DESENVOLVIMENTO DE APLICAÇÃO WEB PARA DIAGNÓSTICO E CONFIGURAÇÃO DE INVERSOR DE FREQUÊNCIA COM SERVIDOR HTTP EMBARCADO

Jackson Michelon Baldin, Luiz Fernando Henning, Marco Antonio Torrez Rojas
Instituto Federal de Santa Catarina
Câmpus Jaraguá do Sul – Rau – Curso de Bacharelado em Engenharia Elétrica
e-mail: jmichelonbaldin@gmail.com, luizh@ifsc.edu.br, marco.rojas@ifsc.edu.br
Trabalho de Conclusão de Curso - TCC - 10/02/2022

***Resumo* – Devido a crescente importância atribuída a dados de operação industrial, a popularização do meio físico Ethernet em redes industriais e a vasta gama de variáveis manipuladas nas rotinas de controle e proteção do equipamento, torna-se viável a utilização de pilhas TCP/IP embarcadas em inversores de frequência. Este trabalho analisa as principais aplicações envolvendo a pilha TCP/IP e o inversor. O estudo enfatiza funcionalidades que empregam os protocolos da camada de aplicação HTTP e MQTT. Uma aplicação web para acesso aos parâmetros do inversor é desenvolvida visando exemplificar uma das funcionalidades apresentadas.**

***Palavras-chave* – Inversor de frequência, pilha TCP/IP, aplicação web.**

DEVELOPMENT OF WEB APPLICATION FOR DIAGNOSIS AND CONFIGURATION OF VARIABLE FREQUENCY DRIVE WITH EMBEDDED HTTP SERVER.

***Abstract* – Due to the the increasing relevance regarding industrial operational data, the popularity of Ethernet physical layer in industrial networks and the wide range of data applied in the device's control and protection routines, the use of TCP/IP stacks in VFDs became feasible. This work go through the main applications involving the TCP/IP stack and the drive. The paper highlights functionalities related to the HTTP and MQTT application layer protocols. An web application to access VFD's parameters is developed to illustrate one of the functionalities presented.**

***Keywords* – Variable frequency drive, TCP/IP stack, web application.**

I. INTRODUÇÃO

Inversores de frequência são amplamente utilizados na indústria para o controle de velocidade de motores de indução. Esses equipamentos trazem benefícios como maior eficiência energética, redução de custos com manutenção e aumento de produtividade [1]. Com a sua popularização, ganharam recursos adicionais como diferentes tipos de controle, módulos para protocolos de comunicação e acessórios que habilitam funções de Computadores Lógicos Programáveis (CLP) no inversor, assumindo, assim, um papel de destaque em aplicações industriais. Atualmente, com a iminência da quarta revolução industrial, um dos principais desafios para os fabricantes de inversores é como utilizar dados do equipamento para aprimorar sua performance.

Para o acesso a dados do inversor e sua integração com outros equipamentos, são empregadas redes de comunicação industrial geralmente controladas por CLPs e/ou sistemas supervisórios. Por muito tempo as interfaces seriais foram o padrão para comunicação industrial com protocolos como CANopen, Modbus-RTU e PROFIBUS [2]. Atualmente, devido ao barateamento de *switches* e a possibilidade de implementar múltiplos protocolos utilizando o mesmo meio físico, a maioria dos novos módulos instalados em redes industriais já são do tipo Ethernet Industrial [3].

A popularização do meio Ethernet em redes industriais e a crescente integração de sistemas de informação a processos operacionais abriu caminho para a utilização de pilhas de protocolos TCP/IP em equipamentos industriais. Criando, assim, uma gama de funcionalidades que podem ser adicionadas a inversores que possuem essa interface integrada.

Em dispositivos industriais, a pilha TCP/IP é frequentemente utilizada para acesso remoto à máquina por meio de servidores web embarcados [4] e aplicações de Internet das Coisas (IoT) com a finalidade de trocar dados com serviços de computação em nuvem [5]. No entanto, não há literatura que aborde benefícios e especificações relacionadas ao uso da pilha em inversores de frequência. Tendo isso em vista, este trabalho visa estudar tais funcionalidades provenientes dessa pilha de protocolos no inversor de frequência. Foca-se nas soluções que envolvem o Protocolo de Transferência de Hipertexto (HTTP), utilizado na troca de dados entre páginas da web e o protocolo de *Message Queuing Telemetry Transport* (MQTT), um protocolo simples e leve muito utilizado em aplicações IoT. Com a finalidade de exemplificar essas vantagens, uma dessas funcionalidades, uma aplicação web, é implementada em um

inversor de frequência comercial.

A aplicação tem a finalidade de facilitar a configuração e diagnóstico do inversor por meio de de uma página web embarcada, a qual pode ser acessada através de um navegador web. Devido às limitações de processamento e memória do inversor, é necessário utilizar uma pilha TCP/IP adaptada para sistemas embarcados. Das diversas implementações de pilhas disponíveis, optou-se pela *Lightweight IP stack* (lwIP) por ser de código aberto e ter sido desenvolvida para microcontroladores [6]. A distribuição da pilha lwIP empregada possui uma implementação de servidor HTTP que é utilizada para desenvolvimento da aplicação web.

O restante desse trabalho é organizado da seguinte forma: na Seção II, apresenta-se a fundamentação teórica do trabalho; as principais funcionalidades e desafios para utilização de protocolos da pilha em inversores são descritas na Seção III; na Seção IV é relatado o desenvolvimento da aplicação web; e as conclusões do trabalho são detalhadas na Seção V.

II. FUNDAMENTAÇÃO TEÓRICA

A. Inversor de Frequência

O inversor de frequência é composto por circuitos com diferentes funções no equipamento. O circuito de potência, com o retificador, barramento CC, e conversor CC/CA; o circuito de controle, que gera o sinal responsável pelo chaveamento dos IGBTs do conversor; o circuito de comunicação, responsável por administrar a troca de dados entre a interface homem-máquina, meios externos (interfaces de comunicação, entradas e saídas digitais e analógicas, módulos adicionais) e as demais placas do inversor. Tendo em vista que este trabalho aborda possíveis funcionalidades para a utilização de uma pilha TCP/IP, é natural que o estudo tenha como foco a placa de comunicação e as formas de acesso a dados da mesma.

As funções de um inversor de frequência são executadas de acordo com parâmetros predefinidos alocados na Unidade Central de Processamento (CPU) [7]. Parâmetros são variáveis de diversos formatos como, *strings*, inteiros, datas, enumerações, entre outros. Através dessas variáveis o usuário pode verificar o estado e configurar o funcionamento do dispositivo. Um inversor têm duas formas principais de acesso aos parâmetros:

- Interface homem-máquina (IHM): Geralmente, são constituídas de botões de navegação, um pequeno *display* LCD e botões com funções especiais (gira/para, mudança de referência), como pode ser visto na Fig. 1. É a forma mais simples de visualizar o valor instantâneo e modificar parâmetros do inversor individualmente. É um meio prático para navegação, mas pode ser confuso, por exemplo, em processos de configuração que necessitam da alteração de diversos parâmetros em série. Inversores mais completos já possuem IHMs maiores com mais opções de navegação, mas isso gera um aumento no custo do produto.
- Interfaces de comunicação: Geralmente empregadas quando a aplicação em que o inversor está inserido possui um Controlador Lógico Programável (CLP), permitindo que o inversor seja controlado e monitorado à distância. São uma forma mais completa de manipulação dos parâmetros, pois possibilitam a integração do inversor com outros equipamentos através de linguagens de programação como *Ladder*, Diagramas de Blocos ou Texto Estruturado. Em contrapartida, para a sua configuração é necessário a utilização de computadores, além de softwares dedicados, o que torna o processo mais complexo e demorado.

Fig. 1. Interface homem-máquina do inversor de frequência CFW11 da fabricante WEG [8].

Também é possível interagir com o inversor por meio de suas saídas e entradas digitais e analógicas. No entanto, para efetivamente alterar parâmetros do inversor, os módulos I/O (*Input/Output*) precisam ser utilizados em conjunto com sistemas supervisórios ou módulos que implementem rotinas de programação de CLPs no próprio dispositivo.

B. Pilha de protocolos TCP/IP

Uma das principais vantagens de possuir uma interface Ethernet no inversor de frequência é poder usufruir da vasta gama de protocolos já utilizados em sistemas da informação. A pilha TCP/IP é a mais utilizada em computadores, sendo responsável pelo acesso à Internet. O modelo TCP/IP é composto de 4 camadas que representam etapas de processamento dos dados entre dispositivos [9]. Neste trabalho, a camada de rede é representada pelo meio físico Ethernet, a camada de Internet pelo Protocolo de Internet (IP), a camada de transporte pelo Protocolo de Controle e Transmissão (TCP) e a camada de aplicação varia de acordo com a funcionalidade implementada no inversor de frequência. Os protocolos abordados neste trabalho estão grifados em negrito na representação do modelo TCP/IP da Fig. 2.

O inversor de frequência que serve de base para o desenvolvimento da aplicação web utiliza a *Lightweight IP stack* (lwIP). Desenvolvida por Adam Dunkles no Instituto Sueco de Ciências da Computação (SICS), a lwIP tem como objetivo gerenciar as camadas intermediárias do modelo TCP/IP (camada

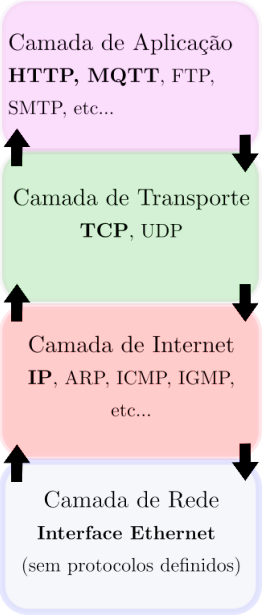


Fig. 2. Modelo TCP/IP com os protocolos de cada camada.

de Internet e transporte), deixando a camada inferior (camada de ligação de dados) para o módulo de interface de rede e a camada superior (camada de aplicação) para o desenvolvedor do produto [10]. Além das funcionalidades das camadas intermediárias a pilha lwIP oferece modelos de implementação dos protocolos de aplicação. Dentre as aplicações, vale destacar um cliente MQTT e um servidor HTTP com suporte a *Server Side Includes* (SSI) e Interface de *Gateway* Comum (CGI).

O servidor HTTP embarcado pode ser acessado com um endereço de IP fixo por meio de um navegador web. A troca de dados é feita através de requisições HTTP em um modelo servidor-cliente. A implementação da pilha lwIP utiliza os conceitos de CGI e SSI para tornar a página dinâmica e, assim, o servidor pode receber e enviar dados do microcontrolador para o navegador. SSI são diretivas inseridas dentro dos arquivos HTML que são avaliadas e substituídas no momento em que as páginas estão sendo servidas [11]. Interface de *Gateway* Comum (CGI) é um padrão de interfaceamento entre servidores de informação e aplicações externas [12]. Quando se faz uma requisição para um arquivo de script reconhecido pelo CGI, em vez de retornar o arquivo, o CGI executa o script e passa a saída para o navegador.

O cliente MQTT é responsável por coletar informações no dispostivo, conectar-se ao servidor e publicar os dados no servidor. Ele também pode se inscrever em tópicos, receber publicações e interagir com o dispositivo [13]. O protocolo MQTT é baseado em um modelo publicação/inscrição criado para dispositivos com restrições de banda larga, alta latência e redes instáveis. Graças a essas caracteristicas, esse protocolo se tornou padrão em aplicações de Internet das Coisas. Tendo em vista dispositivos como o inversor de frequência, o protocolo MQTT pode ser utilizado para que o dispositivo envie dados para um servidor de computação em nuvem. Esses dados, então, podem ser processados e novas soluções podem ser criadas para o sistema, como será abordado no desenvolvimento deste trabalho.

III. DESENVOLVIMENTO

O uso da pilha TCP/IP no inversor de frequência está intimamente relacionada ao fenômeno da indústria 4.0. Em uma era em que se busca extrair o máximo de informação com dados de operação, é esperado que o equipamento empregado para controlar a velocidade do tipo de motor mais utilizado em aplicações industriais acompanhasse essa tendência. No entanto, ainda não há um consenso sobre especificações e soluções que podem ser extraídas da integração dos inversores com sistemas da informação. Nesta seção, busca-se definir as principais funcionalidades que podem ser adicionadas ao inversor, assim como os meios e desafios da utilização dos protocolos de aplicação da camada TCP/IP. A Fig. 3 traz um resumo dos benefícios abordados.

A. MQTT

O protocolo de comunicação MQTT é utilizado, principalmente, para comunicação entre dispositivos periféricos (*edge devices*) com o propósito de transferir dados para serviços de computação em nuvem com as mais diversas finalidades. Considerando o inversor de frequência, o armazenamento e acesso remoto dos parâmetros cria um panorama do estado de funcionamento do dispositivo e do sistema em que está inserido. Além de disponibilizar o histórico de funcionamento do *drive*, os dados podem ser utilizados para auxiliar na manutenção preventiva, visando a maior disponibilidade do equipamento no chão de fábrica; podem ser alimentados em estruturas de inteligência artificial que buscam por padrões que auxiliam na tomada de decisões; e também contribuem na criação de gêmeos digitais que podem acelerar o processo de desenvolvimento de produtos.

Tratando-se de inversores frequência, a manutenção preventiva tem como principal papel evitar a falha de componentes críticos para o funcionamento do dispositivo e, assim, diminuir o tempo de máquina parada. As falhas mais comuns dos inversores são relacionadas a semicondutores de potência como MOSFETs, IGBTs e diodos com cerca de 32%, 19% são relacionados a capacitores e 16% a *gate drivers* [14]. O diagnóstico e prevenção de falhas nesses componentes pode ser realizado através do monitoramento de variáveis. A exemplo disso, o processo de envelhecimento de um capacitor eletrolítico é caracterizado por uma redução da capacitância do componente e um aumento na sua resistência equivalente em série, o que causa um aumento na temperatura dissipada e um aumento na tensão de *ripple* na saída do barramento CC [15]. Esse tipo de análise de sinais é considerada simples e pode se beneficiar do histórico de parâmetros ofertado pela nuvem. Para uma análise mais precisa, os dados também podem ser utilizados em simulações de modelos matemáticos do inversor.

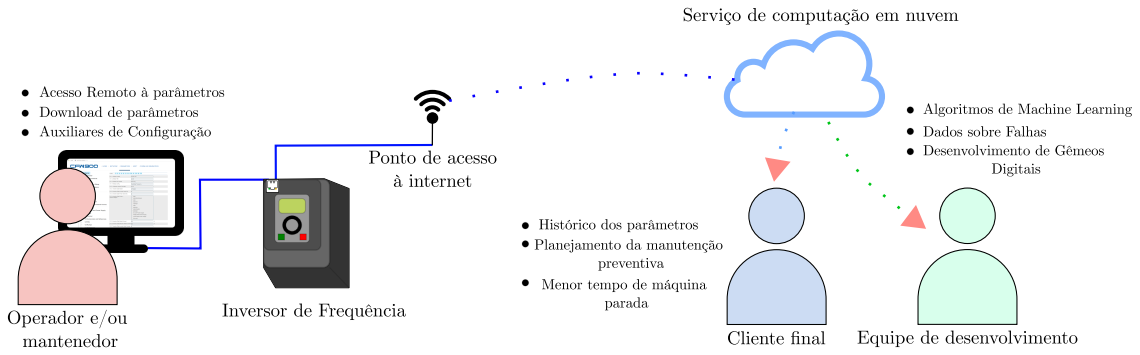


Fig. 3. Representação das aplicações da pila TCP/IP no inversor de frequência.

As análises através de sinais e modelos dos inversores já são amplamente aplicadas na área de eletrônica de potência. Atualmente, com a disponibilidade de grandes conjuntos de dados de operação armazenados em servidores com alta capacidade computacional, é possível uma terceira forma de análise que consiste na utilização dos dados para treinar modelos de *Machine Learning*. Modelos com o objetivo de identificar padrões ocultos e obter aproximações de sistemas dinâmicos complexos. Em [16], por exemplo, um histórico das correntes de fase de saída do inversor, em diversas condições de operação, é utilizado para treinar um algoritmo de diagnóstico de falhas em chaves IGBTs e sensores de corrente do inversor.

Gêmeos digitais são modelos de simulação que utilizam de dados de operação em tempo real do dispositivo para prever seu comportamento em situações específicas. Geralmente são utilizados para estimar variáveis que não são facilmente medidas. O estudo em [17], por exemplo, utiliza a corrente do estator para estimar a tendência da temperatura e resistência do rotor de um motor de indução. A vasta gama de parâmetros de operação manipulados pelo inversor de frequência pode auxiliar no desenvolvimento de gêmeos digitais das aplicações nas quais estão inseridos.

A utilização do protocolo MQTT traz alguns desafios na sua implementação. É necessário definir quais os parâmetros que o inversor deve publicar e como deve ser feito o pré-processamento desses dados no dispositivo, visando a redução no uso da banda larga e às limitações de processamento. Também é necessário garantir a segurança e integridade dos dados que são transmitidos via Internet.

Pode-se separar os dados enviados pelo inversor em: parâmetros de estado, dados intermediários, dados complementares e medições ou valores estimados pelo inversor. Parâmetros de estado são indicações de estados de subsistemas do equipamento como, por exemplo, o status de funcionamento, status de sistema de ventilação, os valores das últimas falhas ou alarmes registrados.

Entre as principais medições e estimações feitas pelo inversor que podem ser enviadas para plataformas de computação em nuvem, destacam-se: velocidade e torque do motor; tensão no barramento CC; correntes nas chaves IGBTs; potência e correntes de fase na saída do inversor; fator de potência e temperatura de semicondutores. A utilidade de cada medição depende da aplicação em que o inversor está inserido e a análise que se busca empregar.

Muitas vezes é possível detectar comportamentos anormais no inversor apenas analisando um sinal do dispositivo. No entanto, para entender a origem desse comportamento e as possíveis causas e formas de evitá-lo geralmente é necessário associar parâmetros com outros dados do inversor ou do ambiente em que está inserido. Esses dados podem ser dados intermediários como valores de sinal de controle que são utilizados em modelos de detecção de falhas ou dados complementares como sinais de sensores externos conectados as entradas analógicas ou digitais do equipamento.

É necessário enfatizar que o intervalo de aquisição e transmissão de dados empregado para a comunicação via MQTT abordada é na casa dos minutos. Um intervalo menor é inviável, pois gera um grande volume de dados, além de ser um gargalo de processamento. Logo, não é possível empregar os dados em métodos como análise espectral dos sinais, a qual requer um período de amostragem na casa dos quilohertz para atender ao teorema de *Nyquist–Shannon*.

Alguns métodos de computação em borda, ou *edge computing*, podem otimizar a performance do inversor na comunicação via MQTT. A implementação de regras, como o envio dos parâmetros de estado apenas caso tenham sido alterados desde o último envio, permite a redução da banda larga utilizada. O estudo em [18] faz uso da tendência dos inversores de frequência operarem em padrões cíclicos, ou seja, apresentam comportamento de variáveis repetidos dependendo da aplicação, assim, pode-se estipular valores de referência que ativam a aquisição dos dados apenas quando certos parâmetros ultrapassam um limiar mínimo ou um padrão conhecido. Essa técnica diminui o tempo de captura com alto período de amostragem apenas para os períodos desejados. Essa abordagem, em conjunto com métodos de *downsampling* [19] e *compressed sensing* [20] pode ser o caminho para viabilizar análises que requerem mais poder computacional.

Quanto à segurança dos dados, o inversor deve ser capaz de autenticar e criptografar as informações transmitidos de forma

que apenas o *broker* selecionado seja capaz de utilizá-las. Na área de dispositivos IoT que utilizam o MQTT é comum o uso do protocolo *Transport Layer Security* (TLS) para garantir a segurança e privacidade dos dados transmitidos. Pensando na implementação de um cliente TLS em um inversor de frequência é necessário atentar-se a capacidade de processamento limitada do dispositivo. O TLS requer uma quantidade significativa de memória RAM para gerar as chaves utilizadas na encriptação dos dados. O emprego de microcontroladores com módulos de criptografia por *hardware* (AES, SHA, RNG) melhora significativamente a performance e torna viável a utilização do protocolo. Outra especificação de projeto recomendada é a utilização de algoritmos de autenticação que fazem uso de chaves pré-compartilhadas, o que reduz o processamento necessário durante o *handshake* do TLS, já que não é necessário o compartilhamento de certificados para geração de chaves públicas [21].

B. HTTP

O HTTP funciona como um protocolo de requisição-resposta no modelo computacional cliente-servidor, sendo a base para a comunicação da *World Wide Web*. Pensando na aplicação do inversor de frequência, pode-se utilizar um servidor HTTP embarcado no dispositivo, no qual os arquivos da aplicação web desenvolvida ficam hospedados dentro do inversor e são acessados através de um navegador web em uma conexão local. A aplicação web tem como objetivo principal ser uma interface de fácil acesso a dados do inversor, sem a necessidade softwares extras, para diagnóstico e/ou configuração.

Para acessar e modificar parâmetros e configurações, geralmente, utiliza-se a Interface Homem-Máquina (IHM) embutida no inversor para acesso rápido ou softwares externos distribuídos pelos fabricantes para acesso a funcionalidades avançadas. Com a disponibilidade da pilha TCP/IP, pode-se utilizar do protocolo HTTP para disponibilizar uma página web que é um meio termo entre as duas opções citadas.

A página pode ter funcionalidades como auxiliares de configuração, interfaces de diagnóstico de sistemas de proteção, exportação dos parâmetros atuais do drive, acesso a dados internos do inversor que não são parâmetros, personalização de interfaces, entre outras possibilidades que são discutidas na seção de implementação da aplicação web.

A principal limitação da aplicação web é o armazenamento dos arquivos da página no servidor HTTP embarcado. O que também limita bibliotecas e ferramentas que podem ser utilizadas no desenvolvimento web da página. Logo, torna-se essencial o uso de ferramentas de compressão dos arquivos da aplicação.

IV. RESULTADOS

Essa seção aborda o desenvolvimento da aplicação web para o inversor de frequência a Fig. 4 mostra o *setup* utilizado no seu desenvolvimento. A lista a seguir demostra as especificações da aplicação desenvolvida:

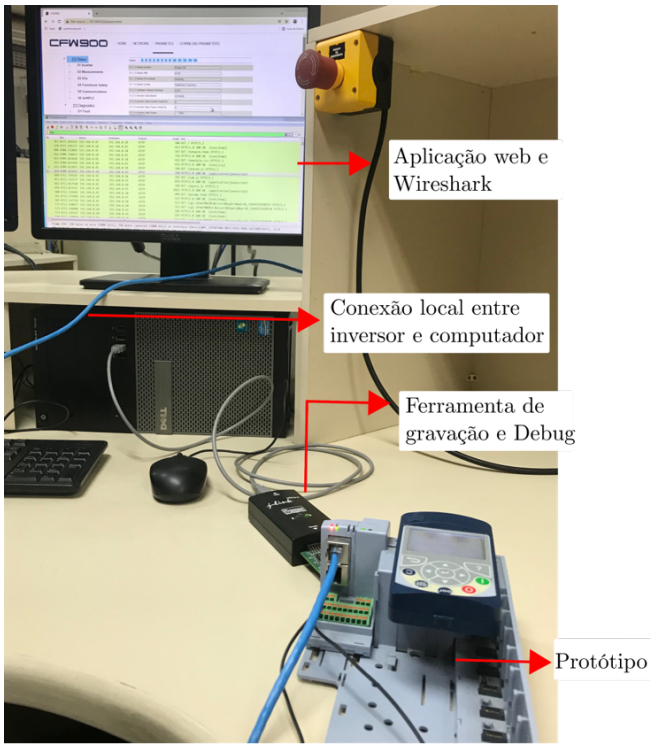


Fig. 4. *Setup* de desenvolvimento da aplicação web.

- A aplicação web é acessada utilizando um navegador web em uma conexão local entre o computador e inversor.
- O endereço de IP para acesso à aplicação é definido através de parâmetros do inversor.
- A troca de dados entre o navegador e o inversor ocorre através de requisições HTTP do tipo GET, uma limitação do uso do servidor web distribuído pela implementação do lwIP.
- A escrita e leitura de parâmetros é feita de forma assíncrona utilizando requisições AJAX. Ferramentas como CGI e SSI são utilizadas no lado do servidor para realizar tais operações.
- Devido a natureza local da conexão e limitações de *hardware* do inversor, o servidor HTTP não possui protocolo de segurança (HTTPS) ou criptografia dos dados transmitidos.

Foram desenvolvidas três páginas com diferentes funcionalidades para o usuário. A página “Home“ é um ficheiro *index* ou uma página padrão que deve ser carregada ao fazer uma requisição ao IP selecionado, tendo como propósito atestar o funcionamento da aplicação. A página “Parameters“ apresenta uma lista dos parâmetros do inversor e tem como objetivo oferecer acesso a todos os parâmetros disponíveis pela IHM. A página “Network“ é um assistente de configuração para redes industriais que busca auxiliar usuários no processo de configuração das diferentes interfaces de rede disponíveis no equipamento.

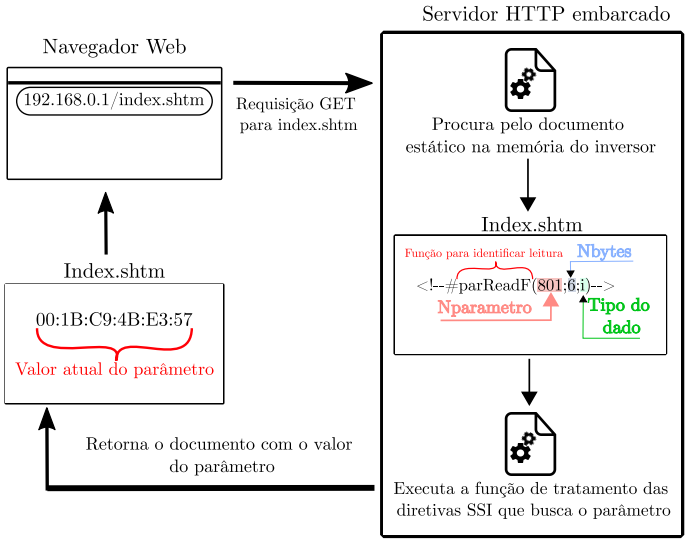


Fig. 5. Funcionamento das diretivas SSI para leitura de parâmetros.

Para desenvolver a aplicação web é necessário compreender os diferentes tipos de dados dos parâmetros do inversor, buscando entender como esses dados podem ser exibidos nas páginas e quais são as transformações necessárias para exportá-los do banco de dados em um formato compatível para visualização nos arquivos HTML.

A página *index* e a página de configuração fazem o uso de um número reduzido de parâmetros do inversor. Logo, é possível adicionar o elemento HTML de cada parâmetro manualmente. No entanto, para a página contendo a lista de todos os parâmetros se torna inviável adicioná-los um por um. Dessa forma, emprega-se o mecanismo de template em *Python* chamado *Jinja* para criação do arquivo HTML da lista de parâmetros seguindo as relações entre tipos definidos na Tabela I.

Após a definição das estruturas das páginas, é preciso estabelecer o processo de leitura e escrita de parâmetros do inversor através da aplicação. A implementação da pilha TCP/IP utilizada pelo inversor em estudo é a *Lightweight IP* (lwIP). A aplicação web desenvolvida neste trabalho faz uso de uma implementação de servidor HTTP distribuída com a pilha lwIP. Essa implementação de servidor tem suporte a SSI que são utilizados para inserir dados no lugar de *tags* marcadas em arquivos com extensões específicas como “.shtml” ou “.shtm”. Esse princípio de substituição de *tags* pode ser utilizado para a exibição de parâmetros de inversores na página. A Fig. 5 traz um exemplo do funcionamento desse sistema.

O navegador web faz a requisição de um documento HTML com a extensão .shtml para o inversor de frequência. Quando o servidor HTTP recebe a requisição ele busca esse documento na memória do inversor. Antes de enviá-lo para o navegador, ele varre o documento buscando por *tags* SSI. Ao encontrar uma *tag* SSI, os parâmetros nele contidos são passados para uma rotina que substitui a *tag* pelo valor atual do parâmetro. Essa abordagem é uma forma simples de ler os parâmetros, mas não é possível alterá-los e é necessário atualizar a página cada vez que

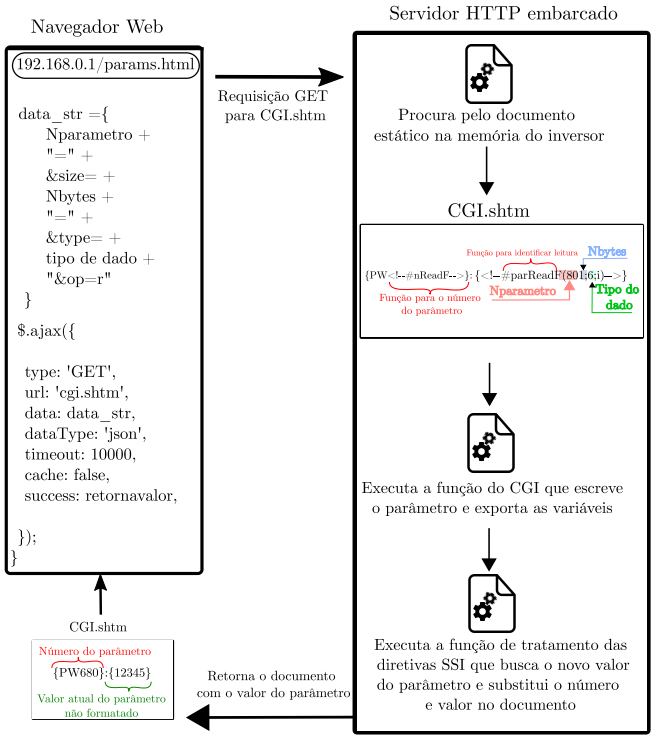


Fig. 6. Funcionamento da escrita e leitura de parâmetros através da página com requisições AJAX (cliente) e script CGI com diretivas SSI (servidor).

se busca um parâmetro.

Além do suporte a SSI, a implementação do servidor HTTP também possui uma Interface de *Gateway* Comum (CGI) que possibilita que o navegador faça requisições para scripts. Variáveis podem ser passadas para o script CGI através da URI da requisição.

Definiu-se que o processo de escrita e leitura é feito através de requisições AJAX, o que permite fazer os pedidos de forma assíncrona sem a necessidade de recarregar a página. As requisições são exclusivamente do tipo GET, uma limitação para o uso do CGI implementado pelo lwIP. Estipulou-se que todas as requisições da página devem ser feitas para o mesmo arquivo chamado “cgi.shtm” que possui duas *tags* SSI em um formato similiar a um arquivo JSON. Quando o servidor recebe a requisição ele passa o nome do arquivo e as variáveis da URI para a função de tratamento CGI. É nessa função que ocorre a escrita do novo valor do parâmetro requisitado, também são salvos em variáveis globais os parâmetros passados pela URI. Após a escrita e/ou armazenamento dos dados, o servidor deve devolver o arquivo requisitado. Como esse arquivo possui as duas *tags* SSI, elas entram na rotina de tratamento de *tags* que tem como objetivo substituir o número do parâmetro e o seu valor atual para que possa ser enviado para o navegador web. Esse processo é ilustrado na Fig. 7.

Para completar a estruturação das páginas foram utilizados Folhas de Estilo em Cascata (CSS) para o estilo da tela e *JavaScript* (JS) para a implementação de funções que

TABELA I

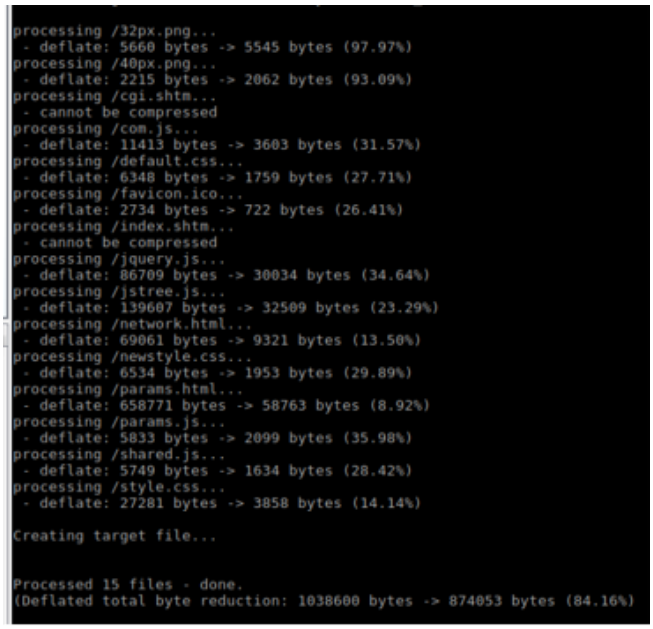
Tipos de dados do inversor e o pseudo-código utilizado para gerar suas respectivas representações com *tags* HTML.

Enumeração (Apenas leitura)	Se enumeração Se ApenasLeitura <input type="texto" id="Nparam" class="enum" disabled="disabled"> Para Nopçoes em Parametro <input type="hidden" id="Nparam_Nopcao" value="valor">
Enumeração	Senão <select size="1" id="Nparam" name="NomeParam"> Para Nopçoes em Parametro <option value="valor"></option>
Bitfield	Se Bitfield <fieldset id="Nparam">> Para Nopçoes em Parametro <input type="checkbox" id="Nparam_Nopcao" name="NomeParam" value="valor" << Se ApenasLeitura 'disabled'>>> <label for="Nparam">>
Outros dados (*Strings*, Inteiros. etc..)	Senão <input type="text" id="Nparam_Nopcao" name="NomeParam" value="valor" << Se ApenasLeitura 'disabled'>>> <input type="hidden" id="Nparam_type" value="Tipo">> <input type="hidden" id="Nparam_size" value="Tamanho">> <input type="hidden" id="Nparam_d" value="Decimal">> Se não ApenasLeitura <input type="hidden" id="Nparam_l" value="Valmin">> <input type="hidden" id="Nparam_u" value="Valmax">>

são executadas no navegador (cliente). Para facilitar no desenvolvimento da dinâmica das páginas foi utilizado a biblioteca *jQuery*, a qual fornece ferramentas que relacionam elementos HTML com funções *JavaScript*. É através de funções em *JavaScript* que alguns tipos de dados como datas e endereços de IP são formatados para exibição na página. As páginas foram desenvolvidas utilizando o recurso de *media queries* do CSS que possibilita a responsividade da página se adaptando aos diferentes formatos de tela do dispositivo em que é carregada.

Após finalizado a etapa de desenvolvimento *front-end*, notou-se que o tamanho total dos arquivos relacionados a página somavam 1038 KBs. Devido às limitações de armazenamento do inversor de frequência foi necessário utilizar uma ferramenta para reduzir ao máximo o tamanho dos arquivos web da página. Fez-se uso da ferramenta *Closure Compiler* [22] que manipula arquivos *JavaScript* retirando espaços em branco e reescrevendo funções. A aplicação da ferramenta resultou em uma redução de 15,48% no tamanho total dos arquivos relacionados à aplicação, como é possível verificar na Fig. **??**.

Para que os arquivos da aplicação possam ser embarcados no inversor é necessário convertê-los dos formatos originais para arquivos binários equivalentes. A pilha lwIP oferece um script que compila todos os arquivos em um diretório indicado para um formato que pode ser utilizado pelo servidor HTTP. Como o desenvolvimento do software embarcado do inversor é



```
processing /32px.png...
 - deflate: 5660 bytes -> 5545 bytes (97.97%)
processing /40px.png...
 - deflate: 2215 bytes -> 2062 bytes (93.09%)
processing /cgi.shtm...
 - cannot be compressed
processing /com.js...
 - deflate: 11413 bytes -> 3603 bytes (31.57%)
processing /default.css...
 - deflate: 6348 bytes -> 1759 bytes (27.71%)
processing /favicon.ico...
 - deflate: 2734 bytes -> 722 bytes (26.41%)
processing /index.shtm...
 - cannot be compressed
processing /jquery.js...
 - deflate: 86709 bytes -> 30034 bytes (34.64%)
processing /jstree.js...
 - deflate: 139607 bytes -> 32509 bytes (23.29%)
processing /network.html...
 - deflate: 69061 bytes -> 9321 bytes (13.50%)
processing /newstyle.css...
 - deflate: 6534 bytes -> 1953 bytes (29.89%)
processing /params.html...
 - deflate: 658771 bytes -> 58763 bytes (8.92%)
processing /params.js...
 - deflate: 5833 bytes -> 2099 bytes (35.98%)
processing /shared.js...
 - deflate: 5749 bytes -> 1634 bytes (28.42%)
processing /style.css...
 - deflate: 27281 bytes -> 3858 bytes (14.14%)

Creating target file...


Processed 15 files - done.
(Deflated total byte reduction: 1038600 bytes -> 874053 bytes (84.16%))
```

Fig. 7. Recorte de tela do *console* com os resultados da aplicação da ferramenta *Closure Compiler*.

feito em um ambiente Linux, parte do trabalho foi dedicado ao desenvolvimento de um *Makefile* que automatiza a exportação, geração, compressão e compilação dos arquivos da página.

Atualmente, uma quarta página está em desenvolvimento. Essa página tem como objetivo mostrar dados de status da conexão MQTT entre o inversor e a nuvem. As seguintes subseções trazem detalhes sobre a implementação das três diferentes páginas já desenvolvidas.

A. Página Home

A Fig. 8 mostra a primeira página visualizada quando o usuário acessa a aplicação pelo IP definido através dos parâmetros do inversor. A sua principal finalidade é comprovar que a conexão entre o inversor e o navegador está funcionando. Logo, determinou-se como pré-requisitos que a página deve ser simples, não deve possuir funções em *JavaScript*, já que este pode não ter suporte no navegador utilizado, e não deve depender de arquivos externos como Folhas de Estilo em Cascata ou bibliotecas. São apresentados alguns parâmetros relacionados à rede local e ao inversor. Os mesmos são carregados através de diretivas SSI como exemplificado na Fig. 5. O estilo da página é definido através de uma *tag* HTML do tipo *style*.

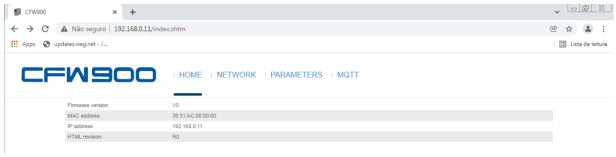


Fig. 8. Página Home da aplicação web (recortada).

B. Página da lista de parâmetros

Esta página dá acesso a todos os parâmetros disponíveis pela IHM. Eles são organizados na mesma estrutura da IHM de quatro menus. Sendo Status, Diagnósticos, Configuração e Assistentes, como visto na Fig. 9. A navegação com o menu lateral é feita usando a biblioteca *Jstree*.

Além de visualizar e alterar todos os parâmetros é possível fazer o download dos valores atuais dos parâmetros em um formato XML que posteriormente pode ser importado utilizando o software especificado pelo fabricante. Essa ferramenta pode ser útil para ajudar no diagnóstico de problemas do equipamento. Esse arquivo de exportação é criado pelo navegador através de um objeto *Blob*. Os valores dos parâmetros são requisitados individualmente através de uma rotina de chamadas AJAX.

C. Página de configurações de rede

Essa página tem como objetivo auxiliar na configuração de redes de comunicação industrial do inversor. O processo de configuração possui diversas etapas que estão sujeitas a erros de configuração pelo usuário. Além disso, é um processo desgastante devido à interface limitada da IHM. Logo, buscou-se desenvolver um assistente que possa elucidar cada etapa da configuração. A página possui subdivisões para três protocolos de rede: DeviceNet, Modbus TCP, Serial 485. A Fig. 10

Fig. 9. Página da lista de parâmetros da aplicação web.

mostra a página para Modbus TCP e Serial 485. Cada um desses protocolos possui uma forma de configuração específica. Para exemplificar a acessibilidade da página, o processo de configuração da rede DeviceNet através da página é mostrado no Anexo 1. O processo segue os seguintes passos:

1. Escolha das referências.
   (a) Selecionar o modo que alterna entre referências. O inversor em estudo possui três referências de comando: Local, Remoto 1 e Remoto 2. É preciso configurar como será alternado entre as referências remotas e configurar como será alternado entre a configuração Local e Remota.
   (b) Configurar referência para a palavra de comando. O inversor possui para cada modo de referência um parâmetro do tipo *Bitfield* no qual cada bit da palavra controla uma função comum do *drive* (gira/para, desabilita/habilita).
   (c) Configurar referência de velocidade.
   (d) (Opcional) Configurar referência de velocidade para comando JOG.
   (e) (Opcional) Configurar palavras de comando com entradas digitais.
2. Escolha de ação em caso de erro. Ação que o inversor deve tomar caso ocorra um erro de comunicação como um desligamento do mestre que controla a rede ou a rede se torne inativa.
3. Configuração de parâmetros específicos do protocolo.
4. Configuração de dados I/O. Além da palavra de comando, protocolos como DeviceNet, Anybus e Ethernet/IP possuem a possibilidade de configurar, manualmente, até

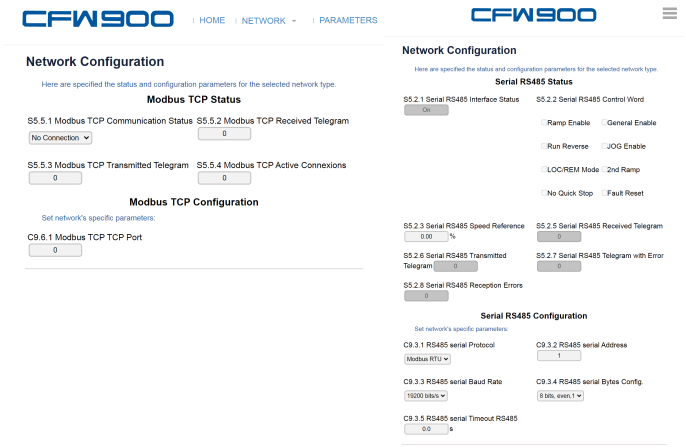


Fig. 10. Página do auxiliar de configuração das interface de rede do inversor. Na esquerda, a aba para Modbus TCP e na direita a aba para serial 485.

100 parâmetros de leitura e 100 parâmetros de escrita do inversor para acesso pela rede.

(a) Selecionar o índice da primeira palavra entre as 100 disponíveis para o protocolo selecionado. É utilizado quando para o caso de mais de um protocolo empregar os dados I/O.

(b) Selecionar a quantidade de palavras reservadas para o protocolo selecionado.

(c) Selecionar os parâmetros para as palavras de escrita e leitura desejadas.

Como é possível notar, o processo de configuração pode ser extenso e com diversas etapas. A configuração através do auxiliar facilita e elucida etapas, diminuindo as chances de erros de configuração.

Além dos textos explicativos auxiliares, a página possui funções em *JavaScript* para mostrar/esconder etapas dependendo do caminho selecionado. Para o passo de escolha dos parâmetros I/O, foi desenvolvido um mecanismo de pesquisa com a capacidade de preenchimento automático de parâmetros. Essa ferramenta foi construída utilizando a página da lista de parâmetros para fazer a busca e seleção dos nomes dos parâmetros e seus respectivos números. Dessa forma, não é necessário aumentar o tamanho dos arquivos do inversor já que a funcionalidade é gerada durante o carregamento da página pelo navegador.

V. CONCLUSÃO

Há diversas aplicações para a vasta quantidade de dados de operação manipulados por inversores de frequência. No estudo em questão, abordou-se as funcionalidades relacionadas a uma pilha TCP/IP embarcada no equipamento. A integração da pilha no inversor traz vantagens de custo e confiabilidade se comparadas com o uso de um acessório, como um *gateway* IoT com a mesma finalidade. Estima-se que os inversores devem seguir a tendência dos equipamentos industriais e ampliar as suas funções de conectividade. É necessária uma maior alocação de recursos para o circuito dedicado à comunicação, na medida em que é notável a relação entre a capacidade de processamento do microcontrolador empregado no inversor e as soluções que dele podem ser extraídas.

O inversor de frequência pode se beneficiar do protocolo MQTT para o monitoramento de parâmetros, manutenção preventiva, construção de modelos de diagnósticos de falhas utilizando algoritmos de inteligência artificial e desenvolvimento de gêmeos digitais. Os maiores desafios na implementação dessas soluções no inversor são a definição dos parâmetros que devem ser transmitidos para a nuvem, bem como o pré-processamento que deve ser feito neles antes do envio. As limitações de *hardware* impedem a aplicação de análises mais complexas que necessitam de altos períodos de amostragem. No entanto, a crescente capacidade de processamento dos microcontroladores e novos métodos de amostragem de sinal tendem a viabilizar análises mais complexas. Também é preciso pensar na segurança e integridade dos dados transmitidos, sendo necessário a implementação de protocolos de segurança. Microcontroladores com módulos de criptografia podem evitar que o protocolo de segurança afete o funcionamento do equipamento.

O protocolo HTTP pode ser utilizado principalmente para a implementação de aplicações web que podem ser acessadas através de uma rede local. A aplicação web desenvolvida neste trabalho possibilita o acesso e modificação de parâmetros do inversor, a exportação dos valores atuais dos parâmetros e a configuração de funções através de assistentes. Além de ser um local para o acesso a dados internos situacionais do inversor que não precisam se tornar parâmetros. A principal limitação da aplicação web com o servidor HTTP é a capacidade de armazenamento dos documentos utilizados nas páginas. Ferramentas de compressão e otimização de código podem auxiliar a reduzir a memória ocupada pela aplicação.

Como trabalho futuro, propõe-se o estudo das alternativas de aquisição de dados voltadas à transmissão MQTT. Outra possibilidade de estudo é a análise da implementação do protocolo da camada de aplicação *Constrained Application Protocol* (CoAP), o qual faz uso do *User Datagram Protocol* (UDP) na camada de transporte no lugar do TCP abordado neste trabalho. Esse protocolo costuma ser utilizado no lugar do MQTT para transferência de dados com um *payload* maior, como é o caso de atualizações de *firmware over-the-air* [23].

REFERÊNCIAS

[1] WEG Motores, "Guia Técnico - Motores de indução alimentados por inversores de frequência PWM", 2016, online. Disponível em: `https://static.weg.net/medias/downloadcenter/h35/h10/WEG-motores-de-inducao-alimentados-`

por-inversores-de-frequencia-pwm-50029351-brochure-portuguese-web.pdf, acesso em 03 de Agosto de 2021.

[2] F. Cavallin, "Estudo sobre redes de comunicação para automação industrial", *Universidade Tecnológica Federal do Paraná*, 2016.

[3] HMS Networks, "Industrial network market shares 2020 according to HMS Networks", 2020, online. Disponível em: https://www.hms-networks.com/news-and-insights/news-from-hms/2020/05/29/industrial-network-market-shares-2020-according-to-hms-networks, acesso em 15 de Agosto de 2021.

[4] J. M. Liu, Y. J. Zhang, L. L. Xu, "Design and Implementation of Embedded Web Server in Industrial Control Systems", *in Applied Mechanics and Materials*, vol. 380, pp. 2658–2661, Trans Tech Publ, 2013.

[5] P. Lea, *IoT and Edge Computing for Architects: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security*, Packt Publishing Ltd, 2020.

[6] L. van der Ploeg, "Small TCP/IP stacks for micro controllers", *Universite Twente*, vol. 11, no. 05, 2014.

[7] C. M. Franchi, *Inversores de frequência: teoria e aplicações*, Saraiva Educação SA, São Paulo, 2009.

[8] WEG, "Catálogo de produtos CFW11 - Dúvidas Frequentes", 2020, online. Disponível em: https://www.weg.net/catalog/weg/BR/pt/Automa%C3%A7%C3%A3o-e-Controle-Industrial/Drives/Inversores-de-Frequ%C3%AAncia/Drives-para-Sistemas-Industriais/Inversor-de-Frequ%C3%AAncia-CFW11/Inversor-de-Frequ%C3%AAncia-CFW11/p/MKT_WDC_BRAZIL_PRODUCT_INVERTER_CFW11, acesso em 20 de Agosto de 2021.

[9] K. R. Fall, W. R. Stevens, *TCP/IP illustrated, volume 1: The protocols*, addison-Wesley, 2011.

[10] A. Dunkels, "Design and Implementation of the lwIP TCP/IP Stack", *Swedish Institute of Computer Science*, vol. 2, no. 77, 2001.

[11] F. M. Pereira, *et al.*, "Protocolos TCP/IP para sistemas embarcados", *Universidade Estadual de Campinas*, 2003.

[12] D. Robinson, K. Coar, "The WWW common gateway interface version 1.1", *University of Cambridge UK*, vol. 15, 1996.

[13] Documentação IBM MQ, "MQTT clients", online. Disponível em: www.ibm.com/docs/en/ibm-mq/8.0?topic=telemetry-mqtt-clients, acesso em 20 de outubro de 2021.

[14] S. Yang, A. Bryant, P. Mawby, D. Xiang, L. Ran, P. Tavner, "An industry-based survey of reliability in power electronic converters", *IEEE transactions on Industry Applications*, vol. 47, no. 3, pp. 1441–1451, 2011.

[15] T. Kamel, Y. Biletskiy, L. Chang, "Capacitor aging detection for the DC filters in the power electronic converters using ANFIS algorithm", *in 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 663–668, IEEE, 2015.

[16] B. Gou, Y. Xu, Y. Xia, Q. Deng, X. Ge, "An Online Data-Driven Method for Simultaneous Diagnosis of IGBT and Current Sensor Fault of Three-Phase PWM Inverter in Induction Motor Drives", *IEEE Transactions on Power Electronics*, vol. 35, no. 12, pp. 13281–13294, 2020, doi: 10.1109/TPEL.2020.2994351.

[17] F. Stocco, "Digital Twin technology for induction motor drives", *Università di Padova*, 2020.

[18] M. Orkisz, M. Wnek, K. Kryczka, P. Joerg, "Variable frequency drive as a source of condition monitoring data", *in 2008 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pp. 179–183, IEEE, 2008.

[19] V. Lenoir, D. Lattard, F. Dehmas, D. Morche, A. Jerraya, "Computational load reduction by downsampling for energy-efficient digital baseband", *in 2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS)*, pp. 333–336, IEEE, 2014.

[20] E. J. Candès, M. B. Wakin, "An introduction to compressive sampling", *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[21] F.-C. Kuo, H. Tschofenig, F. Meyer, X. Fu, "Comparison Studies between Pre-Shared and Public Key Exchange Mechanisms for Transport Layer Security", pp. 1 – 6, 05 2006, doi:10.1109/INFOCOM.2006.52.

[22] Google Developers, "What is the Closure Compiler?", online. Disponível em: https://developers.google.com/closure/compiler, acessado em: 10 de novembro de 2021.

[23] A. Thantharate, C. Beard, P. Kankariya, "CoAP and MQTT Based Models to Deliver Software and Security Updates to IoT Devices over the Air", pp. 1065–1070, 2019, doi:10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00183.

ANEXO 1 - PÁGINA DE CONFIGURAÇÃO DE REDE DEVICENET

1 Commands and References

In order to set the communication network, first off, the drive references must be configured.

Select the command source that will switch between Remote 1 and Remote 2 mode, the Local mode can only be accessed, if enabled, via HMI keypad LOC/REM.

C4.1.1 Config. LOC/REM Mode Configuration [Always Local]

Enable or disable the HMI keypad LOC/REM button in charge of switching between Local mode and the current Remote mode.

C4.1.3 Config. LOC/REM Mode [Disable]

Choos the Remote 1 and Remote 2 mode command word sources:

Quick Selection: [HMI] [Set as Remote 1 Reference] [Set as Remote 2 Reference]

Or select the source for each reference individually:

1.a)

Remote 1 Command Selection	Remote 2 Command Selection
C4.2.1.1 R1 Selection General Enable [HMI]	C4.2.2.1 R2 Selection General Enable [HMI]
C4.2.1.2 R1 Selection Run/Stop [HMI Keypad I / O]	C4.2.2.2 R2 Selection Run/Stop [HMI Keypad I/O]
C4.2.1.3 R1 Selection Speed Direction [HMI Keypad SD]	C4.2.2.3 R2 Selection Speed Direction [HMI Keypad SD]
C4.2.1.4 R1 Selection JOG [Disable]	C4.2.2.4 R2 Selection JOG [Disable]
C4.3.1.2.1 Speed Ref. Source Remote 1 Mode [Keypad]	C4.3.1.2.2 Speed Ref. Source Remote 2 Mode [Keypad]

1.b)

Speed Reference

Select the Speed reference limits:

C4.3.1.1.1 Speed Ref. Range Minimum [ ] rpm

C4.3.1.1.2 Speed Ref. Range Maximum [ ] rpm

Define the speed reference value if the selected Speed Ref. Source (C4.3.1.2) is keypad.

Select the Analogic Input to be used as speed reference.

C4.3.1.3.1 Sp. Ref. HMI [ ] rpm

C4.3.1.3.2 AI Config. Speed Ref. [Inactive]

1.c)

JOG Reference

Define the speed reference for the JOG command

C4.3.2.1 JOG JOG Reference [ ] rpm

1.d)

DI Reference

To use Digital Inputs (DI) as command references, select the corresponding DI for each command:

C4.2.3.1 DI Selection General Enable [Inactive]

C4.2.3.2 DI Selection Run/Stop DI [Inactive]

C4.2.3.3 DI Selection 3-Wire Start [Inactive]

C4.2.3.4 DI Selection 3-Wire Stop [Inactive]

C4.2.3.5 DI Selection Forward [Inactive]

C4.2.3.6 DI Selection Reverse [Inactive]

C4.2.3.7 DI Selection Quick Stop [Inactive]

C4.2.3.8 DI Selection Speed Direction [Inactive]

C4.2.3.9 DI Selection JOG [Inactive]

C4.2.3.10 DI Selection Ramp Selection [Inactive]

C4.2.3.11 DI Selection Fault Reset [Inactive]

C4.2.3.12 DI Selection Disable Flying Start [Inactive]

1.e)

2 Communication Error

Configure how the drive should act in case of communication errors.

Master Offline

Select how the drive should act in a communication interruption with the network's master. If an Alarm is chosen, select what action the drive should take when the alarm is activated.

C9.1.1.1 Master Offline Mode [Off]

C9.1.1.2 Master Offline Alarm Action [Off]

Master Idle/Prog

Select how the drive should act if the network's master enter Prog/Idle mode. If an Alarm is chosen, select what action the drive should take when the alarm is activated.

C9.1.2.1 Master Idle/Prog Mode [Inactive]

C9.1.2.2 Master Idle/Prog Alarm Action [Off]

2)

3 Network Configuration

Here are specified the status and configuration parameters for the selected network type.

DeviceNet Status

Check the CAN interface current status and it's control word.

S5.7.1 CAN/CANop/DNet CAN Controller Status [Disabled]

S5.7.2 CAN/CANop/DNet Control Word

- Ramp Enable
- General Enable
- Run Reverse
- JOG Enable
- LOC/REM Mode
- 2nd Ramp
- No Quick Stop
- Fault Reset

S5.7.3 CAN/CANop/DNet Speed Reference [ ] %

S5.7.5 CAN/CANop/DNet Received Telegram [ ]

S5.7.6 CAN/CANop/DNet Transmitted Telegram [ ]

S5.7.7 CAN/CANop/DNet Bus Off Counter [ ]

S5.7.8 CAN/CANop/DNet Lost Messages [ ]

S5.7.11 CAN/CANop/DNet DNet Network Status [Offline]

S5.7.12 CAN/CANop/DNet DNet Master Status [Run]

DeviceNet Configuration

Set network's specific parameters:

C9.8.1 CAN/CANop/DNet Protocol [Disabled]

C9.8.2 CAN/CANop/DNet Address [ ]

C9.8.3 CAN/CANop/DNet Baud Rate [1 Mbps/Auto]

C9.8.4 CAN/CANop/DNet Bus Off Reset [Manual]

C9.8.5 CAN/CANop/DNet DeviceNet I/O instances [20/70 CIP Basic Speed]

3)

4 I/O Data Configuration

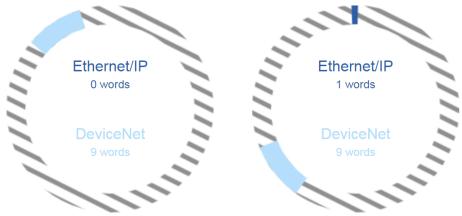
Configure network data exchange area. For Ethernet/IP, DeviceNet, Anybus and Profibus DB networks the user should set the desired read and write parameters through the I/O data words. There is a total of 100 read and 100 write words slots.

Select the first word (between 0 and 100) that will be reserved to the DeviceNet protocol in the I/O Data list.

DeviceNet Write Word 1st Word [ ] DeviceNet Read Word 1st Word [ ]

Set the number of write/read words for data communication (inputs for master), from the first word on. Take into consideration the maximum word limit and the initial word number previously chosen. Each I/O word is 16-bit so, if the desired parameter is 32 bits long, two I/O words should be reserved.

DeviceNet Write Word Quantity [ ] DeviceNet Read Word Quantity [ ]



[Configure I/O words]

4)

JACKSON MICHELON BALDIN

DESENVOLVIMENTO DE APLICAÇÃO WEB PARA DIAGNÓSTICO E CONFIGURAÇÃO DE INVERSOR DE FREQUÊNCIA COM SERVIDOR HTTP EMBARCADO

Este trabalho foi julgado adequado para obtenção do título em Bacharel em Engenharia Elétrica, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Jaraguá do Sul, 17 de fevereiro de 2022.

Prof. Dr. Luiz Fernando Henning
Orientador
IFSC – Campus Jaraguá do Sul – Rau

Prof. Dr. Marco Antonio Torrez Rojas
Co-orientador
IFSC – Campus Jaraguá do Sul – Rau

Engenheiro Guilherme Mazon Hessmann
Chefe do setor de redes industriais do departamento de desenvolvimento de drives seriados da Weg drives e automação.

Prof. MSc Edilson Hipolito da Silva
IFSC – Campus Jaraguá do Sul – Rau

#23b46c0e-a383-4686-9697-79af79d677db

Comprovante de Assinatura Eletrônica

contraktor

Datas e horários baseados no fuso horário (GMT -3:00) em Brasília, Brasil
Sincronizado com o NTP.br e Observatório Nacional (ON)
Certificado de assinatura gerado em 22/02/2022 às 11:13:15 (GMT -3:00)

documento_assinado_guiherme.pdf

ID única do documento: #23b46c0e-a383-4686-9697-79af79d677db

Hash do documento original (SHA256): 98a7d91470486c3bc8e47f0be20539b179d7729ba1cc5e5b926dcd91220b78b7

Este Log é exclusivo ao documento número #23b46c0e-a383-4686-9697-79af79d677db e deve ser considerado parte do mesmo, com os efeitos prescritos nos Termos de Uso.

Assinaturas (3)

- Luiz Fernando Henning (Participante)
  Assinou em 22/02/2022 às 11:14:02 (GMT -3:00)
- Edilson Hipolito da Silva (Participante)
  Assinou em 22/02/2022 às 14:42:31 (GMT -3:00)
- Marco Antonio Torrez Rojas (Participante)
  Assinou em 22/02/2022 às 14:11:12 (GMT -3:00)

Histórico completo

Data e hora	Evento
22/02/2022 às 11:13:23 (GMT -3:00)	Luiz Fernando Henning solicitou as assinaturas.
22/02/2022 às 11:14:02 (GMT -3:00)	Luiz Fernando Henning (Autenticação: e-mail luizh@ifsc.edu.br; IP: 191.36.54.145) assinou. Autenticidade deste documento poderá ser verificada em https://verificador.contraktor.com.br. Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.

Data e hora	Evento
22/02/2022 às 14:11:12 (GMT -3:00)	Marco Antonio Torrez Rojas (Autenticação: e-mail marco.rojas@ifsc.edu.br; IP: 191.36.54.66) assinou. Autenticidade deste documento poderá ser verificada em https://verificador.contraktor.com.br. Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.
22/02/2022 às 14:42:31 (GMT -3:00)	Edilson Hipolito da Silva (Autenticação: e-mail edilson.hipolito@ifsc.edu.br; IP: 179.223.196.202) assinou. Autenticidade deste documento poderá ser verificada em https://verificador.contraktor.com.br. Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.
22/02/2022 às 14:42:32 (GMT -3:00)	Documento assinado por todos os participantes.