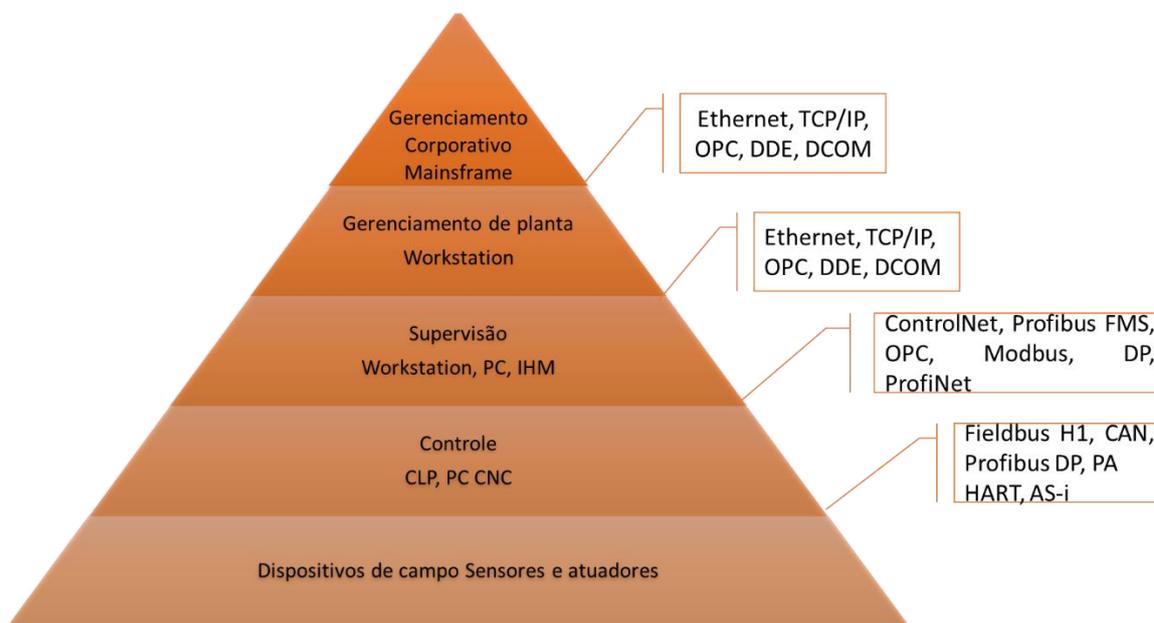


Figura 1 - Níveis da Pirâmide de Automação

Nas palavras de Jacir Luiz Bordim (2006) as redes industriais são:

[...] essencialmente sistemas distribuídos, ou seja, diversos elementos trabalham de forma simultânea a fim de supervisionar e controlar um determinado processo. Tais elementos (sensores, atuadores, CLPs, PCs, etc.), necessitam estar interligados e trocando informações de forma rápida e precisa. (BORDIM, 2006)

1.1.2 Rede RS-485

Nos anos 60 o comitê conhecido como *Electronic Industries Association* (EIA) criou uma interface de comunicação de dados entre equipamentos chamado de RS (*Recommended Standard*), que tinha o objetivo de trocar dados digitais entre um computador central (*Mainframe*) com os terminais remotos ou apenas entre dois terminais, assim surgiu os padrões RS (MARIA, 2007).

As redes RS-485 podem possuir altas taxas de transmissão de dados e grande comprimento de rede, podendo chegar a 1200 metros com um *baud rate* de 9600 bits por segundo. A Figura 2 mostra como varia o comprimento do cabo em relação a velocidade de comunicação. Além disso, o baixo custo de *hardware*, por utilizar só dois condutores do par trançado para troca de informação, faz com que este padrão seja amplamente utilizado.

O padrão RS-485 é uma comunicação multiponto, a qual permite conectar até 32 dispositivos. Em funcionamento, apenas um dispositivo envia dados por vez, enquanto isso, os demais escutam, e somente aquele endereçado pelo pacote de dados interpreta a informação enviada (SANTOS, 2007).

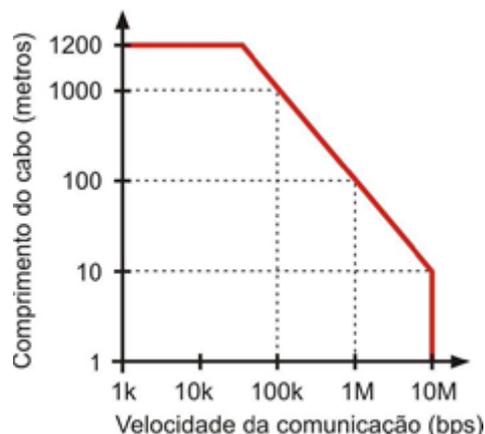


Figura 2 - Gráfico do comprimento do cabo em função da velocidade de comunicação.

1.2 Inversores de Frequência

Os inversores de frequência são equipamentos projetados especificamente para controlar e variar a velocidade ou torque de motores de corrente alternada com eficiência. (BRUGNEROTTO; KUME, 2013)

Para isso, o inversor é capaz de produzir tensão e frequência trifásicas ajustáveis. Como decorrência dessa habilidade, os inversores permitem um acionamento suave do motor, evitando possíveis quebras. Além disso, eles asseguram que o motor não será prejudicado quando há problemas na rede elétrica, como sobrecarga, queda de tensão, entre outros (FERREIRA JÚNIOR, 2012).

Portanto, um inversor de frequência transforma a tensão da rede, de amplitude e frequência constantes, em uma tensão de amplitude e frequência variáveis (WEG, 2006).

O inversor de frequência tem como característica ser parametrizável, o que faz possível sua utilização em uma vasta gama de processos. Neste trabalho essas configurações serão inseridas através da interface de comunicação serial. Franchi (2009) versa que “As funções de um inversor de frequência são executadas de acordo com parâmetros predefinidos alocados na CPU. Os parâmetros são agrupados de acordo com as suas características e particularidades”

De acordo com Capelli (2002), para o inversor funcionar de forma satisfatória, não deve-se somente instalá-lo corretamente, é necessário informá-lo em quais condições de trabalho ele operará, o que consiste em sua parametrização. O número de parâmetros é proporcional a quantidade de recursos oferecidos pelo inversor.

Os parâmetros de um inversor de frequência são agrupados em cinco tipos, assim como mostra a Tabela 1.

Tabela 1 - Tipos de Parâmetros do inversor WEG CFW-08

Parâmetros de Leitura	Variáveis que podem ser visualizadas nos <i>displays</i> , mas não podem ser alteradas pelo usuário.
Parâmetros de Regulação	São os valores ajustáveis a serem utilizadas pela função do inversor.
Parâmetros de Configuração	Definem as características do inversor, as funções a serem executadas, bem como as funções das entradas/saídas do cartão de controle.
Parâmetros do Motor	São os dados do motor em uso: informações contidas nos dados de placa do motor e aqueles obtidos pela rotina de Auto-ajuste.
Parâmetros das Funções Especiais	Inclui os parâmetros relacionados às funções especiais.

1.3 Controlador Lógico Programável

Um sistema controlador é um dispositivo ou um conjunto de dispositivos que conduzem o comportamento de outros aparelhos (atuadores, contadores, etc.). Hoje em dia, todo processo precisa de algum tipo de controlador para garantir uma melhor segurança e viabilidade econômica (PEREIRA, 2011).

O Controlador lógico programável (CLP), também conhecido pela sigla em inglês PLC (*Programmable Logic Controller*), é um dispositivo de estado sólido – um computador industrial, capaz de guardar instruções para implementação de funções de controle (sequência lógica, temporização e contagem, por exemplo), além disso realiza operações lógicas e aritméticas, manipulação de dados e comunicação de rede, sendo utilizado no controle de sistemas automatizados (GEORGINO, 2006).

De acordo com Martins (2007 apud PEDRUZI, 2014), antes do surgimento dos CLP's, os relés eletromagnéticos faziam as tarefas de comando e controle de máquinas e processos industriais. Estes relés eram especialmente projetados para este fim. Além disso, com o advento dos CLP's, os custos com fiação foram diminuídos expressivamente, ao invés de usar centenas

de metros de fios com os painéis, um único par de fios trançados podem ser usados para transmitir os sinais dos subsistemas (CORRADI, 2000 apud MOREIRA, 2009).

Gerson O. Luz Pedruzi destaca as vantagens da aplicação dos CLPs em relação aos painéis a relé utilizados antigamente:

[...] a implementação do CLP diminuiu o espaço físico necessário para a automatização dos processos, entretanto, vale destacar a economia de tempo necessária para a realização de manutenções e melhorias no processo, o menor gasto de potência, o fato de ser reprogramável, a maior confiabilidade, a maior flexibilidade e, principalmente, a permissão de interface através de rede de comunicação com outros CLP's e microcomputadores. (PEDRUZI, 2014)

Conforme os processos se tornam mais complicados torna-se inviável o monitoramento através do programa do CLP, por isso as interfaces homem-máquina (IHM) são propostas, uma vez que torna possível reunir todas as informações necessárias para o controle e monitoramento de um processo de forma amigável.

Uma outra característica do CLP é que ele pode se comunicar com outro CLP e/ou computador, via interface serial, Modbus e *Ethernet*. Esta comunicação é necessária quando se pretende programar o CLP, e também como no caso deste estudo para alimentar uma IHM com os dados atuais do processo.

1.4 Sistemas de Supervisão e Controle

Considerado uma unidade de controle centralizado na linha de produção, o sistema supervisório pode ser avaliado como o nível mais alto de interface homem máquina, pois mostra o que está acontecendo no processo e permitem a atuação sobre este (MARTINS, 2007 apud PEDRUZI, 2014). Estes sistemas podem ser equipados com histórico de eventos, banco de dados, visualização de câmeras, controladores de eventos, dentre outros, para que o operador possa acessá-lo a qualquer instante, com o propósito que for.

Estes sistemas de Controle Supervisório e Aquisição de Dados são, de acordo com Filho (2011 apud RAYSARO, 2012), sistemas de supervisão de processos industriais que coletam dados através de CLP's, unidades terminais remotas ou outros aparelhos de aquisição de dados, e os exibe ao usuário em vários tipos de formas, fornecendo os mais diversos dados do procedimento monitorado. Pode ser visto na Figura 3 um exemplo deste tipo de sistema, para controle e supervisão de silos da empresa FOSPAR.

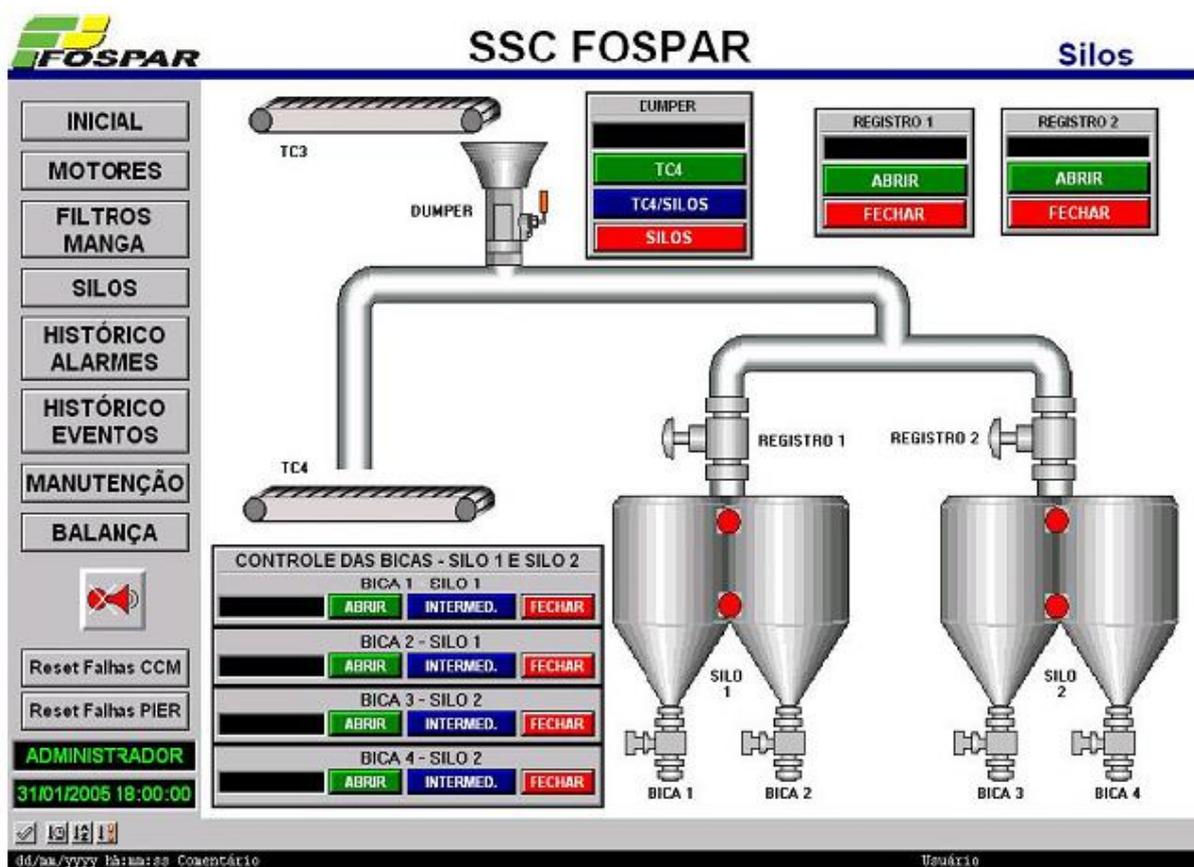


Figura 3 - Exemplo de um sistema supervisório. Controle e manutenção de silos.

Os sistemas SCADA identificam os *tags*, os quais são todas as variáveis numéricas ou alfanuméricas envolvidas na aplicação, podendo executar funções computacionais ou representar pontos de entrada/saída de dados do processo que está sendo controlado, se comportando como a ligação entre o controlador e o sistema. Com base nas *tags* é que os dados coletados são apresentados ao usuário (RADDATZ, 2006 apud BENTHIEN, 2007).

Pires, Oliveira e Barros (2005) definem a importância do SCADA como:

Os sistemas SCADA são de importância estratégica já que são adotados na maioria das indústrias que compõem a infraestrutura de um país. As aplicações da tecnologia SCADA alcançam praticamente todo o espectro do setor produtivo. Para exemplificar, esses sistemas são utilizados na indústria química, petroquímica e de cimentos; na indústria alimentícia; na produção e distribuição de energia elétrica; na distribuição de água; no controle de oleodutos, gasodutos, centrais nucleares, edifícios inteligentes e tráfego.

Na tentativa de se quantificar a importância deste sistema para todo o ambiente industrial atual, pode-se dizer que praticamente todas as indústrias fazem uso desta ferramenta, onde sua importância é afirmada por todos os profissionais da área de automação industrial.

Como é desejável o desenvolvimento de um sistema SCADA de qualidade, é importante seguir algumas etapas de planejamento conforme mostra a Tabela 2 (RESENDE, 2014).

Tabela 2 - Etapas de planejamento de um sistema supervisório.

Etapa	Função
Entendimento do processo a ser automatizado	Reunir o máximo possível de informações sobre o processo, tendo orientação dos operadores ou com especialistas de planejamento.
Tomada de dados	Determinar as variáveis essenciais do processo, para que não sobrecarregue o sistema computacional com excesso de informações.
Determinação das <i>Tags</i>	Determinar a quantidade de <i>tags</i> a serem utilizadas e também o seu tipo.
Planejamento do banco de dados	Designar as variáveis do sistema supervisório, desenvolver um sistema de nome para as variáveis e utilizar pastas de arquivos para organizá-las.
Planejamento dos alarmes	Definir as condições de acionamento dos alarmes, a forma na qual vão ser apresentadas ao operador, esses alarmes tem como principais funções as de atentar o operador para mudanças no processo, sinalizar alguma situação e dar ideia da condição geral do sistema.
Planejamento da hierarquia de navegação entre telas	Consiste na navegação entre as telas que fornecem progressivamente detalhes das plantas à medida que se navega no aplicativo.
Desenho das telas	Determinar símbolos e cores, nome dos botões, posição dos botões na tela, sendo necessário para facilitar o entendimento do processo.
Gráficos de tendência	Apresentar como as variáveis mudam ao longo do tempo.
Acesso e segurança	Necessário para restringir o acesso de pessoas sem autorização ao sistema.

O Eclipse SCADA é um programa para desenvolvimento de interface homem-máquina voltado para supervisão e controle de processos bastante conhecido no mercado, como o nome indica, é baseado em sistemas SCADA.

Por ter uma linguagem de fácil manuseio e por possuir drivers para diversos tipos de protocolos de comunicação, o Eclipse SCADA é um *software* para supervisórios bastante popular. Por utilizar o protocolo Modbus, faz possível a conexão aos inversores de frequência da linha CFW-08 e aos CLP's Clic02 20VT-D, sendo possível o monitoramento em tempo real de variáveis através de objetos e gráficos que estão relacionados com as variáveis físicas do campo.

Para cada *tag* criada pode-se estabelecer condições de trabalho, assim quando o valor da *tag* ultrapassa a condição pré-estabelecida um alarme pode ser ativado.

A Eclipse *Software*, desenvolvedora do Eclipse SCADA, fornece uma versão gratuita de demonstração de seu programa que, por ser uma versão demo, possui um limite de vinte *tags*. Como este trabalho se trata apenas de um protótipo, estas são suficientes para o desenvolvimento do supervisório. A interface do Eclipse SCADA pode ser vista na Figura 4.

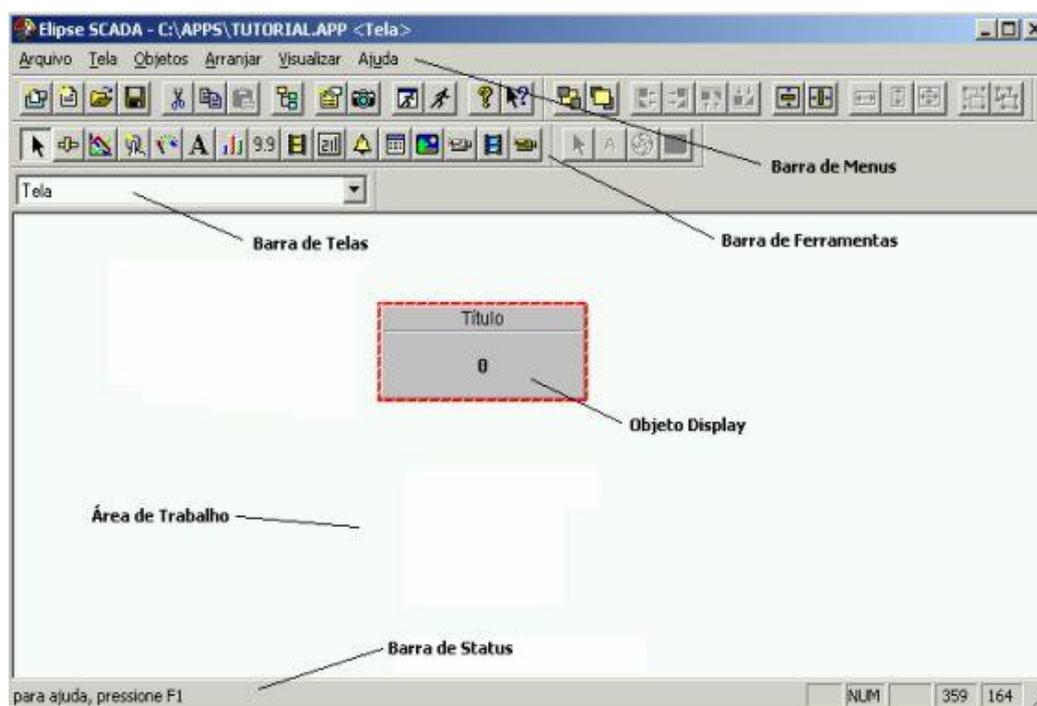


Figura 4 - Tela de Trabalho do Elipse SCADA.

1.5 Objetivos

Este trabalho tem como objetivo geral a implementação de um sistema para comunicação serial para troca de mensagens entre inversor de frequência CFW-08, da WEG e um sistema supervisor. Assim, deve-se criar um sistema que ao aplicar ao conjunto inversor, CLP e motor, realize o controle do processo e faça a parametrização do inversor de frequência.

2 *Materiais e Métodos*

2.1 *Protocolo Modbus RTU*

O protocolo Modbus foi desenvolvido pela Modicon *Industrial Automation Systems*, hoje denominada Schneider, no ano de 1979 e tinha o objetivo de estabelecer uma comunicação entre seus controladores lógicos programáveis. Foi publicado abertamente, de modo que qualquer um pudesse utilizá-lo sem pagar *royalties*, sendo hoje um dos protocolos mais utilizados no ambiente industrial (GUARESE, 2011).

Este protocolo é baseado em uma estrutura de mensagens compostas por *bytes*, reconhecida pelos mais diferentes tipos de dispositivos. Apesar de ser utilizado normalmente sobre conexões seriais, ele também pode ser usado como um protocolo da camada de aplicação de redes industriais tais como TCP/IP sobre *Ethernet* e MAP (NASCIMENTO; LUCENA, 2003).

O Modbus é fundamentado no modelo de comunicação mestre-escravo ou cliente/servidor, onde o dispositivo que pode inicializar a comunicação deve ser somente o único mestre da rede, os demais dispositivos escravos respondem enviando os dados que lhe foram solicitados, ou realizam alguma ação ordenada. Um escravo é todo dispositivo periférico (transdutor de E/S, válvula, inversor, etc.) que processa as informações e responde ao mestre. Já os mestres são dispositivos de comando, como computadores e CLP's. O dispositivo mestre pode endereçar cada dispositivo escravo da rede individualmente ou acessar a todos da rede através de mensagens em *broadcast*. Veja os exemplos de comunicação mestre-escravo na Figura 5.

O modo de transmissão determina como os bits que são transmitidos ao longo da rede são codificados, além de como os dados precisam ser empacotados e decodificados. O modo ASCII e o RTU são os dois modelos de transmissão existentes. Eles são, usualmente, escolhidos juntamente com outros parâmetros de comunicação (tais como taxa de transmissão, paridade, stop bits, etc.) como parte da configuração do dispositivo (ARAÚJO JÚNIOR, 2006).

Os dispositivos utilizados neste trabalho, CLP Clic02, Inversor de Frequência CFW-08 e o Elipse SCADA fazem uso do Modbus RTU (*Remote Termina User*), e segundo Villa (2014), constroem suas mensagens de maneira que cada byte de 8-bits transporta dois caracteres de 4-

bits em hexadecimal. Assim há uma maior densidade de informação, o que o faz capaz de transmitir mais informações que o modo ASCII utilizando a mesma taxa de transmissão.

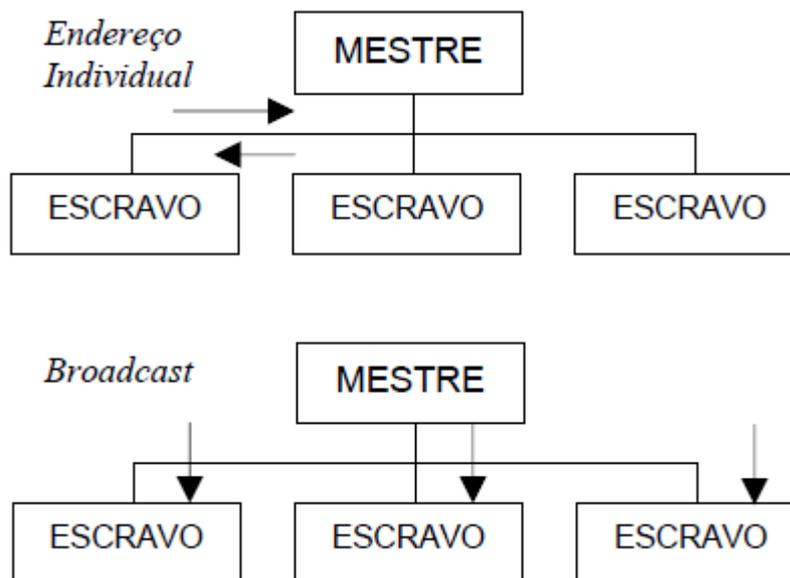


Figura 5 - Modelo Mestre-Escravo.

No modo RTU, cada palavra transmitida possui 1 start bit, oito bits de dados, 1 bit de paridade (opcional) e 1 stop bit (2 stop bits caso não se use bit de paridade). Desta forma, a sequência de bits para transmissão de um byte pode ser vista na Figura 6.

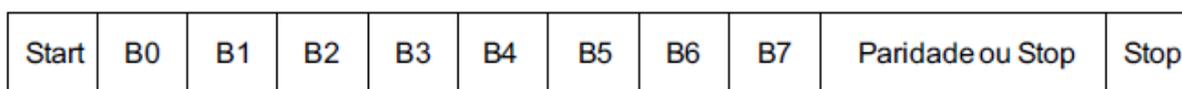


Figura 6 - Transmissão de dados modo RTU.

A estrutura de mensagem utilizada é da seguinte forma: Endereço, Código da Função, Dados e CRC. Somente o campo de dados poderá ter dimensão variável, dependendo do que está sendo requerido (WEG, 2009). Esta estrutura está ilustrada na Figura 7.

O campo de endereço é o campo com o endereço do dispositivo escravo, o campo código da função contém o tipo de serviço ou função solicitada ao escravo (leitura, escrita, etc.). De acordo com o protocolo, cada função é utilizada para acessar um tipo específico de dado. O campo de dados contém os dados a serem escritos ou lidos pelo escravo, dependendo da função. CRC é o campo para checagem de erros de transmissão.

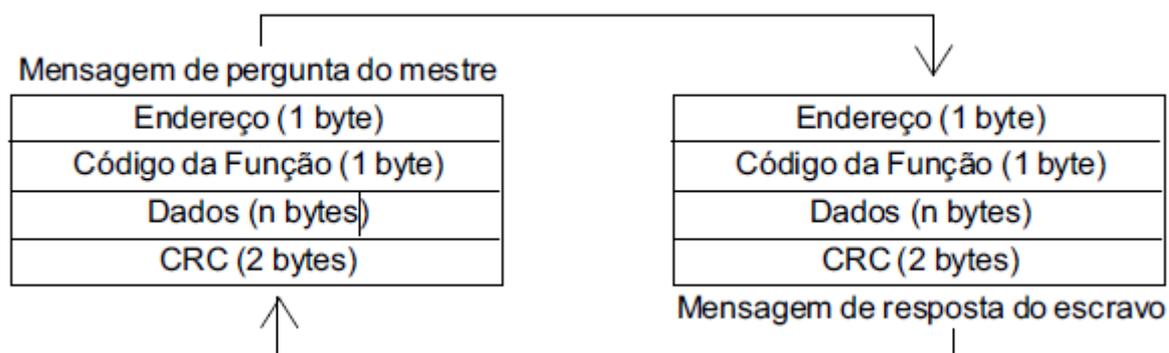


Figura 7 - Estrutura das mensagens.

2.2 CLP WEG Clic-02 20VT-D

A Figura 8 mostra um modelo de CLP da linha Clic02 da WEG, que é um controlador que não possui a robustez de um CLP industrial, mas é muito empregado em automações de pequeno e médio porte, pois é capaz de exercer uma infinidade de aplicações em ambientes que não sejam muito agressivos. O Clic02 20VT-D é simples, fácil de se configurar e ainda possui uma grande quantidade de funções capazes de fazer tarefas como temporização, contagem, PWM, PID, entre outras (VILLA, 2014).

A programação deste controlador pode ser feita por meio de duas linguagens: lógica ladder ou diagrama de blocos. Esta programação pode ser feita tanto no *display* do CLP quanto através do programa CLIC02 Edit via cabos serial RS-232 utilizando um computador. Neste trabalho será utilizado um Clic02 20VT-D, pois possui comunicação serial através do protocolo Modbus, o que possibilita a comunicação com outros dispositivos. No manual do usuário do Clic02, disponível *online* no site da WEG, é possível encontrar essas e outras informações.



Figura 8 - CLP da linha Clic02.

O Clic02 pode ser controlado pelo computador ou outro CLP que possua a função de mestre Modbus. O mestre Modbus pode ter acesso aos estados de Entrada/Saída, valores pré-

ajustados dos Blocos de Função, temporizadores, contadores. Também é possível controlar os modos *Run/Stop* do CLIC-02.

Villa (2014) sustenta que o CLP Clic02 20VT-D só opera na rede Modbus RTU se for configurado corretamente de acordo com os itens destacados na Figura 9. Esta tela pode ser acessada através do programa Clic02 Edit.

A ID do CLP pode ser ajustada de 1~99, mas nunca dois ou mais escravos podem ter o mesmo endereço, neste trabalho a ID do controlador lógico foi definida para 1. Além disso é necessário que a taxa de transmissão (*baud rate*) seja 38400 bps. O modo de operação deve ser igual ao da Figura 9, 8/N/2 o que significa que são utilizados pacotes de 8 bits, sem paridade e com 2 bits de parada.

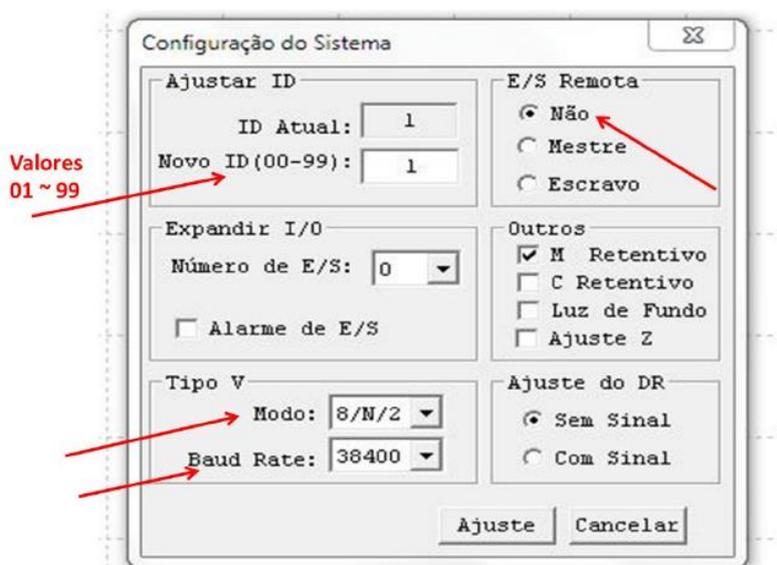


Figura 9 - Configurações do Clic02.

Para acessar os estados dos registradores deve-se saber o endereço destes registradores na memória do CLP, o manual do Clic02 disponibiliza um mapa de memória Modbus como mostrado na Tabela 3.

2.3 Inversor de Frequência CFW-08

Para fazer a comunicação serial, o inversor de frequência WEG CFW-08 precisará do módulo de comunicação serial, neste caso será utilizado o KRS-485-CFW08. Este módulo é inserido na parte frontal do inversor no local da IHM paralela, disponibilizando a conexão RS-485. O controle do inversor de frequência será feito acessando diretamente seus registradores. A Figura 10 ilustra o inversor de frequência com o módulo de comunicação conectado.

Tabela 3 - Endereço das variáveis do CLP.

Endereço Modbus		Função Modbus	Conteúdo das Words																
Decimal	Hexadecimal		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
1536	0600	03H	R10	R0F	R0E	R0D	R0C	R0B	R0A	R09	R08	R07	R06	R05	R04	R03	R02	R01	
1537	0601		R1F	R1E	R1D	R1C	R1B	R1A	R19	R18	R17	R16	R15	R14	R13	R12	R11		
1538	0602		G10	G0F	G0E	G0D	G0C	G0B	G0A	G09	G08	G07	G06	G05	G04	G03	G02	G01	
1539	0603		G1F	G1E	G1D	G1C	G1B	G1A	G19	G18	G17	G16	G15	G14	G13	G12	G11		
1540	0604		T10	T0F	T0E	T0D	T0C	T0B	T0A	T09	T08	T07	T06	T05	T04	T03	T02	T01	
1541	0605		T1F	T1E	T1D	T1C	T1B	T1A	T19	T18	T17	T16	T15	T14	T13	T12	T11		
1542	0606		C10	C0F	C0E	C0D	C0C	C0B	C0A	C09	C08	C07	C06	C05	C04	C03	C02	C01	
1543	0607		C1F	C1E	C1D	C1C	C1B	C1A	C19	C18	C17	C16	C15	C14	C13	C12	C11		
1544	0608		M10	M0F	M0E	M0D	M0C	M0B	M0A	M09	M08	M07	M06	M05	M04	M03	M02	M01	
1545	0609		M20	M1F	M1E	M1D	M1C	M1B	M1A	M19	M18	M17	M16	M15	M14	M13	M12	M11	
1546	060A		M30	M2F	M2E	M2D	M2C	M2B	M2A	M29	M28	M27	M26	M25	M24	M23	M22	M21	
1547	060B		M3F	M3E	M3D	M3C	M3B	M3A	M39	M38	M37	M36	M35	M34	M33	M32	M31		
1548	060C		N10	N0F	N0E	N0D	N0C	N0B	N0A	N09	N08	N07	N06	N05	N04	N03	N02	N01	
1549	060D		N20	N1F	N1E	N1D	N1C	N1B	N1A	N19	N18	N17	N16	N15	N14	N13	N12	N11	
1550	060E		N30	N2F	N2E	N2D	N2C	N2B	N2A	N29	N28	N27	N26	N25	N24	N23	N22	N21	
1551	060F		N3F	N3E	N3D	N3C	N3B	N3A	N39	N38	N37	N36	N35	N34	N33	N32	N31		
1552	0610	10H	-	-	-	-	I0C	I0B	I0A	I09	I08	I07	I06	I05	I04	I03	I02	I01	
1553	0611		-	-	-	-	X0C	X0B	X0A	X09	X08	X07	X06	X05	X04	X03	X02	X01	
1554	0612		-	-	-	-	Y0C	Y0B	Y0A	Y09	Y08	Y07	Y06	Y05	Y04	Y03	Y02	Y01	
1555	0613		-	-	-	-	-	-	-	-	Q08	Q07	Q06	Q05	Q04	Q03	Q02	Q01	
1556	0614		-	-	-	-	-	-	-	-	-	-	-	-	Z04	Z03	Z02	Z01	
1557	0615		H10	H0F	H0E	H0D	H0C	H0B	H0A	H09	H08	H07	H06	H05	H04	H03	H02	H01	
1558	0616		-	H1F	H1E	H1D	H1C	H1B	H1A	H19	H18	H17	H16	H15	H14	H13	H12	H11	
1559	0617		-	-	-	-	-	-	-	-	L08	L07	L06	L05	L04	L03	L02	L01	
1560	0618		-	-	-	-	-	-	-	-	-	-	-	-	-	S01	P02	P01	
1561	0619		W10	W0F	W0E	W0D	W0C	W0B	W0A	W09	W08	W07	W06	W05	W04	W03	W02	W01	
1562	061A		W20	W1F	W1E	W1D	W1C	W1B	W1A	W19	W18	W17	W16	W15	W14	W13	W12	W11	
1563	061B		W30	W2F	W2E	W2D	W2C	W2B	W2A	W29	W28	W27	W26	W25	W24	W23	W22	W21	
1564	061C		W40	W3F	W3E	W3D	W3C	W3B	W3A	W39	W38	W37	W36	W35	W34	W33	W32	W31	

Basicamente existem dois tipos de controle dos inversores eletrônicos: o escalar e o vetorial.

Pires (2006) enuncia as diferenças entre os dois tipos de controle:

As principais diferenças entre os dois tipos de controle são que o controle escalar só considera as amplitudes das grandezas elétricas instantâneas (fluxos, correntes e tensões), referindo-as ao estator, e seu equacionamento baseia-se no circuito equivalente do motor, ou seja, são equações de regime permanente. Já o controle vetorial admite a representação das grandezas elétricas instantâneas por vetores, baseando-se nas equações espaciais dinâmicas da máquina, com as grandezas referidas ao fluxo enlaçado pelo rotor, ou seja, o motor de indução é visto pelo controle vetorial como um motor de corrente contínua, havendo regulação independente para torque e fluxo.

2.3.1 Aplicação do Protocolo Modbus ao Inversor CFW-08

Na especificação do protocolo Modbus RTU são definidas as funções utilizadas para acessar os tipos de registradores descritos na especificação. No CFW-08, tanto parâmetros quanto variáveis básicas foram definidos como sendo registradores do tipo *holding*. Além destes registradores, também é possível acessar diretamente bits internos de comando e

monitoração (WEG, 2009). Para acessar estes bits e registradores, foram disponibilizados os serviços (ou funções) mostrados na Tabela 4.



Figura 10 - Inversor CFW-08 com módulo KRS-485.

Tabela 4 - Funções Modbus disponíveis no CFW-08.

Função	Descrição	Código da função	Tempo de resposta
Read Coils	Leitura de bloco de bits internos ou bobinas.	01	10 a 20ms
Read Holding Registers	Leitura de bloco de registradores do tipo holding.	03	10 a 20ms
Write Single Coil	Escrita em um único bit interno ou bobina.	05	10 a 20ms
Write Single Register	Escrita em um único registrador do tipo holding.	06	10 a 50ms
Write Multiple Coils	Escrita em bloco de bits internos ou bobinas.	15	10 a 20ms
Write Multiple Registers	Escrita em bloco de registradores do tipo holding.	16	10 a 50ms
Read Device Identification	Identificação do modelo do inversor.	43	10 a 20ms

De acordo com WEG (2009), O endereçamento dos dados no CFW-08 é feito com offset igual a zero, o que significa que o número do endereço equivale ao número dado. Os parâmetros são disponibilizados a partir do endereço 0 (zero), enquanto que as variáveis básicas são disponibilizadas a partir do endereço 5000. Da mesma forma, os bits de estado são disponibilizados a partir do endereço 0 (zero) e os bits de comando são disponibilizados a partir do endereço 100.

As tabelas a seguir ilustram o que foi dito por WEG, a Tabela 5 contém o endereçamento dos parâmetros, na Tabela 6 pode ser visto o endereço Modbus das variáveis básicas e na Tabela 7 o endereçamento dos bits.

Tabela 5 - Endereçamento dos Parâmetros.

Parâmetros		
Número do Parâmetro	Endereço Modbus	
	Decimal	Hexadecimal
P000	0	0000h
P001	1	0001h
⋮	⋮	⋮
P100	100	0064h
⋮	⋮	⋮

Tabela 6 - Endereçamento das variáveis básicas.

Variáveis Básicas		
Número da Variável Básica	Endereço Modbus	
	Decimal	Hexadecimal
V00	5000	1388h
V01	5001	1389h
⋮	⋮	⋮
V05	5005	138Dh

Tabela 7 - Endereçamento dos bits de estado e comando.

Bits de Estado		
Número do Bit	Endereço Modbus	
	Decimal	Hexadecimal
Bit 0	00	00h
Bit 1	01	01h
⋮	⋮	⋮
Bit 7	07	07h
Bits de Comando		
Número do Bit	Endereço Modbus	
	Decimal	Hexadecimal
Bit 100	100	64h
Bit 101	101	65h
⋮	⋮	⋮
Bit 107	107	6Bh

2.3.2 Parametrização do Inversor para Funcionamento na Rede Modbus

Como já explicado, para funcionar corretamente na rede Modbus RTU o inversor deve ser parametrizado. Inicialmente sua parametrização deve ser feita manualmente, a fim de habilitar o controle via interface serial. Para isso, o parâmetro P000 (Parâmetro de Acesso) deve ser colocado em 5 para que seja possível alterar os outros. Em seguida, os primeiros parâmetros que devem ser alterados são, P312 e P308, nesta ordem, porque o parâmetro P308 (Endereço Serial) pode assumir valores diferentes dependendo do protocolo escolhido no parâmetro P312 (Protocolo da Interface Serial).

A Tabela 8 mostra como foram ajustados estes dois parâmetros. O endereço serial deve ser diferente de 1, já que 1 foi escolhido para o CLP. Para os dispositivos funcionarem

corretamente na rede Modbus a taxa de transmissão e a paridade devem ser as mesmas, assim como o Clic02 foi configurado com *baud rate* igual a 38400 bps e sem paridade, o parâmetro P312 também deverá ser ajustado desta forma. Com estes parâmetros ajustados, o inversor e o CLP estão prontos para operar na mesma rede Modbus.

Tabela 8 - Ajuste dos parâmetros P308 e P312.

Parâmetro	Função	Ajuste
P308	Endereço Serial	2
P312	Protocolo da Interface Serial	7 = Modbus RTU, 38400 bps sem paridade

Feito isso, ainda não é possível comandar o inversor através da interface serial, ou seja, não dá para habilitar o inversor, mudar o sentido de rotação ou indicar a referência de velocidade remotamente, mas já é possível acessar e modificar parâmetros via serial. Para que os comandos possam ser feitos de forma remota deve-se ajustar os parâmetros a seguir de acordo com a Tabela 9.

Tabela 9 - Ajuste dos Parâmetros para Controle do Inversor.

Parâmetro	Função	Ajuste
P220	Seleção da Fonte Local/Remoto	1 = Sempre Remoto
P222	Seleção da Referência de Velocidade – Situação Remoto	5 = Serial
P230	Seleção de Comandos	2 = Serial ou Teclas HMI-CFW08-RS
P231	Seleção do Sentido de Giro – Situação Local e Remoto	2 = Comandos

O parâmetro P220 foi ajustado para sempre remoto, uma vez que se deseja controlar remotamente o inversor e não era de interesse o controle local. Para a referência de velocidade na situação remoto ser dada pela interface serial, deve-se colocar o parâmetro em 5. A fim de comandar o inversor via serial e escolher o sentido de giro remotamente, deve-se setar os parâmetros P230 e P231 em 2.

3 *Desenvolvimento do Sistema de Automação*

3.1 *Seleção do Tipo de Acionamento*

O protótipo desenvolvido faz o acionamento de um motor de indução de duas maneiras:

- **Partida estrela-triângulo:** Através desta manobra o motor realizará uma partida mais suave, reduzindo sua corrente de partida a aproximadamente $1/3$ da que seria se acionado em partida direta. Ao escolher partida estrela-triângulo energiza-se alguns contatores, fazendo com que o motor conecte-se à rede trifásica na configuração estrela. Após um tempo especificado (cerca de 5s), o contator que liga o motor em estrela é desenergizado e outro contator é ativado, acionando o motor na configuração triângulo, a qual deve permanecer até que se desligue o sistema ou mude o tipo de partida.
- **Partida via inversor de frequência:** Por meio deste acionamento é possível a regulação de velocidade e torque de um motor, além de aumentar a vida útil do sistema, pois evita-se acionamentos bruscos e protege-se o motor de falhas na rede elétrica. Caso faça-se a opção de partida via inversor os contatores da partida estrela-triângulo são desativados e outros dois contatores passam a estar energizados, isto conecta o motor ao inversor de frequência na configuração triângulo.

Estes acionamentos podem ser escolhidos através do CLP que foi controlado por um sistema supervisor, onde também foi possível fazer o monitoramento e controle do motor, além de parametrizar o inversor de maneira remota. O protótipo criado pode ser visto na Figura 11 e o esquema de configuração dos contatores pode ser observado na Figura 12.

O princípio de funcionamento do sistema consiste em, inicialmente o operador do sistema possui a opção de eleger dois tipos de acionamento, ao escolher partida estrela-triângulo energiza-se os contatores C1 e C3. Isto faz com que o motor conecte-se à rede trifásica na configuração estrela. Após um tempo especificado (cerca de 5s), o contator C3 é desenergizado e o contator C2 é ativado, acionando o motor na configuração triângulo, a qual deve permanecer até que se desligue o sistema ou mude o tipo de partida.

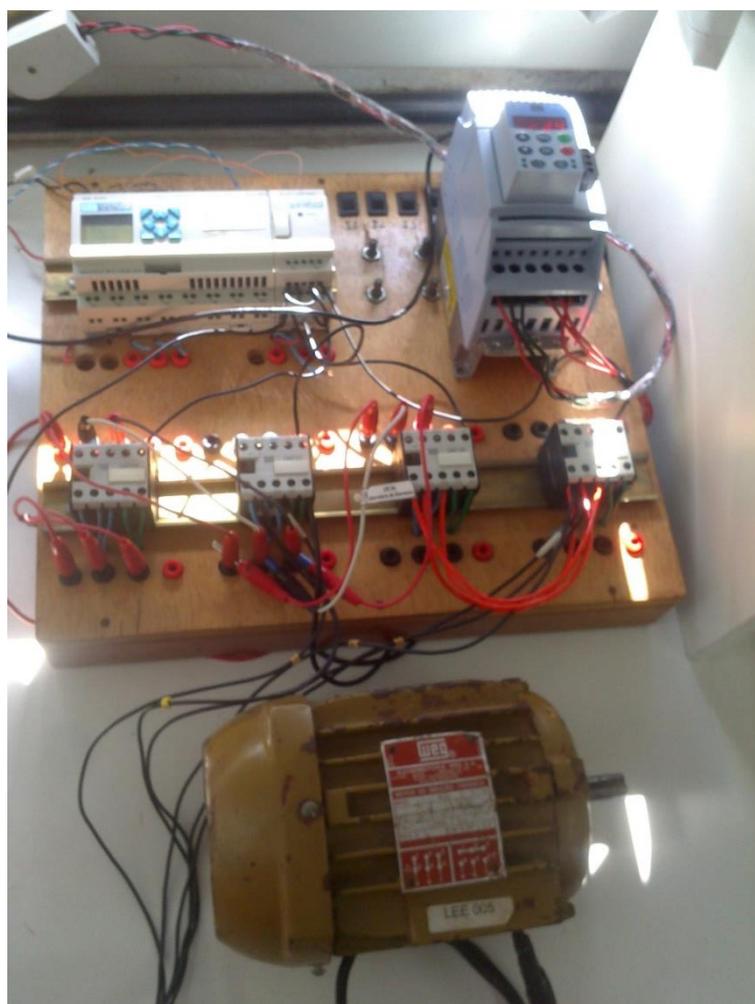


Figura 11 - Sistema de automação desenvolvido.

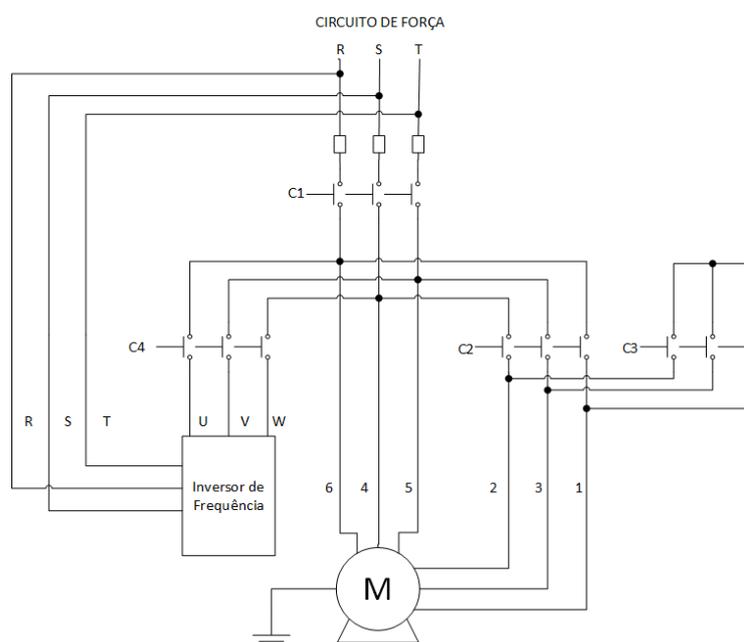


Figura 12 - Esquema de conexão dos contadores para partida estrela-triângulo ou via CLP.

3.1.1 Acionamento via CLP

Para fazer o acionamento via CLP é necessário o uso de contatores, a fim de separar o circuito de potência do circuito de controle, foram utilizados quatro contatores WEG CW07, mostrado na Figura 13, e para fazer o acionamento destes contatores é necessário um módulo de expansão com saídas digitais a relé, por estes serem mais resistentes às altas correntes. O módulo utilizado é do modelo CLW-02/8ER-D, Figura 14.



Figura 13 - Contator WEG CW07.



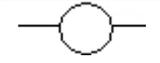
Figura 14 - Módulo de expansão CLW-02/8ER-D.

As saídas à relé Y01~Y04 do módulo de expansão foram conectadas cada uma à um contator de forma que a saída Y01 foi conectada ao contator C1, Y02 ao contator C3, Y03 ao contator C2 e Y04 ao contator C4. Como entradas foram utilizadas as variáveis auxiliares de memória M01~M03, que substituíram as chaves físicas, já que todo acionamento será feito através da escrita na memória do CLP via sistema supervisor.

Para fazer a transição de estrela para triângulo é necessário que se espere um tempo, cerca de 5s, depois desse tempo o contator C3 que liga o motor em estrela é desenergizado e por motivos de segurança há, ainda, um tempo de espera de 0,5s, para que o contator C2 (configuração triângulo) seja acionado. Para isso utiliza-se dois temporizadores de retardo na ligação (TON), T01 programado para ligar após 5s e T02 programado para 0,5s.

O diagrama Ladder criado pode ser visto na Figura 15 e a simbologia utilizada neste diagrama está exposta na Tabela 10.

Tabela 10 - Simbologia utilizada no diagrama ladder.

	Contato normalmente aberto, envia sinal somente quando é acionado.
	Contato normalmente fechado, envia sinal durante todo tempo em que não é acionado. Quando acionado deixará de enviar o sinal.
	Bobina <i>set</i> , liga uma saída quando é excitada com um sinal.
	Bobina <i>reset</i> , desliga uma saída quando é excita com um sinal.
	Bobina vazia, a saída só se mantém ativada, se um sinal for mantido constantemente para mantê-la ligada.

3.1.2 Acionamento via Inversor de Frequência

A utilização de inversores no acionamento de motores elétricos é utilizada por ser a maneira mais viável de fazer o controle de velocidade e torque destes motores, pois possuem uma série de vantagens, como a redução do consumo de energia elétrica por evitar acionamentos bruscos utilizando uma baixa corrente de partida. Outra vantagem é em relação ao aumento da vida útil dos motores, pois estes possuem diversas funções de proteção, como, por exemplo, medidas contra surtos na rede elétrica. Além disso podem contribuir para a automatização de processos, já que alguns deles possibilitam a programação de eventos automáticos.

O acionamento via inversor de frequência pode ser feito via IHM, o operador faz a escolha e o CLP envia um sinal para os contadores C1 e C3 serem desativados e para os contadores C2 e C4 serem energizados, isto conecta o motor ao inversor de frequência na configuração triângulo, assim é possível monitorar e controlar o motor.

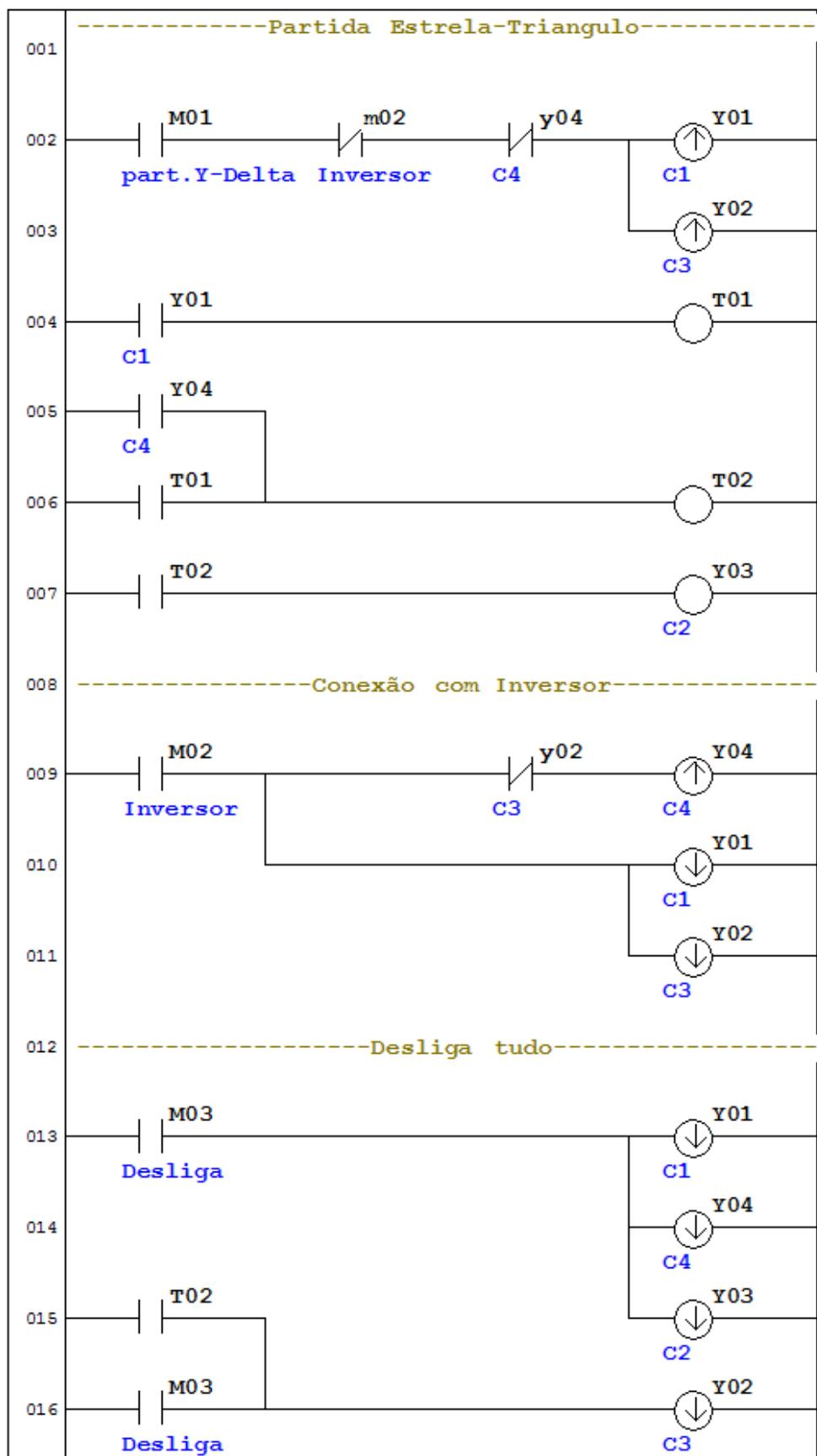


Figura 15 - Diagrama ladder para o sistema de escolha do tipo de partida.

3.1.3 Parametrização do Inversor para Funcionamento do Controle Vetorial

A vantagem do modo de controle vetorial é que a operação do motor é otimizada, pois se obtém um desempenho mais favorável em termos de torque e regulação de velocidade. A regulação da velocidade é na ordem de 0,5% da rotação nominal. Por exemplo, para um motor de quatro polos e 60Hz, obtém-se uma variação da velocidade na faixa de 10rpm.

Para colocar o inversor no modo de controle vetorial deve-se primeiro mudar manualmente, via IHM do CFW-08, o parâmetro P202 (Tipo de Controle) para controle vetorial, ao fazer isso, a IHM do inversor leva o usuário aos parâmetros do motor que devem ser ajustados de acordo com os dados da placa do motor.

A parametrização dos dados do motor de indução trifásico deste trabalho se encontra na Tabela 11. Vale destacar que esses parâmetros só podem ser alterados com o inversor desabilitado (motor parado).

Tabela 11 - Parâmetros do motor.

Parâmetro	Função	Ajuste
P399	Rendimento Nominal do Motor	77,9 %
P400	Tensão Nominal do Motor	220 V
P401	Corrente Nominal do Motor	2,0 A
P402	Velocidade Nominal do Motor	1685 rpm
P403	Frequência Nominal do Motor	60 Hz
P404	Potência Nominal do Motor	4 = 0,75 CV
P407	Fator de potência Nominal do Motor	0,8

3.2 Comunicação Entre os Dispositivos

Como a escolha do tipo de partida feita no motor foi via supervisor, primeiramente foi feita a conexão física entre os dispositivos através de uma rede com padrão RS-485, usando um par de fios trançados, como mostrado na Figura 16. Esta conexão deve ser feita para possibilitar o acionamento e a parametrização via supervisor. O computador, por não possuir porta de comunicação serial, precisa de um conversor RS-485 para USB, utilizou-se para isso o conversor STM-8522 mostrado na Figura 17.

O conversor, assim como o CLP, o inversor de frequência e Elipse SCADA, também foram configurados para atuar na mesma rede Modbus. Para isso, deve-se acessar o gerenciador de dispositivos do Windows e configurar a taxa de comunicação para 38400 bps, 8 bits de dados, nenhuma paridade e 2 *stop* bits (vide Apêndice A).

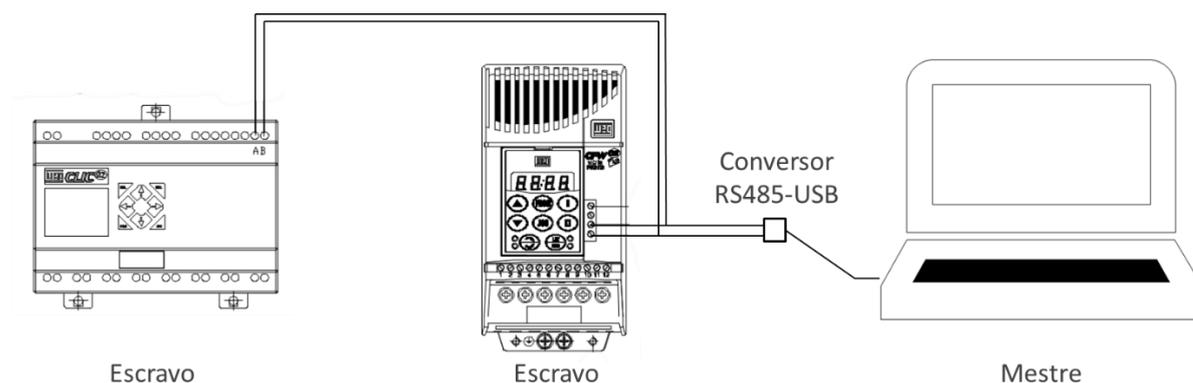


Figura 16 - Conexão entre os dispositivos da rede Modbus.

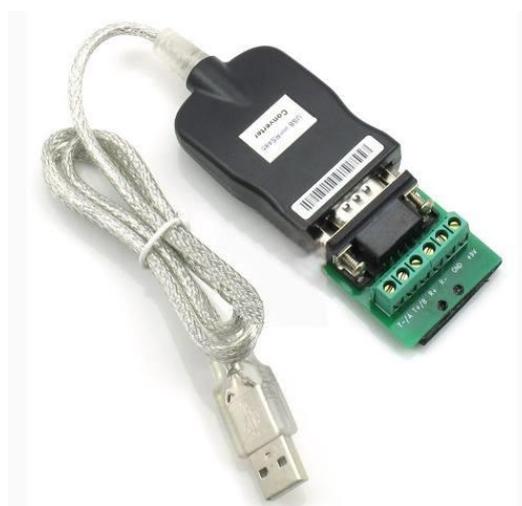


Figura 17 - Conversor RS485-USB STM-8522.

Para configurar o Elixir SCADA para trabalhar como mestre da rede Modbus, basta seguir os passos mostrados, também, no Apêndice A.

3.4 Desenvolvimento do Sistema Supervisório

3.4.1 Etapas do Desenvolvimento do Supervisório

O sistema supervisório deve ser capaz de parametrizar um inversor de frequência e controlar os acionamentos de contadores conectados a um CLP, que, dependendo da sequência, farão a partida em estrela-triângulo de um motor de indução ou conectarão este ao inversor.

A fim de desenvolver o sistema supervisório é preciso saber quais são as variáveis a serem controladas e monitoradas, por isso a escolha das variáveis e parâmetros mais importantes deve ser feita.

É necessário escrever nas variáveis de entrada do CLP, M01~M03, porque estas servem para o controle do tipo de partida. A leitura das saídas Y01~Y04, também é indispensável, já que se deseja saber qual tipo de partida está sendo realizado. Além disso deve-se controlar o estado do CLP remotamente. A Tabela 12 mostra a quantidade de *tags* utilizadas para estas variáveis.

Os parâmetros e variáveis básicas do inversor de frequência escolhidos para fazer parte do sistema supervisor, e a quantidade de *tags* utilizadas por eles constam na Tabela 13.

Tabela 12 - Variáveis do CLP utilizadas no supervisor.

	Função	Tags
Y01	Modificar estado do contator C1	1
Y02	Modificar estado do contator C3	
Y03	Modificar estado do contator C2	
Y04	Modificar estado do contator C4	
M01	Botão de partida Y- Δ	3
M02	Botão de partida via inversor	
M03	Botão desliga	
Run/Stop	Controlar estado do CLP	1

Tabela 13 – Parâmetros e variáveis básicas do inversor utilizados no supervisor.

Parâmetro ou Variável Básica	Função	Tags
P003	Corrente de Saída (Motor)	1
P005	Frequência de Saída (Motor)	1
P100	Tempo de Aceleração	1
P101	Tempo de Desaceleração	1
P104	Rampa S	1
P220	Seleção da Fonte Local/Remoto	1
P400	Tensão Nominal do Motor	1
P401	Corrente Nominal do Motor	1
P402	Velocidade Nominal do Motor	1
V03	Seleção do comando lógico	1
V04	Referência de velocidade dada pela Serial	1

Utilizou-se, também, *tags* do tipo expressão para que se pudesse mostrar os valores de alguns parâmetros, pois a resposta enviada pela rede Modbus não permite o uso de vírgula ou ponto. Por exemplo, deseja-se saber a corrente do motor, então o mestre faz a solicitação e o escravo retorna o valor 150, mas como a resolução da corrente possui duas casas decimais (Tabela 14), configura-se uma *tag* expressão para que ela faça a conversão deste valor para o real, 1,50 A. Foram usadas três *tags* deste tipo, para os parâmetros P003, P005 e P401.

Tabela 14 - Resoluções utilizadas na comunicação serial

Grandeza	Unidade	Resolução
Frequência	Hz	0,01 Hz/unid.
Corrente	A	0,01 Hz/unid.
Tempo	s	0,1 s/unid

Na etapa do planejamento do banco de dados, designou-se as variáveis do sistema supervisorio nomes, e pastas de arquivos para organizá-las. Agrupando tags que se referem a uma mesma etapa do processo, para que nada se perca durante o projeto do supervisorio. A Figura 18 mostra como foi feita a organização das *tags*.

Para não sobrecarregar o sistema foi definidos diferentes ciclos de *scan* para cada tipo de *tag*, pois existem variáveis que não alteram muito com o passar do tempo, como, por exemplo, o estado das saídas Y01~Y04 onde o acionamento mais rápido tem duração de 0,5s, enquanto outras variáveis mudam constantemente, como no caso da velocidade e da corrente do motor. Os ciclos de *scan* definidos estão mostrados na Tabela 15.

Tabela 15 - Ciclo de *scan* definido para cada variável.

Variável	Scan (ms)
Corrente de Saída (Motor)	125
Frequência de Saída (Motor)	125
Tempo de Aceleração	500
Tempo de Desaceleração	500
Rampa S	500
Seleção da Fonte Local/Remoto	500
Tensão Nominal do Motor	500
Corrente Nominal do Motor	500
Velocidade Nominal do Motor	500
Seleção do comando lógico	125
Referência de velocidade dada pela Serial	125
Saídas Y01~Y04	500
Entradas M01~M03	500
Run/Stop	500

Na etapa de planejamento de alarmes deve-se, inicialmente, definir em quais condições os alarmes serão acionados. O segundo passo é escolher como será feita a notificação dos operadores.

A notificação dos alarmes críticos ou urgentes serão feitas em todas as telas do sistema supervisorio, mas os alarmes normais que não implicam numa situação perigosa somente são mostrados na tela de alarmes.

A Figura 19 mostra as condições para que os alarmes associados a tag velocidade são ativados.

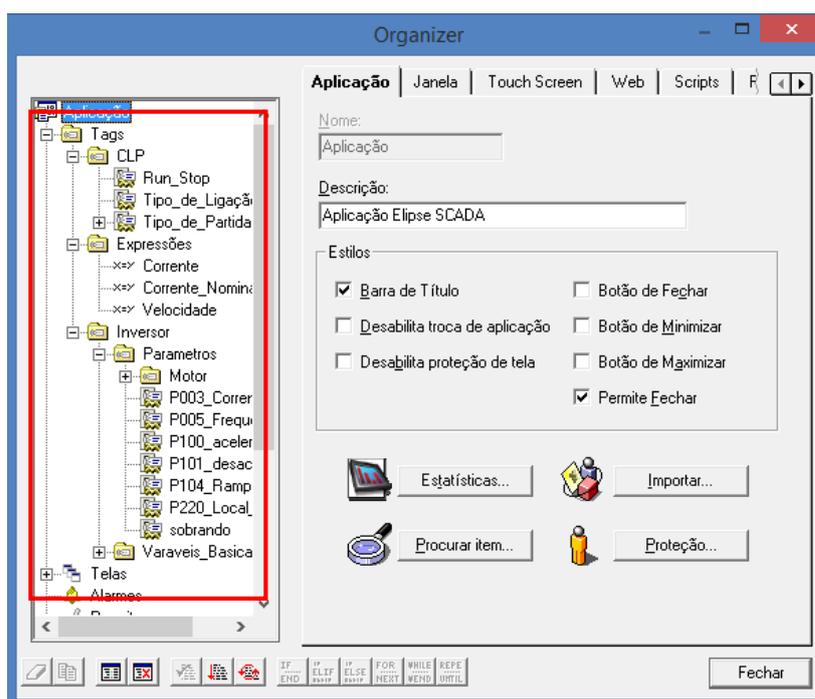


Figura 18 - Organização das variáveis.

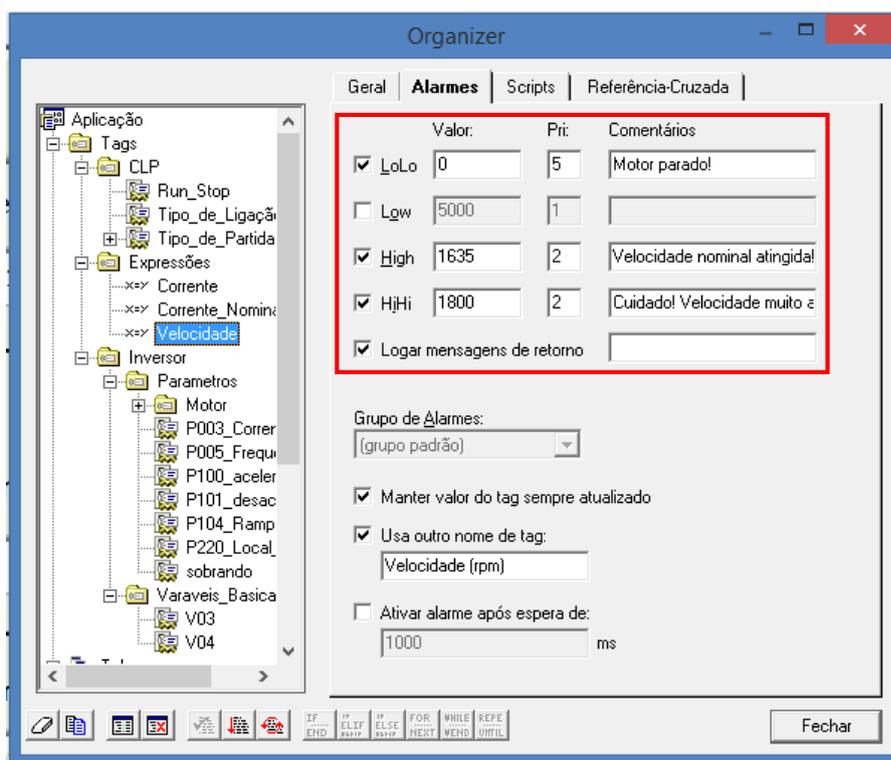


Figura 19 - Condições para ativação de alarmes da tag velocidade.

As telas foram planejadas de forma hierárquica para que o usuário possa navegar progressivamente no sistema, um menu foi criado em todas as telas do projeto para facilitar a navegação entre elas. A Figura 20 ilustra a hierarquia entre as telas do supervisor, em primeiro

plano a tela Principal, seguida da tela do motor, tela de parametrização, alarme e em último plano a tela dos gráficos.

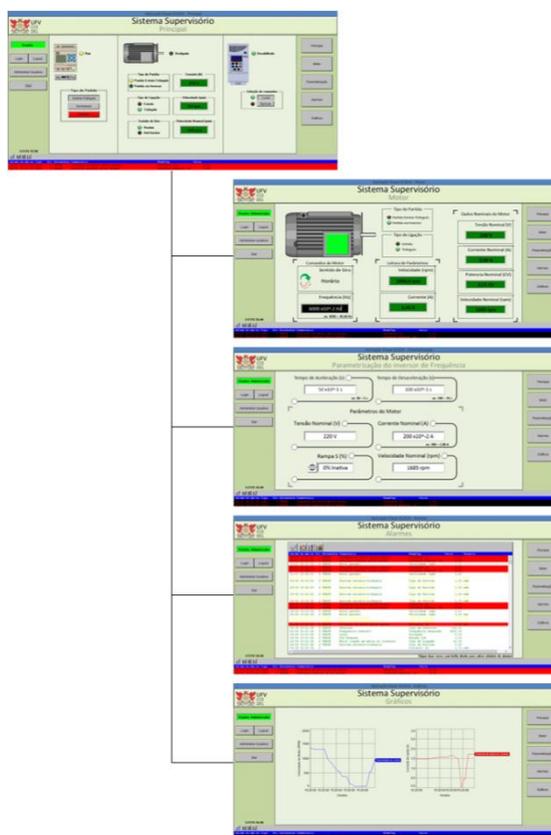


Figura 20 - Hierarquia entre as telas do sistema supervisório

Os *layouts* das telas do supervisório foram desenhados usando símbolos e cores de forma a tornar o processo de supervisão mais agradável ao operador. Além disso nomeou-se os botões para facilitar o entendimento. A seção 3.4 mostra o desenvolvimento das telas do programa.

Os gráficos de tendência são de suma importância em sistemas supervisórios, pois é com eles que o operador pode monitorar a evolução de uma variável do processo. Além disso pode ser usado pela gerência, pois este armazena dados para futuras auditorias.

Foi determinado que as variáveis velocidade e corrente do motor necessitariam de gráficos para que estas possam ser monitoradas. A seção 3.4.8 mostra a criação da tela onde estes gráficos estão localizados.

Deve-se saber, também, que o protocolo Modbus trabalha com pacotes de 8 em 8 bits, mas os endereços de memória utilizam bytes de 16 bits, assim o protocolo faz a leitura de dois pacotes de 8 bits como se fosse um de 16 bits. Com isso, caso seja feita a leitura da saída Q de

um CLP, a resposta será um valor em decimal correspondente a conversão do número binário relacionado ao byte. A variável Q01 está endereçado no ponto menos significativo do byte e Q02 o segundo menos significativo e assim sucessivamente.

3.4.2 Configuração das Tags

Antes de tudo é necessário saber que o inversor de frequência foi endereçado em 2 e o CLP em 1, de forma que as tags que irão ser lidas/escritas no inversor de frequência devem enviar telegramas ao dispositivo 2 da rede Modbus, enquanto as tags que irão ler/escrever no CLP devem ser direcionadas ao dispositivo 1 da rede.

A configuração das *tags* é feita através do *organizer* do Elipse SCADA, sendo as que fazem a comunicação com um dispositivo externo são do tipo *device*, que segundo Pedruzi (2014) “[...] são as variáveis relacionadas às memórias de imagem de entrada e de saída do CLP, além da memória interna.”. Essas *tags* são chamadas pelo Elipse SCADA de “Tag PLC”, mas também podem ser usadas para controlar e parametrizar o inversor de frequência.

Após a instalação e configuração do driver Modbus no Elipse (Apêndice A), é preciso criar as *tags* que farão a comunicação do SCADA com o CLP e o inversor de frequência. Todas as *tags* PLC devem estar associadas ao driver, assim como mostra a Figura 21. O endereçamento destas *tags* pode ser observado na Tabela 16.

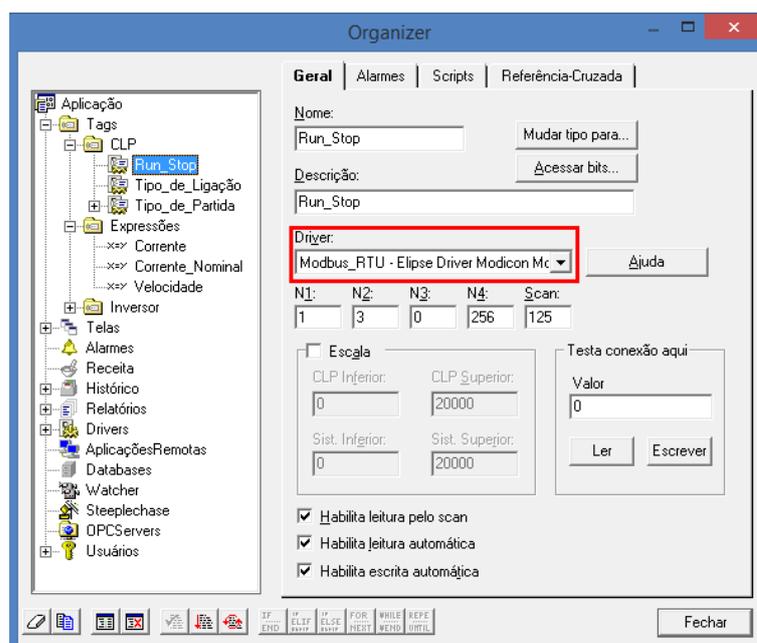


Figura 21 - Associação do driver Modbus a uma tag PLC.

3.4.3 Leitura e Escrita de Parâmetros

A leitura e escrita dos parâmetros deve ser feita obedecendo a resolução da grandeza a ser parametrizada, assim como mostrado na Tabela 14.

O Modbus RTU trabalha em hexadecimal, mas o Elixir SCADA faz a conversão automática dos dados de hexadecimal para decimal quando utilizada a função de leitura e o contrário quando a operação é de escrita, ou seja, os dados de entrada e saída do Elixir estão sempre em decimal.

Tabela 16 - Endereço das *tags* utilizadas.

Tag	Tipo	Endereço	Dispositivo
Run_Stop	PLC	256	CLP
Tipo_de_Ligação	PLC	1554	CLP
Tipo_de_Partida	PLC	1544	CLP
Partida_YD	PLC	1544 (Bit 0)	CLP
Partida_Inversor	PLC	1544 (Bit 1)	CLP
Desacoplar_Motor	PLC	1544 (Bit 2)	CLP
Corrente	Expressão	3	Inversor
Corrente_Nominal	Expressão	401	Inversor
Velocidade	Expressão	5	Inversor
P400_Tensão	PLC	400	Inversor
P401_Corrente	PLC	401	Inversor
P402_Velocidade	PLC	402	Inversor
P003_Corrente	PLC	3	Inversor
P005_Frequencia	PLC	5	Inversor
P100_aceleração	PLC	100	Inversor
P101_desaceleração	PLC	101	Inversor
P104_Rampa_S	PLC	104	Inversor
P220_Local_Remoto	PLC	220	Inversor
V03	PLC	5003	Inversor
V04	PLC	5004	Inversor

Por exemplo, deseja-se fazer com que o tempo de aceleração do motor seja cinco segundos, para isso deve-se configurar uma *tag* PLC da seguinte forma:

- N1 (Endereço do inversor): 2
- N2 (função de leitura e escrita): 3, ler e escrever inteiros de 16 bits com sinal.
- N3 (sem função): 0
- N4 (endereço do parâmetro P100 na memória do inversor): 100, em hexadecimal 0064h
- Scan: 125ms

Associa-se a *tag* PLC a um objeto *setpoint* no Elipse, assim como mostrado na Figura 22. Quando a aplicação estiver rodando basta escrever o número 50, já que a resolução da grandeza tempo é de uma casa decimal.

O telegrama enviado pelo Elipse SCADA para fazer a escrita desta variável é mostrada pela Tabela 17. A função de escrita é a *Write Multiple Register*, número 16 segundo a Tabela 23 (encontrada na seção A.2 do Apêndice A).

Para fazer a leitura do parâmetro P003 (corrente do motor), o Elipse receberá a resposta do escravo, conforme a Tabela 18, o valor em hexadecimal, 0096h, será convertido para decimal pelo Elipse SCADA, 150, como a resolução deste parâmetro é de duas casas decimais, é necessário uma *tag* expressão que faça a conversão para o valor real que é de 1,50 A, que será mostrado pelo *display*, assim como na Figura 23 (b). A Figura 23 (a) mostra a associação da *tag* do parâmetro P003 à *tag* expressão.

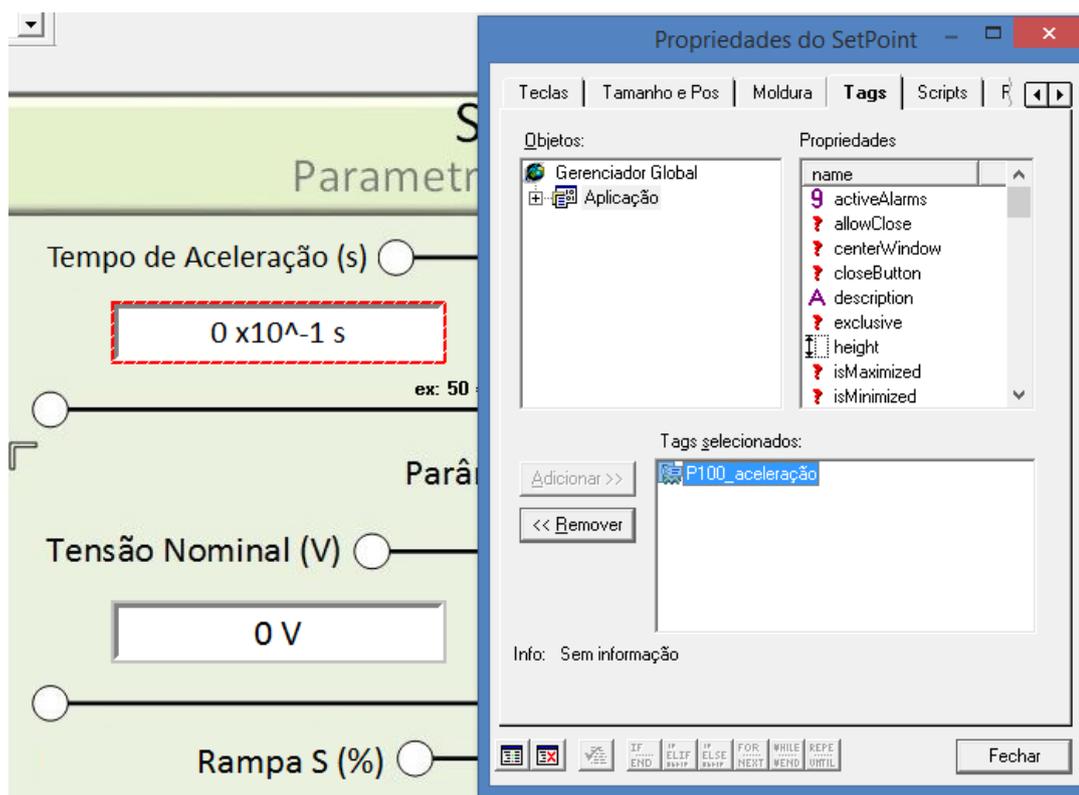


Figura 22 - Associando uma *tag* PLC a um objeto *setpoint*.



Figura 23 - Leitura do parâmetro P003. (a) Associação da tag PLC à tag expressão. (b) Propriedades do display de corrente.

Tabela 17 - Exemplo de telegrama para escrever no parâmetro P100.

Pergunta (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do Escravo	02h	Endereço do Escravo	02h
Função	10h	Função	10h
Registrador inicial (<i>High</i>)	00h	Registrador inicial (<i>High</i>)	00h
Registrador inicial (<i>low</i>)	64h	Registrador inicial (<i>low</i>)	64h
Nº de Registradores (<i>High</i>)	00h	Nº de Registradores (<i>High</i>)	00h
Nº de Registradores (<i>low</i>)	01h	Nº de Registradores (<i>low</i>)	01h
P100 (<i>low</i>)	00h	CRC-	02h
P100 (<i>High</i>)	32h	CRC+	D2h
CRC-	91h		
CRC+	75h		

Tabela 18 - Exemplo de telegrama para ler o parâmetro P003.

Pergunta (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do Escravo	02h	Endereço do Escravo	02h
Função	03h	Função	03h
Registrador inicial (<i>High</i>)	00h	Byte count	04h
Registrador inicial (<i>low</i>)	03h	P100 (<i>low</i>)	00h
Nº de Registradores (<i>High</i>)	00h	P100 (<i>High</i>)	96h
Nº de Registradores (<i>low</i>)	01h	CRC-	38h
CRC-	65h	CRC+	95h
CRC+	CBh		

3.4.4 Leitura e Escrita de Variáveis Básicas

Variáveis básicas são as variáveis internas do inversor de frequência, e que somente podem ser acessadas via serial. É possível através das variáveis básicas, por exemplo, alterar a referência de velocidade, ler o estado, habilitar ou desabilitar o inversor, etc.

A variável básica V03 (Seleção do comando lógico) é uma variável de escrita, cujos bits tem os significados mostrados pela Tabela 19, o endereço dessa variável na memória do CFW-08 é 5003.

Por exemplo, para habilitar o inversor os bits que são relacionados a ação que se deseja fazer devem estar em estado alto, ou seja, o Bit 8 e Bit 9, habilita rampa (gira/para) e habilita geral, respectivamente, devem ser 1, para que a ação desejada (habilitar inversor) possa funcionar, assim o Bit 1 pode ser 1, caso deseja-se ligar o motor e 0 caso queira-se pará-lo.

Sendo assim os valores para leitura e escrita e suas funções, usadas no presente trabalho, podem ser vista através da Tabela 20.

Tabela 19 - Bits da Variável Básica V03.

Byte High: máscara da ação desejada. O bit correspondente deve ser colocado em 1, para que a ação ocorra.	Byte Low: nível lógico da ação desejada
Bit 8: 1 = habilita rampa (gira/para)	Bit 0: 1 = habilita (gira) 0 = desabilita por rampa (para)
Bit 9: 1 = habilita geral	Bit 1: 1 = habilita 0 = desabilita geral (para por inércia)
Bit 10: 1 = sentido de rotação	Bit 2: 1 = sentido de rotação horário 0 = sentido de rotação anti-horário
Bit 11: 1 = JOG	Bit 3: 1 = JOG ativo 0 = JOG inativo
Bit 12: 1 = local/remoto	Bit 4: 1 = remoto 0 = local
Bit 13: não utilizado	Bit 5: não utilizado
Bit 14: não utilizado	Bit 6: não utilizado
Bit 15: 1 = "RESET" do inversor	Bit 7: Transição de 0 para 1 nesse bit provoca "RESET" do inversor, caso o mesmo esteja em alguma condição de erro.

Tabela 20 - Valores de escrita e leitura da variável básica V03.

Funções	Escrita		Leitura
	Decimal	Binário	Decimal
Ligar motor, sentido horário	1799	00000111 00000111	7
Ligar motor, sentido anti-horário	1795	00000111 00000011	3
Desligar Motor (mantendo sentido horário)	1796	00000111 00000100	4

Para fazer o motor ligar o decimal 1799 deve ser escrito na *tag* PLC que está associada ao endereço Modbus correspondente a variável básica V03.

A variável básica V04 (Referência de velocidade dada pela serial), endereçada em 5004, é uma variável de leitura e escrita, ela permite enviar a referência de velocidade (em Hz) para o inversor, desde que a parametrização para este comando esteja correta (P222 = 5), a resolução

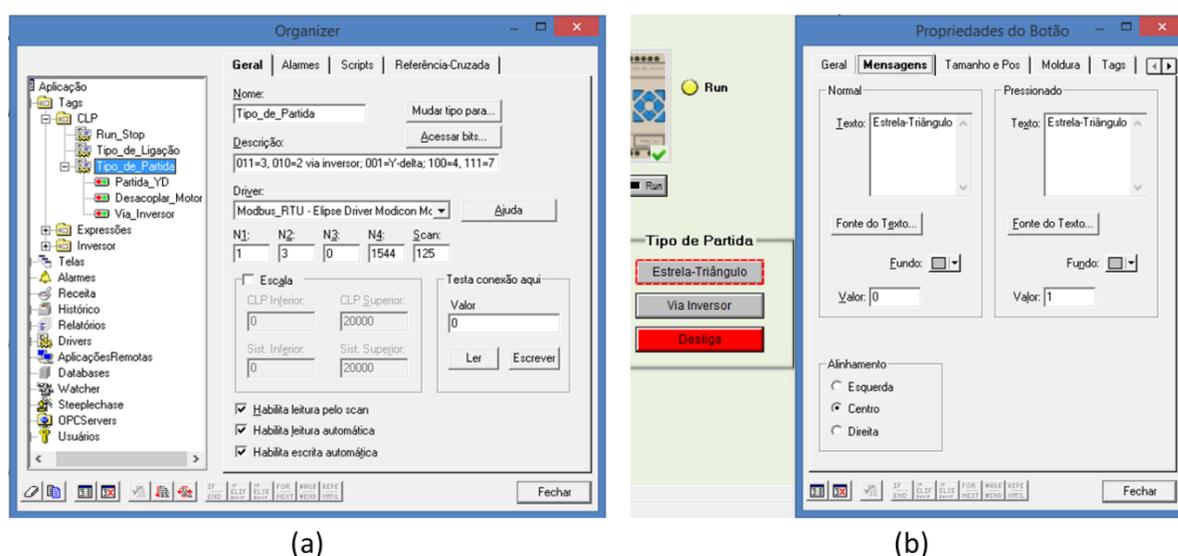
desta variável é de duas casas decimais, portanto se deseja-se que a frequência do motor seja de 25 Hz, deve-se enviar 2500.

3.4.5 Tela Principal

A tela principal é mostrada em detalhe pela Figura 24, nela é possível mudar o estado do CLP (*run/stop*), escolher o tipo de partida do motor e monitorá-lo. Também é possível selecionar a situação dos comandos (local ou remoto), além de poder monitorar os alarmes mais importantes.



Figura 24 - Tela Principal.



(a)

(b)

Figura 25 - Criação dos botões de tipo de partida. (a) Tag PLC relacionada às variáveis M01~M03. (b) Valor escrito no botão Estrela-Triângulo.

Os botões que definem o tipo de partida estão associados com as *tags* PLC correspondentes as entradas M01~M03, bits 0, 1 e 2, respectivamente, da variável M. Que tem como endereço, segundo a Tabela 16, o decimal 1544, assim como ilustrado na Figura 25 (a). A Figura 25 (b) demonstra o valor escrito na entrada quando o botão Estrela-Triângulo é pressionado.

Para criar a animação que indica quando a partida é estrela-triângulo ou via inversor, deve-se criar um objeto de animação adicionando duas zonas, a primeira é onde o LED estará apagado e a segunda onde ele estará aceso. Deve-se associar a este objeto a *tag* PLC relacionada ao tipo de ligação (saídas Y01~Y04), Figura 26 (a).

Na Figura 26 (b) pode-se observar o valor mínimo e máximo em que o LED da partida Estrela-Triângulo acenderá. Na Tabela 21 estão ilustradas as situações em que os LEDs do tipo de partida serão acionados.

Tabela 21 – Situações em que os LEDs do tipo de partida serão ativados.

Tipo de partida	Y04	Y03	Y02	Y01	Decimal
Estrela-Triângulo	0	0	1	1	3
	0	1	0	1	5
Via Inversor	1	1	0	0	12

As indicações do tipo de ligação também seguem a mesma lógica de configuração mostrada pela Figura 26, mas estão ligadas às variáveis Y02 (estrela), Y03 (triângulo). A Tabela 22 mostra as situações para que esta animação seja ativada.

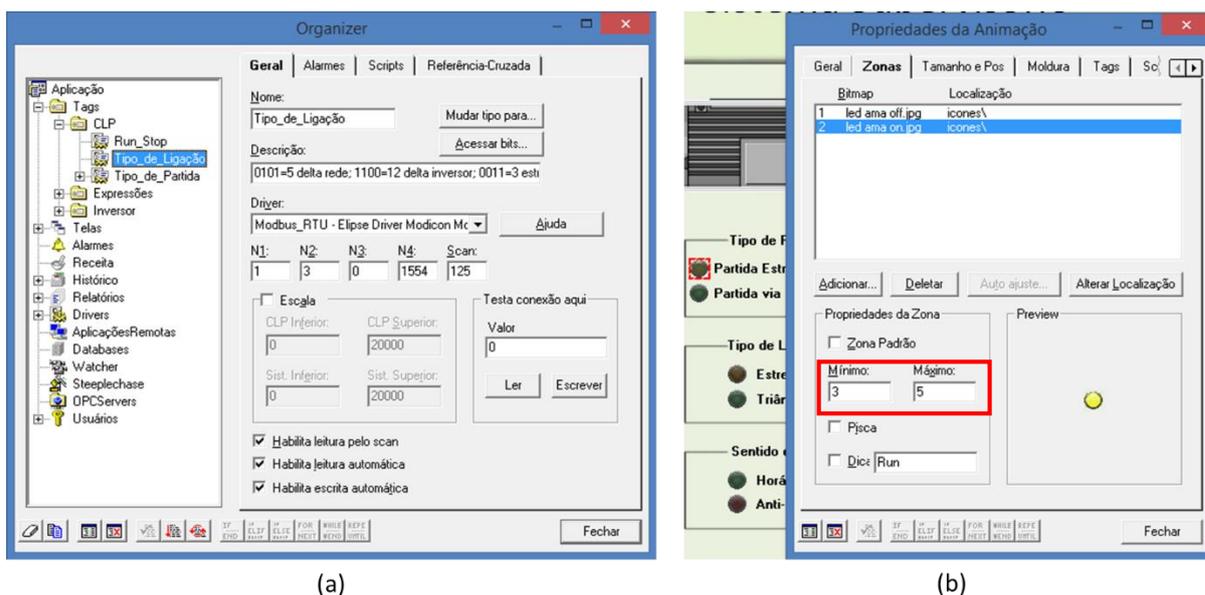


Figura 26 - Configuração de um objeto de animação. (a) *Tag* PLC relacionada às variáveis Y01~Y04. (b) Zonas em que o LED “acenderá”.

Os *displays* mostrados em todas as telas são criados da mesma maneira que mostra a Figura 23, mudando apenas as *tags* as quais são associados.

Tabela 22 - Situações em que os LEDs do tipo de ligação são ativados.

Tipo de Ligação	Y04	Y03	Y02	Y01	Decimal
Estrela	0	0	1	1	3
Triângulo	0	1	0	1	5
	1	1	0	0	12

3.4.6 Tela Motor

A tela em que o operador pode fazer os comandos do motor através do inversor de frequência está ilustrada pela Figura 27, sendo que nesta tela pode-se ligar o motor, variar a velocidade e seu sentido de rotação. É possível, também, visualizar o tipo de partida e configuração, velocidade e corrente do motor. Além disso, é possível ajustar os valores de tensão, corrente e velocidade nominais.

Para ligar o motor o operador deve pressionar o botão vermelho mostrado na Figura 28, este botão foi criado e associado a *tag* da variável básica V03. Quando pressionado, o botão envia para esta variável o valor 1799, e assim o motor é ligado no sentido horário. Para desligar o botão enviará 1796, desligando o motor e deixando-o no sentido horário. Os significados destes valores podem ser vistos na Tabela 20.

O botão para mudar o sentido de rotação destacado na Figura 29 também está associado à *tag* da variável básica V03, sendo que para ligar o motor no sentido anti-horário o botão deve escrever o valor 1796 e 1799 para sentido horário.

dd/mm hh:mm:ss	Tipo	Pri	Estado	Comentário	NomeTag	Valor
02/07 15:57:50	DDO	1	UNACK	Corrente do motor muito baixa!	Corrente (A)	0,00
02/07 15:57:50	DDO	1	UNACK	Corrente nominal muito baixa!	Corrente Nominal (A)	0,00

Figura 27 - Tela Motor.

Para variar a velocidade um objeto *setpoint* teve de ser criado, associando-o à tag da variável básica V04. Para escrever um valor nesta tag deve se ter em mente a resolução desta variável que pode ser vista na Tabela 14.

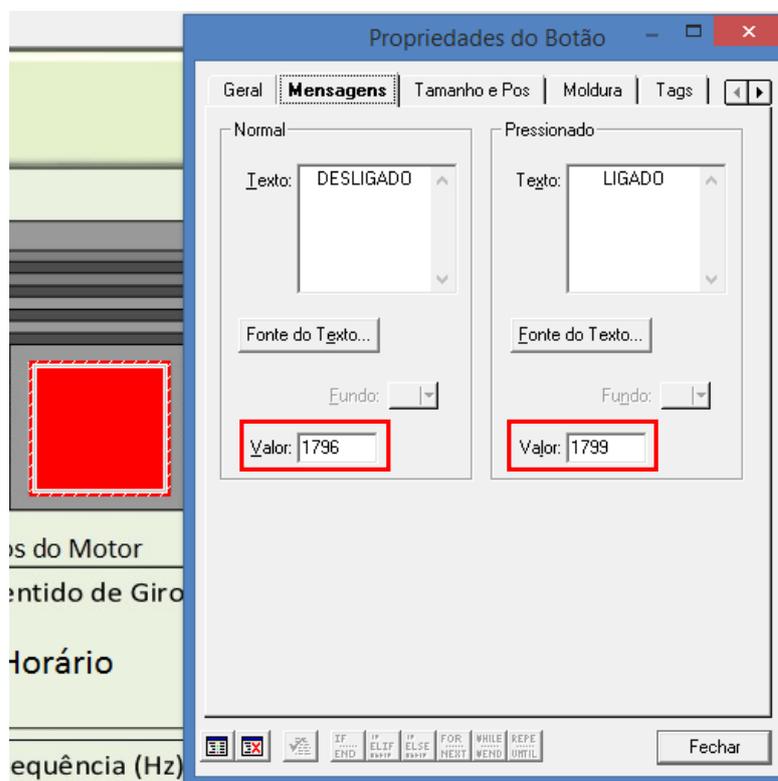


Figura 28 - Configuração do botão liga/desliga motor.

3.4.7 Tela Parametrização

Na tela de Parametrização, Figura 30, é possível acessar os parâmetros descritos na Tabela 13. É possível setar o tempo de aceleração e desaceleração da rampa de partida e frenagem do motor, bem como a porcentagem da rampa S, além de poder definir os valores nominais de tensão, corrente e velocidade do motor.

Os objetos de *setpoint* mostrados na Figura 30 podem ser configurados de maneira parecida com o da Figura 22, mudando apenas a *tag* associada. A tag da rampa S é associada a um objeto *slider* e sua configuração é mostrada pela Figura 31. Onde os valores variam de 0 a 2, com passo de uma unidade, assim o valor 0, 1 e 2, significam, respectivamente, 0%, 50% e 100% de rampa S.

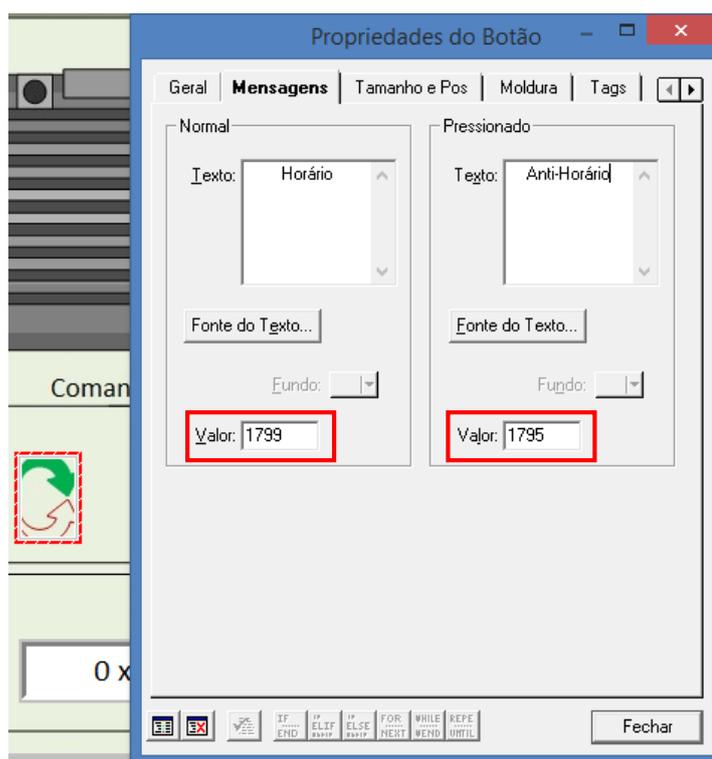


Figura 29 - Configuração do botão sentido de giro.

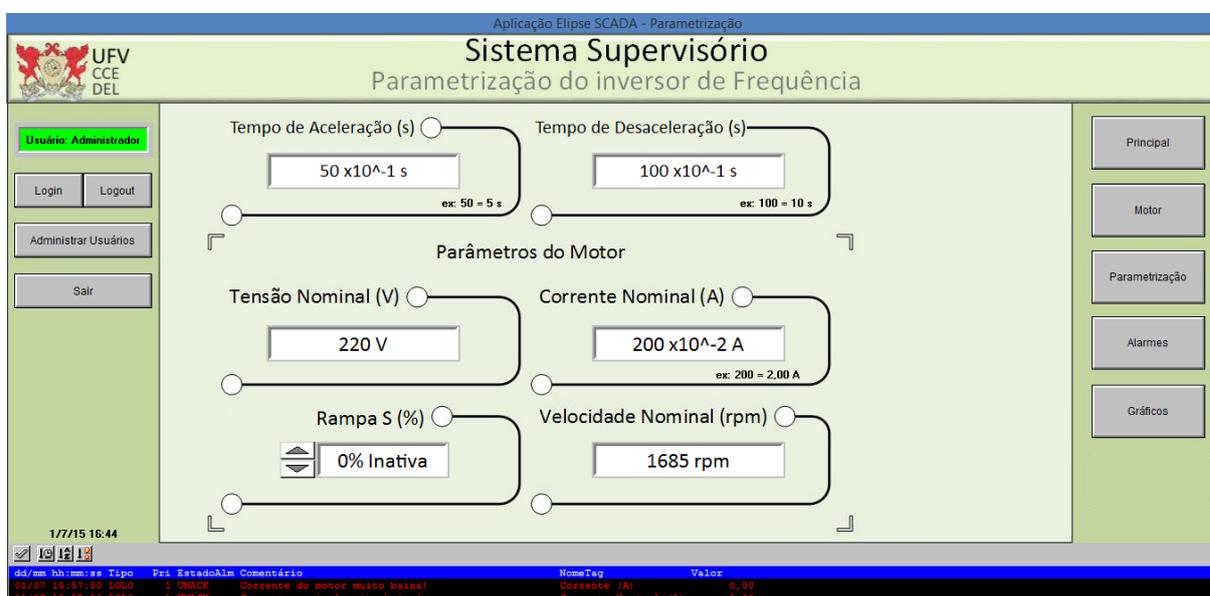


Figura 30 - Tela Parametrização.

3.4.8 Tela Alarmes

A visualização do alarmes deve ser feita em todas as telas, mas na tela Alarmes, Figura 32, pode-se vê-los de maneira mais detalhada.

Às tags julgadas mais importantes foram definidos alguns alarmes, alguns definidos como alarmes normais, já outros foram definidos como alarmes críticos. A configuração de

prioridade no Elipse SCADA é inversamente proporcional a sua importância, ou seja, quanto maior a prioridade menor a importância do alarme. A Figura 33 mostra os alarmes criados para a tag da velocidade do motor.

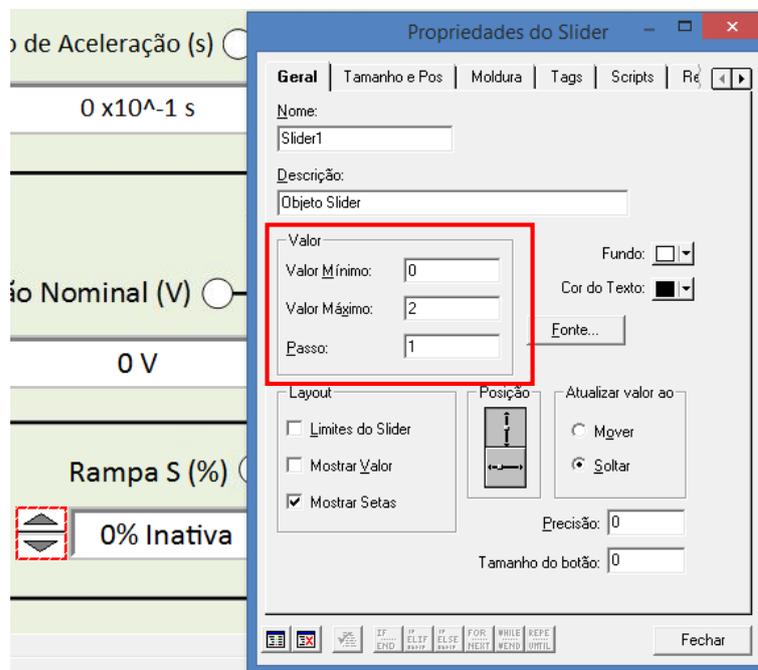


Figura 31 - Configuração de um objeto slider.



Figura 32 - Tela Alarmes.

3.4.9 Tela Gráficos

Os gráficos da corrente de saída e da velocidade do motor são mostradas na tela Gráficos, Figura 34. O gráfico em azul mostra a variação da velocidade quando a motor estava

setado para 1300 rpm e mudou-se o sentido de rotação. Nota-se uma diminuição gradual na velocidade e uma mudança brusca na corrente do motor no momento em que inverteu-se o sentido de giro, e com o tempo a velocidade do motor e sua corrente foram voltando para os valores em que foram inicialmente definidos.

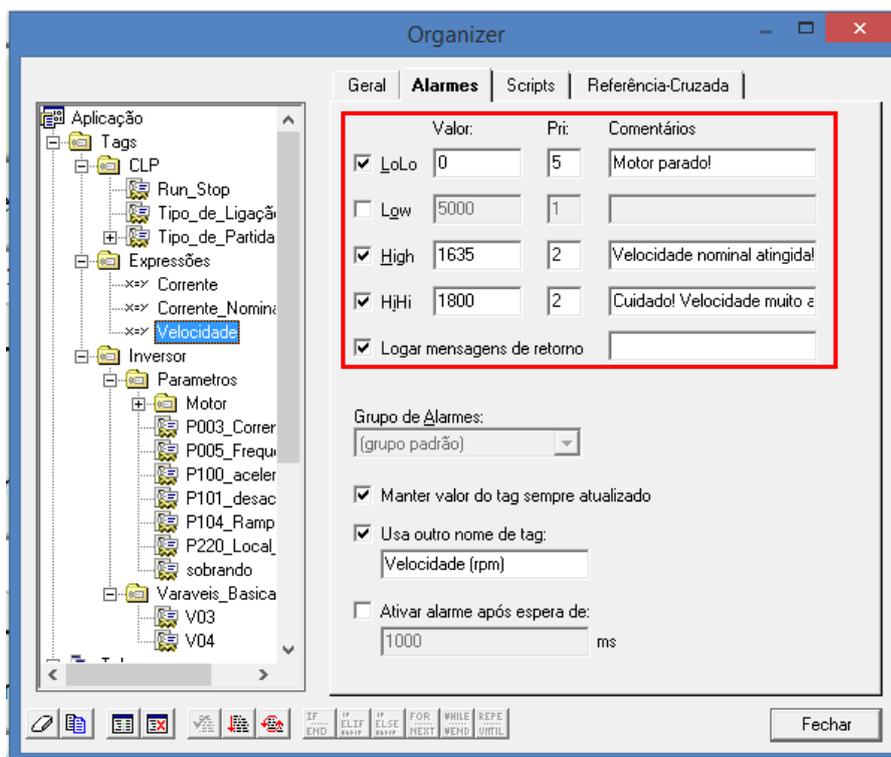


Figura 33 - Alarmes da tag velocidade.

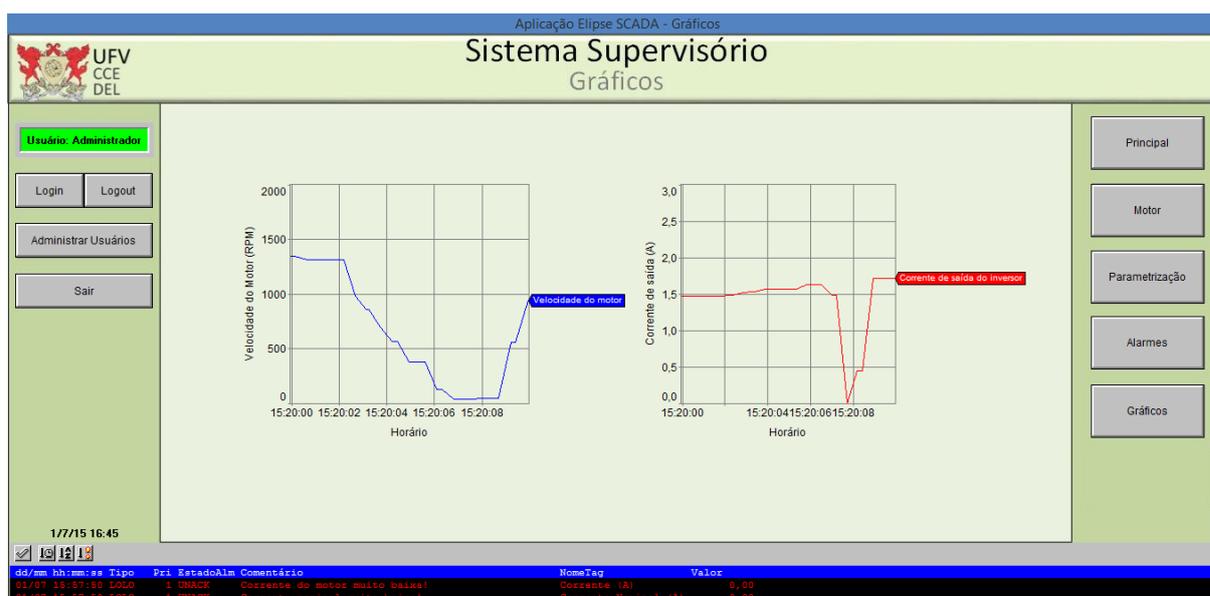


Figura 34 - Tela Gráficos.

Para o gráfico da velocidade do motor criou-se um objeto de tendências e associou a uma pena do gráfico a *tag* referente a velocidade, assim como mostra a Figura 35.

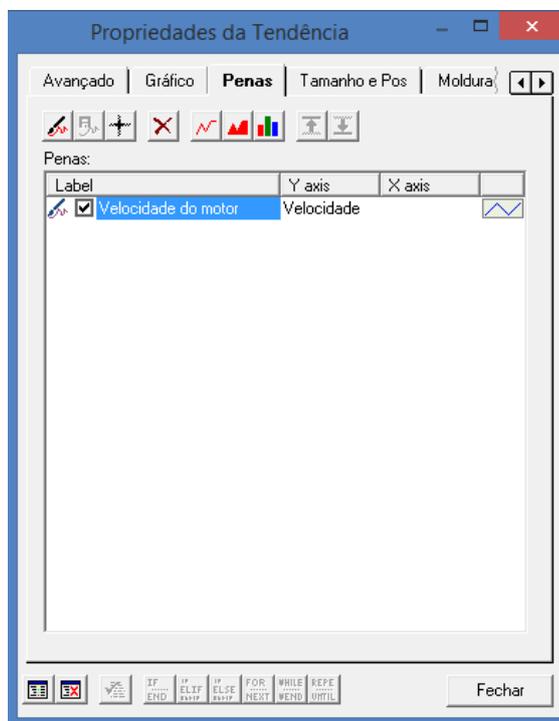


Figura 35 - Propriedades da tendência.

4 *Considerações Finais*

O objetivo do presente trabalho é realizar a comunicação serial entre o computador e o inversor de frequência CFW-08, através do protocolo Modbus RTU, para fazer a parametrização e o controle deste via interface gráfica em uma estação remota.

Para tal, a criação de um sistema supervisório via Elipse SCADA se fez necessária, pois este programa possibilitou a conexão entre o computador e o inversor de frequência usando o protocolo Modbus. Entretanto, para que se possa acessar todos os parâmetros do inversor por meio do sistema supervisório é aconselhável a compra deste *software*, já que este possui uma limitação, que neste trabalho não foi um problema por se tratar de um protótipo.

Visando a aplicação da comunicação serial, criou-se um modelo em que se é possível controlar o tipo de partida de um motor de indução usando um CLP, além de fazer a conexão do inversor de frequência com o motor de indução, controlando o sentido de giro e a velocidade de rotação deste motor.

Referências Bibliográficas

BENTHIEN, Andrey. **Sistema De Aquisição De Dados E Supervisão De Um Misturador Industrial**. 59f. Trabalho de Conclusão de Curso (Graduação em Informática) – Universidade do Planalto Catarinense, Lages, 2007. Disponível em: <https://revista.uniplac.net/ojs/index.php/tc_si/article/viewFile/765/477>. Acesso em: 22/06/2015.

BORDIM, Jacir Luiz. **Transmissão de dado**. Universidade de Brasília, Brasília, 2006. 13 p. Disponível em: <http://www.cic.unb.br/~bordim/TD/Arquivos/G04_Monografia.pdf>. Acessado em: 21/06/2015.

BRUGNEROTTO, André Luis; KUME, Wagner Luiz. **Rede de controle e supervisão para inversores de frequência utilizando labVIEW**. 2014. 96 f. Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2014. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/3048>>. Acesso em: 21/06/2015.

CAPELLI, Alexandre. **Inversores de Frequência. Mecatrônica Atual**. São Paulo: Editora Saber, ano 01, nº 02, fevereiro 2002.

CASSIOLATO, César. **Redes Industriais. Artigos Técnicos Smar**. São Paulo, 2011. Disponível em: <<http://www.smar.com/newsletter/marketing/index150.html>>. Acesso em: 21/06/2015.

FERREIRA JÚNIOR, Milton Gontijo. **Controle de um Inversor de Frequência via CLP**. 2012. 40f. Monografia (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Ouro Preto, Ouro Preto, 2012. Disponível em: <<http://em.ufop.br/cecau/monografias/2012/Milton%20Gontijo%20Ferreira%20Junior.pdf>>. Acesso em: 21/06/2015.

FRANCHI, C. M. **Inversores de Frequência: Teoria e Aplicações**. 2ª edição. Editora Érica, 2009.

GEORGINO, Marcelo. **Automação Aplicada Descrição e Implementação de Sistemas Sequenciais com PLCs**. 6. ed. São Paulo: Editora Érica, 2006.

GUARESE, Giuliano Bruno Martins. **Arquitetura Híbrida de Comunicação para Ambientes de Automação Industrial: Protocolos IEEE 802.15. 4 e Modbus RTU sobre RS485**. 63f. Monografia (Graduação em Engenharia da Computação) – Faculdade de Engenharia e de Informática. Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2011. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/12431>>. Acesso em: 24/06/2015.

LOPES, Marco Aurélio Moreira. **Importância dos Sistemas Supervisórios no Controle de Processos Industriais**. 49f. Monografia (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Ouro Preto, Ouro Preto, 2009. Disponível em: <<http://www.em.ufop.br/cecau/monografias/2009/MARCO%20AURELIO%20M.%20LOPE%20S.pdf>>. Acesso em: 24/06/2015.

MACHADO, Jorge Luis. **Sistema De Aquisição De Dados De Um Processo Industrial**. 61f. Trabalho de Conclusão de Curso (Graduação em Informática) – Universidade do Planalto

Catarinense, Lages, 2006. Disponível em: <https://revista.uniplac.net/ojs/index.php/tc_si/article/download/751/463>. Acesso em: 22/06/2015.

MARIA, R. E. **Estudo e Implementação do Protocolo Modbus para Automação Predial**. 73f. Monografia (Bacharelado em Ciência da Computação) - Centro de Engenharia e Ciências Exatas, Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, 2007. Disponível em: <<http://www.gnoia.org/~holoturoide/mono.pdf>>. Acesso em: 22/06/2015.

MOREIRA, Maicon Vaz. **Automação De Um Processo De Convecção De Calor E Massa**. 50f. Monografia (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Ouro Preto, Ouro Preto, 2009. Disponível em: <<http://www.em.ufop.br/cecau/monografias/2009/Maicon%20Vaz%20Moreira.pdf>>. Acesso em: 22/06/2015.

PEDRUZI, Gerson Ovídio Luz. **Controle de Nível de Tanques conectados Utilizando CLP Clic02 e Elipse SCADA**. 65f. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal de Viçosa, Viçosa, 2014.

PEREIRA, Paulo Henrique Cruz. **Sistema Automatizado de Irrigação por Pivô Central: Uma Proposta para Melhoria da Eficiência Energética Através do Uso de Controladores Lógicos Programáveis em Conjunto com Inversor de Frequência em Rede sem Fio Zigbee**. 133f. Tese (Doutorado em Engenharia Agrícola) – Universidade Federal de Lavras, Lavras, 2011. Disponível em: <<http://repositorio.ufla.br/jspui/handle/1/3517>>. Acesso em: 21/06/2015.

PIRES, Paulo Sérgio Motta; OLIVEIRA, Luiz Affonso H. Guedes de; BARROS, Diogo Nascimento. **Aspectos de segurança em sistemas SCADA** – Uma visão geral. 4º Congresso Internacional de Automação, Sistemas e Instrumentação. In: Controle & Instrumentação, Maio de 2005. Disponível em: <http://www.dca.ufrn.br/~affonso/FTP/artigos/2004/isa_scada_2004.pdf>. Acesso em: 22/06/2015.

PIRES, Waldiberto de Lima. **Motores de indução alimentados por inversores de frequência PWM**. Departamento de Pesquisa e Desenvolvimento do Produto –WEG, Jaraguá do Sul, 2006. Disponível em: <http://www.coe.ufrj.br/~richard/Acionamentos/motores_de_inducao_alimentados_por_conv_ersores_de_frequencia_PWM.pdf>. Acesso em: 11/07/2015.

RADDATZ, Thiago Galli. **Sistema De Monitoramento De Uma Esteira Didática Via Web**. 81f. Trabalho de Conclusão de Curso (Graduação em Informática) – Universidade do Planalto Catarinense, Lages, 2007. Disponível em: <https://revista.uniplac.net/ojs/index.php/tc_si/article/viewFile/762/474>. Acesso em: 22/06/2015.

RAYSARO, Márcio Coiado. **Sistema open-source de Supervisão e Aquisição de Dados**. 99f. Monografia (Graduação em Sistemas de Informação) – Faculdade de Tecnologia. Universidade de Cuiabá, Cuiabá, 2012. Disponível em: <www.scadabr.org.br/sites/default/files/TCC_Marcio_Rayasaro.pdf>. Acesso em: 22/06/2015.

RESENDE, Michael de Oliveira. **Desenvolvimento de um Sistema Supervisório com Acesso Remoto para o Controle de um Processo de Acionamento Elétrico Utilizando Inversor De Frequência**. 66f. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal de Viçosa, Viçosa, 2014.

SANTOS, S. **Conceitos Básicos de RS-485 e RS-422**. Novus, Porto Alegre, 2007. 6p. Disponível em:

<<http://faq.novus.com.br:8080/phpmyfaq/attachments/10/Conceitos%20Basicos%20de%20RS485%20e%20RS422.pdf>>. Acesso em: 21/06/2015.

SOUZA, Adriano de. **Controle De Motores Trifásicos Interligados Através De Redes**. 80f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade do Planalto Catarinense, Lages, 2007. Disponível em: <https://revista.uniplac.net/ojs/index.php/tc_si/article/viewFile/790/500>. Acesso em: 22/06/2015.

TOVAR, Eduardo Manuel de Médicis. **Suporte à Comunicação de Tempo-Real Utilizando Redes Industriais Normalizadas**. 160f. Dissertação (Doutorado em Engenharia Eletrotécnica e de Computadores) – Faculdade de Engenharia da Universidade do Porto, Porto, 1999. Disponível em: <<http://hdl.handle.net/10216/11682>>. Acesso em: 21/06/2015.

VILLA, Daniel Khéde Dourado. **Comunicação Modbus RTU no Clic02-WEG para Criação de Interfaces Gráficas**. 66f. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal de Viçosa, Viçosa, 2014.

WEG. **Guia de aplicação de inversores de frequência**. Jaraguá do Sul, 2006. Disponível em: <http://www.mundoeletrico.com/downloads/Guia_de_Aplicacao_de_Inversores_de_Frequencia.pdf>. Acesso em: 21/06/2015.

WEG. **Manual do Inversor de Frequência CFW-08**. Jaraguá do Sul, 2009. Disponível em: <<http://ecatalog.weg.net/files/wegnet/WEG-cfw-08-manual-do-usuario-0899.5241-5.2x-manual-portugues-br.pdf>>. Acesso em: 20/03/2015.

Apêndice A – Configuração do Elipse SCADA na Rede Modbus RTU

A maioria dos fabricantes de sistemas supervisórios e IHMs disponibilizam *device drives* que agilizam a integração de equipamentos que se comunica através de Modbus em uma rede industrial. No site oficial do Elipse SCADA o usuário pode baixar o driver de comunicação Modbus, basta procurar por Driver Modicon Modbus Master (ASC/RTU/TCP). Após isso, faz-se a configuração do Elipse, seguindo a sequência:

1º Passo: Acessar o componente *Organizer*, onde se encontram os *drivers*, *tags*, etc.

2º Passo: Dentro da caixa de diálogo do *Organizer*, clique em *Drivers*.

3º Passo: Clique em ‘Novo’, para adicionar um novo *driver*.

4º Passo: Depois de abrir uma nova caixa de diálogo, será necessário localizar o *driver*. O *driver* estará localizado na pasta em que foi salvo o *download*.

5º Passo: Abrir o *driver*.

Pode-se ver estes cinco passos ilustrados na Figura 36.

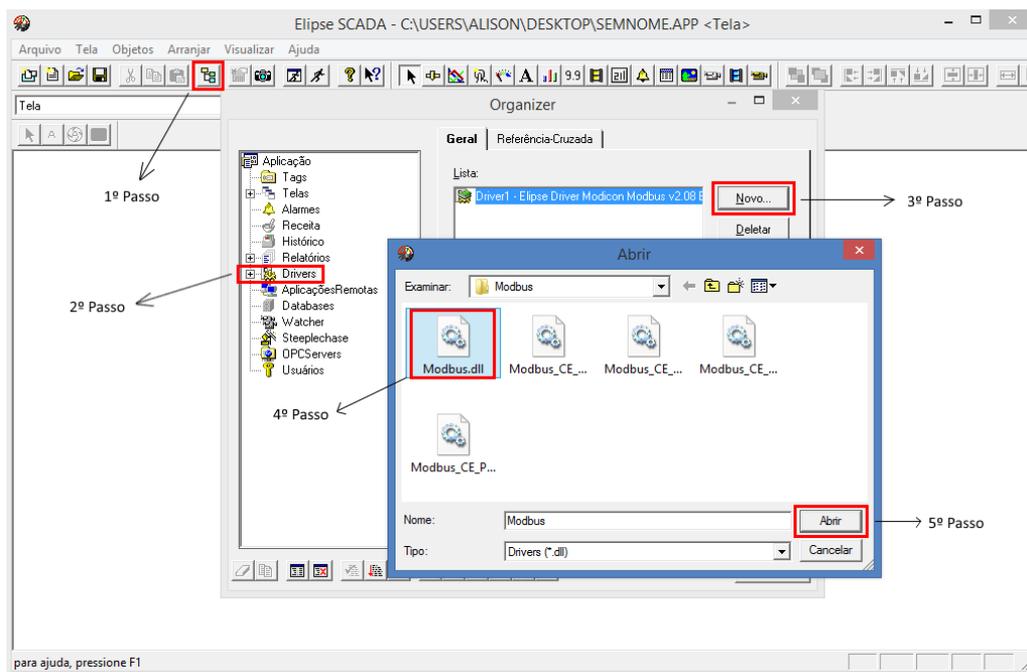


Figura 36 - Tutorial de como adicionar o driver Modbus ao Elipse SCADA.

Agora será necessário configurar o driver, para isto, deve-se seguir a sequência:

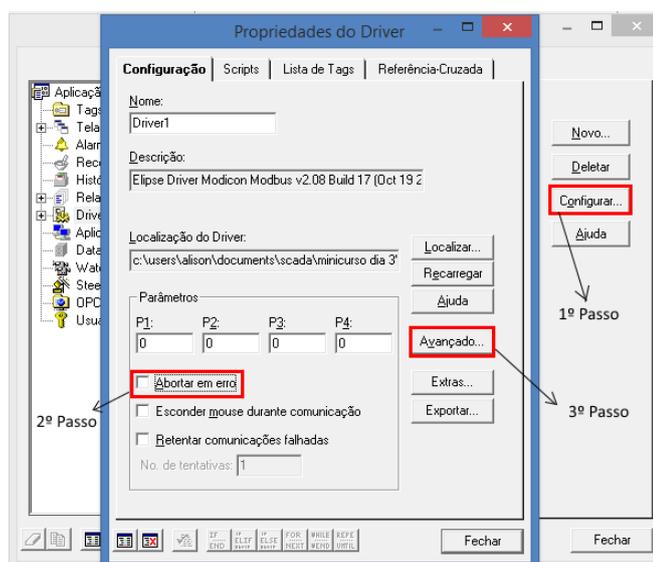
1º Passo: Selecione o *driver* e clique em ‘Configurar’.

2º Passo: Na aba ‘Configuração’ desmarque a opção ‘Abortar em Erro’.

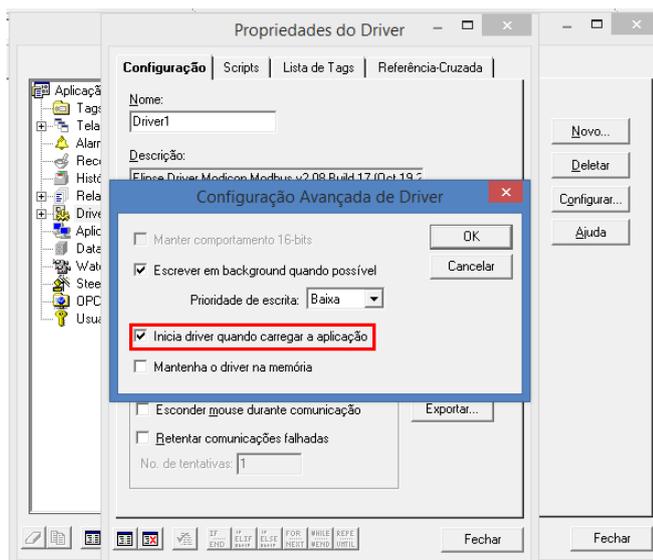
3º Passo: Em propriedades do driver clique em ‘Avançado’.

4º Passo: Marque a opção ‘inicia drive" quando carregar a aplicação e clique em OK. Caso não realize este procedimento o drive não carrega quando a aplicação rodar.

A Figura 37 ilustra os quatro passos mostrados acima.



(a)



(b)

Figura 37 - Configuração do driver Modbus. (a) Três primeiros passos. (b) Quarto passo.

A partir desse momento partir-se-á para a segunda parte da configuração do *driver* clicando em ‘Extras’. Logo ao abrir a nova janela, várias funções adicionadas na Modbus

Operations estarão disponíveis, como pode ser visto na Figura 38. Observe que cada função utilizada possui portas de leitura e escrita diferentes e o tipo de variável também muda.

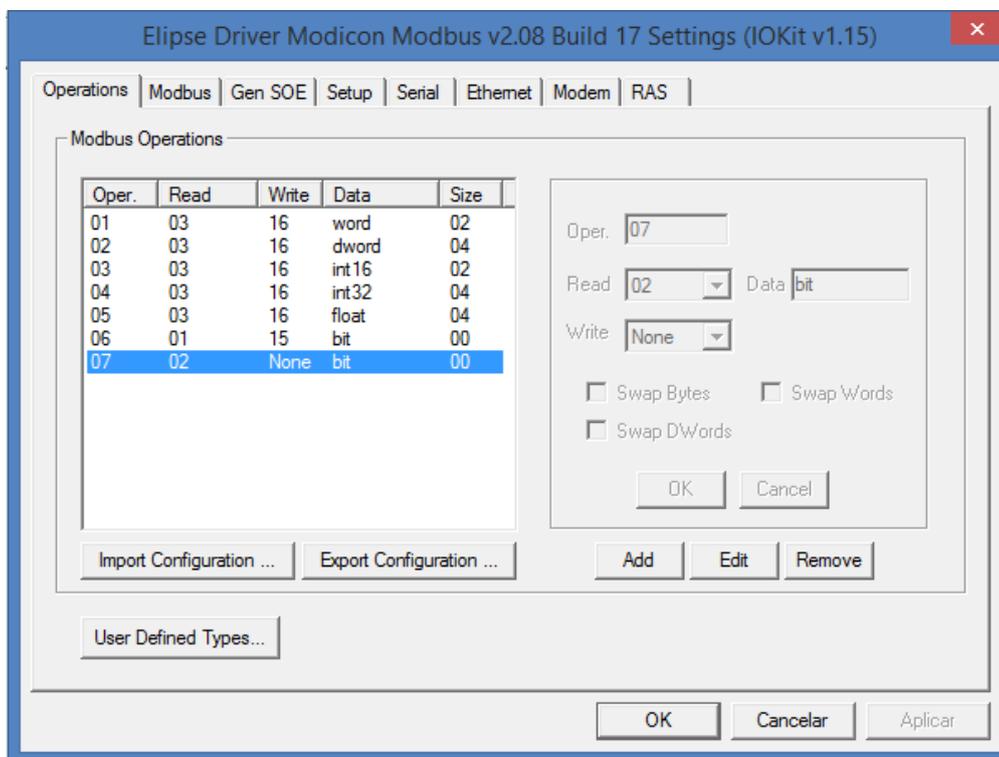


Figura 38 - Tipos de operações feitas pelo driver Modbus do Elipse.

Continuando a configuração deve-se clicar na aba 'Modbus' e escolher o tipo de protocolo para ser 'RTU Mode', ainda nesta aba escolhe-se o *offset* dos endereços para ser igual a zero, assim como pode-se notar na Figura 39. Na aba 'Setup' e mudar o tempo de *Time Out* para 125 ms, conforme ilustrado na Figura 40.

Por fim, é necessário configurar a porta serial do drive. Porém para que isso seja feito, deve-se primeiramente verificar em qual porta do computador o cabo de conexão RS485-USB está conectado, faz-se isto olhando através do Gerenciador de Dispositivos do Windows. Finalmente, na aba 'Serial' efetua-se a configuração da porta serial de acordo com que foi configurado no CLP e Inversor de Frequência, *Baud rate* igual a 38400 bps, 8 bits de dados, sem paridade e 2 *stop* bits, assim como ilustrado na Figura 41.

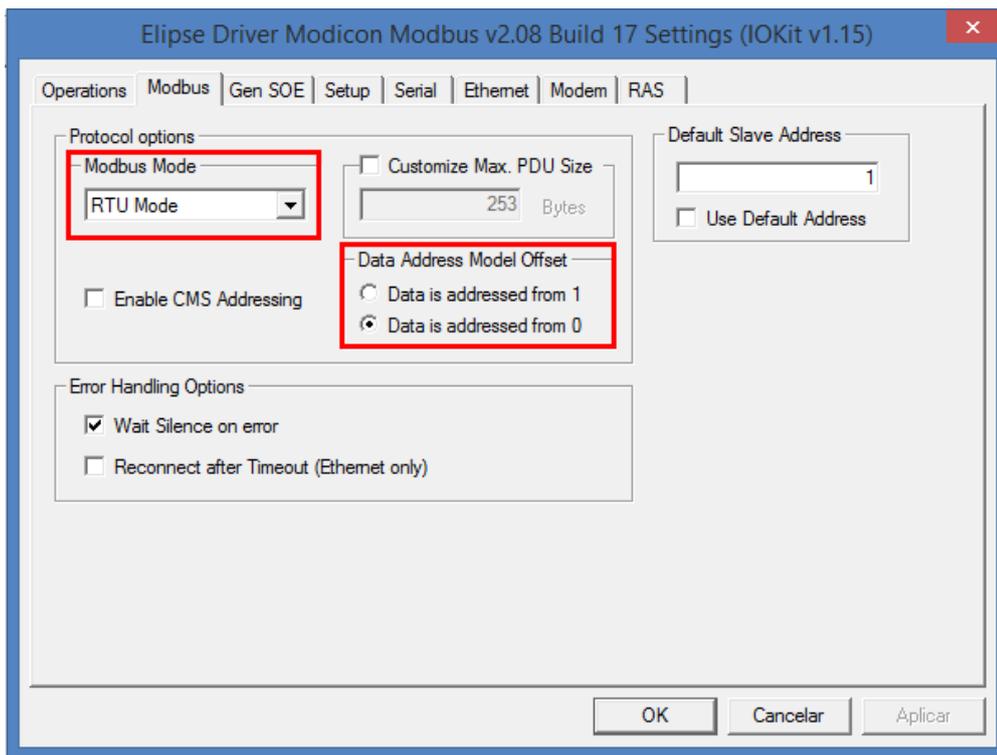


Figura 39 - Configuração do Modo Modbus e *Offset*.

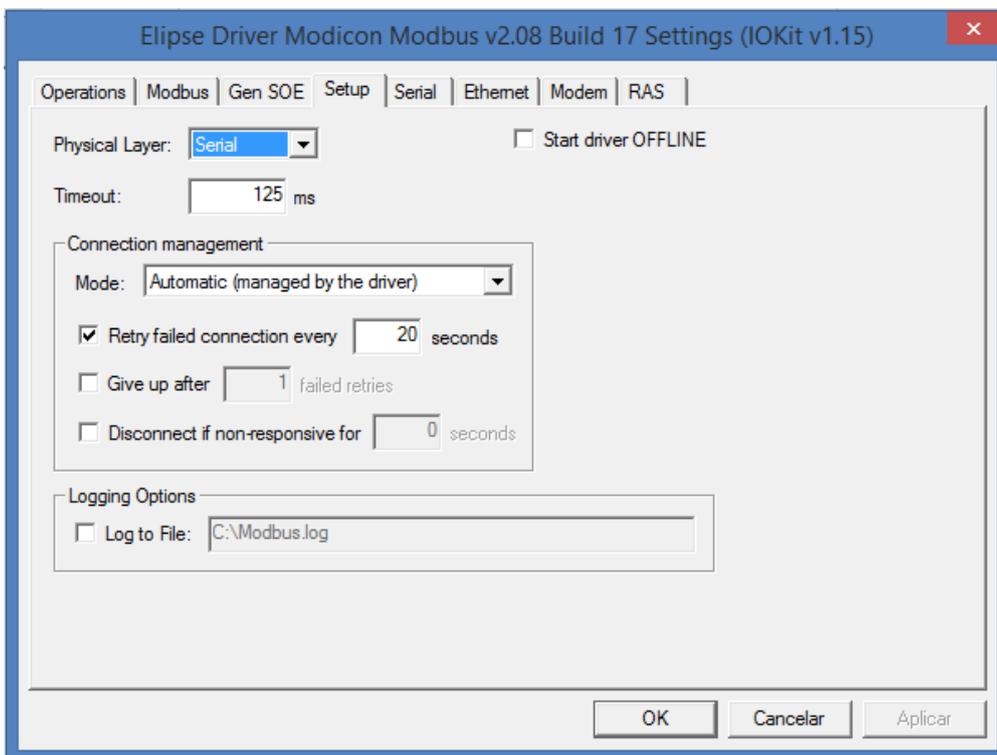


Figura 40 - Mudando o *timeout*.

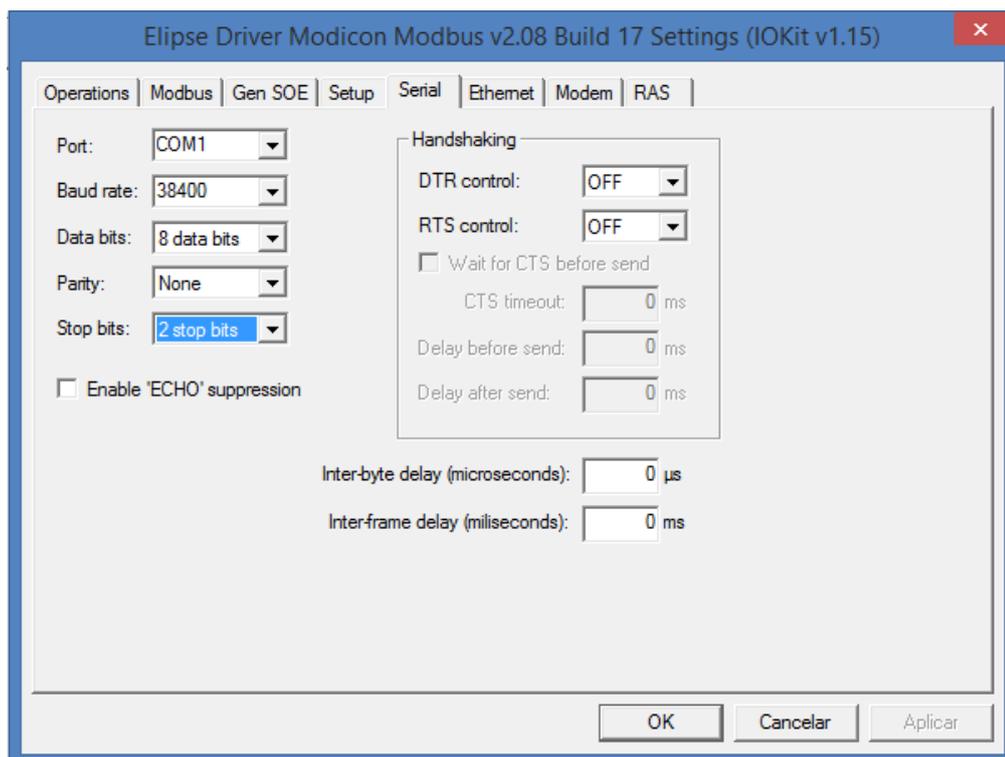


Figura 41 - Configuração da porta serial no Elipse SCADA.

A.1 Configuração do Conversor RS485- USB

Para a rede funcionar corretamente também deve-se configurar a taxa de transmissão, paridade, bits de parada e *Data bits* do cabo conversor.

O primeiro passo é acessar o gerenciador de dispositivos do Windows e lá clicar duas vezes em “Prolific USB-to-Serial Comm Port” como mostra a Figura 42, a janela das propriedades do dispositivos será aberta, na aba ‘Configurações de porta’ devem ser modificados os campos ‘Bits por segundo’, ‘Bits de dados’, ‘Bits de parada’, ‘Paridade’ e ‘Controle de Fluxo’ assim como ilustra a Figura 43.

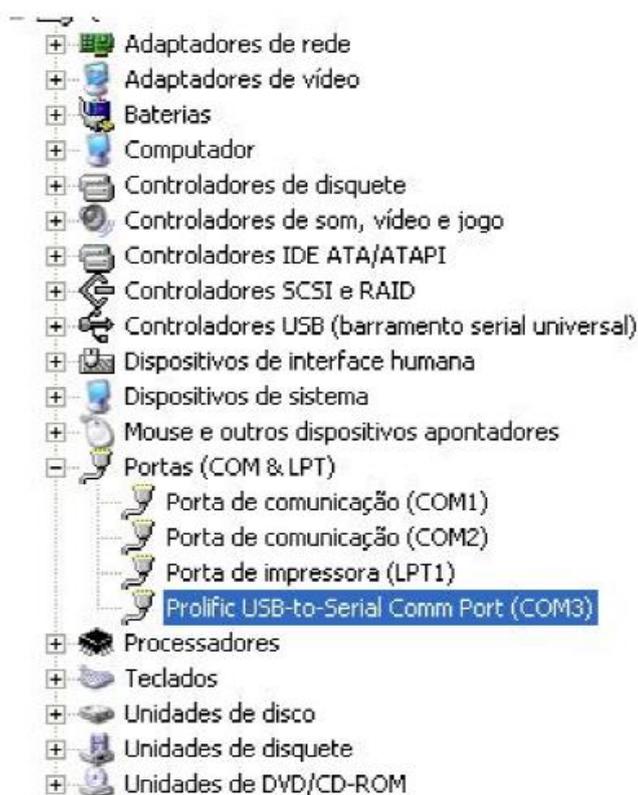


Figura 42 - Localização do cabo conversor no Gerenciador de Dispositivos do Windows.

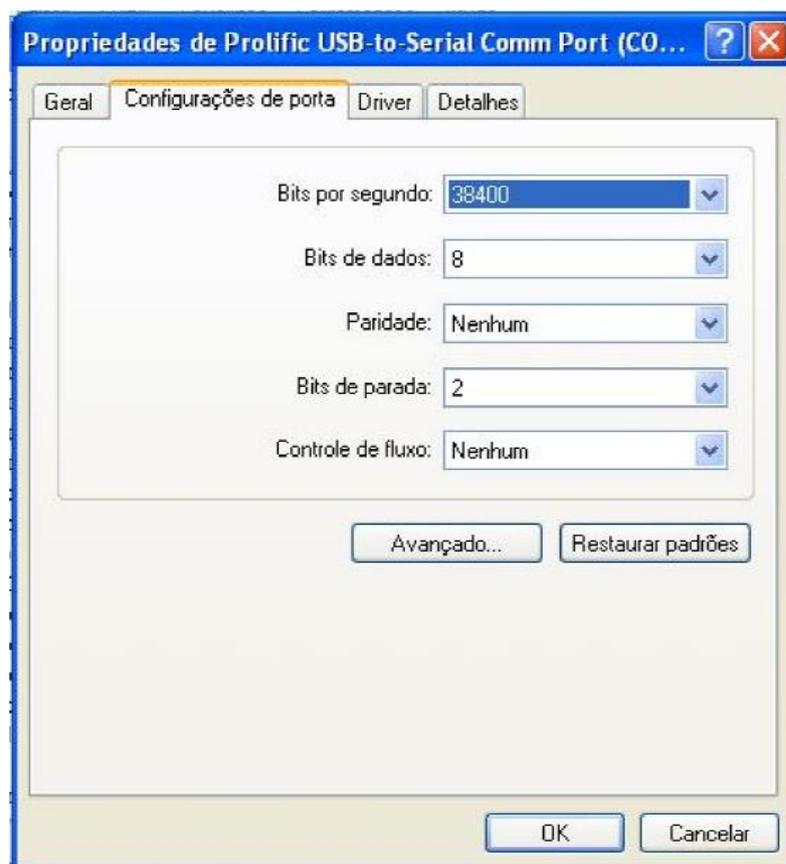


Figura 43 - Propriedade de Prolific USB-to-Serial Comm Port

A.2 Configuração das Tags PLC

Observa-se na Figura 44 cinco campos para configuração: N1, N2, N3, N4 e Scan. O campo N1 é o endereço do escravo que se deseja acessar, neste trabalho configurou-se o Clic02 para trabalhar com o endereço 1 e o CFW-08 com o 2. A figura abaixo mostra a configuração da tag tipo de partida via CLP, por isso está com 1. N2 é o campo onde se define a função de escrita e leitura do driver, as possibilidades são mostradas pela Tabela 23, é mais comum se utilizar a operação 3, pois o CLP e o inversor transmitem dados em 16 bits com sinal. N3 não é usado, podendo deixá-lo em zero. Já N4 é o endereço Modbus em decimal do registrador que se deseja acessar.

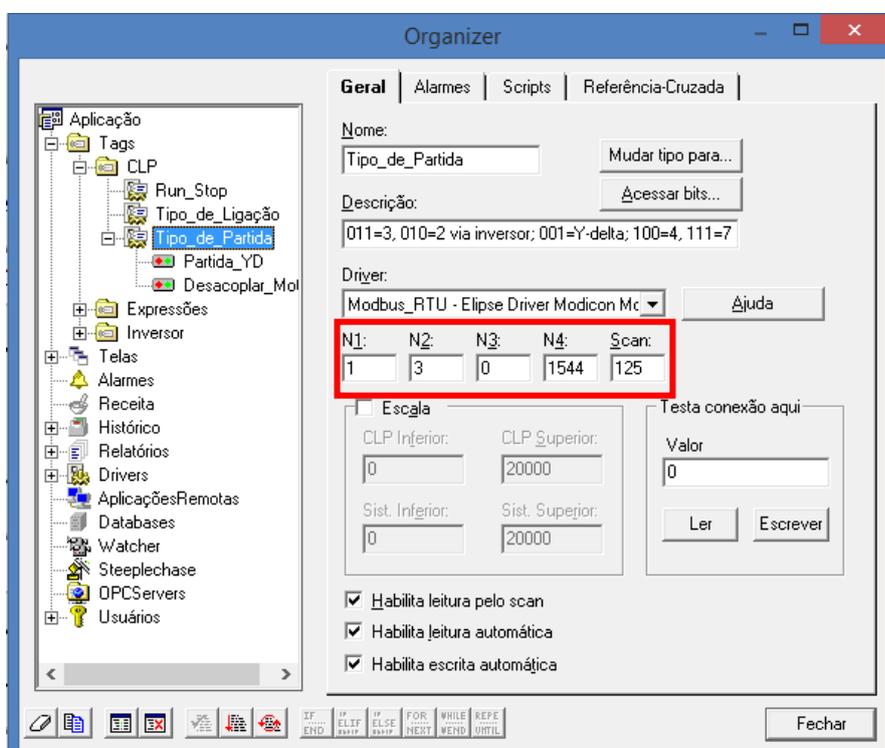


Figura 44 - Configuração do parâmetros N1, N2, N3, N4 e Scan.

Se a configuração do driver Modbus for feita de acordo com os procedimentos adotados neste capítulo, o *offset* será zero, para acessar uma variável do Clic02 basta olhar seu endereçamento na Tabela 3, os endereçamento dos parâmetros e das variáveis básicas do CFW-08 pode ser visto na

Tabela 5 e Tabela 6. O campo Scan é o tempo que se leva para atualização dos dados em milissegundos, com uma taxa de transmissão de 38400 bps o mínimo tempo de *scan* é de 125 ms.

Tabela 23 - Detalhamento das funções Modbus.

Operação	Função de Leitura	Função de Escrita	Tipo de Dado	Finalidade
1	3 – Reading Holding Registers	16 – Write Multiple Registers	word	Para ler e escrever inteiros de 16 bits sem sinal
2	3 – Reading Holding Registers	16 – Write Multiple Registers	dword	Para ler e escrever inteiros de 32 bits sem sinal
3	3 – Reading Holding Registers	16 – Write Multiple Registers	int16	Para ler e escrever inteiros de 16 bits com sinal
4	3 – Reading Holding Registers	16 – Write Multiple Registers	int32	Para ler e escrever inteiros de 32 bits com sinal
5	3 – Reading Holding Registers	16 – Write Multiple Registers	float	Para ler valores com ponto flutuante de 32 bits
6	3 – Reading Holding Registers	15 – Write Multiple Coils	bit	Para ler e escrever bits
7	2 – Read Discrete Inputs	Nenhum	bit	Para ler bits do bloco de dados de entradas discretas

Um exemplo de uma tag PLC para acessar o parâmetro de leitura P003 (Corrente do Motor) pode ser visto na Figura 45. Nota-se que o endereço do escravo (N1) deve ser de acordo com endereço configurado no parâmetro P308 do CFW-08, neste caso 2, é possível visualizar que o endereço do parâmetro (N4) é igual a 3 correspondendo, segundo a Tabela 5, ao parâmetro P003.

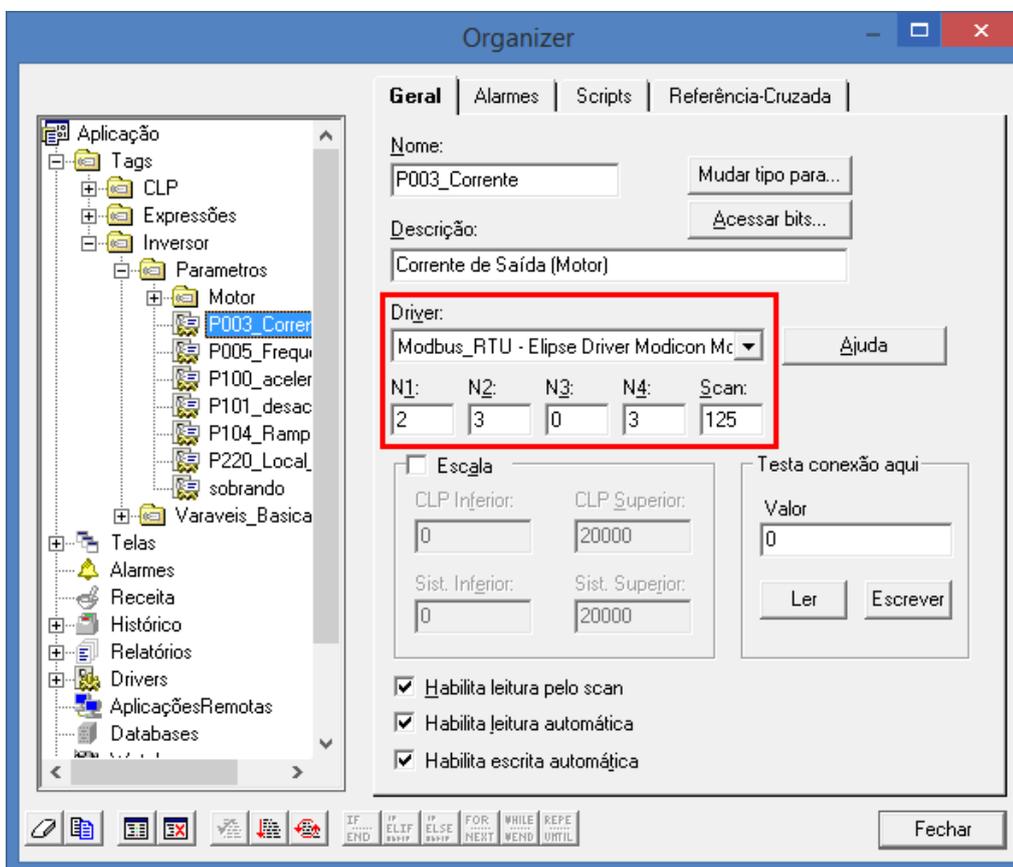


Figura 45 - Configuração de uma tag PLC para acessar o parâmetro P003.