



DISSERTAÇÃO DE MESTRADO

**DESENVOLVIMENTO DE UM SISTEMA DE SUPERVISÃO,
CONTROLE E MANUTENÇÃO BASEADO EM CONDIÇÃO PARA
USINAS DE GERAÇÃO SOLAR FOTOVOLTAICAS USANDO O
CONCEITO IOT**

ANTONIO BERNARDO DE VASCONCELLOS PRAXEDES

Brasília, Novembro de 2020

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

**DESENVOLVIMENTO DE UM SISTEMA DE SUPERVISÃO, CONTROLE E
MANUTENÇÃO BASEADO EM CONDIÇÃO PARA USINAS DE GERAÇÃO
SOLAR FOTOVOLTAICAS USANDO O CONCEITO IOT**

ANTONIO BERNARDO DE VASCONCELLOS PRAXEDES

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA MECÂNICA
DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM SISTEMAS MECATRÔNICOS.**

APROVADA POR:

Prof. Alberto José Álvares, Dr. Eng. (ENM-UnB)
Orientador

Prof. Li Weigang, Dr. Eng. (CIC-UnB)
Membro Interno

Prof. Rafael Amaral Shayani, Dr. Eng. (ENE-UnB)
Membro Externo

BRASÍLIA, Novembro de 2020

FICHA CATALOGRÁFICA

Antonio Bernardo de Vasconcellos Praxedes

Desenvolvimento de um Sistema de Supervisão, Controle e Manutenção Baseado em Condição para Usinas de Geração Solar Fotovoltaicas usando o Conceito IoT. Brasília DF, Brasil, 2020.

246 p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2020).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

- | | |
|-------------------------------|-----------------------------------|
| 1. Geração Solar Fotovoltaica | 2. Manutenção Baseada em Condição |
| 3. Internet das Coisas | 4. Lógica Fuzzy |
| 5. Interoperabilidade | 6. Supervisão e Controle |

RESUMO

Nos tempos atuais, onde a energia elétrica é um bem essencial para a vida humana, praticamente todo tipo de trabalho e lazer dependem dela. Assim, o consumidor investe em micro geração solar visando basicamente dois objetivos: economia na fatura e/ou energia ininterrupta (no-break solar), e espera o retorno satisfatório, que depende fundamentalmente do bom funcionamento dos equipamentos. Entretanto, a maioria dos consumidores não são técnicos e os equipamentos são complexos, resultando na fragilidade da operação e manutenção do sistema. Assim, surgiu a ideia de um sistema de baixo custo que pudesse ser instalado mesmo em pequenas usinas, mas que fosse capaz de proporcionar as funcionalidades de supervisão, controle e monitoramento de condição de forma confiável e amigável, agilizando a detecção e resolução de anomalias que porventura ocorram. Desta forma, este trabalho possui três objetivos principais: 1. Desenvolver um sistema de supervisão, controle e manutenção baseada em condição de baixo custo aplicado a pequenas usinas de geração solar fotovoltaicas; 2. Desenvolver um protocolo de comunicação padronizado para supervisão e controle, dentro do contexto da monitoração de condição, utilizando as redes IP; 3. Integrar o ambiente da usina com a Web usando os recursos do Protocolo HTTP, computação em nuvem e as orientações do Modelo de Arquitetura de Software REST.

A metodologia é baseada em: 1. Desenvolvimento e montagem de módulos eletrônicos de baixo custo para sensoriamento e controle; 2. Sistemas operacionais e ambientes de desenvolvimento de software de última geração, que possam ser obtidos sem custos; 3. Computadores e controladores de placa única de baixo custo e alta confiabilidade; 4. Recursos simples e robustos de inteligência artificial para implementar os procedimentos de manutenção baseada em condição; 5. Protocolos de comunicação dentro do conceito de Internet das Coisas (CoAP e HTTP); 6. Recursos de Computação em Nuvem de baixo custo. O sistema realiza quatro tarefas principais: 1. Monitora as condições de operação de cada equipamento; 2. Toma as medidas necessárias para proteger os equipamentos, se necessário; 3. Disponibiliza o acesso local (Intranet) e remoto (Internet) às informações da usina por clientes HTTP; 4. Permite o envio de comandos para a usina por clientes HTTP locais ou remotos.

Este sistema foi instalado em uma usina de geração fotovoltaica híbrida de 160KWh/mês, e está sendo testado e ajustado, permitindo a visualização em tempo real das informações da planta, auxiliando na verificação das condições dos equipamentos e evitando que as anomalias na operação reduzam a capacidade de geração de energia e causem danos aos equipamentos.

ABSTRACT

Nowadays, where electric energy is an essential asset for human life, practically all types of work and leisure depend on it. Thus, the consumer invests in micro solar generation basically aiming at two objectives: savings in the bill and / or uninterrupted energy (solar UPS), and expects a satisfactory return, which fundamentally depends on the proper functioning of the equipment. However, the complexity of this equipment combined with the fact that most consumers are not technicians, results in the fragility of the operation and maintenance of the system. Thus, the idea of a low-cost system that could be installed even in small plants, but capable of providing the functionalities of supervision, control and condition monitoring in a reliable and friendly way, speeding up the detection and resolution of anomalies that arose, arose. perhaps occur. Thus, this work has three main objectives: 1. To develop a system of supervision, control and maintenance based on a low cost condition applied to small photovoltaic solar generation plants; 2. Develop a standardized communication protocol for supervision and control, within the context of condition monitoring, using IP networks; 3. Integrate the plant environment with the Web using the features of the HTTP Protocol, cloud computing and the guidelines of the REST Software Architecture Model.

The methodology is based on: 1. Development and assembly of low cost electronic modules for sensing and control; 2. Use of state-of-the-art operating systems and software development environments, which can be obtained free of charge; 3. Use of low cost and high reliability computers and single board controllers; 4. Use of simple and robust artificial intelligence resources to implement condition-based maintenance procedures; 5. Use of standardized communication protocols within the concept of Internet of Things (CoAP and HTTP); 6. Use of low-cost Cloud Computing resources. The system performs four main tasks: 1. Monitors the operating conditions of each equipment; 2. Take the necessary measures to protect the equipment, if necessary; 3. Provides local (Intranet) and remote (Internet) access to plant information via HTTP clients; 4. Allows the sending of commands to the plant by local or remote HTTP clients.

This system was installed in an hybrid photovoltaic generation plant of 160KWh / month, and is being tested and adjusted, allowing real-time visualization of the plant information, helping to verify the condition of the equipment and preventing anomalies in operation, reduce the power generation capacity and cause damage to equipment.

Sumário

1 - Introdução.....	1
1.1 Definição do Problema.....	1
1.2 Motivação.....	2
1.3 Objetivos da Dissertação.....	3
1.4 Contribuições.....	4
1.5 Estrutura da Dissertação.....	6
2 - Revisão de Literatura.....	8
2.1 O Conceito de No-Break Solar.....	8
2.2 O Conceito de Internet das Coisas (IoT).....	9
2.3 Sistema de Controle Fuzzy Baseado em Regras para MBC.....	17
2.4 O Conceito de MBC e o Padrão OSA-CBM.....	18
2.5 A Irradiância na Geração Solar.....	19
2.6 Estado de Carga e Estado de Saúde das Baterias.....	20
2.7 Monitoramento em Equipamentos Comerciais.....	22
3 - Trabalhos Correlatos.....	24
3.1 Aplicações no Fornecimento de Energia Ininterrupta.....	24
3.2 Aplicações no Monitoramento de Geração Solar e IoT.....	26
3.3 Monitoramento com Controle Fuzzy baseado em Regras.....	32
4 - Metodologia.....	33
4.1 Apresentação do Cenário.....	34
4.2 Arquitetura da Usina.....	36
4.3 Recursos de Comando, Operação e Controle da Usina.....	38
4.4 Procedimentos de Controle Automático das Cargas.....	41
4.5 Arquitetura do Sistema de Supervisão, Controle e MBC.....	43
4.6 Arquitetura para Montagem do Hardware e do Software.....	46
4.7 Projeto do Inversor SPWM de um Estágio.....	49
4.8 Desenvolvimento do Software dos Controladores Arduíno.....	54
4.9 Desenvolvimento dos Softwares em Linguagem Java.....	54
4.10 Algoritmos para Detecção e Identificação de Falhas.....	55
4.11 Arquitetura de Comunicação para Supervisão e Controle.....	57
4.12 Protocolo de Comunicação para Supervisão e Controle.....	59
4.13 Metodologia de Implementação do Protocolo CoAP.....	59
4.14 Desenvolvimento dos Serviços WEB.....	61

4.15 Servidor HTTP para acesso local (Intranet).....	61
4.16 Cliente HTTP para atualização do Servidor em Nuvem.....	64
4.17 Servidor HTTP na Nuvem e Conversor.....	64
4.18 Monitoramento de Condição dos Inversores.....	66
4.19 Apresentação das Informações de MBC do Inversor.....	72
4.20 Monitoramento de Condição dos Painéis Fotovoltaicos.....	73
4.21 Monitoramento de Condição do Banco de Baterias.....	75
4.22 Monitoramento de Condição dos Controladores de Carga.....	80
5 - Implementação.....	81
5.1 Implementação das Camadas 1 e 2 do Modelo OSA-CBM.....	81
5.2 Montagem das Mensagens Padrão CoAP-OSA-CBM.....	88
5.3 Montagem das Mensagens em Protocolo HTTP.....	98
5.4 Montagem do Arquivo FCL (Fuzzy Control Logic).....	101
6 - Resultados Obtidos.....	113
6.1 Testes dos Recursos de Comunicação.....	113
6.2 Análise da Comunicação – Protocolo CoAP-OSA-CBM.....	114
6.3 Análise da Comunicação - Protocolo HTTP na Intranet.....	117
6.4 Análise usando o Protocolo HTTP para Acesso Remoto.....	120
6.5 Análise do Monitoramento de Condição dos Inversores.....	125
6.6 Análise do Monitoramento de Condição dos Painéis.....	132
6.7 Análise do Monitoramento de Condição das Baterias.....	133
7 - Conclusões.....	136
7.1 Arquitetura de Controladores da Usina.....	136
7.2 Uso do Computador Raspberry PI 3.....	138
7.3 Implementação do Protocolo CoAP.....	138
7.4 Implementação dos Recursos de Acesso Web.....	139
7.5 Recursos de Detecção de Falhas Graves.....	139
7.6 Monitoramento de Condição dos Inversores.....	139
7.7 Monitoramento de Condição dos Painéis Fotovoltaicos.....	140
7.8 Monitoramento de Condição das Baterias.....	140
7.9 Custo da Usina Solar.....	141
8 – Sugestões para Trabalhos Futuros.....	142
9 - Referências Bibliográficas.....	143
Apêndice A - Descrição dos Softwares.....	147
A.1 Considerações Gerais.....	147

A.2 Software da Unidade Terminal Remota (UTR).....	148
A.3 Software do Concentrador.....	155
A.4 Considerações sobre os Softwares Java.....	156
A.5 Softwares Servidores HTTP (Local e na Nuvem).....	156
A.6 Software Cliente HTTP de Atualização do Servidor em Nuvem.....	157
A.7 Programa Javascript.....	158
Anexo A - Arquitetura OSA-CBM.....	160
A.1 Blocos Funcionais da Arquitetura OSA-CBM.....	160
Anexo B – Hardware de Processamento.....	164
B.1 Características do Computador Raspberry PI 3.....	164
B.2 Características do Controlador Arduíno Mega 2560.....	165
Anexo C - Protocolos de Comunicação.....	167
C.1 Características Detalhadas do Protocolo CoAP.....	167
C.2 Características Gerais do Protocolo HTTP.....	186
C.3 Características do Modelo Arquitetural REST.....	193
Anexo D - Características dos Inversores.....	196
D.1 Inversores com Saída em Onda Modificada.....	196
D.2 Inversores SPWM com Saída em Onda Senoidal.....	198
Anexo E – Sistemas de Controle Fuzzy.....	208
E.1 Características Gerais.....	208
E.2 Características da Linguagem FCL.....	214
Anexo F – Tipos Básicos de Usinas Solares.....	221
F.1 Usinas Solares do tipo “On Grid”.....	221
F.2 Usinas Solares do tipo “Off Grid” e Híbridas.....	223
F.3 Características do Controlador de Carga.....	224
Anexo G - Baterias de Chumbo Ácido.....	226
G.1 Características Gerais.....	226
G.2 Cálculo da Capacidade de Carga.....	226
G.3 Informações Fornecidas pelo Fabricante.....	227

Índice de Figuras

Figura 2.1 - Exemplo de Curva de Descarga para Bateria de Chumbo Ácido Selada de 12V.....	21
Figura 4.1 - Componentes Principais da Usina de Geração Solar Híbrida.....	34
Figura 4.2 - Diagrama Elétrico da Usina Solar e Fonte Ininterrupta de Energia.....	36
Figura 4.3 - Diagrama Esquemático das Chaves de Transferência.....	38
Figura 4.4 - Diagrama IDEF0 A-0 com O Bloco Principal A0.....	44
Figura 4.5 - Diagrama A0 com os Blocos A1 (Nível de Processo) e A2 (Nível de Rede).....	44
Figura 4.6 - Diagrama A1 (Nível de Processo).....	45
Figura 4.7 - Diagrama Funcional A2 (Nível de Processo).....	45
Figura 4.8 - Diagrama de Blocos do Nível de Processo.....	48
Figura 4.9 - Diagrama de Blocos do Inversor de Um Estágio SPWM.....	51
Figura 4.10 - Diagrama de Comunicação do Sistema de Supervisão, Controle e MBC.....	57
Figura 4.11 - Fluxo de Mensagens Atualiza: Browser, Servidor Local e Concentrador.....	62
Figura 4.12 - Fluxo de Mensagens entre o Concentrador e os Dispositivos do Processo.....	63
Figura 4.13 - Fluxo de Mensagens entre o Servidor HTTP em Nuvem e os Clientes.....	65
Figura 4.14 - Painel Fotovoltaico 5W.....	74
Figura 4.15 - Esquemático do Banco de Baterias.....	75
Figura 4.16 - Exemplo de Curva de Descarga para Bateria de 12V 115Ah.....	77
Figura 4.17 - Curva Teórica de descarga do Banco de Baterias para a corrente de 10 Amperes.....	78
Figura 4.18 - Comparação da Reta de Descarga Real com a Reta de Descarga Teórica.....	79
Figura 5.1 - Função Entrada/Saída do CI Sensor de Corrente ACS713.....	85
Figura 5.2 - Especificações dos Sensores/Condicionadores das Medidas de Corrente.....	86
Figura 5.3 - Funções de Pertinência Associadas à Variável V_e = Tensão de Saída.....	103
Figura 5.4 - Funções de Pertinência Associadas à Variável W_e = Potência de Entrada.....	104
Figura 5.5 - Funções de Pertinência Associadas à Variável W_s = Potência de Saída.....	105
Figura 5.6 - Funções de Pertinência Associadas à Variável E_i = Eficiência do Inversor.....	107
Figura 5.7 - Funções de Pertinência Associadas à Variável T_d = Temp. dos MOSFETs.....	108
Figura 5.8 - Funções de Pertinência Associadas à Variável T_t = Temp. do Transformador.....	109
Figura 5.9 - Funções de Pertinência Associadas à Variável de Saída SaudeInversor.....	111
Figura 6.1 - Diagrama de Comunicação Simplificado.....	114
Figura 6.2 - Mensagem de Requisição CoAP apresentada pelo programa Wireshark.....	115
Figura 6.3 - Mensagem de Resposta CoAP-OSA-CBM do Concentrador Arduino Mega.....	116
Figura 6.4 - Análise da Resposta do Servidor Local à Requisição GET /.....	117
Figura 6.5 - Análise da Resposta do Servidor Local à Requisição GET / atualiza.....	118

Figura 6.6 - Tela do Navegador ao carregar a página de Supervisão e Controle Local.....	119
Figura 6.7 - Análise da Mensagem de Atualização do Conversor HTTP ao Servidor em Nuvem. .	120
Figura 6.8 - Resposta à Mensagem de Atualização do Servidor em Nuvem.....	121
Figura 6.9 - Análise da Resposta do Servidor em Nuvem à Requisição GET /.....	122
Figura 6.10 - Análise da Resposta do Servidor em Nuvem à Requisição GET/atualiza.....	123
Figura 6.11 - Tela do Navegador ao carregar a página de Supervisão do Servidor em Nuvem.....	124
Figura 6.12 - Onda de Tensão na Saída do Inversor SPWM.....	128
Figura 6.13 - Esquema de Ventilação Forçada dos Drivers do Inversor SPWM.....	129
Figura 6.14 - Diagrama Esquemático Simplificado da Ponte H com Falha de Conexão.....	131
Figura 6.15 - Gráfico de Descarga Real do Banco de Baterias de 10 horas.....	134
Figura 6.16 - Ilustração do Método de Estimativa do Tempo Máximo Projetado de Descarga.....	134
Figura A.1 - Blocos Funcionais do Padrão OSA-CBM.....	160
Figura B.1 - Computador Raspberry PI 3.....	165
Figura B.2 - Controlador Arduíno Mega 2560.....	166
Figura B.3 - Controlador Arduíno Mega 2560 com Shield Ethernet.....	166
Figura C.1 - Cabeçalho da Mensagem do Protocolo UDP.....	169
Figura C.2 - Fluxo de Mensagem do tipo CON: sem erro.....	183
Figura C.3 - Fluxo de Mensagem do tipo CON: com erro.....	183
Figura C.4 - Fluxo de Mensagens do tipo CON com informação não disponível.....	184
Figura C.5 - Transmissão e Recepção de Mensagem em Modo Confiável com Resposta.....	185
Figura C.6 - Requisição do tipo GET com Resposta Incorporada.....	185
Figura C.7 - Mensagens do tipo Não-Confiável (NON).....	186
Figura D.1 - Forma de Onda de Saída de um Inversor de Onda Modificada.....	197
Figura D.2 - Forma de Onda da Modulação de Largura de Pulso.....	200
Figura D.3 - Forma de Onda da Modulação de Largura de Pulso Senoidal.....	201
Figura D.4 - Diagrama de um Inversor SPWM Básico.....	202
Figura D.5 - Módulo Controlador SPWM EGS002 com o Circuito Integrado EG8010.....	203
Figura D.6 - Diagrama Simplificado de um Inversor SPWM de um Estágio.....	204
Figura D.7 - Diagrama Simplificado de um Inversor SPWM de dois Estágios.....	206
Figura E.1: Funções de Pertinência Contínuas.....	210
Figura E.2 - Diagrama do Sistema de Inferência Fuzzy.....	212
Figura E.3 - Funções de Pertinência associadas à variável Temperatura.....	216
Figura E.4 - Funções de Pertinência associadas à variável Ventilacao.....	218
Figura F.1 - Diagrama Simplificado de uma Usina Solar On Grid.....	223
Figura F.2 - Diagrama Simplificado de uma Usina Solar Off Grid.....	224
Figura G.1 - Gráfico do Número de Ciclos x Profundidade da Descarga.....	229

Índice de Tabelas

Tabela 2.1 - Principais Protocolos usados em Internet das Coisas (IoT).....	11
Tabela 2.2 - Classificação de Dispositivos com Recursos Restritos.....	13
Tabela 4.1 - Valores de Tensão para Habilitação/Desabilitação das Cargas CA.....	43
Tabela 4.2 - Recursos Funcionais do Sistema de Gerenciamento de Saúde.....	55
Tabela 5.1 - Especificações dos Sensores/Condicionadores das Medidas de Tensão CC.....	84
Tabela 5.2 - Especificações dos Sensores/Condicionadores das Medidas de Corrente CC.....	84
Tabela 5.3 - Especificações dos Sensores/Condicionadores das Medidas de Tensão CA.....	85
Tabela 5.4 - Escalas de Medidas de Tensão CA para o Arduíno.....	86
Tabela 5.5 - Especificações do CI Sensor de Temperatura LM35.....	87
Tabela 5.6 - Especificações dos Sensores/Condicionadores das Medidas de Temperatura.....	87
Tabela 5.7 - Medidas Obtidas de Cada Controlador de Carga.....	87
Tabela 5.8 - Bytes do Início da Mensagem CoAP de Requisição do Cliente para o Servidor.....	89
Tabela 5.9 - Bytes do Payload da Mensagem CoAP de Requisição do Cliente para o Servidor.....	89
Tabela 5.10 - Estrutura Geral da Mensagem Padrão OSA-CBM Binária.....	92
Tabela 5.11 - Especificação do Cabeçalho da Mensagem Padrão OSA-CBM Binária.....	93
Tabela 5.12 - Byte do Cabeçalho da Mensagem Padrão OSA-CBM Binária já Configurado.....	93
Tabela 5.13 - Campo de Data, Hora e Fuso Horário.....	94
Tabela 5.14 - Campo de Estados do Equipamento.....	94
Tabela 5.15 - Campo de 80 Estados Digitais.....	95
Tabela 5.16 - Campo de 48 Estados das Saídas Digitais.....	96
Tabela 5.17: Campo de 60 Medidas.....	97
Tabela 5.18 - Estrutura Geral da Mensagem CoAP-OSA-CBM Binária.....	98
Tabela 5.19 - Variáveis do Campo Geral da Mensagem XML.....	99
Tabela 5.20 - Variáveis do Campo AGUA da Mensagem XML.....	100
Tabela 5.21 - Variáveis do Campo GERCONS da Mensagem XML.....	100
Tabela 5.22 - Variáveis do Campo INV da Mensagem XML.....	101
Tabela 6.1 - Bytes do Payload da Mensagem de Requisição CoAP.....	116
Tabela C.1 - Formato das Mensagens do Protocolo CoAP.....	168
Tabela C.2 - Camadas do Protocolo CoAP.....	168
Tabela C.3 - Modelo de Camadas do Protocolo CoAP.....	171
Tabela C.4 - Formato Básico das Mensagens do Protocolo CoAP.....	175
Tabela C.5 - Formato do Cabeçalho das Mensagens do Protocolo CoAP.....	176
Tabela C.6 - Códigos dos Tipos de Mensagens de Solicitação.....	176
Tabela C.7 - Códigos dos Tipos de Mensagens de Resposta de Sucesso.....	177

Tabela C.8 - Códigos dos Tipos de Mensagens de Resposta com Erro de Cliente.....	177
Tabela C.9 - Códigos dos Tipos de Mensagens de Resposta com Erro de Servidor.....	177
Tabela C.10 - Formato do Campo Opção.....	179
Tabela C.11 - Tipos de Opções para as Mensagens CoAP.....	180
Tabela C.12 - Tipos de Dados definidos para a Opção 12: Content-format.....	180

Abreviaturas, Símbolos e Acrônimos

1-Wire - é um tipo de comunicação, um sistema de barramento projetado pela Dallas Semiconductor Corp. que provê dados de baixa velocidade, sinalização e sinal único de energia.

6LoWPAN - (IPv6 over Low power Wireless Personal Area Networks) é o grupo de desenvolvimento da IETF, que cria e mantêm as especificações que nos permitem usar IPv6 nas redes IEEE 802.15.4.

ABSOLAR - (Associação Brasileira de Energia Solar Fotovoltaica). Fundada em 2013, a Associação Brasileira de Energia Solar Fotovoltaica (ABSOLAR) é uma pessoa jurídica de direito privado sem fins lucrativos que congrega empresas de toda a cadeia produtiva do setor fotovoltaico (FV) com operações no Brasil.

BTLE - (Bluetooth Low Energy – Bluetooth de Baixa Energia) é uma tecnologia de comunicação sem fio projetada e comercializada pelo Bluetooth SIG (Special Interest Group) destinada a novas aplicações onde os equipamentos são alimentados por bateria.

CA – (Corrente Alternada) refere-se a um sinal elétrico de alimentação que varia o seu valor instantâneo periodicamente entre um valor máximo positivo e um valor máximo negativo em uma frequência determinada.

CBM – (Condition Based Maintenance) é um tipo de metodologia de manutenção baseado na análise das informações de sensores conectados a determinados pontos de um equipamento. Através desta análise, é possível detectar diversos tipos de degradação no funcionamento.

CC – (Corrente Contínua) refere-se a um sinal elétrico de alimentação que possui valor fixo no tempo positivo ou negativo.

CoAP - (Constrained Applications Protocol) significa Protocolo para Aplicações Restritas, que é um protocolo de comunicação para redes IP que trafega em telegramas UDP usado por pequenos controladores dentro do conceito de Internet das Coisas.

CRC - (CRC - Cyclic Redundancy Check ou Verificação Cíclica de Redundância). CRC16 e **CRC32** é um mecanismo de verificação da integridade de uma mensagem digital enviada de um equipamento para outro. O equipamento transmissor efetua o cálculo do CRC e grava os valores no

final da mensagem. O equipamento receptor efetua o cálculo do CRC da mensagem recebida e compara com o valor gravado pelo equipamento transmissor.

CSS - (Cascading Style Sheet) é uma linguagem que descreve os estilos presentes em um documento em formato HTML. Por exemplo, um estilo pode ser o tamanho e a cor de um caracter ou a cor de fundo de um determinado campo.

FCL – (Fuzzy Control Language). É uma linguagem de programação orientada para sistemas de controle fuzzy baseado em regras. As instruções desta linguagem devem ser escritas em um arquivo em formato texto, o qual deve ser lido e executado por um programa específico.

HTML – (Hypertext Markup Language). Linguagem em formato texto para tráfego de informações na Internet. Usada principalmente para montagem de telas em navegadores.

HTTP – (Hypertext Transfer Protocol) significa Protocolo de Transferência de Hipertexto. É o protocolo mais usado no tráfego de mensagens na Internet. Usa mensagens com um cabeçalho padrão em formato texto, que identifica o protocolo HTTP e seu conteúdo, seguido com conteúdo em texto ou binário. Existem diversos tipos de conteúdo texto, tais como HTML, XML e Javascript.

ICSP - (In Circuit Serial Programming) significa Interface Serial de Programação no Circuito. É uma interface serial de excelente performance que permite o tráfego de informações para controladores, inclusive carga de programas em memória interna.

IDE – (Integrated Development Environment – Ambiente de Desenvolvimento Integrado). Programa usado para desenvolvimento de software que normalmente possui recursos para edição dos arquivos fonte, compilação e configuração. Alguns possuem o recurso de carga do programa compilado no controlador, como o Arduíno IDE.

IPV4 – (Internet Protocol Version 4) é a quarta e mais difundida versão do protocolo IP que possui endereços no padrão de 32 bits (4 bytes). A visualização é feita em quatro números decimais de 0 a 255 separados por pontos, por exemplo: 192.168.0.170.

IPV6 - (Internet Protocol Version 6). é a sexta revisão do Internet Protocol que possui endereços no padrão de 128 bits.

IP – (Internet Protocol) é um endereço numérico atribuído a cada dispositivo conectado a uma rede de computadores que utiliza o Protocolo de Internet para comunicação.

IHM – (Interface Homem Máquina) = HMI (Human Machine Interface). Consiste em telas formatadas para supervisão e controle por um usuário. Estas telas podem ser geradas diretamente por um software supervisorio (SCADA) ou podem ser montadas em um navegador através dos recursos do protocolo HTTP.

Javascript – é uma linguagem de programação, carregada normalmente junto com um arquivo HTML, que permite a execução de instruções em um navegador.

JSON - (JavaScript Object Notation - Notação de Objetos JavaScript) é um formato de dados em caracteres texto usado para envio de informações na Web. O formato JSON tem sido bastante utilizado por aplicações Web devido a sua capacidade de estruturar informações de um modo mais compacto do que o formato XML.

MQTT – (MQ Telemetry Transport) significa Protocolo de Telemetria MQ. É um protocolo de comunicação para redes IP que usa conexão TCP/IP dentro do conceito de Internet das Coisas.

RS485 – é um tipo de interface elétrica diferencial usada em interfaces de comunicação serial. É usada atualmente para conexão de dispositivos microprocessados tais como medidores em geral, inversores e controladores de carga a controladores para leitura de informações e o envio de comandos.

SCADA - (Supervisory Control and Data Acquisition - Sistemas de Supervisão e Aquisição de Dados) é também denominado Software Supervisorio, o que é usado para efetuar a comunicação com equipamentos de supervisão e controle e apresentar as informações em uma Interface Homem Máquina em uma tela.

SPI - (Serial Peripheral Interface - Interface Periférica Serial) é um tipo de interface de comunicação serial síncrona usada em sistemas embarcados. A interface SPI possui quatro sinais físicos: SCLK (Serial Clock), MOSI (Master Output Slave Input), MISO (Master Input Slave Output) e SS (Slave Select).

TCP – (Transport Control Protocol) significa Protocolo de Controle de Transporte. É um protocolo amplamente usado em redes IP, o qual opera na camada de transporte do modelo OSI. As conexões TCP/IP permitem envio e recebimento confiável de vários pacotes de dados em sequência.

UDP - (User Datagram Protocol) significa Protocolo de Datagrama do Usuário. É um protocolo usado em redes IP, composto por datagramas, que são normalmente mensagens mais curtas. O protocolo UDP é orientado à máxima velocidade do fluxo de dados.

URI - (Uniform Resource Identifier) significa Identificador Uniforme de Recurso, que é uma cadeia de caracteres compacta usada para identificar ou denominar um recurso na Internet.

Vca – (Volts de Corrente Alternada) é um valor de tensão normalmente fornecido em Volts RMS (Root Mean Square).

Vcc – (Volts de Corrente Contínua) é um valor de tensão fornecido diretamente em Volts.

W5100 - é o circuito integrado controlador Ethernet 10/100 usado pelo controlador Arduíno. Este circuito integrado, composto por um único chip, possui funções embutidas IP, com todos os recursos, projetado para aplicações nas quais é necessária facilidade de integração, estabilidade e desempenho.

WiFi - marca registrada da Wi-Fi Alliance. É utilizada por produtos certificados que pertencem à classe de dispositivos de rede local sem fios (WLAN) baseados no padrão IEEE 802.11. Por causa do relacionamento íntimo com seu padrão de mesmo nome, o termo Wi-Fi é usado frequentemente como sinônimo para a tecnologia IEEE 802.11.

1 - Introdução

De acordo com as informações fornecidas pela ABSOLAR, no início de 2020, o investimento acumulado de consumidores residenciais em energia solar fotovoltaica alcançou a marca de R\$ 5,1 bilhões no Brasil com a capacidade instalada de 2,3 gigawatts. Os principais usuários são: instalações residenciais com 38,9%, empresas de comércio e serviços com 18%, os consumidores rurais com 8,7% e as indústrias com 2,7%. Os consumidores rurais tiveram um aumento expressivo nos últimos anos com um investimento total de R\$1,2 bilhão. Assim, os números mostram que, no Brasil, a capacidade de geração e o número de usinas fotovoltaicas está em crescimento acelerado.

Conseqüentemente, esta significativa expansão do número de instalações de geração solar reduziu os custos e facilitou a aquisição dos componentes básicos para a montagem das usinas. Isto viabilizou o uso destas usinas não só para economia na fatura, mas também para outras aplicações importantes como no-break solar e fornecimento de energia em locais desassistidos pela rede da concessionária.

1.1 Definição do Problema

Nos tempos atuais, onde a energia elétrica é um bem essencial para a vida humana, praticamente todo tipo de trabalho e lazer dependem dela. Assim, existem basicamente três objetivos quando se investe em uma usina solar de pequeno porte:

1. Fornecimento de energia ininterrupta (no-break solar);
2. Fornecimento de energia em localidades não atendidas pela rede da concessionária;
3. Proporcionar economia na fatura.

Após a entrada em operação da usina, o usuário espera o retorno satisfatório, que depende fundamentalmente do correto e eficiente funcionamento dos equipamentos instalados. Entretanto, a complexidade desses equipamentos, aliada ao fato de que a maioria dos consumidores não são técnicos, resulta na fragilidade da operação e manutenção do sistema. Uma usina de geração solar, por mais simples que seja, possui equipamentos muito mais complexos que uma instalação convencional, que basicamente só tem disjuntores termomagnéticos. Portanto, defeitos e degradação no funcionamento dos equipamentos são muito mais difíceis de detectar, ou seja, a capacidade de geração pode ser reduzida devido a

diversos tipos de falhas, causando a conseqüente redução da economia e das funcionalidades esperadas pelos usuários.

Historicamente, as instalações tradicionais de geração de energia elétrica são gerenciadas por empresas da área, as quais possuem pessoal especializado nas tarefas de operação e manutenção. Entretanto, a maioria das usinas solares de pequeno porte são instaladas em empresas e residências, locais que normalmente não tem disponibilidade de pessoal técnico que possa verificar o correto funcionamento dos equipamentos.

Os problemas nas usinas on-grid, que operam sempre conectadas à rede da concessionária podem passar despercebidos, porque na falta da geração solar, as cargas continuam recebendo energia normalmente. Assim, na falta de supervisão do sistema, o usuário só descobre que há algo errado quando recebe a fatura de energia. No caso das usinas fotovoltaicas instaladas em locais desassistidos pela concessionária, a ocorrência de uma falha grave pode deixar o local sem energia, fato que normalmente gera transtornos e prejuízos.

Desta forma, faz-se necessário um recurso que permita aos usuários, mesmo não técnicos, monitorarem os equipamentos da usina em tempo real, verificando, assim, as condições operacionais.

1.2 Motivação

As interrupções no fornecimento de energia elétrica nas áreas rurais normalmente são bem mais demoradas que em área urbana, o que aumenta o tempo para identificação e correção da falha. Isto ocorre em primeiro lugar por ser maior a distância a ser percorrida pela equipe de manutenção até o local, e em segundo lugar pela grande extensão das redes de distribuição. Assim, para que essas ocorrências não causem restrições ao trabalho realizado em casa e também, falhas de segurança, foi necessário a instalação de um recurso do tipo no-break solar.

Desta forma, foi elaborado o projeto elétrico, realizada a aquisição dos componentes e feita a instalação de uma usina solar híbrida. Este tipo de usina destina-se a fornecer prioritariamente energia ininterrupta. Entretanto, surgiu a necessidade de um recurso capaz de realizar a supervisão, controle e monitoramento de condição dos equipamentos.

O segundo fator importante para motivar este trabalho é a disponibilidade dos recursos de hardware e software de última geração, baixo custo e alta confiabilidade. Os equipamentos

mais difundidos atualmente em aplicações de baixo custo são o controlador Arduíno e o computador Raspberry PI (Anexo B).

O terceiro fator importante na motivação foi a ampliação do uso de sistemas de supervisão e controle do tipo Web-SCADA, onde as informações são formatadas usando o protocolo HTTP para apresentação em navegadores. Esta solução resulta em redução de custos do sistema de supervisão como um todo, já que permite que qualquer computador ou smartfone possa ser usado para visualização das informações, já que todos possuem navegadores de internet (VARGAS-SALGADO, 2019).

A Internet das Coisas é um paradigma de comunicação recente que prevê um futuro próximo, no qual os objetos da vida cotidiana serão equipados com microcontroladores, interfaces e protocolos adequados. Estes recursos os tornarão capazes de se comunicar uns com os outros e com os usuários, tornando-se parte integrante da Internet (ZANELLA, 2014). Desta forma, o uso cada vez maior de recursos dentro do conceito IoT, motivou a escolha do protocolo CoAP neste trabalho, o qual é amplamente usado em sistemas IoT (GREEN, J., OTTERDAHL, B., 2017).

1.3 Objetivos da Dissertação

1.3.1 Objetivo Geral

O desenvolvimento de um conjunto de recursos de baixo custo que permitam a supervisão, controle e monitoramento de condição dos equipamentos, que seja aplicável às usinas de geração fotovoltaicas de qualquer tamanho, e que seja aderente ao conceito de Internet das Coisas (IoT).

1.3.2 Objetivos Específicos

O primeiro objetivo deste trabalho é desenvolver os recursos e efetuar o estudo de viabilidade de um sistema de detecção de falhas graves e monitoramento de condição de baixo custo aplicado a usinas de geração solar fotovoltaicas. As ferramentas de análise devem ser eficientes e simples, permitindo a sua implementação nos controladores e computadores disponíveis no mercado, tais como o Arduíno e o Raspberry PI (Anexo B). Assim, os algoritmos usados para estimar as condições de funcionamento de cada equipamento devem ser baseados em regras se-então e no conceito de Sistema de Controle Fuzzy. Estas

estimativas devem ser calculadas em tempo real e ter a robustez necessária para monitorar as condições de funcionamento dos equipamentos prioritários.

O segundo objetivo é o desenvolvimento de um protocolo de comunicação padronizado, baseado em mensagens binárias OSA-CBM encapsuladas no payload CoAP (Anexo C). Esta ideia foi motivada pela necessidade de um recurso que fosse viável para controladores de pequeno porte como o Arduíno, e ao mesmo tempo fosse compatível com os padrões de protocolos de comunicação usados em supervisão, controle e monitoramento de condição de última geração. O uso do protocolo CoAP justifica-se pela sua fácil conversão para o protocolo HTTP, viabilizando a integração imediata com praticamente todos os sistemas da Internet.

O terceiro objetivo é o desenvolvimento de recursos de software para acesso Web do tipo Web-SCADA através da implementação do protocolo HTTP. Esses softwares devem ser instalados em um Servidor HTTP Local e um Servidor HTTP em computador virtual na nuvem. Estes servidores devem disponibilizar as informações de supervisão e permitir o envio de comandos para clientes HTTP locais e remotos.

Assim, deseja-se com este estudo mostrar a viabilidade de um recurso de comunicação de baixo custo capaz de disponibilizar as informações localmente ou remotamente através dos aplicativos em nuvem dentro do conceito de Internet das Coisas (IoT – Internet of Things). Esses recursos visam aumentar a disponibilidade e facilitar as tarefas de manutenção.

1.4 Contribuições

1.4.1 Monitoramento de Condição de Baixo Custo para Usinas Solares

Neste trabalho serão desenvolvidos recursos para monitoramento de condição dos equipamentos de uma usina solar usando componentes de baixo custo e fácil aquisição. Estes recursos foram mais focados no equipamento inversor, por sua maior vulnerabilidade às falhas. As informações de supervisão podem ser obtidas diretamente por sensores ou serem lidas por interface de comunicação no caso de equipamentos mais complexos.

No caso específico deste trabalho, os inversores usados na usina solar serão projetados e montados já com os sensores de tensão, corrente e temperatura instalados. No caso dos inversores adquiridos de terceiros, o sistema de monitoração funciona como um recurso de proteção adicional, evitando o estresse e aumentando a vida útil dos equipamentos.

1.4.2 Supervisão e Controle de Usina Solar por Web SCADA (HTTP)

O sistema desenvolvido neste projeto de pesquisa e desenvolvimento permite a supervisão e controle de qualquer tipo de equipamento, disponibilizando as informações de modo universal em protocolo HTTP, que pode ser usado por qualquer navegador de internet. A integração dos equipamentos comerciais depende da disponibilidade de protocolos abertos e interfaces padronizadas, recursos já disponibilizados por diversos fabricantes, tal como o controlador de carga usado neste trabalho.

O uso do protocolo HTTP em conjunto com os recursos de computação em nuvem, permitiram a disponibilização da supervisão e controle de uma usina solar de qualquer lugar que tenha acesso à Internet. A importância desse fato é bem reconhecida pelos profissionais da área, quando há não muito tempo era uma operação cara e complicada conectar equipamentos em uma rede de supervisão e controle. Atualmente, a contratação do serviço de um servidor básico em nuvem não custa mais que R\$ 20,00 por mês, o qual pode concentrar e disponibilizar as informações de várias usinas, proporcionando estas funcionalidades a instalações de qualquer porte devido ao seu baixo custo.

Os softwares para supervisão e controle do tipo SCADA (Supervisory Control and Data Acquisition) usados para sistemas de geração solar normalmente possuem custo de licença. Desta forma, a capacidade de supervisionar e controlar uma ou mais usinas através do recurso de Web-SCADA usando um navegador em um computador ou celular, sempre tem custo mais baixo que qualquer software supervísório. Outro ponto importante é a compatibilidade, pois, o uso de um navegador evita a instalação de softwares e aplicativos. Assim, o sistema desenvolvido neste trabalho proporciona aos usuários a independência em relação a softwares específicos de supervisão e controle, os quais normalmente tem custo associado e requerem procedimentos de instalação.

1.4.3 Protocolo CoAP com Campos de Dados Padronizados

O desenvolvimento do protocolo CoAP-OSA-CBM, feito neste trabalho, gerou um payload em padronizado, a qual é adequado à supervisão e controle de usinas solares. O payload é um campo do protocolo CoAP destinado a transportar informações genéricas em modo texto ou binário. Assim, esta área de dados, em condições normais de uso do protocolo CoAP, pode carregar qualquer informação, o que requer tratamento específico. Desta forma, a proposta de formatar esta área de dados em campos padronizados de mensagens OSA-CBM, os quais são próprios para supervisão controle e monitoramento de condição e facilitam os procedimentos

de desenvolvimento e integração de sistemas de supervisão e controle para usinas solares, principalmente as de pequeno porte.

1.5 Estrutura da Dissertação

O capítulo dois apresenta uma revisão de literatura, apresentando alguns conceitos básicos para compreensão do trabalho. Os conceitos apresentados são: Internet das Coisas, protocolos de comunicação, padrão OSA-CBM, irradiância na geração solar e estado de carga das baterias.

O capítulo três apresenta os três principais trabalhos correlatos ao tema desta dissertação. O capítulo quatro contém a metodologia concebida para desenvolvimento do sistema. O capítulo cinco apresenta os procedimentos usados para implementação das funcionalidades. O capítulo seis mostra as análises sobre os testes de funcionalidades e os resultados obtidos. O capítulo sete apresenta as conclusões sobre as análises do resultados de cada teste mostrado no capítulo seis. O capítulo oito contém as referências bibliográficas.

No Apêndice A são apresentadas as descrições dos cinco softwares desenvolvidos neste trabalho.

No Anexo A é apresentada a arquitetura OSA-CBM, padrão usado para montagem das mensagens de supervisão e controle.

No Anexo B são apresentados os principais equipamentos de processamento usados no projeto: o controlador Arduíno Mega 2560 e o computador Raspberry PI 3.

No Anexo C são apresentados os protocolos de comunicação usados no trabalho: CoAP e HTTP. O protocolo CoAP, por ser usado em novos sistemas dentro do conceito de Internet das Coisas, foi descrito de forma detalhada.

No Anexo D são apresentadas as características dos equipamentos inversores usados em sistemas de geração solar. Os inversores transformam a corrente contínua proveniente da geração solar em corrente alternada para a rede elétrica padrão dos consumidores. Estes equipamentos são os mais complexos e sujeitos a falhas, recebendo, portanto, destaque nos procedimentos de monitoração.

O Anexo E apresenta uma breve descrição dos sistemas de controle Fuzzy baseado em regras e da linguagem FCL. A linguagem FCL é usada para implementar sistemas de controle fuzzy baseado em regras.

O Anexo F mostra alguns princípios básicos da geração solar, descrevendo os tipos de usinas e equipamentos.

O Anexo G apresenta características das baterias de chumbo ácido.

2 - Revisão de Literatura

Neste capítulo serão apresentados os conceitos básicos para a compreensão deste trabalho. Estes conceitos incluem:

1. O Conceito de No Break Solar;
2. A Internet das Coisas, paradigma recente que prevê que, mesmo pequenos controladores, que podem ser incorporados em qualquer equipamento, podem se conectar à Internet;
3. Protocolos de comunicação padronizados, que permitem o acesso às informações de supervisão e controle através da Internet;
4. Sistema de Controle Baseado em Regras Fuzzy para Monitoramento de Condição;
5. O padrão OSA-CBM de mensagens e procedimentos usando o conceito de manutenção baseada em condição;
6. A definição de irradiância como recurso útil para o monitoramento de condição de painéis fotovoltaicos;
7. O estado das baterias de chumbo ácido.

2.1 O Conceito de No-Break Solar

Atualmente, o uso crescente de recursos eletrônicos no trabalho e lazer tornaram a energia elétrica um bem indispensável. Conseqüentemente, as interrupções no fornecimento causam maiores prejuízos e transtornos. Em locais já atendidos pela rede da concessionária, surge a necessidade de uma fonte de energia elétrica ininterrupta, principalmente em zonas rurais, onde o atendimento das equipes de manutenção é mais demorado. Assim, surgiu o conceito de no-break solar, onde a geração fotovoltaica é usada para complementar a energia fornecida pelo banco de baterias na falta da rede. Isto traz uma série de vantagens, principalmente a economia, porque em sistemas convencionais, a energia complementar normalmente é fornecida por geradores a gasolina ou diesel. Outra vantagem expressiva é a confiabilidade, pois, um sistema de geração solar é bem mais confiável que os geradores porque tem somente os ventiladores como parte móvel.

Na usina que é estudo de caso deste trabalho, a função principal é fornecer energia ininterrupta através da geração fotovoltaica e um banco de baterias. Durante a noite ou excepcionalmente durante o dia, quando o céu está muito nublado, as cargas são alimentadas

pela rede e as baterias mantidas em flutuação. Assim, as cargas em corrente alternada são alimentadas pelos inversores ou pela rede. Desta forma, caso o consumo seja menor que a capacidade de geração, a energia adicional não pode ser injetada na rede, pois, os inversores operam em modo isolado (off-grid). Assim, o benefício de economia na fatura não é totalmente explorado.

Venkatramanan e Vinod (2019) propõem uma solução para esta questão, onde um sistema de geração fotovoltaica on-grid, na falta de energia da rede da concessionária, altera o seu modo de operação para off-grid e injeta corrente nas baterias de um no-break. As cargas foram divididas em dois tipos: não críticas e críticas. As cargas não críticas podem ficar sem energia no caso de falta da rede da concessionária. As cargas críticas são alimentadas pelo equipamento no-break e permanecem alimentadas mesmo na falta de energia da rede. No caso, foram usados inversores especiais do tipo “dual mode” que podem operar nos modos on-grid ou off-grid.

Desta forma, o sistema apresentado neste paper permite obter o máximo de geração dos painéis fotovoltaicos em sua operação on-grid, injetando corrente na rede normalmente. Entretanto, na ocorrência de interrupção no fornecimento da concessionária durante o dia, os inversores alimentam as cargas críticas através do no-break, aumentando significativamente a disponibilidade de energia para estas cargas.

Na comparação desta solução com a usina solar, estudo de caso desta dissertação, o ponto mais significativo é a capacidade de aproveitamento total da geração fotovoltaica em sua operação on-grid, permitindo inclusive injetar corrente na rede externa caso o consumo seja baixo.

2.2 O Conceito de Internet das Coisas (IoT)

A Internet das Coisas (IoT) pode ser definida basicamente como a capacidade de comunicação com objetos cotidianos, onde estes são acessíveis globalmente a partir da Internet, e as suas consequências na forma dos programas aplicativos que tratam as informações. A disponibilidade de controladores e computadores de baixo custo, baixo consumo e dimensões reduzidas, que possuem razoável capacidade de memória e processamento, com os recursos de computação em nuvem, permitem a conexão à Internet de diversos dispositivos em diversas áreas de aplicação, como, por exemplo, sensores, atuadores, veículos, câmeras de vigilância, eletrodomésticos, etc (LEE, 2015).

Assim, o conceito ou paradigma de Internet das Coisas pode ser aplicado em diversas áreas tais como automação industrial, automação residencial, automação na agricultura (KRISHNAN et al., 2020), monitoramento de equipamentos médicos, gerenciamento inteligente de energia (GOLPIRA e BAHRAMARA 2020), gerenciamento de redes, monitoramento automotivo, gerenciamento de tráfego e muitos outros. Desta forma, as informações obtidas através da comunicação com essa ampla gama de dispositivos pela Internet, permite a elaboração de muitos programas aplicativos para fornecer novos serviços a cidadãos, empresas e administrações públicas (ZANELLA, 2014).

Rahman (2019) apresenta um sistema de iluminação pública que usa geração solar e eólica monitorado em uma intranet usando recursos da Internet das Coisas (IoT). O gerenciamento das fontes de energia e a comunicação é feito por um controlador Arduíno UNO, que possui processador de oito bits. Assim, para viabilizar o uso deste controlador nas funções de comunicação, é necessário usar protocolos mais leves, que exigem menos processamento, daí a escolha dos recursos de IoT.

Budiman (2019) propõe uma solução usando um controlador Arduíno UNO para efetuar o controle e monitoramento das chaves de transferência em um sistema de geração híbrido. A comunicação com o servidor em nuvem é feita usando o protocolo de Internet das Coisas MQTT. Neste exemplo, também os recursos de Internet das Coisas viabilizam o uso do controlador Arduíno em uma aplicação de monitoramento e controle incluindo os procedimentos de comunicação.

Telrandhe, B. (2019) apresenta um equipamento controlador de carga microprocessado com interface de rádio WiFi. Este equipamento efetua a conexão e envia para um servidor em nuvem as informações de monitoramento. O tipo de controlador usado nesta aplicação é de pequeno porte, necessitando também usar um protocolo com pouca carga de processamento para evitar sobrecarga na realização das funções principais.

2.2.1 O Conceito de Dispositivos com Recursos Restritos

Em uma perspectiva de arquitetura de alto nível, existem dois tipos de dispositivos: 1) Dispositivos Restritos; 2) Dispositivos do tipo Gateway. Uma arquitetura típica de solução de Internet das Coisas (IoT) consiste em:

1. **Dispositivos com Recursos Restritos.** São normalmente pequenos controladores conectados diretamente ao processo. Esses pequenos controladores possuem interface

de comunicação que permitem a sua conexão a equipamentos do tipo Gateway ou a equipamentos Roteadores para acesso direto à Internet;

2. **Gateways.** Esses equipamentos normalmente executam as funções de concentrador de comunicações e conversor de protocolos, efetuando a interface entre os dispositivos com recursos restritos e a rede (Internet). Nas arquiteturas típicas podem existir um ou mais equipamentos com a função de Gateway, executando funções específicas;
3. **Roteadores e Pontos de Acesso.** Esses equipamentos proporcionam a conexão dos Gateways e demais Dispositivos à Internet.
4. **Plataforma em Nuvem.** A plataforma normalmente é composta por um computador virtual fornecido como um serviço por uma empresa de provedor. Este computador virtual deve rodar um programa, normalmente um servidor HTTP, que recebe as conexões dos Dispositivos, dos Gateways e dos clientes. Além do software servidor, o computador na nuvem pode rodar programas aplicativos com funções de tratamento, formatação e apresentação dos dados.

2.2.2 Os Protocolos Mais Usados em Internet das Coisas (IoT)

Visando obter o máximo aproveitamento dos pequenos controladores, que normalmente tem restrição na capacidade de processamento e memória, nos últimos anos foram desenvolvidos alguns protocolos de comunicação específicos para aplicações em Internet das Coisas como o CoAP e MQTT. Estes protocolos fornecem compatibilidade e conectividade desses pequenos controladores com o ambiente da Internet. Dentro do conceito IoT, os dispositivos que possuem mais capacidade de processamento e memória, tal como o computador Raspberry PI, podem operar com praticamente todos os recursos do protocolo HTTP. A Tabela 2.1 apresenta os principais protocolos usados em IoT:

Tabela 2.1 - Principais Protocolos usados em Internet das Coisas (IoT)

Camada	Protocolo das Camadas Superiores
Aplicação	CoAP, MQTT, HTTP/REST, DDS
Transporte	TCP, UDP
Rede	IPV4, IPV6, IPV6 w 6LOWPAN
Enlace	Ethernet, Wireless (802.15.4, WiFi, BTLE, etc.)

O Protocolo de Aplicação Restrita (CoAP – Constrained Applications Protocol) foi desenvolvido especialmente para pequenos controladores, usando mensagens binárias pequenas encapsuladas sobre datagramas UDP. Além disso, o protocolo CoAP suporta mecanismos de segurança como DTLS (Datagram Transport Layer Security). O uso de

mensagens binárias curtas sobre datagrama UDP proporciona baixo carregamento de memória e processamento para os pequenos controladores. Entretanto, é necessário implementar os mecanismos de verificação de erros e retransmissão no caso falha na mensagem, porque o protocolo UDP não tem esses recursos (SHELBY et al. 2014).

O Protocolo MQTT (Message Queuing Telemetry Transport) é do tipo cliente servidor orientado para aplicações em controladores com recursos restritos e aplicações máquina a máquina (M2M). O protocolo MQTT usa mensagens pequenas e conexão TCP/IP, a qual garante a entrega, não necessitando, portanto, de mecanismos de verificação de integridade da mensagem na camada de aplicação (OASIS, 2014).

2.2.3 A Definição de Dispositivo com Recursos Restritos

Os chamados Dispositivos com Recursos Restritos (Constrained Devices) são normalmente pequenos controladores com restrição de memória, capacidade de processamento e disponibilidade de energia, usados como sensores, atuadores e demais dispositivos inteligentes de pequeno porte, os quais, inseridos em uma rede, formam o chamado Nó com Recursos Restritos (Constrained Node). A restrição não se limita somente às características próprias do dispositivo, mas também à largura de banda e a disponibilidade do canal de comunicação. Este último pode limitar a quantidade de dados por tempo, principalmente em topologias que usam meio sem fio e sofrem de variações nas condições de tráfego (SEITZ et al. 2016).

Esses pequenos controladores podem ser embarcados nos mais diversos objetos, desde eletrodomésticos até veículos, proporcionando a interface entre o processo e os equipamentos de comunicação de dados. Os dados do processo são medidas lidas por entradas analógicas, estados lidos por entradas digitais, ajustes (set points) controlados por saídas analógicas e dispositivos acionados por saídas digitais. Além desses pontos de sinal, existem também dispositivos conectados por interfaces de comunicação, normalmente seriais, tais como ICSP, SPI, 1-Wire, serial assíncrona com interfaces elétricas RS485, RS232, 0 a +5V.

Os dispositivos restritos, dentro de uma arquitetura IoT, são classificados conforme as suas características de acordo com a norma RFC 7228 (BORMANN et al. 2014). A Tabela 2.2 apresenta as classes definidas para esses dispositivos:

Tabela 2.2 - Classificação de Dispositivos com Recursos Restritos

Nome	Característica
Classe 0, C0	Restrição severa, usados como sensores
Classe 1, C1	Restrição alta, protocolos específicos (CoAP)
Classe 2, C2	Restrição baixa, pode-se utilizar mais de um protocolo

Os Dispositivos Classe 0 tem limitações que os impedem de conectar diretamente com a Internet. As limitações do dispositivo podem ser devidas ao tipo de interface de comunicação, de quantidade de memória e/ou capacidade de processamento. Normalmente, esses dispositivos classe 0 operam como sensores, mas podem operar também como equipamentos específicos que possuem interface de comunicação serial para monitoramento e controle. O modo de comunicação dos dispositivos Classe 0 normalmente é do tipo mestre-escravo, onde um equipamento Concentrador ou Gateway faz requisições periódicas ao dispositivo Classe 0 e espera a resposta. A interface de comunicação mais comum é do tipo serial. O programa de controle e a configuração desses dispositivos é fixa, sendo carregada em sua memória não volátil por interface específica.

Os Dispositivos Classe 1 também possuem restrições de memória e processamento, mas tem interface de comunicação que permite a sua conexão a equipamentos do tipo Gateway, normalmente através de rede do tipo Ethernet. Os dispositivos Classe 1 podem implementar protocolos mais simples em redes IP tais como o UDP, o que permite o uso de protocolo específico para IoT tal como o protocolo CoAP. O protocolo MQTT também é muito usado neste tipo de dispositivo, apesar de usar conexão TCP/IP que exige mais processamento e memória. Assim, devido às restrições de memória, processamento e capacidade da interface de comunicação, a conexão desses dispositivos na rede deve ser feita levando-se em conta essas limitações. Pode-se citar como exemplo de dispositivo Classe 1 o controlador Arduíno com placa shield Ethernet, o qual tem sido usado em diversas aplicações IoT concentrando as funções de aquisição, controle e comunicação (Anexo B).

Os Dispositivos Classe 2 possuem mais recursos de processamento, memória e interfaces de comunicação, suportando os protocolos mais usados para o tráfego de mensagens na Internet, normalmente o protocolo HTTP, além de usar também os protocolos específicos para dispositivos restritos como o CoAP e o MQTT. Atualmente, após o lançamento dos pequenos computadores completos em uma placa de baixo custo Raspberry PI e seus similares de outros fabricantes, essa classe, implementada nesse tipo de computador, pode ser usada praticamente sem restrições.

2.2.4 O Protocolo CoAP (Constrained Applications Protocol)

O padrão RFC 7252, que descreve o protocolo CoAP, é uma norma proposta pelo Internet Engineering Task Force (IETF) em junho de 2014. A sigla CoAP significa Protocolo para Aplicações Restritas, sendo muito difundido em aplicações IoT M2M (Máquina para Máquina), principalmente para sistemas de gerenciamento de energia e automação predial.

O CoAP é um protocolo com estrutura de mensagem compacta, baixa sobrecarga e procedimentos simples para troca de mensagens em modo assíncrono. As principais características do protocolo CoAP são a simplicidade e a eficiência, principalmente por usar mensagens binárias compactas e datagramas UDP para trafegar essas mensagens em redes. Desta forma, o protocolo CoAP é uma opção natural para integração de dispositivos que possuem recursos limitados de capacidade de memória e processamento (GREEN, 2017).

Portanto, o protocolo CoAP foi projetado especificamente para pequenos controladores com recursos restritos de processamento e memória, e também para redes com restrição de velocidade. As mensagens binárias são curtas, o que viabiliza o uso do protocolo UDP, mais simples e compacto que o TCP, para o tráfego nas redes IP. Este protocolo possui recursos de solicitação e resposta com verificação de erros e re-tentativas, busca de recursos e fácil compatibilidade com sistemas Web (protocolo HTTP) através de URIs e formato de dados.

O CoAP possui um subconjunto de recursos compatíveis com o HTTP otimizado para aplicações M2M. Além disso, o protocolo CoAP possui recursos de descoberta inteligente de recursos, suporte a multicast e troca de mensagens em modo assíncrono. O modo de operação do protocolo CoAP funciona basicamente através do envio, por um cliente, de uma mensagem de solicitação de execução de uma ação em um recurso presente em um servidor. A ação é identificada por um código e o recurso é identificado por um URI. Após receber a mensagem de solicitação, o servidor responde com um código e o recurso solicitado, caso esteja disponível.

Desta forma, este procedimento é semelhante ao usado pelo protocolo HTTP. As mensagens do protocolo CoAP são enviadas usando normalmente datagramas UDP, o que exige procedimentos de verificação e correção de erros, já que o protocolo UDP não possui esses recursos. Estes procedimentos são basicamente a verificação de integridade da mensagem recebida e a solicitação de novo envio de mensagem. No caso de uma solicitação não ser respondida, são definidos tempos de “timeout” padronizados, e no caso de re envio de

mensagens, o tempo é calculado usando algoritmo de back-off exponencial, o qual é semelhante ao usado nos procedimentos da rede ethernet (CSMA/CD).

O protocolo CoAP possui os mesmos métodos de acesso do protocolo HTTP (GET, PUT, POST e DELETE), o que facilita o mapeamento entre os dois protocolos. O CoAP é de uso geral, ou seja, em sua área de carga denominada Payload, pode transportar informações referentes a diversas aplicações e diversos equipamentos. Assim, neste caso foi definido um conjunto de mensagens próprias para supervisão, controle e monitoramento de condição dentro do padrão OSA-CBM.

2.2.5 Considerações sobre a Escolha do Protocolo CoAP

Em relação aos protocolos de comunicação, atualmente existe a escolha entre as opções mais difundidas de recursos dentro do conceito de Internet das Coisas que são o MQTT e o CoAP. Neste caso, foram realizados testes que comprovaram a melhor performance do protocolo CoAP no controlador usado na função de concentrador de comunicações da usina solar (Controlador Arduíno Mega 2560).

No caso do trabalho desta dissertação, o hardware e o software do Arduíno fornecem maior eficiência no uso do protocolo CoAP porque o tráfego de dados entre o controlador e o circuito integrado W5100 usa interface ICSP, que proporciona excelente velocidade neste tipo de aplicação. Neste caso, foram implementados procedimentos para verificação de erro na mensagem binária CoAP encapsulada em datagrama UDP usando o método CRC16 conforme orientado pela norma RFC7252. Assim, o Protocolo de Aplicação Restrita (CoAP) permite reduzir a sobrecarga de comunicação e alcançar baixa complexidade sem comprometer a confiabilidade nessas aplicações usando pequenos controladores como o Arduíno. As características detalhadas do protocolo CoAP estão no Anexo C.

De acordo com Green e Otterdahl (2017), o protocolo CoAP foi escolhido para integração com o protocolo CiP devido à sua simplicidade e eficiência. Isto deve-se principalmente ao tamanho das mensagens binárias e o uso do protocolo UDP para carregar as mensagens CoAP. Este paper apresenta o uso do protocolo CoAP para transporte das mensagens do protocolo CIP visando conectar dispositivos com recursos limitados. A conclusão é de que o protocolo CoAP está em conformidade com as mensagens do protocolo CIP no que se refere ao modelo de objeto e esquema de endereços.

Entretanto, existem algumas limitações no intercâmbio de informações de entrada/saída, as quais podem ser contornadas com o uso de recursos do CoAP para substituir a troca de dados cíclica do protocolo CIP. O tema deste paper assemelha-se à proposta de uso do protocolo CoAP para carregar as mensagens padrão OSA-CBM desta dissertação. O uso cada vez mais difundido de dispositivos com recursos restritos dentro do conceito de Internet das Coisas tende a demandar mais soluções de uso de protocolos específicos como o CoAP.

2.2.6 O Protocolo HTTP (Hypertext Transfer Protocol)

O Protocolo de Transferência de Hipertexto (HTTP - Hypertext Transfer Protocol) , é implementado na camada de aplicação sobre TCP/IP, sendo amplamente usado para envio e recebimento de dados em aplicações da Internet (Web). A aplicação mais comum é a conexão de um cliente rodando o programa aplicativo denominado Navegador (Browser) com um servidor trocando informações usando o Protocolo HTTP. O programa Navegador, que vem instalado em qualquer computador ou dispositivo móvel, permite a apresentação das informações enviadas pelo servidor e a entrada de dados feita pelo usuário.

Além disso, os navegadores podem executar scripts enviados pelo servidor, que são programas em linguagens específicas tal como o Javascript. Assim, o navegador realiza diversas funções de interface com o usuário que são definidas apenas pelas informações enviadas usando o protocolo HTTP. As características do protocolo HTTP estão no Anexo C.

2.2.7 O uso do Protocolo HTTP em Sistemas SCADA

Atualmente, o protocolo HTTP tem sido cada vez mais usado em aplicações SCADA (Supervisory Control and Data Acquisition – Supervisão, Controle e Aquisição de Dados). Isto deve-se ao fato de que é o protocolo mais usado nas aplicações da Internet (Web), possuindo amplo suporte, documentação e compatibilidade, o que resulta em redução acentuada de custo de implementação, fator determinante em sistemas baseados no conceito de Internet das Coisas.

Uma das funções principais de qualquer sistema SCADA é a Interface Homem Máquina (IHM), a qual permite a visualização pelo usuário das telas do supervísório e o envio de comandos. Nos programas tradicionais, esta interface é montada usando-se os recursos de impressão em tela, os quais são disponibilizados pela linguagem e pelo ambiente de desenvolvimento do software. No caso das IHM geradas em navegadores, a montagem das

telas é feita normalmente pelos recursos proporcionados pelo protocolo HTTP, tais como HTML5, CSS, Javascript, JQuery e JSON.

Aguenta e Iqbal (2019) propõem um sistema SCADA de baixo custo open source baseado em IoT. O projeto e implementação deste sistema baseia-se em recentes recursos disponíveis para sistemas baseados no conceito de Internet das Coisas, tal como a plataforma de software ThingsBoard instalada em um computador Raspberry PI 3. Esta plataforma realiza a função prioritária de proporcionar conectividade usando o protocolo MQTT, além do armazenamento de dados históricos.

As conclusões deste trabalho mostram que usando o conceito de Internet das Coisas e recursos de última geração de hardware e software, é possível ter um sistema SCADA totalmente funcional, de baixo custo e open source. Este sistema pode ser utilizado em larga escala, supervisionando diversos sistemas em qualquer local que disponibilize acesso à Internet.

Vargas-Salgado (2019) apresenta um sistema de supervisão e controle para uma usina híbrida que possui várias fontes de geração, incluindo geração eólica, biomassa e solar. A disponibilização das informações é feita através de Web-SCADA em protocolo HTTP. Neste trabalho, o sistema Web-SCADA desenvolvido foi comparado com um SCADA padrão, e foi comprovada a sua eficiência, confiabilidade e baixo custo, sendo em média 86% menor neste tipo de aplicação em microrrede, inclusive permitindo a adição de novos sistemas e dispositivos. Foi demonstrado que o sistema pode ser supervisionado por um computador local ou pela interface Web-SCADA em conjunto com um banco de dados remoto MySQL.

Este trabalho possui diversos pontos em comum com o tema desta dissertação: um sistema de baixo custo para supervisão e controle de pequenas usinas de geração usando recursos do protocolo HTTP para implementação de um Web-SCADA, controladores Arduíno e computador Raspberry PI (Anexo B) operando como dispositivo central.

2.3 Sistema de Controle Fuzzy Baseado em Regras para MBC

Em um sistema de geração solar, os inversores são os equipamentos mais complexos e sujeitos a defeitos, em primeiro lugar pela sua complexidade e em segundo lugar por estarem conectados à rede elétrica de baixa tensão. Assim, estão sujeitos a falhas causadas por fatores internos, como paradas de ventiladores e externos, como sobrecargas, curto-circuitos e surtos de tensão. Desta forma, além das informações de supervisão convencionais, fez-se necessário

um recurso para implementar o monitoramento de condição deste tipo de equipamento. Esta informação deveria ser útil tanto para usuários técnicos como para não técnicos, ou seja, que fosse capaz de fornecer um valor referente ao estado geral do equipamento. Assim, através de pesquisa em publicações, foi escolhido o Sistema de Controle Fuzzy Baseado em Regras, onde uma variável de saída fornece o valor de “saúde” do equipamento na faixa de 0 a 100%.

Neves e Duarte (2019) propõem uma metodologia para obtenção da qualidade da água usando o Sistema de Controle Fuzzy baseado em Regras. Aqui usou-se como variáveis de entrada a cor aparente, o potencial hidrogeniônico e a turbidez, e como variável de saída a potabilidade da água. As funções de pertinência são: boa, adequada e inadequada para o consumo. Esta metodologia é idêntica à usada no monitoramento de condição dos inversores, ou seja, um conjunto de variáveis de supervisão de entrada que são associadas às suas funções de pertinência, geram uma variável de saída também associada à sua função de pertinência correspondente. Um conjunto de regras fuzzy permitem tratar estas informações usando atributos não numéricos, o que facilita a tarefa.

2.4 O Conceito de MBC e o Padrão OSA-CBM

A Manutenção Centrada em Confiabilidade (MCC) é um processo usado para determinar os requisitos de manutenção de qualquer ativo físico em seu contexto operacional. Um subconjunto do MCC é a Manutenção Baseada em Condição (MBC), a qual é uma técnica de manutenção baseada em evidências de necessidade e o MCC fornece as regras de evidência. A MBC é usada para monitorar ativamente a saúde dos componentes ou do sistema, a fim de otimizar os custos operacionais e determinar as necessidades de manutenção antes que uma falha importante ocorra.

No caso da manutenção não programada ou corretiva, enfrenta-se o problema de atrasos inesperados, falta de peças de reposição e de pessoal. A manutenção preventiva pode gerar inconvenientes e desperdício, já que a intervenção em um sistema causa riscos desnecessários e também a substituição de peças em bom estado, ainda em sua fase de vida útil. Assim, a MBC procura reduzir os custos operacionais e aumentar a segurança, otimizando o intervalo entre a manutenção e minimizando o tempo de inatividade. O resultado destes procedimentos é o aumento da disponibilidade e confiabilidade geral do sistema. Em um ambiente típico de MBC, os softwares de monitoramento recebendo informações de um conjunto de sensores incorporados ao processo, determinam quando as tarefas de manutenção são necessárias com base na evidência de necessidade (GILLESPIE, 2015).

O uso de sistemas abertos é fundamental para as técnicas mais modernas de MBC. A implementação de sistemas MBC demanda a integração de diversos equipamentos, muitas vezes de diversos fornecedores, a qual se torna mais fácil através da padronização, principalmente no que se refere à comunicação entre os dispositivos. Portanto, a padronização na troca de informações resulta na melhoria da interoperabilidade e na redução de custos. Assim, algumas instituições têm trabalhado no sentido de consolidar padrões que podem ser usados por sistemas MBC. Entre essas instituições estão a International Organization for Standardization (ISO), Institute of Electronics and Electrical Engineers (IEEE) e a Society of Automotive Engineers (SAE).

Entretanto, a organização denominada Machinery Information Management Open Systems Alliance (MIMOSA) desenvolveu uma arquitetura padronizada, com o objetivo da transferência de dados, especificamente para o ambiente de MBC denominado OSA-CBM (Open Systems Architecture for Condition Based Maintenance). Desta forma, o formato definido pela OSA-CBM foi usado neste trabalho para compor a mensagem com os dados de supervisão e controle a ser carregada no campo de dados do protocolo CoAP. A descrição da arquitetura OSA-CBM está no Anexo A.

2.5 A Irradiância na Geração Solar

A radiação solar total que cai sobre uma superfície horizontal é conhecida como a radiação global de ondas curtas. O fluxo global de radiação de ondas curtas, isto é, fluxo de energia radiante por unidade de tempo, é conhecido como a irradiância, símbolo G .

A unidade normalmente usada para a Irradiância é a potência de irradiação incidente por metro quadrado. A Irradiância é dada na unidade: Watts por metro quadrado (W/m^2). A energia recebida do Sol, denominada Irradiação, é calculada através da integral do fluxo de irradiância em um determinado período de tempo. O cálculo da Irradiação normalmente é feito usando-se períodos de tempo padronizados, que são: a hora (unidade W/m^2 por hora), o dia (unidade W/m^2 por dia) e mês (unidade W/m^2 por mês) (MARKVART, 2003).

A radiação global pode ser dividida em seus dois componentes, radiação solar de incidência direta (I_d) e radiação solar de incidência difusa (I_b). A radiação solar difusa é resultante da difusão e reflexão causadas por vapor de água, partículas suspensas e poluição. Esta radiação não produz sombras definidas como a radiação direta.

A radiação solar em uma superfície horizontal é dada pela Expressão 2.1:

$$I_h = I_d + I_b \quad (2.1)$$

A constante I , é a irradiância extraterrestre do feixe solar à distância média. O valor aceito para a irradiância extraterrestre é 1367W/m^2 . Devido aos efeitos de absorção e difusão causados pela atmosfera, o valor padronizado para a irradiância na superfície da Terra é de 1000 W/m^2 . Assim, a potência máxima de saída de um painel fotovoltaico ou W_p é dada aproximadamente pela eficiência multiplicada pela área multiplicada pela irradiância padrão na superfície. Por exemplo, um painel fotovoltaico que tem área de 2m^2 e eficiência de 16%, tem capacidade para gerar uma potência máxima de $320W_p$:

2.6 Estado de Carga e Estado de Saúde das Baterias

Em sistemas de fornecimento de energia ininterrupta, ou em usinas de geração solar do tipo off-grid que não possuem conexão à rede da concessionária, é necessário um meio de armazenamento em baterias quando a energia solar é insuficiente. Isto ocorre quando a quantidade de luz solar diminui, ou pela hora do dia, ou por condições climáticas quando o céu está nublado. Os tipos de baterias de chumbo ácido mais comumente utilizadas no Brasil são as baterias sem manutenção com liga de Chumbo-Cálcio nas placas positivas (PINHO, 2014). As características das baterias de chumbo ácido estão no Anexo G.

2.6.1 O Gerenciamento do Estado das Baterias

O gerenciamento do estado das baterias tem se mostrado uma tarefa difícil devido à dificuldade de estimativa dos parâmetros denominados Estado de Carga (SOC – State of Charge), Estado de Saúde (SOH - State of Health) e Capacidade em Amperes Hora (AHC – Amper Hour Capacity). Estes parâmetros formam o conjunto mais importante referente às baterias de chumbo ácido.

O parâmetro Estado de Carga reflete basicamente a capacidade de energia que pode ser fornecida, e o estado de saúde reflete as condições da bateria referentes ao tempo de vida útil restante. Estas são informações fundamentais dentro do contexto de manutenção baseada em condição, pois a partir deles, o usuário pode tomar conhecimento de condições não satisfatórias que podem ser corrigidas, visando o aumento da vida útil do banco de baterias (MARCHILDON, 2015).

2.6.2 Os Modelos para Descrever o Funcionamento das Baterias

Basicamente existem três modelos que podem ser usados para descrever o funcionamento das baterias: o químico, o elétrico e de descarga. O modelo químico de bateria tende a ser complexo porque uma série muito grande de valores instantâneos é necessária para estabelecer o modelo. Além disso, são necessários instrumentos especializados para realizar as medições necessárias. Em relação aos modelos elétricos, embora os parâmetros elétricos da bateria possam ser modelados por métodos diferentes e de acordo com a teoria de operação proposta, todos eles são baseados em circuitos elétricos equivalentes. No entanto, a maioria dos modelos elétricos não inclui parâmetros capazes de modelar o Estado de Saúde (SOH).

O modelo mais usado para descrever o funcionamento das baterias é o chamado Modelo de Descarga, onde é feito um gráfico da variação da tensão da bateria em função do tempo para alguns valores da corrente de descarga. Os fabricantes usam este modelo para testar as baterias, submetendo-as a uma corrente de descarga constante durante um certo tempo que depende da capacidade (Ah). Se ao final do teste a bateria apresentar tensão maior que o limite mínimo permitido, então a capacidade especificada foi atendida (MARCHILDON, 2015). A Figura 2.1 apresenta um exemplo de gráfico de descarga em corrente constante para uma bateria de chumbo ácido do tipo selada de 12 Volts.:

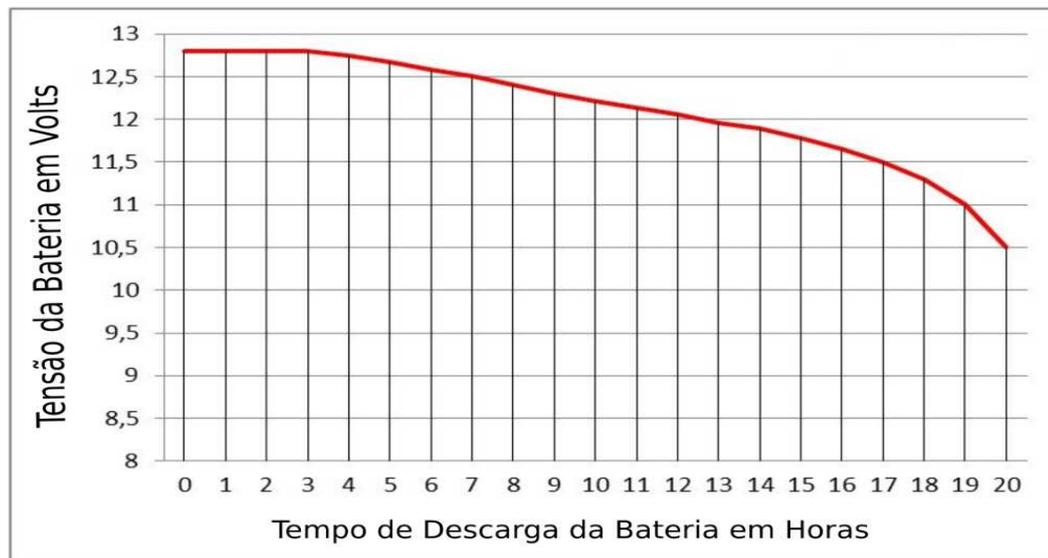


Figura 2.1 - Gráfico de Descarga de Corrente Constante para Bateria de 12V

2.7 Monitoramento em Equipamentos Comerciais

Atualmente, diversos equipamentos usados em sistema de geração solar possuem recursos de monitoramento através de aplicativo de celular ou navegador. Serão mostrados três tipos de equipamentos amplamente usados e disponíveis no mercado nacional que possuem estes recursos. A descrição destes tipos de equipamentos está no Anexo F:

1. Controladores de Carga para aplicações Off-Grid;
2. Inversores String para aplicações On-Grid;
3. Micro Inversores para aplicações On-Grid.

2.7.1 Controladores de Carga

Controladores de Carga Marca EPEVER® Modelos XTRA® e Triron®: estes modelos de controladores tem uma interface serial assíncrona padrão elétrico RS485 que permite o acesso às informações de supervisão e o ajuste de parâmetros usando protocolo MODBUS-RTU. Através do uso de cabos específicos e um módulo WiFi, é possível ao usuário efetuar a comunicação do equipamento com um programa de computador ou um aplicativo de celular para visualização das informações e o envio de comandos.

Os recursos fornecidos pelo fabricante permitem a comunicação individual com cada controlador. Dois controladores marca EPEVER modelo TRIRON 3210 foram usados na usina que é estudo de caso desta dissertação, fornecendo as informações de supervisão da geração solar para o equipamento Concentrador em interface serial assíncrona usando protocolo MODBUS-RTU.

2.7.2 Inversores String

Inversores Marca Fronius® Modelos Primo®: estes modelos de inversores trabalham com potências na faixa de 3KWa 8,2KW, possuindo interface de comunicação Ethernet. De acordo com as informações fornecidas pelo fabricante, os equipamentos possuem os seguintes recursos para monitoramento: Fronius Solar Web, protocolo MODBUS-TCP Sunspec e API Fronius Solar.

Através dos recursos disponíveis na Internet na página Fronius Solar Web® é possível ver as informações de monitoramento de um ou mais inversores da instalação usando-se um navegador (<https://www.solarweb.com/>). Existe também o aplicativo para celular. Usando-se

o protocolo MODBUS-TCP é possível integrar estes equipamentos a um sistema de monitoramento de outros fabricantes.

As Ferramentas para Monitoramento, Análise, Visualização e Apresentação dos Sistemas Fotovoltaicos são apresentadas no endereço:

<https://www.fronius.com/pt-br/brasil/energia-solar/produtos/residencial/monitoramento-de-sistema/visualização/fronius-solar-web-desktop/fronius-solar-tv>

2.7.3 Micro Inversores

Os micro inversores mais usados atualmente no Brasil são fabricados pela empresa Hoymiles® (<https://www.hoymiles.com/>). Este fabricante disponibiliza um sistema de monitoramento baseado em:

1. Um equipamento denominado DTU® (Data Transfer Unit) que deve ser instalado próximo dos micro inversores
2. O Servidor de Monitoramento disponível na Internet denominado Hoymiles Monitoring Server®.

A apresentação e a descrição dos componentes do sistema está no endereço:

<https://www.hoymiles.com/keqiao/products/monitoring>.

O funcionamento deste sistema pode ser resumido nos passos a seguir:

1. A DTU® efetua a comunicação sem fio em 2,4GHz com todos os micro inversores da instalação e faz a aquisição das informações de supervisão (a distância máxima de um micro inversor para a DTU® é 50 metros);
2. A DTU® efetua a conexão à Internet através do roteador disponível e realiza a comunicação com o Hoymiles Monitoring Server® (Servidor de Monitoramento);
3. O Hoymiles Monitoring Server® armazena e disponibiliza as informações de supervisão dos micro inversores na Internet. Estas informações podem ser acessadas através de navegadores e aplicativo de celular.

Uma característica importante deste sistema é a capacidade de monitoramento individual dos painéis fotovoltaicos, mesmo se o micro inversor é conectado a mais de um painel. Isto permite identificar redução da capacidade de geração de um painel específico, facilitando os procedimentos de manutenção.

3 - Trabalhos Correlatos

Este capítulo apresenta os trabalhos publicados mais relevantes dentro do escopo desta dissertação. Usando a plataforma CAFE CAPES e a pesquisa normal da Web, foi feita a pesquisa dos mais recentes trabalhos acadêmicos contendo os seguintes assuntos:

1. Aplicações de Geração Solar Fotovoltaica no fornecimento de energia ininterrupta;
2. Aplicações de Monitoramento de Geração Solar usando o conceito de Internet das Coisas (IoT);
3. Aplicações de Monitoramento de Geração Solar usando o conceito de Web-SCADA;
4. Uso de Sistema de Controle Fuzzy Baseado em Regras para realizar procedimentos de monitoramento de condição;
5. Aplicações usando pequenos controladores como o Arduíno e computadores em uma placa de última geração como o Raspberry PI.

3.1 Aplicações no Fornecimento de Energia Ininterrupta

3.1.1 Arquitetura com Inversor Reconfigurável On-Grid Off-Grid

O trabalho com título original: “**A Reconfigurable Solar Photovoltaic Grid-Tied Inverter Architecture for Enhanced Energy Access in Backup Power Applications**”, foi publicado no periódico IEEE Transactions on Industrial Electronics em 6 de Julho de 2019.

Venkatramanan e Vinod (2019) apresentam uma proposta onde um sistema convencional de geração solar fotovoltaica do tipo on-grid altera seu modo de operação para off-grid durante uma falta da rede da concessionária, sendo usado para complementar o fornecimento de energia do banco de baterias de um no-break. Este equipamento é usado para fornecer energia para cargas críticas.

Além disso, este sistema deve operar em conjunto e de modo transparente com os procedimentos de gerenciamento das baterias do no-break, ou seja, é independente do tipo de no break utilizado. Um dos objetivos é ter um sistema de controle que gerencie a capacidade de geração solar em conjunto com as variações de carga para reduzir ao máximo a descarga das baterias.

A questão principal aqui é compatibilizar a operação dos inversores entre os modos on-grid e off-grid para manter a compatibilidade com as normas de geração solar on-grid. No modo on-

grid os inversores somente injetam a energia gerada pelos painéis fotovoltaicos se a tensão da rede da concessionária está presente. Esta função, chamada de anti-ilhamento, é definida pela norma IEEE 1547.

A operação pretendida de complementar a energia do banco de baterias do no break necessita que os inversores operem na falta de energia da rede. Os autores propuseram o uso de inversores do tipo “multi-port dual-mode inverters” que podem operar tanto no modo on-grid como no modo off-grid. Assim, na falta de energia da rede, os inversores operando no modo off-grid, enviam energia somente para as baterias do no break, atendendo, desta forma, ao requisito anti-ilhamento.

3.1.2 Conversor CC-CC Bidirecional em Geração Solar

O trabalho com título original: “**Bidirectional DC-DC Converter for Solar Battery Backup Applications**” foi publicado no periódico *International Journal for Modern Trends in Science and Technology*. ISSN: 2455-3778 :: Volume: 06, Issue No: 02, February 2020.

Neste trabalho é proposta uma solução para fornecimento de energia ininterrupta usando um conversor CC-CC bidirecional. Este conversor, quando a geração solar é maior que a carga, opera no modo step-down (abaixador), carregando as baterias. No caso contrário, quando a geração solar é insuficiente, o conversor opera no modo step-up, transferindo energia das baterias para o circuito de alta tensão CC e ajudando a alimentar as cargas. No modo step-down, entra em operação um controlador MPPT para obtenção da máxima potência dos painéis fotovoltaicos.

3.1.3 Controlador de Carga Digital Monitorado usando IoT

O trabalho com título original: “**IoT Based Digital Charge Controller**” foi publicado no *International Research Journal of Engineering and Technology (IRJET)*, Volume: 06 Issue: 04. E -ISSN: 2395-0056. Abril de 2019.

A proposta deste paper é um equipamento controlador de carga microprocessado para sistemas de geração solar off-grid com capacidade de monitoração remota. É realizada a monitoração de duas informações básicas:

1. O modo de carga das baterias;
2. A tensão das baterias.

O envio das informações de supervisão é feito através de módulo de rádio WiFi conectado ao controlador principal usando interface serial assíncrona (UART).

Assim, o controlador pode conectar-se a uma rede WiFi disponível que tenha acesso à Internet, efetuar os procedimentos de conexão em um servidor em nuvem e enviar suas informações de supervisão.

3.2 Aplicações no Monitoramento de Geração Solar e IoT

3.2.1 Sistema SCADA Open Source de Baixo Custo baseado em IoT

O trabalho com título original: **Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol**. Foi publicado no AIMS Electronics and Electrical Engineering, 4(1): 57–86 em 12 de Dezembro de 2019.

Este trabalho apresenta o projeto e a implementação de um sistema SCADA open source de baixo custo baseado no mais recente conceito de sistemas SCADA usando recursos de Internet das Coisas. Visando testar a performance e os recursos deste SCADA, ele foi configurado para monitorar um sistema de geração solar fotovoltaica, efetuando a aquisição das medidas elétricas de tensão, corrente e potência.

O sistema SCADA proposto obtém as informações de supervisão através sensores de tensão e corrente, os quais são conectados a um microcontrolador ESP32 que recebe e processa os sinais. Este microcontrolador é conectado a um computador Raspberry PI no qual está instalada a plataforma ThingsBoard¹ em conjunto com um software de banco de dados PostgreSQL. A plataforma ThingsBoard opera como servidor IoT em protocolo MQTT, efetuando o armazenamento de dados históricos e realizando os procedimentos de tratamento das solicitações dos clientes.

O resumo das conclusões deste trabalho mostra que é possível ter um sistema SCADA de baixo custo, alta confiabilidade e com excelentes recursos de conectividade usando o conceito de Internet das Coisas, especificamente neste caso, o uso do protocolo MQTT.

1 A ThingsBoard é uma plataforma de software open source que opera no conceito de Internet das Coisas (IoT) e possui as funções de aquisição e processamento de dados, visualização e gerenciamento de dispositivos. Esta plataforma possui recursos de conectividade com os protocolos de IoT: MQTT, CoAP e HTTP (<https://thingsboard.io/>).

3.2.2 Web-SCADA, Raspberry Pi e Arduíno em Usina Híbrida

O trabalho com título original: “**Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications**”, foi publicado no periódico Heliyon 5 (2019) e02474 (www.heliyon.com), disponível no ScienceDirect, publicado por Elsevier Ltd.

Este trabalho trata do desenvolvimento e implementação de um sistema de supervisão, controle e aquisição de dados de baixo custo para uma usina de energia renovável híbrida usando o recurso de Web-SCADA. O projeto, orientado para a pesquisa sobre estabilidade de micro-redes e geração de energia, tem por objetivo fornecer recursos de supervisão, controle e informações confiáveis a baixo custo para o Laboratório de Energia Renovável da Universidade Politécnica de Valência na Espanha (LabDER-UPV). A usina híbrida possui os seguintes equipamentos principais:

1. Um conjunto de doze painéis fotovoltaicos com potência máxima total de 2,1kW;
2. Um inversor string on-grid Schneider Electric Xantrex GT. Este inversor possui interface serial assíncrona disponível em padrão elétrico RS485 e protocolo MODBUS;
3. Sistema de Geração Eólica com potência máxima de 3,5kW a 12m/s;
4. Sistema de Geração por Biomassa com potência máxima de 10 kW;
5. Banco de Baterias com capacidade de 12 kWh;
6. Gerador de Backup de 9 kW;
7. Carga de Teste de 10 kW.

Os sinais gerados pelos sensores e medidores conectados aos equipamentos da usina são aplicados em controladores Arduino, onde cada controlador efetua as medidas de tensão, corrente, temperatura e umidade relativa; efetua os cálculos de fator de potência, potência ativa, potência reativa, potência aparente, produção de energia e consumo de energia; realiza o armazenamento dos dados em módulo de memória micro SD; apresenta informações em display LCD; envia os dados para o dispositivo mestre por interface sem fio em rádio digital usando banda ISM.

Os controladores Arduíno efetuam a comunicação com um computador Raspberry PI usando interface serial conectada a rádio digital em banda ISM. O Raspberry PI 3 opera como

dispositivo mestre, concentrando todas as informações, enviando-as para um banco de dados local MySQL Web-DB vinculado a uma interface Web SCADA em formato HTML5.

3.2.3 Monitoramento de Geração Solar usando Web SCADA e IoT

O trabalho com título original: “Development of an IoT Based Open Source SCADA System for PV System Monitoring” foi apresentado na 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE) em 5-8 May 2019. ISBN:978-1-7281-0320-4. Disponível no IEEE Xplore.

O objetivo deste trabalho é o desenvolvimento de um sistema de Supervisão, Controle e Aquisição de Dados (SCADA) de baixo custo para sistemas de geração solar fotovoltaica. O sistema possui acesso do tipo Web SCADA, usa controlador Arduino para aquisição, computador Raspberry PI para análise, além de uma plataforma com recursos de Internet das Coisas em um servidor local para armazenamento dos dados, monitoramento e controle remoto.

3.2.4 Controle e Monitoramento de Sistema Híbrido de Geração

O trabalho com título original: “IoT Technology Monitoring, Controlling and Data Logging for ATS on Grid Connected Solar-Wind Hybrid System” foi publicado no IOP Conf. Series: Journal of Physics: Conf. Series 1167 012021 - 2019.

Este trabalho propõe um sistema de controle, monitoramento e registro de dados aplicado a chaves de transferência automáticas (ATS). Estas chaves têm por finalidade selecionar a fonte de energia a ser aplicada à rede, priorizando as fontes limpas e renováveis como solar e eólica em detrimento do gerador a combustível. O monitoramento do sistema é feito usando recursos de Internet das Coisas (IoT) através de meio de comunicação GPRS GSM para acesso à Internet e conexão em um servidor em nuvem.

O controlador usado para realizar os procedimentos de controle, monitoramento e comunicação é um Arduino UNO. Assim, torna-se necessário um protocolo de comunicação usando Internet das Coisas que permita a esse controlador de oito bits realizar todas as suas funções com pouca sobrecarga. No caso deste sistema, usa-se o protocolo MQTT.

O protocolo MQTT usa os recursos de conexão e transporte TCP/IP, o que traz como desvantagem a maior sobrecarga de processamento do controlador. Entretanto, o uso do TCP/IP traz uma vantagem muito importante, que é a capacidade de conexão direta do

controlador a um servidor em nuvem pela Internet. No caso do uso dos datagramas UDP para transportar as mensagens, como ocorre nas aplicações convencionais do protocolo CoAP, podem existir restrições de tráfego em algumas redes, por isso, a norma [RFC0793] aceita também o uso do TCP/IP para carregar as mensagens CoAP (Item 1.5 do Anexo C).

3.2.5 Sistema de Iluminação Pública de energia limpa Monitorado em IoT

O trabalho com título original: **IoT Based Hybrid Green Energy Driven Highway Lighting System** foi apresentado no: *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCoM/CyberSciTech)* em Fukuoka, Japão. Data da Conferência: 5-8 de Agosto de 2019. Disponível no IEEE Xplore. ISBN:978-1-7281-3025-5.

Este trabalho apresenta a proposta de um sistema para alimentação de iluminação pública em auto estradas usando geração solar e eólica em conjunto com duas baterias. Neste sistema, destinado a ser instalado em um poste de iluminação, um painel fotovoltaico de 20Wp, uma turbina vertical acoplada a um gerador trifásico de 10W e duas baterias de 12V de chumbo-ácido ligadas em série fornecem energia para a lâmpada LED. A turbina vertical tem por objetivo gerar energia a partir do deslocamento de ar causado pelos veículos.

O gerenciamento dos recursos de geração, dos procedimentos de carga das baterias e da comunicação é feito por um controlador Arduíno UNO. O monitoramento é feito através de uma intranet usando recursos dentro do conceito de Internet das Coisas (IoT).

3.2.6 IoT p/Monitoramento de Geração Solar em Aplicações Residenciais

O trabalho com título original: “**Internet of Things Towards Smart Solar Grid Systems for Home Appliances**” foi apresentado no evento International Conference on Intelligent Data Communication Technologies and Internet of Things 2019 e publicado no ICICI 2019: pp 771-778.

Este trabalho apresenta o projeto de um sistema de gerenciamento de geração solar usando o conceito de Internet das Coisas. O trabalho foi dividido em três seções principais: 1) Monitoramento em tempo real e controle da geração solar; 2) Aquisição e registro de dados de engenharia; 3) Análise de falhas.

O objetivo principal é agilizar os procedimentos de manutenção em sistemas de geração solar do tipo residencial. O sistema é acessado por celular através da Internet, proporcionando os recursos de monitoramento e controle.

3.2.7 Monitoramento de Sistema Remoto de Geração Solar usando IoT

O trabalho com título original: “**Solar Photovoltaic Remote Monitoring System Using IOT**” foi publicado no Proceeding International conference on Recent Innovations in Signal Processing and Embedded Systems (RISE -2017) 27-29 October, 2017.

Este trabalho apresenta um sistema de monitoramento de geração solar instalada em locais remotos usando controlador de baixo custo e recursos de comunicação baseados em Internet das Coisas (IoT). O controlador usado é um Arduíno UNO e a comunicação é feita através de mensagens de texto enviadas e recebidas através de um módulo SIM 900 GSM/GPRS 900/1800MHz.

No projeto, os sensores de temperatura, tensão e corrente foram montados usando módulos próprios para o Arduíno, tal como o sensor de corrente por efeito Hall e o sensor de temperatura baseado no circuito integrado LM35. Assim, através da conexão à Internet através do modem GSM/GPRS, é possível enviar e receber mensagens do controlador Arduíno remotamente, viabilizando o monitoramento da estação remota de geração solar.

3.2.8 Monitoramento de Sistema de Geração Solar usando IoT

O trabalho com título original: “**Solar Energy Monitoring System using IoT**” foi publicado no Indian Journal of Scientific Research, 2017, p.149 (7). ISSN: 2250-0138 (Online).

Este trabalho propõe um sistema de monitoramento de geração solar baseado nos seguintes componentes principais: 1) Um controlador Arduíno UNO na função de aquisição de dados; 2) Um computador Raspberry PI concentrando os dados e enviando as informações para um servidor em nuvem.

3.2.9 Integração do Protocolo CoAP ao Protocolo Industrial CiP

O trabalho com título original: **Resource-constrained Industrial Things – Proposal for the Adaptation of CoAP to EtherNet/IP™** foi apresentado no evento: ODVA 2017 Industry Conference & 18th Annual Meeting, realizado de 21 a 23 de Fevereiro de 2017 em Palm Harbor, Florida, USA. A ODVA é uma associação, fundada em 1995, que possui como

membros as principais empresas de automação do mundo. A principal tecnologia desenvolvida pela ODVA é o protocolo de rede denominado CIP (Common Industrial Protocol) e as interfaces de rede EtherNet / IP, DeviceNet, CompoNet e ControlNet.

Green e Otterdahl (2017) apresentam uma proposta de integração do protocolo CIP ao protocolo CoAP para uso em Internet das Coisas (IoT), mantendo a confiabilidade. Em resumo, o trabalho tem como objetivo permitir a fácil e eficiente integração de dispositivos com recursos limitados, presente no ambiente de Internet das Coisas, ao protocolo de automação industrial CIP. Entendam-se como recursos que podem ser limitados em um dispositivo IoT como memória, processamento e energia no caso de alimentação por bateria.

3.2.10 Monitoramento de Geração Híbrida usando Web SCADA

O trabalho com título original: “**Web-SCADA for Monitoring and Controlling Hybrid Wind-PV Power System**” foi publicado no periódico TELKOMNIKA (Telecommunication Computing Electronics and Control) June 2014.

Neste trabalho é apresentado um sistema de supervisão e controle de uma usina híbrida experimental de pequena capacidade que possui geração solar e eólica. Os equipamentos de geração são: 1) Um gerador eólico de 200W; 2) Seis painéis fotovoltaicos de 50W; 3) Uma bateria de 12V 100Ah. Os equipamentos de supervisão e controle são: duas Unidades Terminais Remotas (Remote Terminal Unit) denominadas RTU-1 e RTU-2; 2) Um Controlador Lógico Programável (Programmable Logic Controller) denominado PLC; 3) Um computador com software IntegraXor.

As Unidades Terminais Remotas RTU-1 e RTU-2 foram desenvolvidas usando controladores AVR ATmega8535 usados em conjunto com módulos de sensores ambos de baixo custo. O PLC usado é também de baixo custo, do modelo Zelio Smart Relay[®] fabricado pela Schneider Electric[®]. As funções dos equipamentos são mostradas a seguir:

- A RTU-1 lê os sensores de medições ambientais: 1) Irradiação Solar; 2) Temperatura; 3) Velocidade do Vento e Direção do Vento;
- A RTU-2 lê os sensores de medições elétricas: 1) Tensão e Corrente da Geração Eólica; 2) Tensão e Corrente da Geração Solar; 3) Tensão e Corrente do Barramento de Corrente Contínua; 4) Tensão e Corrente das Baterias; 5) Tensão e Corrente das Cargas;

- O PLC faz o gerenciamento das conexões das fontes de geração e das cargas no barramento principal de corrente contínua de 12 Volts.
- O computador com o software IntegraXor efetua duas tarefas principais:
1) Comunicação com os equipamentos RTU-1, RTU-2 e PLC; 2) Disponibilização de interface do tipo Web SCADA na Internet. Esta interface Web SCADA permite ao operador a visualização das informações de supervisão e o envio de comandos para o PLC usando um navegador.

3.3 Monitoramento com Controle Fuzzy baseado em Regras

3.3.1 Sistema de Avaliação da Qualidade da Água

O trabalho com título: **Sistema baseado em regras fuzzy para avaliação da qualidade da água**, foi publicado na Revista Eletrônica Paulista de Matemática. ISSN 2316-9664 Volume 14, fev. 2019. Edição Ermac (NEVES e DUARTE, 2019).

Neves e Duarte (2019) propõem uma metodologia para avaliação da qualidade da água que usa sistema de controle fuzzy baseado em regras. Para isso, usa-se como variáveis de entrada os parâmetros fornecidos por uma empresa de abastecimento: Cor Aparente (medida em UH - Unidade Hazen), pH (potencial Hidrogeniônico) e Turbidez (UT - Unidade de Turbidez - unidades de cor). O cálculo do valor numérico da variável de saída foi feito através do método de inferência de Mamdani usando-se as regras presentes na Base de Regras. A variável de saída é a Potabilidade da água ou qualidade para o consumo. Os conjuntos fuzzy de saída são: boa, adequada e inadequada. Uma breve descrição da Lógica Fuzzy e dos Sistemas de Controle Fuzzy Baseado em Regras estão no Anexo E.

Os resultados apresentados no paper mostram que este método é de fácil implementação, fornecendo um recurso útil e automatizado para auxiliar nos procedimentos de tratamento de água pelas companhias de abastecimento. O objetivo do trabalho é ter uma ferramenta automatizada de auxílio na avaliação da qualidade da água, contribuindo na melhora geral do fornecimento de água de boa qualidade para a população.

Neste trabalho, o método usado para estimar a qualidade da água através da variável de saída Potabilidade é o mesmo aplicado para calcular a condição de saúde dos inversores através da variável de saída SaudeInversor no tema desta dissertação.

4 - Metodologia

Este capítulo descreve a metodologia concebida para o desenvolvimento do sistema de supervisão, controle e manutenção baseada em condição para usinas de geração solar fotovoltaicas usando o conceito IoT. A metodologia baseia-se nos seguintes itens:

1. **Metodologia para o desenvolvimento da Arquitetura de Supervisão e Controle:** inicialmente foram elaborados diagramas do método de modelagem funcional padrão IDEF0. O diagrama A-0 contém o bloco funcional principal do sistema A0, com todas as entradas, saídas, mecanismos e controles. O bloco A0 foi expandido em dois blocos A1 e A2, correspondentes à divisão funcional principal: A1 = Funções do Nível de Processo e A2 = Funções do Nível de Rede. Os próximos dois diagramas mostram os blocos A1 e A2 expandidos mostrando as funções internas. A partir dos diagramas funcionais IDEF0 foi elaborado um Diagrama de Bloco do Nível de Processo e um Diagrama de Bloco do Sistema de Supervisão, Controle e Monitoramento de Condição.
2. **Metodologia para o desenvolvimento e montagem dos dispositivos eletrônicos para condicionamento de sinal e controle:** Estes dispositivos, destinados a permitir a leitura dos sinais de supervisão e o envio de comandos pelo controlador Arduino Mega 2560, devem ser montados com componentes eletrônicos facilmente acessíveis no mercado nacional e de baixo custo;
3. **Metodologia para implementação dos procedimentos de Monitoramento de Condição:** Os procedimentos de monitoramento de condição dos inversores foram feitos usando-se sistema de controle fuzzy baseado em regras. O procedimento de avaliação de saúde do banco de baterias foi feito através da comparação entre o tempo máximo de descarga para baterias novas e o tempo máximo de descarga projetado. As rotinas que implementam essas funções foram desenvolvidas em linguagem Java e rodam em um computador Raspberry PI 3;
4. **Metodologia para implementação dos protocolos CoAP e HTTP:** visando o máximo aprendizado sobre os protocolos, as rotinas que implementam esses protocolos foram desenvolvidas passo a passo a partir das normas que regem o formato das mensagens e os procedimentos de comunicação. A conformidade das mensagens e procedimentos dos protocolos foram verificados usando-se o programa analisador Wireshark. Foram usadas as linguagens C e Java e os ambientes de desenvolvimento IDE Arduino e Eclipse.

4.1 Apresentação do Cenário

A metodologia proposta será validada através de um estudo de caso é feito em uma usina de geração híbrida, alimentando uma residência de médio porte e uma bomba de água na área rural de Brasília-DF. O objetivo principal da usina é fornecer energia ininterrupta. O projeto elétrico, a aquisição dos componentes, a montagem e instalação da usina foram realizados pelo autor.

Visando atender os requisitos de fornecimento energia, foram instalados quatro painéis fotovoltaicos com capacidade máxima de potência de 320W cada, que podem fornecer em conjunto, a potência máxima de 1280W. O fornecimento de energia ininterrupta é garantido por um banco de baterias seladas de chumbo ácido de 345Ah em 24Vcc, o qual pode alimentar a casa por 12 horas na falta de geração solar e da energia da rede da concessionária. A Figura 4.1 apresenta os componentes principais da usina de geração solar híbrida:

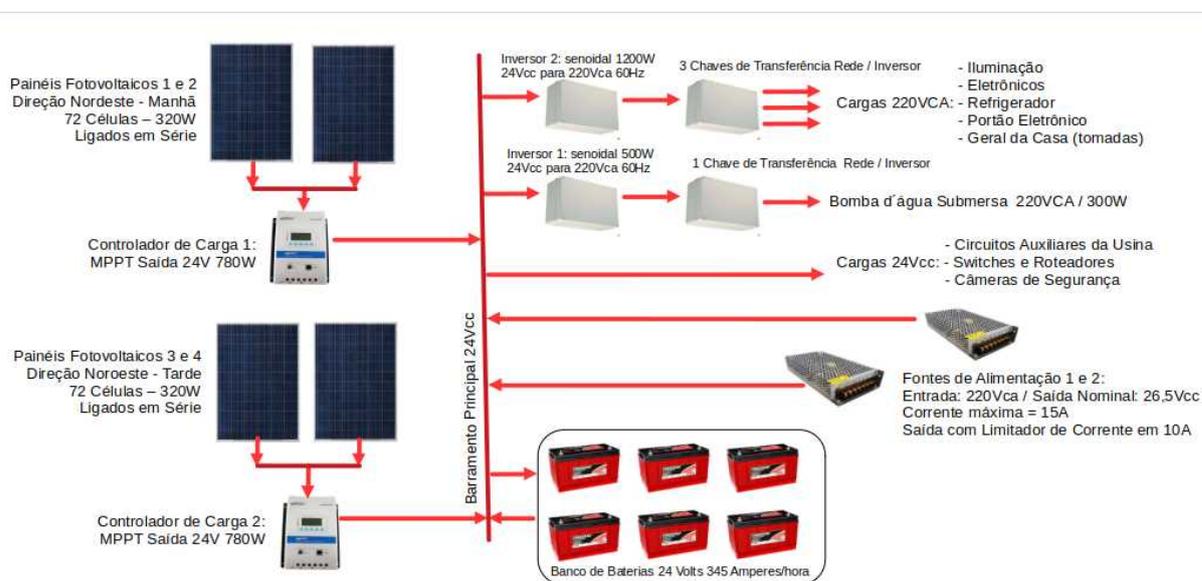


Figura 4.1 - Componentes Principais da Usina de Geração Solar Híbrida

O fornecimento de energia da usina solar é feito em conjunto com a rede da concessionária através de um sistema de gerenciamento de cargas, o qual proporciona energia ininterrupta e economia da seguinte maneira: durante o dia, as cargas são alimentadas pela usina solar. À noite as cargas são transferidas para a energia da concessionária. Caso a capacidade de geração solar se reduza durante o dia por céu encoberto, a energia da rede é injetada no barramento de 24Vcc através de duas fontes de alimentação CA/CC para compensar a redução. A usina possui as seguintes características:

1. Dois conjuntos de dois painéis fotovoltaicos policristalinos de 320Wp / 72 células, Marca Yngli Modelo YGE72. Um conjunto de dois painéis foi instalado no telhado na direção nordeste. O outro conjunto foi instalado na direção noroeste. Os quatro painéis podem fornecer uma potência máxima de 1280W (1280Wp).
2. Dois Controladores de Carga microprocessados de tecnologia MPPT, Marca EPEVER Modelo Triron 3210, com potência máxima de 780W e tensão nominal de saída de 24Vcc cada. Cada Controlador de Carga recebe em sua entrada dois painéis fotovoltaicos ligados em série e tem a sua saída conectada ao barramento principal de 24Vcc. Este equipamento possui interface de comunicação do tipo serial assíncrona, a qual usa protocolo MODBUS RTU para acesso às informações de geração solar;
3. Um Banco de Baterias de chumbo-ácido seladas com tensão nominal de 24Vcc e capacidade de armazenamento de 345Ah. O Banco de Baterias tem por finalidade fornecer a corrente contínua para alimentar os circuitos 24Vcc e os inversores no caso de falta de geração solar e interrupção no fornecimento de energia da rede da concessionária. Para máxima durabilidade do banco de baterias, este é mantido sempre carregado na tensão de flutuação na presença da rede e da geração solar;
4. Dois Inversores de tecnologia SPWM de um estágio com entrada em 24Vcc e saída em 220VCA 60Hz. O Inversor 1 alimenta a bomba d' água e o Inversor 2 os demais equipamentos. A potência máxima de saída é: Inversor 1 = 500W e Inversor 2 = 700W. Cada Inversor fornece em sua saída onda senoidal de 60Hz, contando com os recursos de proteção contra sobrecarga, sobrecorrente, subtensão, sobretensão, sobretemperatura dos transistores e sobrecorrente do transformador;
5. Duas Fontes de Alimentação com entrada em 220V e saída em 27,0Vcc e corrente máxima 15A. Estas fontes alimentam das cargas em 24Vcc na ausência de geração solar durante a noite e também na diminuição temporária durante o dia. Neste último caso, para evitar chaveamento de cargas na redução da geração solar, as fontes fornecem a energia suplementar a partir da rede para os dispositivos da usina;
6. Quatro chaves de transferência. Estas chaves permitem selecionar a fonte de energia de cada conjunto de cargas entre o inversor e a rede da concessionária. A seleção pode ser feita de forma manual ou automática. A seleção automática depende basicamente de dois fatores: do valor da tensão do barramento principal e da hora do dia.

4.2 Arquitetura da Usina

4.2.1 Barramento Principal de Corrente Contínua

A usina solar possui um barramento principal de corrente contínua com tensão nominal de 24Vcc. Neste barramento estão conectados os dispositivos injetores de corrente contínua e os consumidores de corrente contínua (Cargas CC). O banco de baterias é o único dispositivo que pode operar das duas maneiras: injetor de corrente, no caso de operação em descarga das baterias, e consumidor de corrente, no caso de operação em carga das baterias. O diagrama elétrico da usina é mostrado na Figura 4.2.

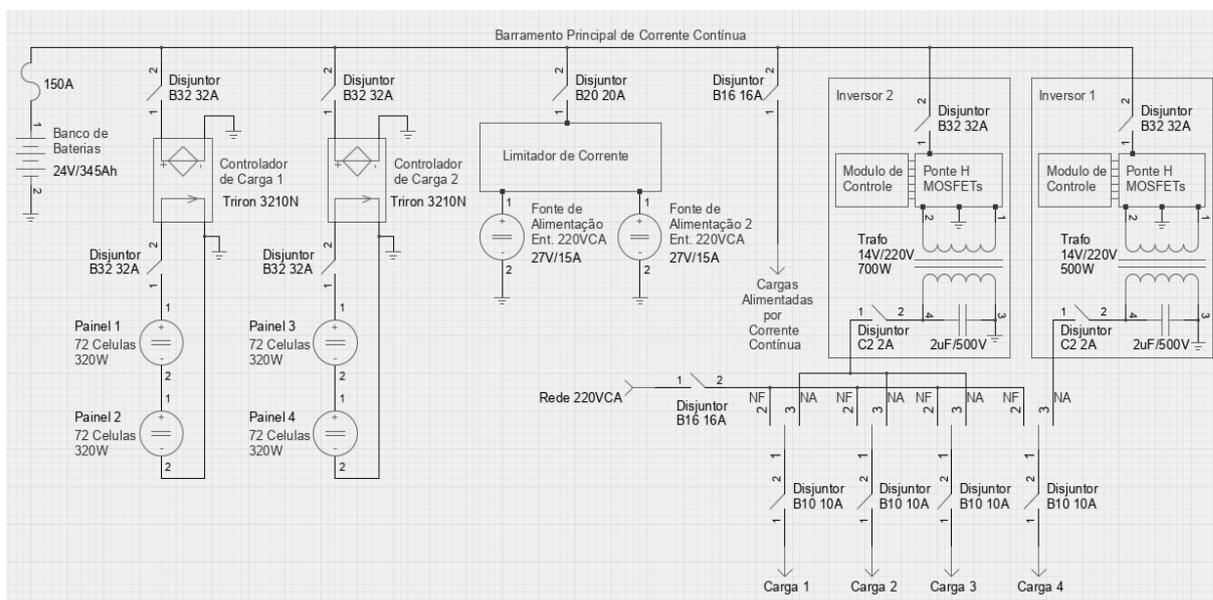


Figura 4.2 - Diagrama Elétrico da Usina Solar e Fonte Ininterrupta de Energia

A tensão no barramento pode variar de 23Vcc a 29Vcc dependendo das condições de fornecimento e consumo de corrente. Em operação normal, com a geração solar operando e/ou com a rede da concessionária disponível, a tensão no barramento principal sempre permanece acima de 26,0Vcc.

Os Controladores de Carga injetam corrente no barramento principal com o objetivo de manter a tensão sempre maior ou igual a 27,6Vcc. O valor 27,6Vcc corresponde ao valor de tensão de flutuação, ideal para manter as baterias carregadas e prontas para operação. Caso a geração solar seja maior que o consumo das Cargas CC, os Controladores de Carga aumentam a tensão do barramento até próximo de 29,0Vcc durante um tempo determinado com o objetivo de carregar completamente as baterias.

As fontes de alimentação que injetam a corrente proveniente da rede da concessionária tem tensão de saída ajustada para 27,0Vcc. Isto é feito para que a geração solar tenha prioridade sobre a energia fornecida pela rede da concessionária, pois como os Controladores de Carga injetam corrente visando manter a tensão do barramento principal maior ou igual a 27,6Vcc, esta tensão sendo maior que a tensão de saída das Fontes de Alimentação alimenta as cargas conectadas ao barramento.

Na Figura 4.2, o barramento principal de corrente contínua está acima e o banco de baterias mais à esquerda. Da direita para a esquerda após o banco de baterias, os dispositivos injetores de corrente: os controladores de carga e as fontes de alimentação. À direita estão os dispositivos consumidores de corrente: as cargas de corrente contínua, o inversor 2 e o inversor 1 mais à direita. Abaixo dos inversores estão as chaves de transferência, alimentando as cargas em corrente alternada 220 volts.

4.2.2 Injeção de Corrente no Barramento Principal de Corrente Contínua

Existem cinco dispositivos que podem injetar corrente no barramento principal de corrente contínua, os quais são mostrados a seguir:

1. Controlador de Carga 1 → Injeta corrente proveniente da Geração Solar feita pelos Painéis Fotovoltaicos 1 e 2;
2. Controlador de Carga 2 → Injeta corrente proveniente da Geração Solar feita pelos Painéis Fotovoltaicos 3 e 4;
3. Fonte de Alimentação 1 → Injeta corrente da Rede da Concessionária;
4. Fonte de Alimentação 2 → Injeta corrente da Rede da Concessionária;
5. Banco de Baterias → Injeta corrente na falta de geração solar e rede simultaneamente.

4.2.3 Cargas Conectadas no Barramento Principal de Corrente Contínua

Existem três cargas principais conectadas ao barramento principal de corrente contínua:

1. As Cargas de Corrente Contínua: circuitos condicionadores de sinal, circuitos de controle da usina, chaves de transferência, controladores Arduíno, computadores Raspberry PI 3, ventiladores, switches, roteadores, câmeras de segurança;
2. O Inversor 1 que alimenta a bomba d'água;
3. O Inversor 2 que alimenta as demais cargas CA

4.3 Recursos de Comando, Operação e Controle da Usina

A usina solar usada neste projeto de pesquisa possui quatro conjuntos de cargas de corrente alternada em 220 volts alimentados por quatro chaves de transferência. Cada chave é composta basicamente por um relé reversível de potência com bobina de 12Vcc e os circuitos de interface para compatibilização com as saídas digitais da UTR. Cada chave de transferência, sob comando do software da UTR, seleciona a fonte de energia de cada conjunto de cargas entre a rede ou o inversor.

A chave de transferência é um dispositivo obrigatório em usinas híbridas, pois, permite que as cargas de corrente alternada sejam alimentadas normalmente pela rede da concessionária no caso de falta de geração solar, baterias sem carga ou na ocorrência de falha de inversor. Assim, seguindo ao conceito de falha segura, a posição NF das chaves de transferência são conectadas na rede da concessionária.

Em uma usina híbrida, as chaves de transferência podem operar em dois modos: assíncrono ou síncrono. No modo assíncrono, não é feita a sincronia de fase entre a onda de tensão de saída do inversor e a da rede. Por este motivo, é necessário um período de desligamento de pelo menos dois segundos na transferência de uma fonte para a outra. O modo assíncrono é usado na usina tema deste trabalho. No modo síncrono, existe o recurso de sincronismo de fase que faz a transferência da fonte de tensão no ponto correto da onda, o que reduz o tempo de interrupção a alguns milissegundos, tornando-o imperceptível para as cargas. A Figura 4.3 mostra o diagrama esquemático das quatro chaves de transferência com quatro relés reversíveis de um polo:

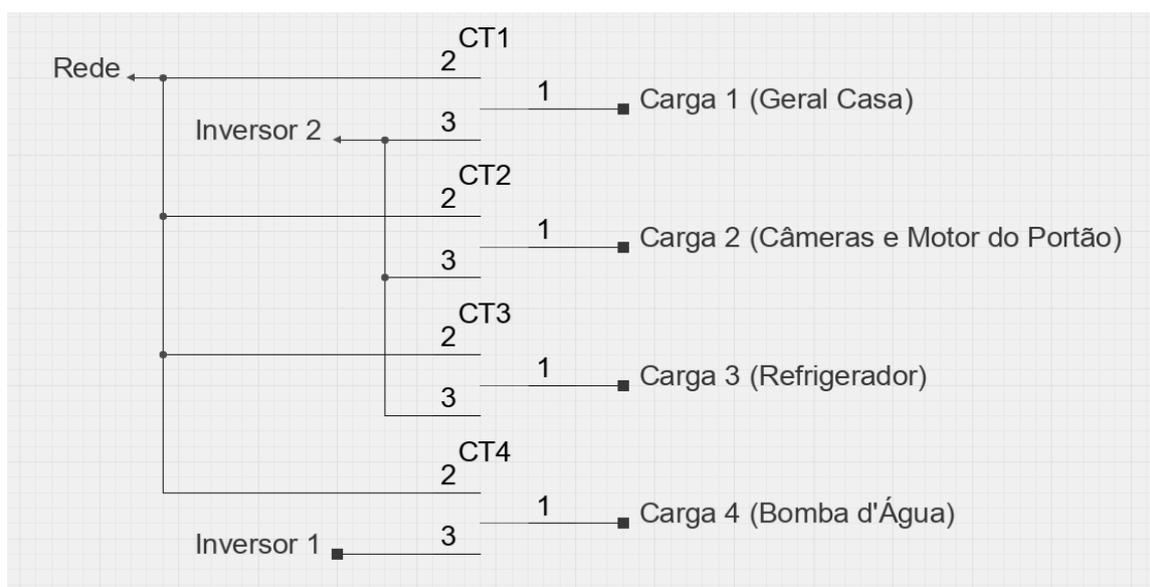


Figura 4.3 - Diagrama Esquemático das Chaves de Transferência

4.3.1 Modos de Operação – Economia e Normal

O Modo de Operação Economia desabilita os inversores e mantém somente as cargas alimentadas com corrente contínua, mesmo na ocorrência de falta de tensão da rede da concessionária. O Modo de Operação Normal permite que os inversores sejam ligados. O Modo de Operação Economia é selecionado sempre após a inicialização do controlador da UTR. Isto é feito por questões de segurança para manter os inversores desligados. A reinicialização da UTR ocorre após três eventos:

- 1 A alimentação CC da UTR é desligada e ligada;
- 2 O botão de reset da placa Arduíno Mega 2560 da UTR é pressionado e solto;
- 3 É feita uma carga de programa na memória da UTR.

4.3.2 Modos de Comando – Local e Remoto

O Modo de Comando Remoto permite a execução pela Unidade Terminal Remota (UTR) de comandos recebidos pela interface de comunicação. Esses comandos quais são enviados pelo usuário através da tela do supervisório apresentada em um Programa Navegador (Browser). Em operação normal, o Modo de Comando Remoto sempre deve estar selecionado para permitir a supervisão e controle através dos servidores HTTP.

O Modo de Comando Local, quando ativado, impede que o software da UTR execute comandos recebidos através da interface de comunicação. Isto é feito para evitar que comandos enviados remotamente sejam executados durante os procedimentos de manutenção da usina.

4.3.3 Modo de Operação Especial – Interrupção no Fornecimento da Rede

Na ocorrência de interrupção no fornecimento de energia da rede da concessionária, o software da UTR entra em um Modo de Operação Especial. Neste modo, se está selecionado o Modo de Operação Normal, as Cargas 1 e 2 são habilitadas e alimentadas pelo Inversor 2 enquanto a tensão do barramento principal de corrente contínua permanecer maior ou igual a 23,0Vcc a qualquer hora do dia, ou da noite. Isto tem por objetivo proporcionar conforto, funcionalidade e segurança. A Carga 1 corresponde à iluminação interna e dispositivos da casa. A Carga 2 ao motor do portão e parte da iluminação externa. À noite, as Cargas 1 e 2 são alimentadas pelo banco de baterias.

Em relação às Cargas 3 e 4: durante o dia, no caso de interrupção no fornecimento da rede, a geração solar seja suficiente para alimentá-las e estejam habilitadas, estas cargas recebem

energia dos inversores 2 e 1 respectivamente. A decisão de acionar o inversor e alimentar a carga é feita pelo software da UTR através da comparação da medida da tensão do barramento principal de corrente contínua com limites pré estabelecidos. Neste caso, se esta tensão se mantiver acima de um valor limite, significa que há geração solar suficiente. Em resumo, no caso de interrupção no fornecimento de energia da rede são realizados os seguintes procedimentos:

1. A Carga 1 e a Carga 2 são alimentadas pelo Inversor 2, em qualquer hora do dia, em Modo de Operação Normal, enquanto a tensão do barramento principal permanecer maior ou igual a 23,0Vcc;
2. Se a geração solar for suficiente durante o dia, a Carga 3 é alimentada pelo Inversor 2 e a Carga 4 é alimentada pelo Inversor 1.

4.3.4 Modo de Controle Manual

No Modo de Controle Manual, as cargas CA são alimentadas pelo inversor se a tensão do barramento principal de corrente contínua permanecer maior ou igual a 23,0Vcc. Caso caia abaixo de 23,0Vcc, os inversores são desligados e todas as cargas são desconectadas do banco de baterias para proteção contra descarga excessiva.

O valor de tensão mínima de 23,0Vcc foi escolhido através da análise das curvas de descarga e do tempo estimado de vida visando a máxima durabilidade do banco de baterias. No caso específico desse trabalho, o banco de baterias foi superdimensionado para maximizar a sua vida útil. Assim, a tensão de 23,0 Volts não é normalmente atingida durante a descarga.

No Modo de Controle Manual, na falta de geração solar, as cargas em CA são alimentadas pelos inversores, os quais recebem energia da rede da concessionária através das fontes de alimentação CA/CC. Nesse caso, ocorrem perdas em torno de 20%, já que são feitas duas conversões de energia para alimentar as cargas 220VCA (220VCA para 24Vcc e 24Vcc para 220VCA).

O Modo de Operação Manual tem por objetivo evitar o chaveamento entre a rede e o inversor no caso de falta da concessionária, evento que causa interrupção de dois segundos no fornecimento de energia. Este modo é usado normalmente para a Carga 1 durante atividades que necessitem do fornecimento de energia 100% ininterrupta.

4.3.5 Modo de Controle Automático

O Modo de Controle Automático permite colocar as cargas 220VCA em alimentação pela rede quando a geração solar não for suficiente, principalmente à noite, visando a economia no consumo da energia da rede da concessionária. Para implementar este procedimento, o software da UTR verifica a hora do dia e a tensão do barramento principal, efetuando a energização do inversor e conectando cada carga à sua saída caso a carga esteja habilitada.

A Carga 1 possui Modo de Controle exclusivo. A Carga 2, a Carga 3 e a Carga 4 possuem um Modo de Controle comum às três cargas. Isto foi feito para permitir o modo de energia ininterrupta somente para a Carga 1, a qual alimenta a iluminação interna e os equipamentos eletrônicos da casa. No modo automático, o chaveamento da alimentação entre o inversor e a rede provoca uma interrupção de dois segundos no fornecimento de energia para a carga na hora da operação. Desta forma, pode ser indesejável esta interrupção para determinadas atividades de trabalho ou lazer, daí a disponibilidade desse recurso.

4.3.6 Recurso de Habilitação / Desabilitação das Cargas

Para cada carga mostrada na figura 4.3, existe um sinalizador de Habilitação ou Desabilitação no software da UTR. A carga estar habilitada significa que esta carga pode ser alimentada por um equipamento inversor. A carga estar desabilitada significa que esta carga é alimentada pela rede da concessionária. Cada sinalizador pode ser colocado nas posições: habilitado ou desabilitado de forma independente através de comando enviado pelo usuário através do Servidor HTTP usando Navegador. É possível também habilitar e desabilitar as cargas usando as chaves instaladas na caixa dos circuitos condicionadores da usina. A restrição é que usando as chaves, as cargas são habilitadas por grupos e não de modo independente.

4.4 Procedimentos de Controle Automático das Cargas

Quando está selecionado o Modo de Operação Normal em conjunto com o Modo de Controle Automático na Unidade Terminal Remota (UTR), o software efetua o controle das chaves de transferência e o acionamento dos inversores. Isto é feito através da seleção entre a fonte de energia de corrente alternada das cargas entre a rede da concessionária e a saída de um inversor, usando variáveis da própria UTR para tomada de decisão. Visando a máxima confiabilidade deste procedimento, foram usadas somente variáveis geradas localmente na UTR. Desta forma, evita-se que falhas de comunicação possam interferir nos procedimentos

de chaveamento de cargas. As três variáveis geradas localmente na UTR usadas para tomada de decisão são mostradas a seguir:

1. A hora e minuto do nascer e do pôr do Sol na localidade da usina. Estas informações são disponíveis através de uma rotina que calcula os horários do nascer e pôr do Sol de todos os dias do ano em Brasília;
2. A hora e minuto do dia. Através da hora e minuto do dia, o software habilita a alimentação da carga pelo inversor correspondente;
3. A tensão do barramento principal de corrente contínua. Através da medida da tensão do barramento, é possível verificar se há injeção de corrente proveniente da geração solar suficiente para alimentar as cargas.

A seguir serão apresentados os recursos e procedimentos para habilitação e desabilitação das cargas em corrente alternada.

4.4.1 Habilitação de Desabilitação das Cargas CA pela Hora do Dia

A seguir são apresentadas as regras para habilitação e desabilitação das cargas em corrente alternada alimentadas pelos Inversores 1 e 2. A Unidade Terminal Remota possui um Circuito Integrado Relógio de Tempo Real (RTC – Real Time Clock) do tipo PCF8563, o qual fornece a data e hora atual. A rotina que obtém os horários do nascer e do pôr do Sol na localidade da usina, no caso, em Brasília DF, possui uma tabela indexada pelo dia do ano, a qual foi obtida do site www.sunrise-and-sunset.com. As pequenas variações dos horários que podem ocorrer de um ano para o outro não tem influência significativa na geração solar. Por este motivo, a tabela do ano de 2020, que é bissexto, foi usada para esta finalidade. A seguir são apresentadas as regras para habilitação/desabilitação das cargas condicionadas à hora do dia:

1. As Cargas 1 e 2 são habilitadas meia hora após o horário do nascer do Sol.
2. A Carga 3 é habilitada uma hora após o horário do nascer do Sol.
3. A Carga 4 é habilitada duas horas e meia após o horário do nascer do Sol.
4. As Cargas 1 e 2 são desabilitadas meia hora antes do pôr do Sol.
5. A Carga 3 é desabilitada uma hora antes do horário do pôr do Sol.
6. A Carga 4 é desabilitada duas horas antes do horário do pôr do Sol.

4.4.2 Habilitação de Desabilitação das Cargas CA pela Tensão CC

Quando a Unidade Terminal Remota (UTR) está no Modo de Operação Normal e no Modo de Controle Automático, se a geração fotovoltaica está normal, ao ser habilitada cada carga, a tensão do barramento tende a ser maior que 27,5V. No caso da ocorrência de céu encoberto que reduz a geração fotovoltaica, a tensão do barramento fica mais baixa, e o software verifica se é possível ligar o inversor e habilitar as cargas em corrente alternada.

A Tabela 4.1 mostra os valores dos limites da tensão do barramento principal de corrente contínua para habilitação e desabilitação das cargas de corrente alternada:

Tabela 4.1 - Valores de Tensão para Habilitação/Desabilitação das Cargas CA

Carga	Tensão que Habilita Ligar	Tensão que Desabilita Ligar
1	27,30 Vcc	25,40 Vcc
2	27,30 Vcc	25,40 Vcc
3	27,40 Vcc	26,00 Vcc
4	27,50 Vcc	26,30 Vcc

4.5 Arquitetura do Sistema de Supervisão, Controle e MBC

A Arquitetura do Sistema de Supervisão, Controle e Monitoramento de Condição da Usina Solar estudo de caso foi feita inicialmente com o auxílio do método dos diagramas funcionais IDEF0. O IDEF0 é uma ferramenta de modelagem funcional, consistindo de um subconjunto da Técnica de Análise e Projeto estruturados (Structured Analysis and Design Technique - SADT).

4.5.1 Diagramas Funcionais IDEF0

No método dos Diagramas Funcionais IDEF0 deve ser definida uma Função Principal para o Sistema, que é o título do Bloco Principal denominado A0. O Bloco A0 recebe, à esquerda todas as entradas, fornece à direita todas as saídas, recebe embaixo todos os mecanismos e recebe em cima todos os controles.

O Bloco A0 é apresentado no diagrama principal denominado A-0 apresentado na Figura 4.4:

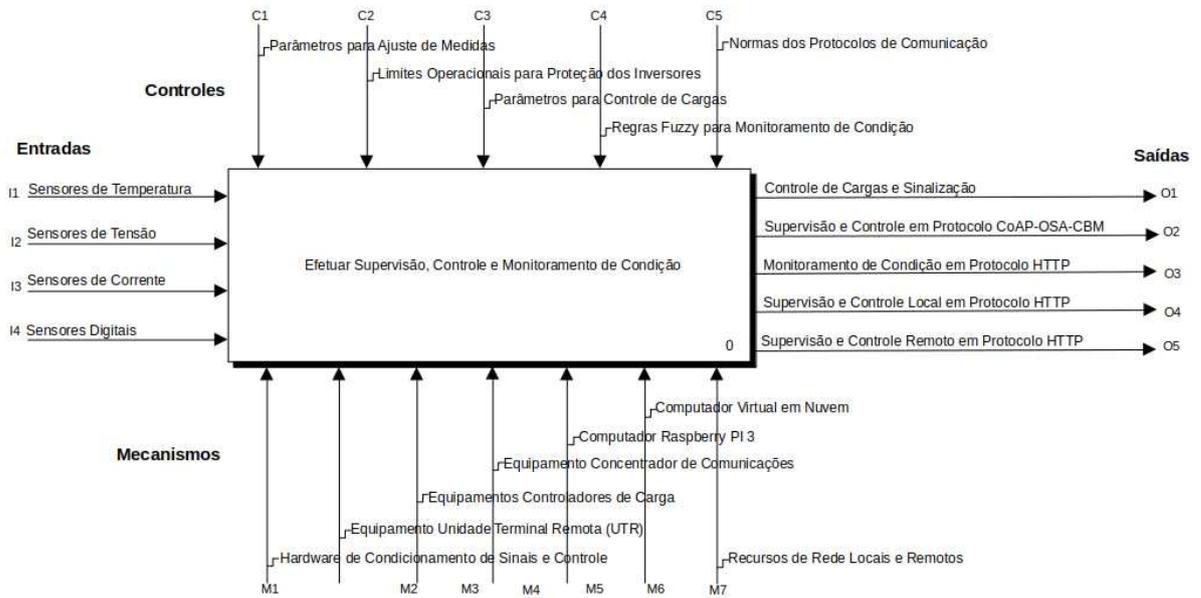


Figura 4.4 - Diagrama IDEF0 A-0 com O Bloco Principal A0

O Bloco A0 é expandido em dois blocos: A1 correspondente às funções do Nível de Processo e A2 correspondente às funções do nível de rede. O diagrama A0 mostrado na Figura 4.5 apresenta os Blocos A1 e A2:

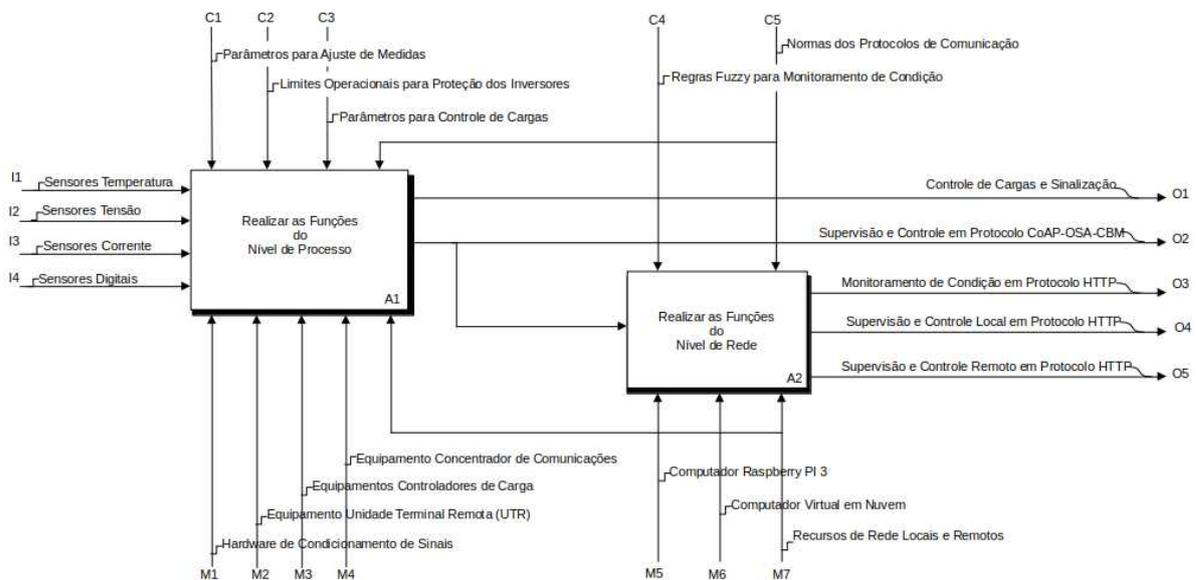


Figura 4.5 - Diagrama A0 com os Blocos A1 (Nível de Processo) e A2 (Nível de Rede)

O Diagrama Funcional A1 apresenta as funções internas do Bloco A1 (Nível de Processo) denominadas A11, A12, A13, A14 e A15. A Figura 4.6 mostra este diagrama:

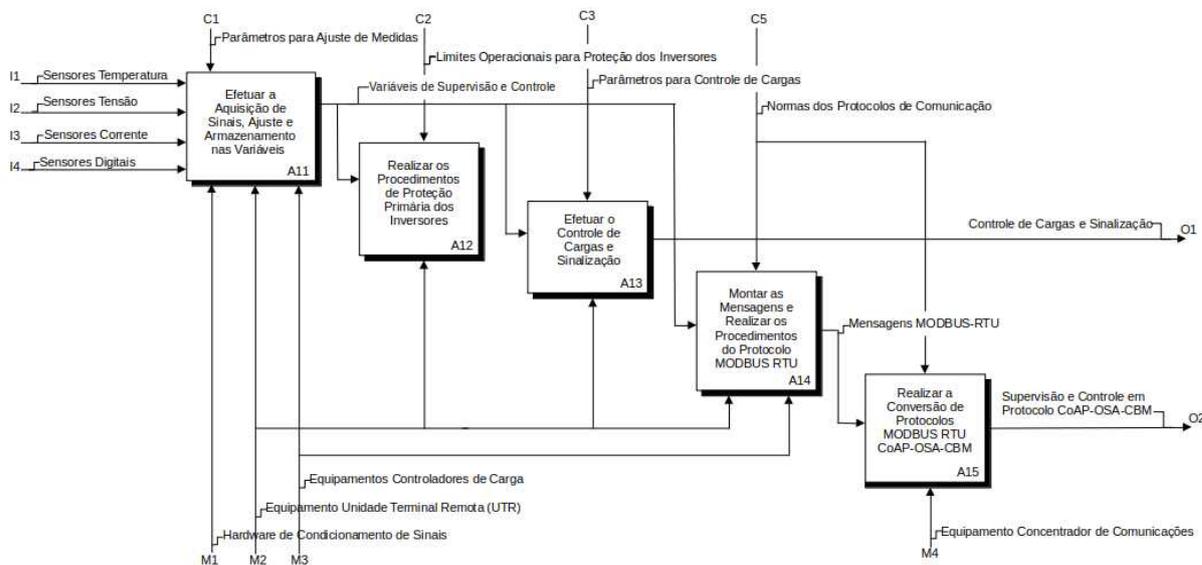


Figura 4.6 - Diagrama A1 (Nível de Processo)

O Diagrama Funcional A2 apresenta as funções internas do Bloco A2 (Nível de Rede) denominadas A21, A22, A23, A24 e A25. A Figura 4.7 mostra este diagrama:

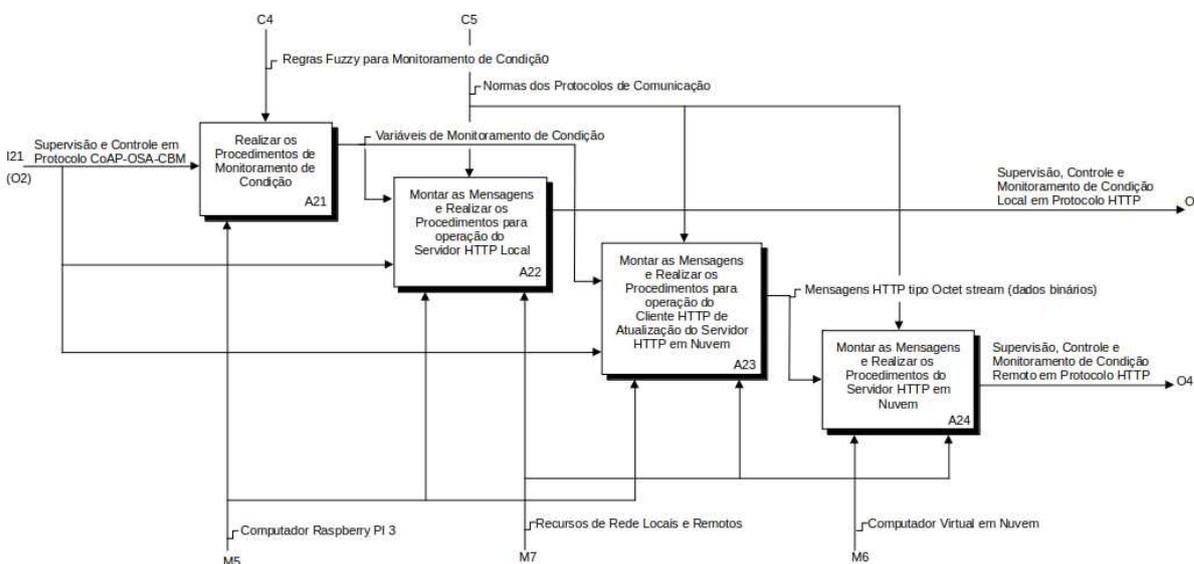


Figura 4.7 - Diagrama Funcional A2 (Nível de Processo)

4.6 Arquitetura para Montagem do Hardware e do Software

Neste projeto foi usado o conceito de processamento distribuído de forma simples. O uso da rede Ethernet e o protocolo CoAP em UDP como meio de comunicação entre o computador Raspberry PI 3 e o controlador Arduíno Mega, permite alta velocidade e alta confiabilidade no tráfego dos dados na rede local, tornando viável esta solução de baixo custo.

Um Arduíno Mega efetua a interface com o processo (aquisição e controle) e o processamento de controle básico. O computador Raspberry PI 3 efetua o processamento de informações mais complexo, executando os procedimentos de inferência fuzzy, a comunicação com o nível do processo e com os níveis de supervisão e controle local e remoto. O nível remoto pode ser um computador na nuvem, o qual faz a intermediação entre os clientes remotos e o sistema.

No nível de processo, foi montada uma arquitetura expansível para realizar os procedimentos de supervisão e controle da usina de geração fotovoltaica, da caixa e da bomba d'água, principalmente por questões de segurança, confiabilidade e capacidade de ampliação. Esta arquitetura permite a conexão de mais um Arduíno Mega destinado a receber pontos físicos, e a mais dois Controladores de Carga ao Concentrador, mantendo a mesma estrutura de comunicação com o Servidor Raspberry PI 3.

Nesta arquitetura foi usado um Arduíno Mega 2560 com placa Shield Ethernet na função de Concentrador, o qual efetua a comunicação em interface serial e protocolo MODBUS RTU com os dispositivos conectados diretamente ao processo. O Concentrador faz a conversão das informações do processo para o Protocolo CoAP-OSA-CBM em UDP em Rede Ethernet. Os três dispositivos do processo são mostrados nos itens a seguir:

4.6.1 Equipamento Unidade Terminal Remota (UTR)

A Unidade Terminal Remota (UTR) faz a interface com todos os pontos físicos do processo de entradas digitais, entradas analógicas e saídas digitais da Usina Solar e da Caixa d'Água.

O software embarcado na UTR efetua a leitura e controle desses pontos, executa os procedimentos de proteção primária dos inversores e faz a conversão das informações para o formato usado pelo protocolo MODBUS RTU.

A UTR é montada com um Arduíno Mega 2560 sem a placa Shield Ethernet, o que facilita a ligação dos pinos dos conectores na placa base. Na placa base da UTR estão instalados os

elementos de proteção contra surtos, as régua de bornes para conexão dos cabos e a fonte de alimentação de 24Vcc para 9Vcc.

A placa base da UTR é instalada em uma caixa elétrica para proteção mecânica. A carga de programa na placa Arduino Mega 2560 da UTR é feita por interface serial USB conectada ao Computador Raspberry PI 3 instalado na usina solar.

4.6.2 Equipamentos Controladores de Carga (CC1 e CC2)

Os Equipamentos Controladores de Carga MPPT são fabricados pela EPEVER modelo Triron 3210N, sendo denominados:

- Controlador de Carga 1 (CC1)
- Controlador de Carga 2 (CC2).

Cada controlador de carga recebe a energia de dois painéis fotovoltaicos de 72 células e 320Wp, efetuando a comunicação com o Concentrador Arduino Mega 2560 através de interface serial assíncrona padrão elétrico RS485. As informações fornecidas referem-se à geração solar.

4.6.3 Equipamento Concentrador

O Equipamento Concentrador é composto pelos seguintes dispositivos:

1. Um controlador Arduino Mega 2560 com uma placa do tipo piggy-back Shield Ethernet;
2. Uma placa com duas interfaces padrão elétrico RS485 para comunicação serial com os equipamentos controladores de carga;
3. Uma placa conversora de energia de 24Vcc para 9Vcc.

O Equipamento Concentrador tem por função principal intermediar a comunicação entre nível de rede e nível do processo. Assim, o concentrador efetua a comunicação com o nível de rede em interface Ethernet 10/100BaseT com protocolo CoAP-OSA-CBM, e efetua a comunicação com os equipamentos do nível de processo em interface serial assíncrona com protocolo MODBUS RTU. Os equipamentos do nível do processo são a Unidade Terminal Remota (UTR) e os dois Controladores de Carga Triron 3210. A carga de programa na placa Arduino Mega 2560 do Concentrador é feita por interface serial USB conectada ao Computador Raspberry PI 3 instalado na usina solar.

Observação: Neste projeto de pesquisa não será feita análise do protocolo MODBUS RTU nem da comunicação serial assíncrona. A razão disso é porque são serem temas já bem consolidados na área de automação, supervisão e controle.

4.6.4 Diagrama de Blocos do Nível de Processo

A Figura 4.8 mostra o Diagrama de Blocos do Nível de Processo da usina solar e da caixa e bomba d'água. No bloco mais à esquerda estão listados todos os equipamentos sob supervisão e controle. A UTR faz a supervisão e controle dos equipamentos através de circuitos condicionadores de sinal, os quais compatibilizam os sinais para o nível de tensão das entradas e saídas do Arduino Mega.

Por fim, o concentrador efetua a comunicação com a UTR e com os controladores de carga através de interface serial assíncrona e protocolo MODBUS RTU. A conexão da interface serial do concentrador com a UTR é direta, pois ambos usam o padrão 0 a +5Vcc e os equipamentos estão bem próximos, o que viabiliza esse tipo de conexão. A conexão do concentrador com os controladores de carga é feita através de conversor para o padrão elétrico RS485.

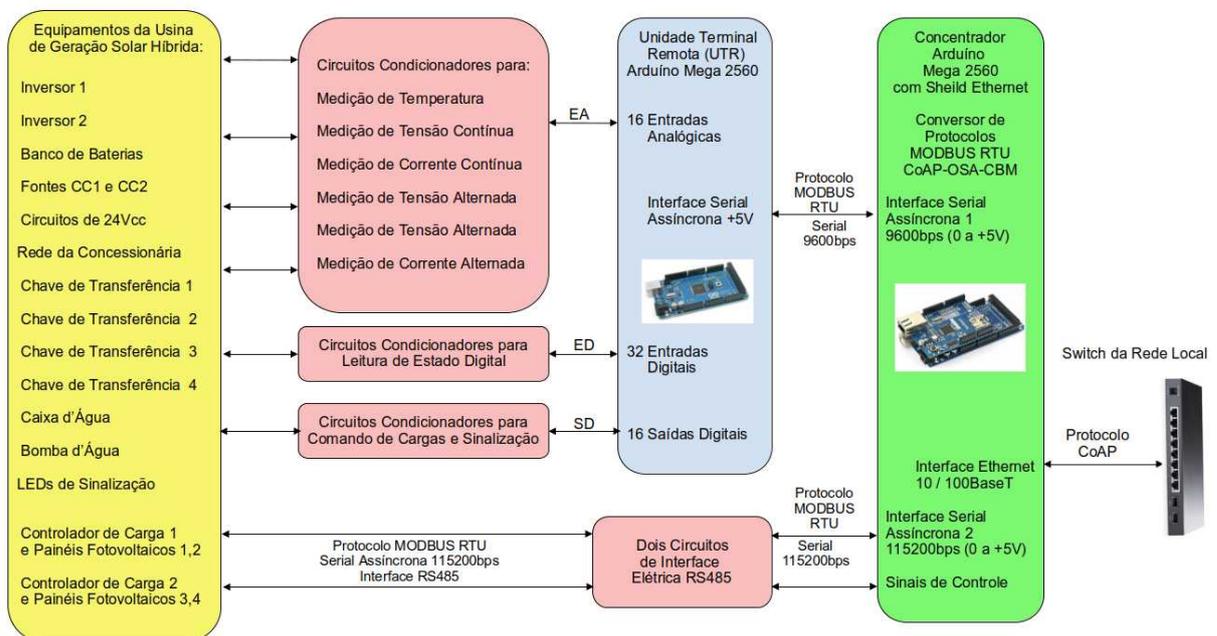


Figura 4.8 - Diagrama de Blocos do Nível de Processo

4.6.5 Projeto e Montagem dos Condicionadores de Sinal e Conversores

Todos os condicionadores de sinal foram projetados e montados em placas protótipo levando-se em consideração a necessidade de confiabilidade e razoável precisão aliada ao baixo custo. Assim, todos os sinais de supervisão digitais e analógicos foram convertidos para tensão na faixa de 0 a +5 volts em corrente contínua. No caso dos sinais obtidos em tensão alternada, esta foi retificada, filtrada e convertida para o padrão 0 a +5 Volts.

A Unidade Terminal Remota (UTR), baseada no controlador Arduíno Mega 2560, possui dezesseis entradas analógicas com resolução de dez bits para medição. Estas entradas fornecem valores brutos lidos pelo software, na faixa de 0 a 1023 contagens, referentes às tensões de 0 e +5 Volts respectivamente. No controlador Arduíno Mega 2560 estão disponíveis também, cinquenta e quatro pinos de entradas ou saídas digitais, que são configurados pelo software de acordo com sua função (entrada ou saída).

4.7 Projeto do Inversor SPWM de um Estágio

Os dois inversores usados neste projeto de pesquisa foram projetados e montados levando-se em consideração as condições particulares de operação. O Inversor 1 é usado exclusivamente para alimentar a bomba de água do poço profundo. Esta bomba, alimentada com 220VCA, é do tipo vibratória submersa, sendo usado um capacitor para correção do fator de potência. O Inversor 2 é usado para alimentar as demais cargas em 220VCA. As características principais dos inversores são as seguintes:

1. Entrada de alimentação fornecida por controladores de carga, banco de baterias e fontes CA/CC com tensão nominal de 24 Volts CC. A faixa de operação da tensão de entrada para o inversor neste caso é de 21 a 30 Volts. O valor mínimo de 21 Volts refere-se ao valor mínimo no caso de operação pelo banco de baterias, quando cada bateria do banco está totalmente descarregada com tensão de 10,5 Volts. O valor máximo refere-se à tensão máxima de equalização do banco de baterias;
2. Saída em tensão senoidal de 220V RMS 60Hz. Respeitando-se o limite de potência, este inversor deve ser capaz de alimentar qualquer equipamento destinado à rede elétrica de baixa tensão fase - neutro na tensão e frequência especificadas. O inversor deve ser capaz de suportar a corrente de partida de motores de indução monofásicos, que pode atingir até quatro vezes a corrente nominal;

3. Tecnologia SPWM (Sinusoidal Pulse Width Modulation - Modulação por Largura de Pulso Senoidal) com os objetivos de obter onda senoidal pura na saída com o máximo de eficiência. Esta tecnologia é baseada na geração de um sinal de 24KHz modulado em largura de pulso (PWM) por um sinal senoidal de 60Hz. Este sinal modulado, após passar por um filtro LC, resulta em um sinal senoidal de 60Hz;
4. Arquitetura de um estágio com transformador de baixa frequência. Nesta arquitetura, a modulação SPWM é feita diretamente usando a tensão de entrada de 24Vcc, necessitando, portanto, de um transformador de baixa frequência para elevar a tensão e fornecer na saída os 220VCA. O transformador também serve como indutor para, em conjunto com um capacitor na sua saída, realizar a filtragem do sinal modulado SPWM de 24KHz;
5. O Inversor 1 deve fornecer potência contínua máxima de 500W para alimentar uma bomba vibratória submersa que tem consumo máximo de potência ativa de 350W;
6. O Inversor 2 deve fornecer potência contínua máxima de 800W e até 1600W por curtos períodos para suprir a corrente de partida de motores de indução monofásicos;
7. Através da instalação de sensores e condicionadores para tensão, corrente e temperatura, disponibilizar os sinais de supervisão para o Monitoramento de Condição;
8. Usar o Módulo de Comando: EGS002 - Sinusoid Inverter Driver Board;
9. A entrada de alimentação +12Vcc para o módulo de comando é separada da entrada de alimentação de +24Vcc para o Módulo de Potência, sendo controlada pelo software da UTR. Isto permite à UTR ligar e desligar o inversor;
10. Usa Módulo de Potência do tipo ponte completa (full-bridge) com quatro conjuntos de dois MOSFETs em paralelo. Este módulo recebe a alimentação de 24Vcc através de disjuntor termomagnético do tipo “quick-lag”, ou seja, de comutação rápida. O MOSFET é um componente eletrônico de potência usado como chave de alta velocidade, ou seja, opera ou aberto com alta resistência ou fechado com baixa resistência. Os tempos de chaveamento estão na faixa de nanossegundos;
11. A proteção de curto circuito e sobrecorrente na saída de 220VCA é feita por disjuntor termomagnético do tipo Quick-Lag Siemens modelo B2. A proteção de curto circuito e sobrecorrente na entrada de corrente contínua é feita por disjuntor termomagnético do tipo Quick-Lag Siemens modelo B32.

4.7.1 Arquitetura do Inversor

A arquitetura do inversor, com um estágio e transformador de baixa frequência foi escolhida por ser mais tolerante a falhas tais como curto-circuitos, surtos de tensão e sobrecargas, além de usar componentes de menor custo e ter manutenção mais fácil. Entretanto, esta arquitetura apresenta corrente quiescente maior e menor rendimento que os inversores que usam dois estágios. A característica de suportabilidade é necessária devido à necessidade de alimentar equipamentos a distâncias de até cem metros, tais como luzes externas, portão eletrônico e torre de comunicações, em cabeamento suscetível a receber surtos de descargas atmosféricas.

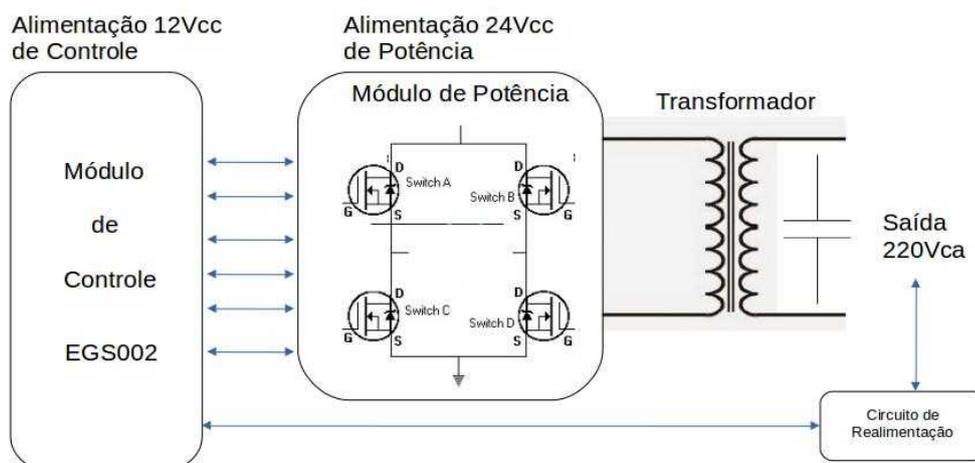


Figura 4.9 - Diagrama de Blocos do Inversor de Um Estágio SPWM

A Figura 4.9 mostra um diagrama de blocos simplificado de um inversor SPWM de um estágio. O funcionamento deste tipo de inversor e as suas características são mostradas no Anexo D – Características dos Inversores.

Normalmente os DPS (Dispositivos de Proteção contra Surtos) conseguem absorver os surtos de tensão provenientes das descargas atmosféricas nos cabos externos, não ocorrendo interrupção na operação do inversor. Entretanto, na incidência de descarga atmosférica muito próxima, evento raro, ocorre o desarme do disjuntor termomagnético na saída do transformador do inversor de um estágio sem maiores danos. Isso ocorre por causa da isolação e do amortecimento que o transformador proporciona aos circuitos eletrônicos de chaveamento. Um evento deste ao atingir um inversor de dois estágios, com grande probabilidade, causaria danos aos MOSFETs da ponte de saída.

O Módulo de Comando EGS002 é baseado no Circuito Integrado EG8010, que é um ASIC (Application Specific Integrated Circuit – Circuito Integrado de Aplicação Específica) fabricado pela empresa EG Microelectronics Corporation (www.EGmicro.com). O Módulo

EGS002 gera todos os sinais necessários para comandar os MOSFETs da ponte completa do módulo de potência e disponibiliza as outras funcionalidades essenciais para a montagem de um inversor SPWM de um ou dois estágios.

4.7.2 Dissipação de calor nos MOSFETs

Em um inversor de um estágio, a tensão de alimentação em corrente contínua é aplicada a um circuito denominado ponte completa (full-bridge). Este circuito efetua a conexão e desconexão do primário do transformador aos terminais positivo e negativo de acordo com um conjunto de sinais modulados em uma frequência de 24KHz. O circuito ponte completa inverte a polaridade da tensão contínua ao aplicá-la ao primário do transformador, obtendo, com isso, um sinal de corrente alternada a partir de uma alimentação CC.

A conexão e desconexão da alimentação ao primário do transformador é realizada por dispositivos eletrônicos de potência denominados MOSFETs, os quais operam como chaves, possuindo somente os estados aberto e fechado. No estado aberto, cada MOSFET possui resistência muito alta, e no estado fechado como uma resistência baixa de 0,008 Ohms (IRF3205). A perda em cada MOSFET, em Watts, é proporcional ao quadrado da corrente média que circula pelo componente, sendo dada pela expressão: $P = RI^2$ onde R é a resistência no estado fechado em Ohms e I é a corrente média em Amperes. Para reduzir ainda mais a resistência, e conseqüentemente a dissipação, em cada extremo da ponte são usados dois MOSFETs em paralelo.

O Módulo de Comando sempre aciona dois conjuntos de MOSFETs da ponte simultaneamente para aplicar a tensão de alimentação no primário do transformador. Assim, a corrente consumida na entrada do inversor é a média do total das correntes que circulam em todos os MOSFETs da ponte completa. Desta forma, o aquecimento dos MOSFETs é gerado majoritariamente pela soma das perdas ôhmicas da circulação de corrente na resistência em estado fechado de cada MOSFET.

Esse calor gerado nos MOSFETs precisa ser dissipado para que a temperatura se mantenha dentro de limites razoáveis. Assim, são usados dissipadores de calor, que são peças de alumínio com aletas onde os MOSFETs são fixados de modo a obter a menor resistência térmica possível, e conseqüentemente, conseguir a melhor transferência de calor. Além dos dissipadores, o inversor possui também dois ventiladores que geram um fluxo de ar através das aletas dos dissipadores retirando o ar quente pela parte de cima da caixa reduzindo a temperatura.

Os ventiladores são acionados por circuitos do tipo termostato somente quando a temperatura nos dissipadores ultrapassa determinado valor. Isso é feito para preservar os ventiladores, porque mantê-los em funcionamento contínuo rapidamente leva-os ao fim de sua vida útil. O módulo EGS002 possui um circuito do tipo termostato que controla os ventiladores, ou seja, quando a temperatura dos dissipadores dos MOSFETs ultrapassa determinado valor, os ventiladores são ligados. O módulo possui também um dispositivo de segurança que desliga o inversor caso a temperatura ultrapasse um valor máximo. Estes dispositivos proporcionam proteção térmica primária.

Além da proteção térmica primária feita pelo módulo EGS002 através de um NTC, cada inversor possui instalado um sensor de temperatura no dissipador dos MOSFETs. A Unidade Terminal Remota efetua a monitoração da temperatura com dois objetivos: supervisão e proteção secundária de temperatura. Neste caso, se a medida de temperatura dos dissipadores dos MOSFETs ultrapassar 55°C a UTR desliga o inversor correspondente.

4.7.3 Dissipação de Calor no Transformador do Inversor

O calor gerado na operação do transformador usado no inversor de um estágio reflete basicamente dois tipos de perdas:

1. As perdas devido à circulação de corrente nas resistências dos enrolamentos primário e secundário do transformador. Estas perdas em Watts são proporcionais ao quadrado da corrente média, e são dadas pela expressão: $P = RI^2$, onde R é a resistência do enrolamento em Ohms e I é a corrente média em Amperes;
2. As perdas no ferro devido à histerese e às correntes parasitas. Estas perdas tendem a ser maiores por causa da frequência de 24KHz de chaveamento dos MOSFETs. Neste caso, a escolha do tipo de ferro do núcleo do transformador pode minimizar estas perdas.

A corrente no enrolamento primário atinge valores em torno de 50 Amperes na potência máxima deste inversor, valor que ao quadrado multiplicado pela resistência do enrolamento pode gerar calor suficiente para fazer a temperatura atingir valores que podem causar danos. O aumento da temperatura também aumenta a resistência dos enrolamentos, gerando mais calor.

O transformador é um dispositivo elétrico e magnético que pode sofrer degradação no funcionamento devido a diversos fatores, tais como falhas de isolamento nos enrolamentos e

corrosão do núcleo de ferro. Estas condições de degradação tem como consequência a diminuição da eficiência, ou seja, mais energia se transforma em calor no processo de conversão de tensão e filtragem.

Cada inversor possui instalado um sensor de temperatura no transformador de núcleo de ferro. A Unidade Terminal Remota efetua a monitoração da temperatura com dois objetivos: supervisão e proteção de temperatura. Neste caso, se a medida de temperatura do transformador de núcleo de ferro ultrapassar 55°C a UTR desliga o inversor correspondente.

4.8 Desenvolvimento do Software dos Controladores Arduíno

Os softwares que rodam nos controladores Arduíno Mega 2560 são destinados à UTR e ao Concentrador. Estes softwares foram desenvolvidos obedecendo a um conjunto de diretrizes que definem basicamente: a organização do arquivo fonte; o modo de implementar conceitos de programação em tempo real e máquina de estados. A descrição destas diretrizes e das rotinas dos softwares estão no Apêndice A. Este apêndice também contém o link de acesso à pasta de armazenamento na nuvem que contém os arquivos.

O software da UTR deve realizar os procedimentos de leitura e tratamento dos sinais de todos os pontos físicos de supervisão digital e analógica. O software da UTR deve também, e simultaneamente, efetuar os procedimentos de proteção primária dos inversores, supervisão do banco de baterias, supervisão dos controladores de carga e o controle das chaves de transferência.

O software do equipamento Concentrador deve atuar entre o Nível de Rede e o Nível de Processo, efetuando a comunicação em protocolo MODBUS RTU com a UTR e dois equipamentos Controladores de Carga, no Nível de Processo, e também com os softwares do computador Raspberry PI 3 usando protocolo CoAP em rede Ethernet no Nível de Rede.

4.9 Desenvolvimento dos Softwares em Linguagem Java

Visando disponibilizar os recursos de supervisão e controle de modo universal para as aplicações Web, devem ser desenvolvidos softwares que implementem servidores HTTP para acesso local (Intranet) e remoto (Nuvem). Assim, os softwares servidores HTTP devem atender aos seguintes requisitos básicos: 1) Permitir a conexão de diversos clientes HTTP simultaneamente usando os recursos de Threads da linguagem Java; 2) Fornecer para o cliente HTTP página HTML 5 em formato tabela e script em linguagem de ampla utilização.

O procedimento de atualização dos valores das variáveis pelo servidor HTTP local deve ser feito através de comunicação direta com o Concentrador em protocolo CoAP-OSA-CBM pela rede ethernet. Este procedimento somente deve ser realizado quando houver solicitação de um ou mais clientes HTTP já conectados ao servidor local. O procedimento de atualização dos valores das variáveis do servidor HTTP em nuvem deve ser feito periodicamente por iniciativa de um software cliente HTTP rodando no computador Raspberry PI 3.

Os softwares destinados a rodar no computador Raspberry PI 3 devem realizar os procedimentos de monitoramento de condição dos equipamentos. Estes procedimentos visam detectar degradação no funcionamento dos equipamentos, e devem ser implementados através de máquinas de inferência fuzzy. Estas máquinas devem usar a linguagem FCL (Fuzzy Control Language) descrita na Norma IEC 61131-7. O pacote jFuzzyLogic_v3.0 instalado no Ambiente de Desenvolvimento Eclipse Java permite fazer programas que efetuam a execução dos comandos de um arquivo texto do tipo FCL.

4.10 Algoritmos para Detecção e Identificação de Falhas

O conceito de Gerenciamento de Saúde e Prognóstico (Prognostic and Health Management – PHM) pode ser considerado um recurso mais abrangente da Manutenção Baseada em Condição, o qual é aplicado de modo simplificado no sistema desenvolvido neste trabalho. A aplicação do PHM neste trabalho é basicamente conceitual, sendo feito de modo bastante simplificado, pois o monitoramento é feito somente com grandezas elétricas e temperatura.

De acordo com Vachtsevanos (2006), são apresentados os Recursos Funcionais do Sistema (System Functional Capabilities) dentro do conceito de Gerenciamento de Saúde e Prognóstico (PHM). Um subconjunto destes recursos, adaptado para esta aplicação em forma simplificada, é mostrado na Tabela 4.2:

Tabela 4.2 - Recursos Funcionais do Sistema de Gerenciamento de Saúde

Item	Método	Descrição
1	Detecção de Falhas	Comparação com limites pré estabelecidos.
2	Isolamento de Falhas	Ativação de Indicadores e Bloqueio de Equipamento em Falha.
3	Métodos de Diagnóstico	Indicadores de falhas e valores das variáveis de supervisão.
4	Método de Prognóstico	Cálculo do Parâmetro “Saúde” de cada Equipamento na faixa: 0 a 100%.

4.10.1 Desenvolvimento do Algoritmo para Detecção de Falhas Graves

Os procedimentos para detecção de falhas graves, os quais são denominados de Proteção Primária, são realizados pelo software do Equipamento Unidade Terminal Remota (UTR) no nível de processo. Esses procedimentos foram desenvolvidos especificamente para os equipamentos Inversores, por serem os mais suscetíveis à ocorrência de falhas.

Os procedimentos de proteção contra falhas graves implementados são basicamente algoritmos para detecção dos seguintes eventos: sobrecorrente, sobretensão, subtensão e sobretemperatura realizados por comparação com limites preestabelecidos. No caso de detecção de falha em inversor, o controlador realiza os seguintes passos:

1. Desliga o equipamento que esteja em risco imediato e o bloqueia para evitar propagação da falha;
2. Aciona um indicador, o qual identifica a falha e pode ser lido através de supervisão remota. Enquanto este indicador estiver acionado, não é possível religar o inversor que apresentou falha;
3. O indicador de falha somente por ser zerado através de comando enviado através de comando por interface HTTP ou reiniciando a UTR.

Observação: Não foi implementado recurso de religamento após falha grave por questões de segurança. Assim, o bloqueio do inversor permite que o usuário faça uma inspeção detalhada e identifique as causas da ocorrência da falha grave.

4.10.2 Desenvolvimento dos Algoritmos para Monitoramento de Condição

Considerando os equipamentos principais presentes em uma usina solar, diversos tipos de defeitos não causam uma falha grave imediatamente, mas se iniciam com uma degradação progressiva no funcionamento do equipamento. Mais especificamente em equipamentos de potência como inversores, os dispositivos que efetuam a ventilação forçada são componentes que possuem partes móveis, sujeitos a falhas mecânicas.

Por exemplo, a ocorrência de aquecimento anormal por falha parcial de ventilação reduz a vida útil do equipamento, mas não causa interrupção na operação imediatamente. Assim, é necessária uma ferramenta de análise que permita identificar estas situações. Para isso, foi escolhido o Sistema de Controle Fuzzy Baseado em Regras, no qual as saídas de controle são usadas como variáveis de monitoramento de condição, indicando em forma de medida percentual, a saúde dos equipamentos.

As regras do Sistema de Controle Fuzzy não somente modelam o sistema através das suas próprias características, mas são definidas usando os conhecimentos de um técnico sobre o processo (ROSS, 2010). A principal utilidade dos sistemas de inferência fuzzy está no fato dos resultados serem aproximados à linguagem humana, facilitando a análise pelo usuário do sistema.

Desta forma, o valor resultante de uma variável de saída fuzzy, a qual indica a condição de saúde do equipamento, é apresentada ao usuário leigo, o qual na redução do valor percentual pode solicitar a avaliação inicial feita por um técnico através da observação das variáveis de supervisão. Assim, o técnico pode fazer uma identificação prévia de falhas e recomendar procedimentos de manutenção.

Neste projeto, foram implementados procedimentos de detecção de falhas graves somente para os inversores. O monitoramento de condição foi implementado para: painéis fotovoltaicos; controladores de carga; inversores; banco de baterias.

4.11 Arquitetura de Comunicação para Supervisão e Controle

A Figura 4.10 mostra o Diagrama de Comunicação para Supervisão e Controle da usina:

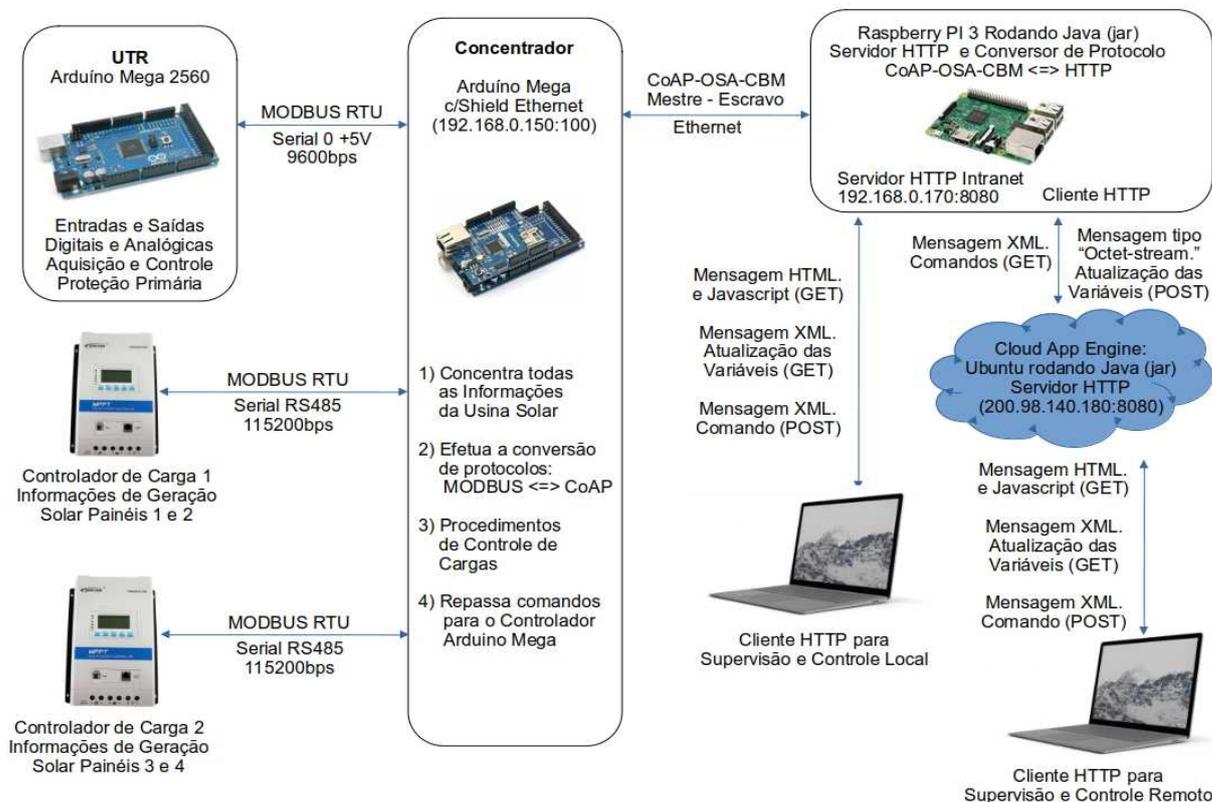


Figura 4.10 - Diagrama de Comunicação do Sistema de Supervisão, Controle e MBC

Na Figura 4.10, à esquerda estão os dispositivos do Nível de Processo, que são a Unidade Terminal Remota (UTR) e os dois Controladores de Carga. Os três dispositivos do Nível de Processo efetuam comunicação através de interface serial assíncrona em protocolo MODBUS RTU com o equipamento Concentrador Arduíno Mega 2560. O equipamento UTR possui quatro funções principais: 1. Ler todos os pontos físicos de supervisão analógicos e digitais e formatar as informações; 2. Receber solicitações de comando e atuar nas saídas digitais; 3. Efetuar os procedimentos de proteção primária dos equipamentos; 4. Efetuar os procedimentos de controle de cargas, realizando a transferência entre a rede e os inversores.

Os Controladores de Carga são responsáveis por efetuar a conversão da energia fornecida pelos painéis fotovoltaicos para a tensão do barramento principal da usina (24Vcc). Estes equipamentos são microprocessados, e possuem interface de comunicação serial assíncrona em padrão elétrico RS485, disponibilizando as informações da geração solar em registros lidos através do protocolo MODBUS RTU.

No centro da Figura 4.10 está o Concentrador, que tem a função de efetuar a comunicação em protocolo MODBUS RTU com os dispositivos da usina e armazenar todas as informações de supervisão. O Concentrador também recebe as solicitações de comando e repassa para a UTR. Neste equipamento, a placa Arduíno Mega 2560 recebe a conexão de uma placa do tipo “shield” Ethernet 10BaseT com CI Wiz5100. Este hardware em conjunto com a biblioteca EthUDP fornecem as funcionalidades de comunicação usando datagramas UDP em interface Ethernet. Neste projeto o protocolo CoAP é usado para comunicação entre o controlador Arduíno Mega 2560 e o Computador Raspberry PI 3. Assim, o Concentrador tem a função de Servidor CoAP, recebendo as requisições do computador Raspberry PI 3.

À direita está o Raspberry PI 3 que roda dois softwares que fazem dele servidor HTTP de clientes na Intranet, e cliente HTTP do Servidor em Nuvem. Na função de servidor HTTP, o Raspberry PI 3 disponibiliza uma página HTML em formato de tabela e um script Javascript que faz a atualização automática das informações através da solicitação periódica de mensagem do tipo XML. Na função de cliente HTTP, o Raspberry PI 3 envia periodicamente ao servidor em nuvem uma mensagem HTTP do tipo Octet-stream com os valores de todas as variáveis de supervisão atualizadas. A mensagem do tipo octet-stream é aproximadamente dez vezes menor em número de bytes que uma mensagem do tipo XML, tornando a atualização da base de dados do servidor em nuvem bem mais eficiente e confiável.

4.12 Protocolo de Comunicação para Supervisão e Controle

Este projeto de pesquisa foi feito dentro do conceito de Internet das Coisas (IoT), que tem por finalidade integrar os equipamentos e sistemas em rede. Esta funcionalidade permite a supervisão e controle remotos, além da funcionalidade de edição, compilação e carga de programas no controlador através da rede usando o recurso VNC.

Assim, no desenvolvimento do protocolo, escolheu-se buscar o modo mais simples e eficiente de comunicação que tivesse todas as funcionalidades oferecidas pelas redes de computadores atualmente, e que permitisse a fácil integração com o protocolo HTTP, permitindo o acesso de todo tipo de clientes, principalmente os navegadores (Browsers). A questão que surge é: quais recursos são mais adequados ao sistema, ou seja, que permitam a realização das funções desejadas com o máximo de performance dentro do hardware e software disponíveis? Qual a melhor maneira de efetuar a conversão do protocolo usado pelos controladores para o protocolo HTTP?

Desta forma, o protocolo escolhido foi o CoAP encapsulado em datagramas UDP carregando mensagens OSA-CBM em formato binário para maior simplicidade e performance. As descrições dos protocolos CoAP e HTTP estão no Anexo C – Protocolos de Comunicação.

4.13 Metodologia de Implementação do Protocolo CoAP

4.13.1 Considerações Gerais

Na implementação do sistema tema desta dissertação, o Concentrador Arduíno Mega 2560 é o Servidor em Protocolo CoAP e o Computador Raspberry PI 3 é o Cliente CoAP. Ambos são conectados por interface Ethernet 10/100BaseT e protocolo UDP. A mensagem de resposta do servidor envia todos os estados e medidas da usina. Isto é feito para otimizar o procedimento de comunicação, pois, devido à aplicação em pequenas usinas, todos os pontos de supervisão digitais e analógicos podem ser enviados em somente uma mensagem binária. O byte com o código de comando também é enviado em todas as requisições feitas pelo cliente.

As mensagens terminam com dois bytes contendo o valor de verificação de erro calculado pelo método CRC16: Verificação Cíclica de Redundância (CRC - Cyclic Redundancy Check). Este valor é calculado pelo emissor, usando todos os bytes da mensagem CoAP, sendo usado pelo receptor para verificação da mensagem. O receptor aplica o mesmo método

de cálculo sobre os bytes recebidos e compara com os dois últimos bytes da mensagem recebida. Caso a comparação seja verdadeira, indica que a mensagem recebida está correta.

4.13.2 Procedimentos de Envio e Recebimento de Mensagens

1. As mensagens de requisição do cliente para o servidor são sempre do tipo Confirmável (CON), exigindo a mensagem de resposta do servidor.
2. As mensagens de resposta do servidor para o cliente são do tipo Não Confirmável (NON). Isto é feito porque as requisições feitas pelos clientes são periódicas, ou seja, caso ocorra uma falha na recepção da mensagem com os estados e medidas da usina, logo depois é feita outra solicitação, e caso o canal de comunicação esteja funcionando normalmente, a atualização dos dados é restabelecida. Caso ocorram muitas falhas de recepção, os clientes têm procedimentos para indicar aos usuários.
3. O recurso de Token não é usado. Neste sistema, as informações sempre estão disponíveis no servidor CoAP (Concentrador Arduíno Mega 2560). Caso ocorra falha de comunicação entre o Concentrador e algum dispositivo do processo, isto é sinalizado na própria mensagem enviada para o cliente CoAP (Raspberry PI 3).

4.13.3 Definições sobre o Payload OSA-CBM do Protocolo CoAP

A especificação das mensagens OSA-CBM que mapeiam a UML abstrata podem ser montadas em uma mensagem binária, ou seja, composta de bytes com as informações inseridas da maneira mais eficiente possível. Uma mensagem binária, composta por uma sequência de bytes, é bem mais eficiente que uma mensagem no formato texto composta por caracteres ASCII (formato XML).

A eficiência na composição da mensagem binária padrão OSA-CBM e o uso dos datagramas UDP em rede Ethernet, torna este conjunto possível de ser usado por pequenos controladores como o Arduíno. Assim, é possível padronizar as mensagens destinadas à supervisão, controle e monitoramento de condição, facilitando os procedimentos de integração (MIMOSA, 2010). Esta mensagem padrão OSA-CBM é carregada no campo payload CoAP, sendo enviada por um controlador Arduíno Mega 2560 com placa Shield Ethernet.

4.14 Desenvolvimento dos Serviços WEB

Visando à máxima compatibilidade e integração, principalmente com clientes do tipo Navegadores (Browsers), que são disponíveis para computadores, celulares e tablets, foram implementados os seguintes recursos:

4.14.1 Servidor HTTP e Conversor para acesso local (Intranet)

Este software tem por finalidade aguardar a conexão de clientes HTTP locais e efetuar a comunicação com o Concentrador Arduino Mega 2560. Assim, envia comandos, requisita as informações de supervisão em protocolo CoAP-OSA-CBM, converte para XML e as repassa ao cliente HTTP. Os clientes HTTP são navegadores instalados em computadores e dispositivos móveis operando na área da rede local.

4.14.2 Servidor HTTP e Conversor para uso na Nuvem (Internet)

Este software, instalado em um computador na nuvem, tem por finalidade aguardar a conexão de dois tipos de cliente: o cliente de atualização da base de dados e os clientes HTTP remotos (navegadores).

4.14.3 Cliente HTTP para atualização dos Dados do Servidor em Nuvem

Este software tem por função atualizar periodicamente os valores das variáveis de supervisão do Servidor HTTP na nuvem, e caso seja necessário, trazer uma solicitação de comando feita do browser. Para isso, o software efetua a comunicação com o Concentrador, envia comandos caso seja solicitado, requisita as informações de supervisão e envia a mensagem CoAP-OSA-CBM para o servidor em nuvem. O envio desta mensagem binária é feita usando o protocolo HTTP com dados do tipo `application/octet-stream`.

Assim, este recurso permite a supervisão e controle de qualquer local pela Internet. A base de dados é quase idêntica à do servidor HTTP local, sendo atualizada periodicamente pelo Software Cliente HTTP para atualização dos dados do Servidor em Nuvem.

4.15 Servidor HTTP para acesso local (Intranet)

O Servidor HTTP para acesso local (Intranet) foi implementado através de um software aplicativo desenvolvido em linguagem Java. Este software é instalado na forma de um arquivo em formato executável (.jar), o qual é executado em sistema operacional Raspbian no

computador Raspberry PI 3, que opera como cliente CoAP e servidor HTTP. Este computador está instalado na usina solar, sendo acessado através de aplicativo VNC. O programa Conversor de Protocolos e Servidor HTTP para acesso local aguarda conexões na porta 8080 usando protocolo HTTP. Quando um cliente HTTP do tipo browser faz uma requisição ao servidor, ocorre a sequência de eventos mostrada nos itens a seguir:

4.15.1 Acesso do Cliente Local com Método GET /

No acesso do navegador ao endereço IP = 192.168.0.170 na porta 8080 com o método GET / (sem URI), o servidor envia uma página HTML em formato de tabela para apresentação das informações de supervisão ao usuário. No mesmo arquivo texto que contém o código HTML, existe um programa em linguagem Javascript. Este programa tem por função principal fazer a atualização periódica e de forma automática de todos os valores das variáveis de supervisão na tabela.

4.15.2 Acesso do Cliente Browser com Método GET/atualiza

Em seguida à carga no browser da página HTML juntamente com programa Javascript, este programa efetua o envio periódico ao servidor de uma mensagem de requisição usando o objeto Javascript XMLHttpRequest com o método GET seguido da URI “atualiza”. O servidor, então, responde com uma mensagem HTTP do tipo XML com os valores de todas as variáveis da usina.

Este procedimento permite a atualização da tabela de forma transparente ao usuário com tráfego em torno de dois mil caracteres. A Figura 4.11 apresenta o diagrama do fluxo de mensagens completo entre o Navegador (Browser), o Software Servidor HTTP Local e o Concentrador. A Figura 4.11 mostra o fluxo de mensagens que ocorre a cada ativação do objeto XMLHttpRequest no browser.

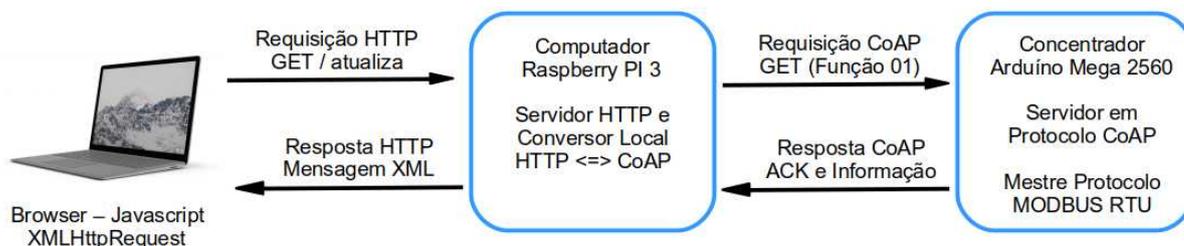


Figura 4.11 - Fluxo de Mensagens Atualiza: Browser, Servidor Local e Concentrador

Ao receber do Navegador a requisição da mensagem XML de atualização de variáveis, o programa Servidor HTTP Local envia uma solicitação ao Concentrador em protocolo CoAP. Caso tenha sido pressionado algum botão de comando na tela do Navegador pelo usuário, o código desse comando é inserido na mensagem de requisição em uma URI no formato texto “cmd=nnnn”, onde nnnn é o código de 0000 a 9999. Neste caso, o Software Servidor HTTP Local converte este código para um byte e o carrega no Payload da mensagem de requisição CoAP. Caso não exista solicitação de comando, é inserido o código com valor igual a um.

A Figura 4.12 apresenta o fluxo de mensagens entre o Concentrador à esquerda e os três dispositivos do processo à direita:

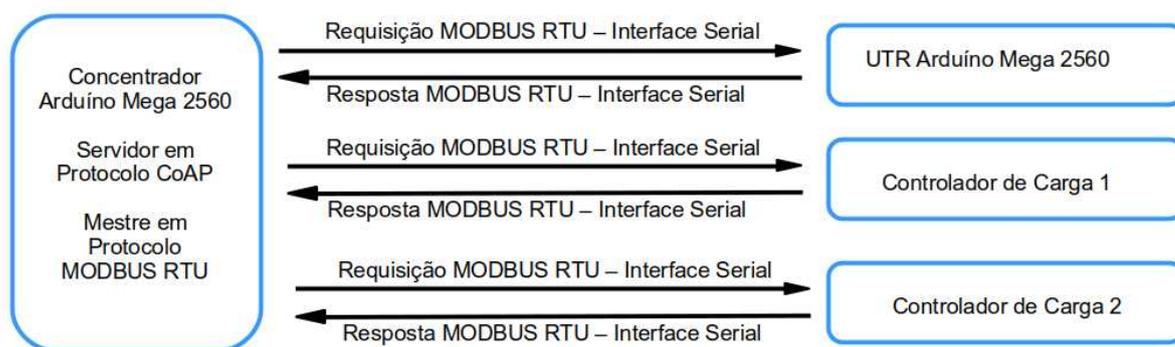


Figura 4.12 - Fluxo de Mensagens entre o Concentrador e os Dispositivos do Processo

Conforme mostra a Figura 4.12, após receber a mensagem de requisição CoAP, o Concentrador envia sequencialmente, em protocolo MODBUS RTU, uma solicitação a cada dispositivo do processo conectado por interface serial. No caso de ter sido feita uma requisição incluindo comando, este é repassado para a UTR, que é o único dispositivo do processo capaz de executar comandos digitais. Em seguida ao envio de cada solicitação, o Concentrador espera a mensagem de resposta de cada dispositivo, o que não demora mais que 250ms. Os dispositivos do processo são:

- 1 Unidade Terminal Remota (UTR);
- 2 Controlador de Carga 1;
- 3 Controlador de Carga 2.

Após receber de cada dispositivo a mensagem de resposta MODBUS RTU, o Concentrador lê e formata os dados em padrão OSA-CBM. Em seguida, envia a mensagem de resposta para o Software Servidor HTTP Local usando o protocolo CoAP-OSA-CBM.

4.16 Cliente HTTP para atualização do Servidor em Nuvem

O Cliente HTTP para atualização do Servidor em Nuvem foi implementado através de um software aplicativo desenvolvido em linguagem Java. Este software é instalado na forma de um arquivo em formato executável (.jar), o qual é executado em sistema operacional Raspbian em um computador Raspberry PI 3, o qual está instalado na usina solar, sendo acessado através de aplicativo VNC. O programa, a cada quatro segundos, efetua os procedimentos mostrados nos itens a seguir:

4.16.1 Requisição e Leitura das Informações ao Concentrador

A cada quatro segundos, o programa Cliente HTTP para atualização do Servidor em Nuvem envia uma solicitação CoAP ao Concentrador e recebe a mensagem de resposta em formato CoAP-OSA-CBM usando o mesmo procedimento mostrado no item 4.15.2.

4.16.2 Envio da Mensagem para Atualização do Servidor em Nuvem

O Software Cliente HTTP para atualização do Servidor em Nuvem, que roda no computador Raspberry PI 3, recebe do Concentrador a mensagem binária em protocolo CoAP-OSA-CBM e a envia para o Servidor HTTP em Nuvem. Isto é feito usando o tipo de mensagem application/octet-stream, que possui cabeçalho padrão HTTP composto por caracteres ASCII seguido por um área de dados binária composta por uma sequência de bytes. Esta sequência de bytes é a própria mensagem CoAP recebida do Concentrador.

4.17 Servidor HTTP na Nuvem e Conversor

O Servidor HTTP na Nuvem e Conversor para acesso remoto (Internet) foi implementado através de um software aplicativo desenvolvido em linguagem Java. Este software é instalado na forma de um arquivo em formato executável (.jar), o qual é executado em sistema operacional Ubuntu 18.04 em computador na nuvem (Virtual Machine).

O programa Servidor HTTP na Nuvem aguarda conexões na porta 8080. Este programa pode receber o acesso de dois tipos de cliente: Cliente do tipo Navegador (Browser) e Cliente HTTP para Atualização do Servidor em Nuvem. A Figura 4.13 mostra o fluxo de mensagens entre o Servidor em Nuvem e os clientes:

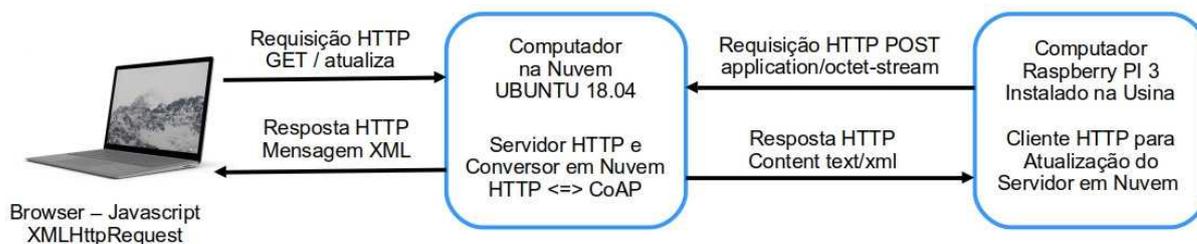


Figura 4.13 - Fluxo de Mensagens entre o Servidor HTTP em Nuvem e os Clientes

A partir do primeiro acesso feito pelo Software Cliente HTTP para Atualização do Servidor em Nuvem, ocorre a seguinte sequência de eventos apresentada nos itens a seguir:

4.17.1 Acesso do Cliente HTTP para Atualização do Servidor em Nuvem

A cada quatro segundos, o Programa Servidor HTTP na Nuvem recebe uma mensagem do tipo octet-stream (binária) enviada pelo Software Cliente HTTP para atualização do Servidor em Nuvem rodando no computador Raspberry PI 3. Esta mensagem, em sua área de dados, contém a resposta do Concentrador em protocolo CoAP-OSA-CBM.

Desta forma, o software do servidor HTTP no computador em nuvem atualiza a sua base de dados (variáveis de supervisão) com as informações recebidas na mensagem binária. Em seguida, este software converte estas informações numéricas para texto e monta a mensagem de atualização dos clientes HTTP em formato XML.

4.17.2 Acesso do Cliente Remoto com Método GET /

Quando um navegador acessa o endereço IP = 200.98.140.180 na porta 8080 de qualquer lugar, com o método GET / (sem URI), o servidor envia o código HTML em formato de tabela para apresentação das informações de supervisão ao usuário. No mesmo arquivo do código HTML é fornecido um programa em linguagem Javascript. Existe uma diferença entre as tabelas para acesso local e remoto. A tabela fornecida com acesso remoto possui um campo denominado “Conexão Atualiza”, onde é apresentado o estado de comunicação do Software Cliente para Atualização do Servidor HTTP na Nuvem. Caso o Servidor HTTP em Nuvem pare de receber mensagens de atualização por mais de oito segundos, este campo na tabela deve indicar falha na conexão.

4.17.3 Acesso do Cliente Navegador com Método GET /atualiza

Em seguida à carga no browser da página HTML juntamente com programa Javascript, este programa efetua o envio periódico ao servidor na nuvem de uma mensagem de requisição usando o objeto Javascript XMLHttpRequest com o método GET seguido da URI “atualiza”.

O Programa Servidor HTTP na Nuvem responde à requisição Javascript rodando no navegador enviando uma mensagem HTTP do tipo text/xml contendo os valores atualizados de todas as variáveis de supervisão. Esta mensagem XML contém o valor atualizado de todas as variáveis de supervisão a ser enviada para o navegador. A solicitação da mensagem XML pelo navegador é feita pelo objeto Javascript XMLHttpRequest.

4.18 Monitoramento de Condição dos Inversores

Os Inversores são os equipamentos mais complexos e sujeitos a falhas de um sistema de geração fotovoltaica, pois, estão diretamente conectados à rede elétrica de baixa tensão. Assim, estão sujeitos a sobrecargas, curto-circuitos e transientes de tensão em sua saída. Além destas condições anômalas, o próprio funcionamento do inversor gera calor que deve ser dissipado com o auxílio de ventiladores, que são dispositivos mecânicos, portanto, mais sujeitos a falhas.

Desta forma, os procedimentos de monitoramento de condição desenvolvidos para os inversores foram baseados nos seguintes princípios: 1) A proteção primária com atuação direta para desligar o inversor em caso de falhas graves; 2) A proteção secundária que usa lógica fuzzy para detectar condições anômalas de operação, as quais causam estresse e podem levar o equipamento a sofrer danos mais graves.

4.18.1 Sinais para Detecção de Falhas e Monitoramento de Condição

Nos inversores SPWM de um estágio, as informações de supervisão analógicas obtidas através dos sensores e condicionadores são: 1) Tensão CC de Entrada; 2) Corrente CC de Entrada; 3) Tensão CA de Saída; 4) Corrente CA de Saída; 5) Temperatura dos Dissipadores dos MOSFETs; 6) Temperatura do Transformador.

A partir das tensões e correntes, são calculadas as demais informações analógicas que são: potência de entrada, potência de saída e eficiência. Juntamente com as informações analógicas existem as informações digitais que somente informam estado 1 ou 0.

As informações de supervisão são usadas em três funções principais: 1) Detectar e realizar as ações necessárias no caso de falhas muito graves; 2) Efetuar o diagnóstico e tomar as ações necessárias no caso de Monitoramento de Condição; 3) Disponibilizar as informações de operação do equipamento para supervisão e controle. Estas informações são mostradas a seguir:

1. **Tensão de Entrada:** o valor da tensão de entrada do inversor deve estar dentro da faixa especificada: Limite Mínimo = 21Vcc e Limite Máximo = 30Vcc. Se estiver abaixo do limite mínimo, o circuito de regulação de tensão de saída perde a capacidade de manter a tensão de saída em seu valor nominal $V_s = 220V_{ca}$ RMS;
2. **Corrente de Entrada:** é usada para detecção de sobrecorrente instantânea e para calcular a potência e a energia consumida;
3. **Potência de Entrada:** é calculada a partir da tensão e da corrente de entrada, sendo usada nos procedimentos de detecção de perdas excessivas e sobrecarga;
4. **Tensão de Saída:** possui valor regulado através de realimentação, mantendo-se no valor de 220Vca RMS com variação máxima de $\pm 2\%$ em operação normal;
5. **Corrente de Saída.** É usada para calcular a potência e a energia fornecida às cargas, as quais fazem parte dos procedimentos de detecção de perdas e sobrecarga;
6. **Potência de Saída:** é calculada a partir da tensão e corrente de saída, sendo usada nos procedimentos de detecção de perdas e sobrecarga;
7. **Eficiência do Inversor:** é calculada a partir da potência consumida e da potência fornecida. Permite detectar diversos tipos de anomalias que causam dissipação extra;
8. **Temperatura do Módulo de Potência.** Esta medida é feita no dissipador de calor dos MOSFETs de potência da ponte completa, sendo útil nos procedimentos de detecção de sobrecarga ou defeito. Existem dois ventiladores para os dissipadores que são acionados automaticamente por um circuito termostato. Este circuito liga os ventiladores quando a temperatura ultrapassa os 45°C e desliga quando a temperatura cai abaixo de 41°C;
9. **Temperatura do Transformador.** Esta medida é feita no ferro do transformador em um ponto mais próximo dos enrolamentos, sendo útil nos procedimentos de detecção de sobrecarga ou defeito. Existe um ventilador do transformador que é acionado automaticamente por um circuito termostato. Este circuito liga o ventilador quando a temperatura ultrapassa os 45°C e desliga quando a temperatura cai abaixo de 41°C.

4.18.2 Tratamento das Falhas Graves

Os procedimentos de tomada de decisão e ações relativas às falhas muito graves são feitos por hardware ou pelo software da UTR usando lógica binária. Trata-se de condições extremas, onde é necessário efetuar o imediato desligamento do inversor para evitar o agravamento e a propagação dos danos. Estes procedimentos são os seguintes:

Falha Grave 1: Sobrecorrente

Descrição: o valor da corrente na entrada de alimentação do inversor ultrapassou o limite máximo permitido. Este limite máximo da UTR para detecção de sobrecorrente deve ser menor que três vezes o valor nominal do disjuntor de entrada. Isto ocorre porque este disjuntor possui curva de atuação B e corrente nominal de 32 Amperes. Na curva B, a proteção térmica atua até três vezes a corrente nominal e acima deste valor atua a proteção instantânea. Portanto, a sobrecarga detectada pela UTR está na faixa 32 a 96 Amperes, porque acima deste valor, a proteção instantânea do disjuntor atua antes da UTR.

Prováveis Causas: sobrecarga por excesso de corrente na entrada do inversor.

Modo de Detecção: circuito detector de sobrecorrente na entrada do inversor.

Ações imediatas: desligamento do inversor através do acionamento da entrada de sobrecorrente I_{fb} (Current Feedback) do Módulo de Comando EGS002 por circuito detector.

Indicação: após a ocorrência e detecção da falha, o indicador na memória do controlador permanece ativado até ser zerado por comando.

Recuperação: após ao primeiro desligamento, o módulo de comando tenta partir por mais duas vezes, pois, o curto pode ter sido momentâneo. Caso a falha persista, o módulo de comando mantém o inversor desligado até que o módulo de comando seja desligado e ligado novamente.

Ações de Manutenção: se a sobretensão persiste mesmo com a saída desconectada, a causa mais provável é um ou mais MOSFETs estarem em curto. Portanto, deve ser providenciada a substituição destes componentes. A falha do módulo de comando é muito rara, entretanto, todos os sinais de controle devem ser verificados.

Falha Grave 2: Sobretensão Máxima na Saída

Descrição: a tensão CA de saída do inversor ultrapassou o limite máximo permitido.

Prováveis Causas: defeito no inversor.

Modo de Detecção: comparação da medida de tensão com o valor máximo pelo controlador.

Ações imediatas desligamento do inversor pelo controlador.

Indicação: após a ocorrência e detecção da falha, o indicador na memória do controlador permanece ativado até ser zerado por comando.

Recuperação: o inversor permanece desligado até que o indicador de falha no software do controlador seja zerado.

Ações de Manutenção: provável defeito no circuito de realimentação da tensão de saída para a entrada V_{fb} (Voltage Feedback) do Módulo de Comando EGS002. Somente ligar o inversor após o defeito estar corrigido, porque a ocorrência de sobretensão na saída do inversor pode danificar todos os equipamentos alimentados.

Falha Grave 3: Subtensão Mínima na Saída

Descrição: a tensão CA de saída do inversor caiu abaixo do limite mínimo permitido.

Prováveis Causas: subtensão na entrada, sobrecarga na saída ou defeito no inversor.

Ações imediatas desligamento do inversor pelo controlador.

Indicação: após a ocorrência e detecção da falha, o indicador na memória do controlador permanece ativado até ser zerado por comando.

Recuperação: o inversor permanece desligado até que o indicador de falha no software do controlador seja zerado.

Ações de Manutenção: se as causas não forem externas como subtensão na entrada ou sobrecarga, é defeito no inversor. A subtensão pode ter basicamente três causas: 1) Defeito no circuito de realimentação da tensão de saída para a entrada V_{fb} (Voltage Feedback) do Módulo de Comando EGS002; 2) Defeito nos MOSFETs; 3) Problema com os sinais de controle dos MOSFETs. Somente ligar o inversor após o defeito estar corrigido, pois a ocorrência de subtensão na saída do inversor pode causar mau funcionamento dos equipamentos alimentados.

Falha Grave 4: Sobretemperatura Máxima dos Dissipadores dos MOSFETs

Descrição: a temperatura nos dissipadores dos MOSFETs ultrapassou o limite máximo permitido.

Prováveis Causas: falha na ventilação forçada.

Modo de Detecção: comparação da medida de temperatura com o valor máximo.

Ações imediatas desligamento do inversor pelo controlador.

Indicação: após a ocorrência e detecção da falha, o indicador na memória do controlador permanece ativado até ser zerado por comando.

Recuperação: o inversor permanece desligado até que o indicador de falha no software do controlador seja zerado.

Ações de Manutenção: verificar funcionamento do circuito termostato e dos ventiladores dos dissipadores dos MOSFETs. Somente ligar o inversor após o defeito estar corrigido, porque a ocorrência de sobretemperatura nos dissipadores pode danificar os MOSFETs.

Falha Grave 5: Sobretemperatura Máxima do Transformador

Descrição: a temperatura no transformador ultrapassou o limite máximo permitido.

Prováveis Causas: falha na ventilação forçada ou defeito no transformador.

Modo de Detecção: comparação da medida de temperatura com o valor máximo.

Ações imediatas desligamento do inversor pelo controlador.

Indicação: após a ocorrência e detecção da falha, o indicador na memória do controlador permanece ativado até ser zerado por comando.

Recuperação: o inversor permanece desligado até que o indicador de falha no software do controlador seja zerado.

Ações de Manutenção: verificar funcionamento do circuito termostato e dos ventiladores do transformador. Verificar o funcionamento do transformador. Somente ligar o inversor após o defeito estar corrigido, porque a ocorrência de sobretemperatura nos dissipadores pode danificar os MOSFETs.

4.18.3 Falhas que causam Degradação no Funcionamento do Inversor

As falhas menos graves, que causam a degradação progressiva no funcionamento do inversor são mais difíceis de detectar e diagnosticar. Portanto, faz-se necessário efetuar o monitoramento de condição usando o recurso de lógica fuzzy com o objetivo de realizar ações de detecção, que podem ser corretivas ou preventivas, visando os seguintes objetivos principais: 1) Evitar que condições anormais de operação causem danos; 2) Identificar defeitos para auxiliar os procedimentos de manutenção.

O método para fazer o monitoramento de condição do inversor baseia-se no uso do conhecimento especialista para montar um conjunto de variáveis de entrada fuzzy, regras fuzzy e variáveis de saída fuzzy para compor um sistema de inferência Fuzzy baseado em regras. Este sistema de inferência tem por objetivo principal a implementação dos

procedimentos automáticos de identificação das condições de falha em um inversor de um estágio. A seguir são apresentados os tipos de falha já identificados que causam degradação no funcionamento:

Falha 1: Sobreaquecimento dos Dissipadores dos MOSFETs

Descrição: aquecimento anormal dos dissipadores dos MOSFETs

Prováveis Causas: 1) Mal funcionamento dos ventiladores dos dissipadores, pois os ventiladores possuem parte móvel e estão sujeitos a poeira, resíduos e desgaste; 2) Falha de um dos MOSFETs de um conjunto de dois em paralelo.

Modo de Detecção: detecção de aquecimento anormal para as condições de operação.

Indicação: LED indicador, mensagem no supervisório e evento com data/hora da ocorrência.

Ações de Manutenção: verificar o funcionamento dos ventiladores, do sensor e do circuito termostato.

Falha 2: Sobreaquecimento do Transformador

Descrição: aquecimento anormal do transformador

Prováveis Causas: Mal funcionamento dos ventiladores dos dissipadores, pois os ventiladores possuem parte móvel e estão sujeitos a poeira, resíduos e desgaste.

Modo de Detecção: detecção de aquecimento anormal para as condições de operação.

Indicação: LED indicador, mensagem no supervisório e evento com data/hora da ocorrência.

Ações de Manutenção: verificar o funcionamento dos ventiladores, do sensor e do circuito termostato.

Falha 3: Sobreaquecimento Geral

Descrição: aquecimento anormal dos dissipadores dos MOSFETs e do transformador

Prováveis Causas: Muita potência sendo fornecida por muito tempo, sobrecarga.

Modo de Detecção: detecção de aquecimento normal para as condições de operação.

Indicação: LED indicador, mensagem no supervisório e evento com data/hora da ocorrência.

Ações de Manutenção: verificar a causa do excesso de potência sendo fornecida.

Falha 4: Baixo Rendimento

Descrição: relação baixa entre a potência fornecida e a potência consumida.

Prováveis Causas: 1) Defeito no transformador; 2) Defeito no módulo de potência.

Modo de Detecção: comparação entre as medidas de potência de entrada e saída.

Indicação: LED indicador, mensagem no supervisório e evento com data/hora da ocorrência.

Ações de Manutenção: verificar a causa da falha. Realizar todos os testes de desempenho.

Falha 5: Tensão de Saída Alta

Descrição: a tensão de saída tende a permanecer acima do valor nominal.

Prováveis Causas: 1) Calibração fora no circuito de realimentação; 2) Interferência no sinal de realimentação.

Modo de Detecção: verificação da tensão de saída

Indicação: LED indicador, mensagem no supervisório e evento com data/hora da ocorrência.

Ações de Manutenção: verificar a calibração da realimentação de tensão.

Falha 6: Tensão de Saída Baixa

Descrição: a tensão de saída tende a permanecer abaixo do valor nominal.

Prováveis Causas: 1) Calibração fora no circuito de realimentação; 2) Tensão de entrada baixa.

Modo de Detecção: verificação da tensão de saída

Indicação: LED indicador, mensagem no supervisório e evento com data/hora da ocorrência.

Ações de Manutenção: 1) Verificar a calibração da realimentação de tensão; 2) Verificar as condições da tensão de entrada.

4.19 Apresentação das Informações de MBC do Inversor

O valor da função de pertinência a cada conjunto da variável de saída mostra o resultado Fuzzy do procedimento de inferência. Os resultados da aplicação da defuzzificação à variável de saída SaudeInversor é apresentada de forma numérica em porcentagem para o usuário. Esta informação, não técnica, permite que usuários não técnicos possam identificar uma condição de operação anômala de um equipamento. Assim, a variável Saúde tem seu valor na faixa de 0 a 100%, onde os valores mais próximos de 100 indicam operação normal.

No caso da Operação do Inversor, o valor defuzzificado informa a “saúde” do equipamento, ou seja, quanto menor este valor, menor a “saúde” e maiores são as evidências de funcionamento anormal. O valor das pertinências aos conjuntos é uma indicação deste nível de saúde. Vários defeitos podem ocorrer com um inversor antes da interrupção da operação. Os mais comuns são falhas nos ventiladores, sobrecarga na saída e falha em um transistor do driver. Assim, os procedimentos fuzzy permitem diferenciar um aquecimento normal devido às condições de carga de um aquecimento anormal devido à degradação no funcionamento do inversor.

4.20 Monitoramento de Condição dos Painéis Fotovoltaicos

Para efetuar a verificação da capacidade de geração dos painéis fotovoltaicos, são necessárias duas informações essenciais: a irradiância e a potência máxima fornecida. Estas duas informações devem ser obtidas seguindo o conceito OSA-CBM nas camadas 1 e 2: aquisição de dados e processamento de sinal. Desta forma, faz-se necessário um recurso independente que permita obter o valor de irradiância no conjunto de painéis, para que seja possível a detecção de anomalias referentes à capacidade de geração.

Os problemas mais comuns que causam redução da geração são o aquecimento excessivo e o bloqueio da luz solar por poeira ou sombras sobre os painéis. O aquecimento pode ser causado somente por condições climáticas ou também por problemas de instalação. Os problemas críticos são defeitos nos painéis que podem ser causados por descargas atmosféricas. Neste último caso deve ser feita a substituição dos mesmos.

4.20.1 Procedimento de MBC dos Painéis em Usinas do tipo “Off-Grid”

Em usinas do tipo off-grid, cada conjunto de painéis fotovoltaicos é conectado a um equipamento Controlador de Carga. Este equipamento tem por função receber a energia de um conjunto de painéis e entregá-la na tensão e corrente requeridos pelas cargas e pelo banco de baterias. Assim, a potência solicitada dos painéis fotovoltaicos pelo controlador de carga depende do estado de carga do banco de baterias e do consumo das cargas, ou seja, o controlador precisa solicitar potência aos painéis para que estes forneçam. Desta forma, o monitoramento de condição dos painéis fotovoltaicos em usinas do tipo off-grid precisa ser feito em conjunto com a demanda de potência da usina.

4.20.2 Procedimento de MBC dos Painéis em Usinas do tipo “On-Grid”

Em usinas do tipo on-grid o conjunto de painéis é ligado normalmente a equipamentos integrados com função MPPT e Inversor, no qual a saída de corrente alternada do inversor é conectada diretamente à rede do consumidor. Neste caso, é sempre exigida a potência máxima dos painéis, já que nesse tipo de sistema quando o consumidor não demanda toda a potência fornecida, a energia excedente é injetada na rede da concessionária. Portanto, a verificação de anomalias na geração no caso de usinas on-grid é mais simples, porque necessita apenas do cálculo da potência máxima que pode ser entregue pelos painéis fotovoltaicos, a qual é comparada com a energia injetada pelo inversor na rede.

4.20.3 Método para obtenção da Irradiância

Para efetuar a verificação da capacidade de geração dos painéis fotovoltaicos, é necessária a instalação de um recurso que permita fazer uma medida aproximada da irradiância no seu valor máximo. A irradiância na unidade W/m^2 , é a potência incidente do Sol por área no local da instalação dos painéis. Desta forma, a medida aproximada da irradiância máxima permite estimar o valor de potência máximo esperado. Este é um recurso muito simples para obtenção da irradiância. Por isso, somente o valor máximo de potência fornecido pelo pequeno painel pode ser considerado aproximadamente correto.

A medida aproximada da irradiância máxima é realizada através da medição da potência de saída de dois pequenos painéis fotovoltaicos de 1,25Wp. Cada pequeno painel deve ser instalado em local livre e no mesmo ângulo de inclinação em relação ao Sol do conjunto de painéis fotovoltaicos principais a ser monitorado, recebendo, portanto, a mesma quantidade de energia solar por área. A instalação também deve ser feita em local de fácil acesso para permitir periodicamente a inspeção e limpeza. Este tipo de painel, com dimensões de 69 x 110mm, fornece a potência máxima de 1,25Wp com tensão de saída próxima de 5Vcc e corrente de 0,25A. Portanto, ao ser usado um resistor $R = 20$ Ohms em paralelo com a saída do painel, obtém-se $P =$ potência gerada, normalizada na escala de 0 a 100%. é dada pela Expressão 4.1:

$$P = (100 * V^2) / 25 \quad (4.1)$$

Assim, na ocorrência de irradiância próxima de 100% indicada pelos pequenos painéis, é calculada a potência máxima que pode ser fornecida por cada conjunto de painéis. A Figura 4.14 mostra o pequeno painel fotovoltaico de 1,25Wp / 5Vcc / 0,25A:



Figura 4.14 - Painel Fotovoltaico 5W

4.21 Monitoramento de Condição do Banco de Baterias

No caso das usinas híbridas, o banco de baterias serve para fornecer energia durante os períodos em que a geração solar não está disponível ou não é suficiente para alimentar totalmente as cargas na ocorrência de interrupção no fornecimento pela concessionária. Isto ocorre à noite, quando o céu está nublado ou quando há um aumento repentino de carga devido ao acionamento de algum equipamento com alta corrente de partida.

Neste sistema, duas fontes de alimentação e um circuito limitador de corrente mantém o banco de baterias sempre em flutuação quando não há geração fotovoltaica, deixando o seu uso para quando forem necessárias. Este procedimento aumenta ao máximo o tempo de vida útil do banco de baterias. Neste trabalho foi constatado que a energia armazenada neste tipo de bateria custa mais caro que a energia fornecida pela concessionária. Portanto, à noite, em operação normal, quando não há geração fotovoltaica, as cargas são alimentadas pela rede.

4.21.1 Definição da Curva Teórica de Descarga do Banco de Baterias

O Modelo de Descarga é o mais usado para descrever o funcionamento das baterias. Ele é expresso através de um gráfico da tensão da bateria em função do tempo para alguns valores padronizados de corrente de descarga. Assim, para criar de uma metodologia para estimar o estado de saúde das baterias, é preciso definir um modelo teórico da variação da tensão em função do tempo para um determinado valor de corrente média de descarga.

O banco de baterias usado na usina possui seis baterias estacionárias de chumbo ácido marca Freedom modelo DF2000, cada uma com tensão de 12 Volts e capacidade de 115Ah. As baterias são conectadas na forma de dois conjuntos em série de três baterias conectadas em paralelo. A conexão em paralelo dos conjuntos é um esquema usado para equalizar a tensão entre os grupos. O diagrama elétrico do banco de baterias é mostrado na Figura 4.15:

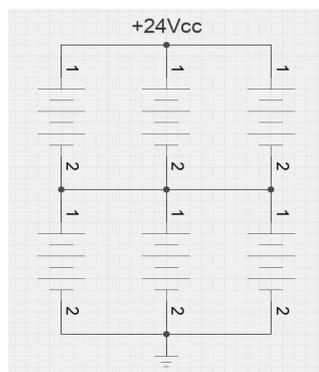


Figura 4.15 - Esquemático do Banco de Baterias

Para viabilizar o cálculo da estimativa de saúde do banco de baterias, é necessário usar um valor de corrente média de descarga que corresponda às cargas usadas no período de uso das baterias, normalmente à noite, quando não há geração solar. Assim, usando as informações de supervisão coletadas durante as faltas de energia, foi determinado o valor de dez Amperes para a corrente média de descarga. Este valor é apropriado para alimentar as cargas, correspondendo a uma potência média de 250 Watts.

A Figura 4.16 apresenta o Gráfico de Curvas de Descarga – Corrente Constante – DF2000 referente à bateria Freedom DF2000 com três curvas, fornecidas pelo fabricante, para uma bateria de 12V / 115Ah, com a escala horizontal de tempo, dada em horas. A curva mais à esquerda corresponde a uma corrente de descarga de 25 Amperes, a do meio a uma corrente de descarga de 10 Amperes, e a curva da direita, a uma corrente de descarga de 5 Amperes.

Considerando uma corrente média de descarga do banco de baterias de 10 Amperes, que corresponde à potência solicitada pelas cargas no período noturno, o valor de 5 Amperes foi escolhido. O banco de baterias usa três baterias em paralelo, portanto, a curva de 5 Amperes, à direita, mostrada na Figura 4.16 corresponde, na verdade, a uma corrente de descarga total de 15 Amperes. Foi definido que a tensão final de descarga de uma bateria é 11,50V, que corresponde a um tempo de descarga de 18:30h = 66600 segundos. Este valor de tensão mínima foi escolhido para evitar descarga excessiva e a diminuição da vida útil da bateria. Consultando o gráfico, o tempo de descarga no qual a tensão chega a 11,5 Volts é de 18:30 horas = 66600 segundos.

Por outro lado, a corrente de descarga média foi definida como sendo igual a 10 Amperes, portanto, o tempo obtido do gráfico deve ser multiplicado por 15 / 10, ou seja, o tempo de descarga do banco de baterias com uma corrente média de 10 Amperes é igual a $1,5 * 66600$ segundos = 99900 segundos.

Desta forma, a curva de cinco Amperes deve ser usada para definir os parâmetros do modelo teórico que descreve o comportamento das baterias em descarga. Assim, esse modelo será usado para comparação com os dados experimentais com o objetivo de calcular a saúde do banco de baterias em valor percentual.

CURVAS DE DESCARGA - CORRENTE CONSTANTE - DF2000

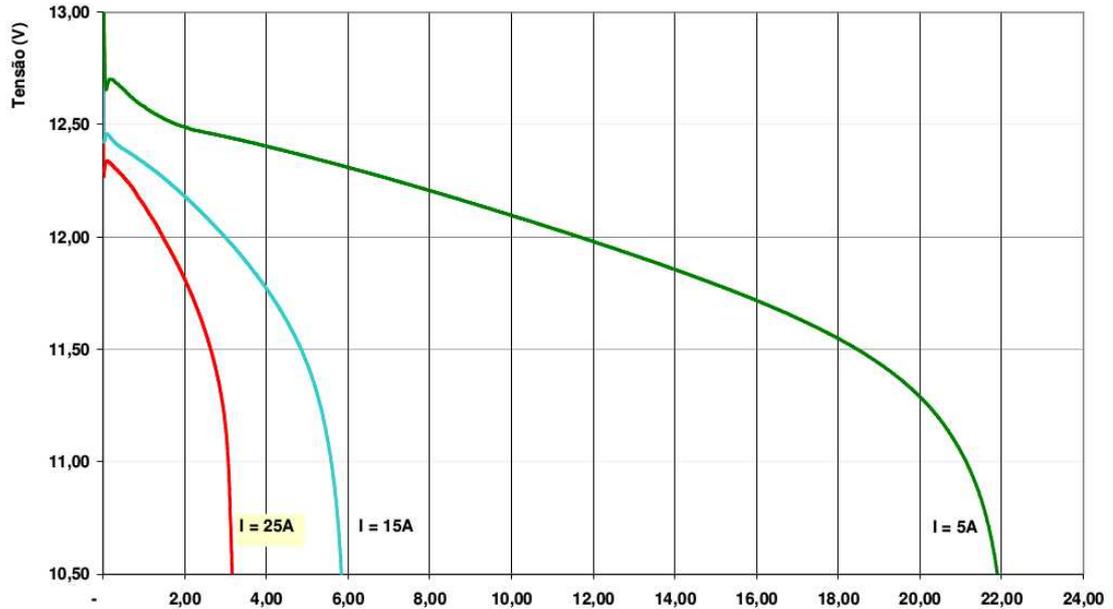


Figura 4.16 - Exemplo de Curva de Descarga para Bateria de 12V 115Ah

A definição do valor mínimo de tensão de descarga igual a 11,50 Volts para uma bateria, ou de 23,0 Volts para o banco de baterias, simplifica os mecanismos de análise, pois, a curva de descarga se aproxima de uma reta. Portanto, considerando que a curva de descarga nesta faixa é uma reta, obtemos a Expressão 4.3 referente à tensão do banco de baterias em função do tempo para uma descarga de 10 Amperes:

$$V_{bat} = V_i \left(1 - \frac{(V_i - V_m)t}{V_i T} \right) \quad 4.3$$

Onde:

V_{bat} : tensão do banco de baterias em Volts;

$V_i = 25,20$ Volts: tensão inicial do banco de baterias (baterias carregadas);

$V_m = 23,00$ Volts: tensão final do banco de baterias (baterias descarregadas);

$T = 99900$ s: tempo de descarga máximo em segundos.

t : tempo de descarga em segundos.

O gráfico da Figura 4.17 mostra a reta teórica de descarga do banco de baterias para a corrente média de descarga de 10 Amperes. A escala vertical é a tensão do banco de baterias na faixa de 20,00 a 26,25 Volts.

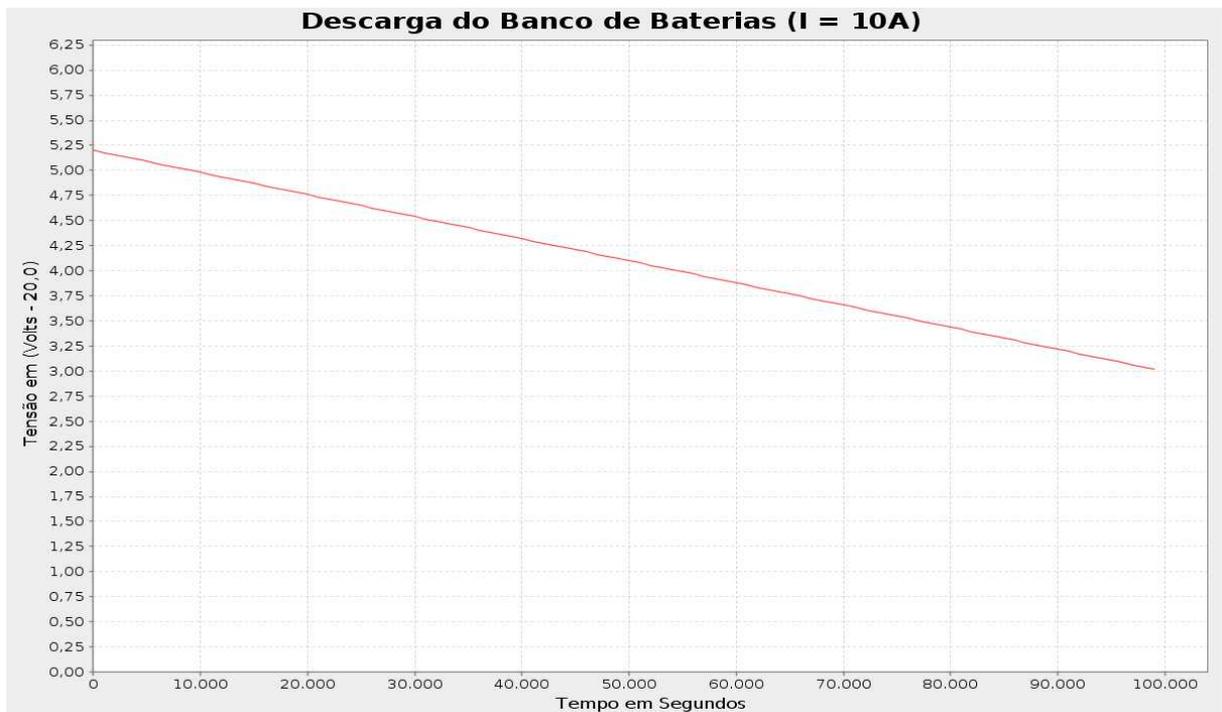


Figura 4.17 - Curva Teórica de descarga do Banco de Baterias para a corrente de 10 Amperes

4.21.2 Metodologia para Estimativa da Saúde do Banco de Baterias

É preciso desenvolver um procedimento para comparar a reta de descarga teórica com a reta de descarga obtida através das medições. O procedimento deve ser simples e eficiente, representando de modo realista o estado de armazenamento de carga das baterias. Normalmente, a descarga do banco de baterias ocorre durante uma interrupção no fornecimento de energia da rede. O software da UTR faz registro em memória da tensão do banco de baterias exatamente antes do início ($t = 0$), que corresponde às baterias carregadas. Da mesma forma, é feito o registro do tempo e da tensão exatamente antes do fim do período de descarga. Durante a descarga é calculada a média da corrente consumida pelas cargas.

O algoritmo desenvolvido gera uma reta de descarga no gráfico com o valor de tensão no eixo vertical e o valor do tempo no eixo horizontal. Isto é feito desde o momento inicial ($t = 0$), até o momento final da descarga das baterias ($t = T_f$). A projeção da reta de descarga real, ao cruzar o eixo horizontal, fornece uma estimativa do Tempo Máximo Projetado de Descarga das Baterias (T_p). Assim, ao fim do período de descarga, o software da UTR tem estas informações armazenadas, as quais permitem fazer os cálculos para estimar o T_p . A projeção da reta de descarga real pode ser vista na Figura 4.18 no formato tracejado. A corrente média de descarga é igual a 10 Amperes.

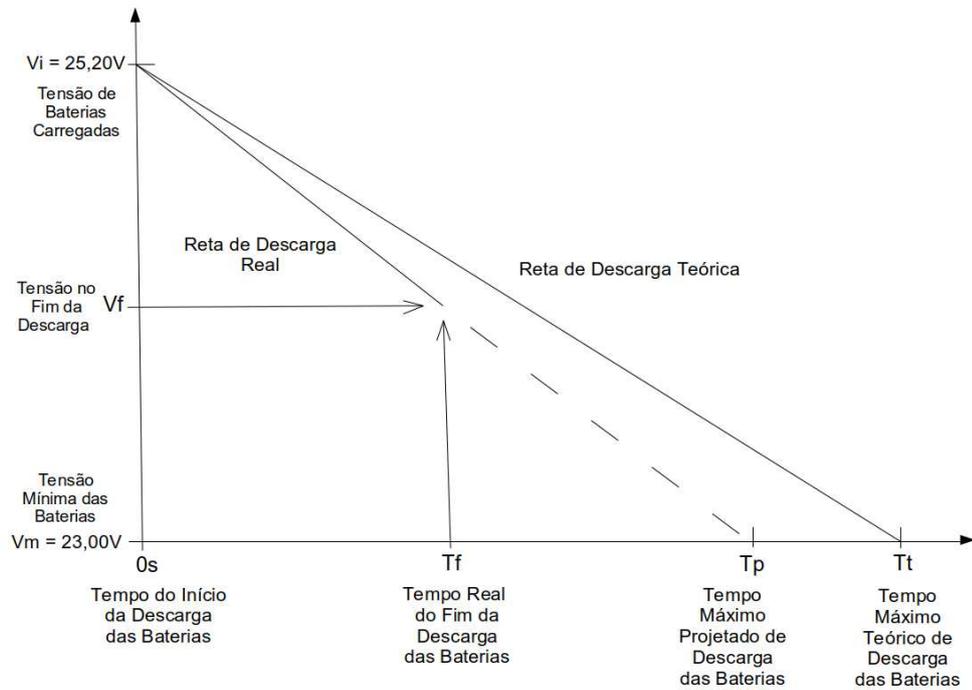


Figura 4.18 - Comparação da Reta de Descarga Real com a Reta de Descarga Teórica

Para realizar o cálculo da saúde do banco de baterias, primeiramente é preciso calcular a estimativa do tempo máximo de descarga (T_p) através dos dados de supervisão. Isto é feito projetando-se a reta de descarga real no eixo horizontal (tempo) através da Expressão 4.4 :

$$T_p = \frac{T_f (V_i - V_m)}{V_i - V_f} \quad (4.4)$$

Onde:

- T_f = Tempo de descarga do banco de baterias;
- V_i = Tensão inicial com o banco de baterias carregado (25,20V);
- V_m = Tensão mínima do banco de baterias (23,00V);
- V_f = Tensão do banco de baterias no fim do período de descarga.

A expressão para estimar a saúde do banco de baterias deve efetuar a comparação entre o Tempo Máximo Projetado de Descarga das Baterias (T_p) e o Tempo Máximo Teórico de Descarga das Baterias (T_t), ambos mostrados na Figura 4.14. As condições de contorno do estado de saúde do banco de baterias são as seguintes:

Condição 1: se $T_p = 0 \Rightarrow$ Saúde = 0%;

Condição 2: se $T_p = T_t \Rightarrow$ Saúde = 100%.

A Condição 1 significa que o banco de baterias não retém praticamente nenhuma carga, resultando em um valor de saúde igual a zero. Isto resulta do fato da tensão cair abaixo do

valor mínimo em um tempo muito curto quando é solicitada a corrente de descarga. A Condição 2 significa: O banco de baterias armazena carga da mesma forma que baterias novas com uma corrente média de descarga igual ou menor a 10 Amperes, resultando em um valor de saúde igual a 100%. Desta forma, o cálculo da saúde do banco de baterias (S_{bb}) é mostrada na Expressão 4.5:

$$S_{bb} = 100 * \left(1 - \frac{(T_t - T_p)}{T_t} \right) \quad (4.5)$$

4.22 Monitoramento de Condição dos Controladores de Carga

Os Controladores de Carga recebem a energia proveniente de um ou mais painéis fotovoltaicos, convertem para o padrão de tensão de corrente contínua e fornecem energia para alimentar as cargas e carregar o banco de baterias. São equipamentos microprocessados que otimizam a corrente consumida dos painéis fotovoltaicos e realizam todos os procedimentos para o correto carregamento das baterias através do controle da tensão de saída. A verificação do funcionamento destes equipamentos baseia-se na análise da eficiência de conversão da energia fornecida pelos painéis e da energia fornecida pelo controlador de carga para o barramento principal de corrente contínua. Normalmente nesses equipamentos de última geração, a eficiência de conversão é superior a 97%.

Assim, a verificação da condição de funcionamento dos controladores de carga é feita verificando a relação entre a potência de entrada fornecida pelo conjunto de painéis e a potência de saída fornecida para o barramento principal de 24Vcc. Estas medidas são fornecidas pelo Controlador de Carga ao Concentrador através de comunicação serial assíncrona em interface elétrica RS485. Portanto, a saúde de um equipamento controlador de carga (S_{cc}) é dada pela Expressão 4.6:

$$S_{cc} = 100 * \left(\frac{P_s}{P_g} \right) \quad (4.6)$$

Onde:

P_g : Potência em Watts Gerada pelo conjunto de Painéis Fotovoltaicos conectados à entrada do Controlador de Carga;

P_s : Potência em Watts entregue pelo Controlador de Carga ao Barramento Principal de Corrente Contínua.

5 - Implementação

Neste capítulo são apresentados os procedimentos para implementação dos seguintes recursos: 1) Projeto dos circuitos de condicionamento de sinal e dos conversores; 2) Montagem das Mensagens do Protocolo CoAP-OSA-CBM; 3) Montagem das mensagens em Protocolo HTTP; 4) Montagem do arquivo FCL (Fuzzy Control Logic).

5.1 Implementação das Camadas 1 e 2 do Modelo OSA-CBM

Seguindo o conceito OSA-CBM, a Camada 1 do modelo corresponde aos procedimentos de aquisição de dados. As informações de supervisão analógicas e digitais são obtidas através dos sensores e condicionadores de sinal. Neste projeto, existem três grandezas básicas que são adquiridas através de sensores específicos: tensão, corrente e temperatura. O sinal de saída de cada sensor é aplicado a um circuito condicionador de sinal, o qual tem por finalidade fornecer um valor de tensão na faixa de zero a +5 Volts, valor compatível com as entradas analógicas do controlador Arduino Mega 2560.

A Camada 2 do modelo OSA-CBM corresponde ao processamento de sinal destes dados. A UTR Arduino Mega 2560 realiza basicamente três procedimentos de processamento de sinal nos dados analógicos. Ao final desses procedimentos o valor da medida já está na unidade de engenharia. Os procedimentos são apresentados a seguir:

1. **Cálculo da Média.** Para cada medida, é feito o cálculo da média de vinte amostras obtidas a intervalos de 50ms. O valor de 50ms corresponde a três ciclos completos de 60Hz. Assim, o cálculo da média com este intervalo entre as amostras proporciona redução da interferência de 60Hz causada pela rede elétrica;
2. **Cancelamento de tensão de offset.** Alguns circuitos de condicionamento podem eventualmente gerar uma pequena tensão de offset que deve ser subtraída do valor medido. Isto ocorre porque não foram incluídos recursos de cancelamento de offset no hardware por se tratar de um sistema de baixo custo, assim, a calibração de precisão de cada medida deve ser feita através de um parâmetro fixo no software;
3. **Ajuste de escala.** O valor bruto lido do conversor analógico/digital do Arduino opera somente com entrada de tensão positiva e possui resolução de 10 bits. Usando a referência de 5Vcc, cada entrada analógica fornece valores brutos na faixa de 0 a 1023 para tensão na entrada na faixa de 0 a 5Vcc. Assim, esses valores são convertidos para um valor inteiro na escala x100 ou x1000. O valor de escala x100 é usado somente

para medidas de corrente de menor amplitude. Cada medida já ajustada para a escala da grandeza correspondente pode assumir valores positivos de até 16 bits ($2^{16} = 65.535$).

5.1.1 Condicionadores de Sinal

Todos os sinais de supervisão digitais e analógicos devem ser lidos pelo controlador Arduino Mega 2560. O padrão do Arduino é tensão positiva na faixa de zero a cinco volts. Assim, todos os condicionadores de sinal devem fornecer a saída em tensão na faixa de 0 a +5Vcc.

5.1.2 Considerações sobre Aterramento, Proteção e Referência

O modo mais eficiente de conseguir proteção contra surtos e descargas atmosféricas é conectar um dos terminais do barramento de corrente contínua à terra de proteção, no caso o terminal negativo. Assim, em todos os circuitos da usina solar, o terminal negativo da alimentação é conectado ao terra de proteção, o que também conecta o negativo de todos os painéis fotovoltaicos instalados no telhado. Além disso, as carcaças dos painéis fotovoltaicos são conectadas por cabo de descida exclusivo com bitola de 25mm² conectado ao terra de proteção.

Devido a condições próprias da instalação elétrica do local da usina, uma chácara com área de 30.000m², que possui vários circuitos externos com até 100 metros de extensão, o neutro da rede é aterrado em vários pontos para fornecer proteção extra contra descargas atmosféricas. Portanto, os circuitos alimentados com corrente contínua e os circuitos alimentados com corrente alternada em 220V possuem a mesma referência, o terra.

5.1.3 Considerações sobre Falha Segura

O projeto do hardware e software de transferência de cargas da usina foi feito de acordo com os procedimentos de falha segura. Assim, as falhas de alimentação em corrente contínua e nos inversores sempre revertem as cargas para a energia fornecida pela rede da concessionária. A filosofia de hardware é bem simples: todas as chaves de transferência fornecem alimentação da rede da concessionária na posição normalmente fechada dos relés de transferência. Desta forma, na falta de alimentação corrente contínua na usina, todas as chaves de transferência alimentam as cargas pela rede. O software de proteção, da mesma forma, retorna as chaves de transferência do inversor sob falha para a rede e mantém nesta posição até que seja feita a limpeza dos indicadores.

Os procedimentos de detecção de falhas graves tem por filosofia o seguinte: desativar os equipamentos que porventura, entrem em condição de falha, neste caso os inversores 1 e 2, e mantê-los desativados até que o operador apague os indicadores. Quando ocorre a detecção de falha grave, as cargas do inversor correspondente são transferidas para a rede da concessionária. Desta forma, não foi implementado mecanismo de restabelecimento automático no caso de falha grave, ficando a critério do usuário da usina analisar a situação para reativar o equipamento.

5.1.4 Entradas Analógicas do Arduíno Mega 2560

A placa do controlador Arduíno Mega 2560 possui dezesseis entradas analógicas destinadas a receber os sinais de medição de tensão na faixa de 0 a +5 Volts. Esses pontos de entrada para medição são identificados como A0 a A15 no “silkscreen” da placa. Cada ponto de entrada analógica da placa é conectado a um multiplexador, que tem a sua saída tratada por um circuito conversor analógico para digital unipolar com resolução de dez bits.

Este conversor fornece ao software valores brutos na faixa de 0 a 1023 contagens referentes aos valores 0 e +5 Volts de cada entrada respectivamente. Assim, a faixa de medição de cada grandeza deve ser ajustada para ocupar ao máximo a faixa de tensão de entrada do controlador com o objetivo de maximizar a resolução.

5.1.5 Entradas e Saídas Digitais do Arduíno Mega 2560

A placa do controlador Arduíno MEGA 2560 possui cinquenta e quatro pinos de supervisão e controle digital que podem ser configurados individualmente como entrada ou saída. A programação da função de cada pino digital é feita na área de inicialização do software da UTR e do Concentrador (Setup()).

A aplicação de uma tensão maior que quatro Volts em uma entrada digital gera certamente a indicação de nível um ou nível alto quando lida pelo software da UTR. A aplicação de uma tensão menor que meio Volt em uma entrada digital gera certamente a indicação de nível zero ou nível baixo para o software da UTR.

5.1.6 Proteção contra Surtos das Entradas e Saídas do Arduíno Mega 2560

Todas as entradas analógicas e digitais do Arduíno Mega 2560 usado na Unidade Terminal Remota possuem proteção contra surtos, a qual é composta por um resistor de 4,7 KOhms em

série e um diodo zener de 5,1 Volts e potência máxima de 400mW em paralelo. Este tipo de proteção se mostrou bastante eficiente, pois a impedância das entradas do Arduino está na faixa dos mega ohms, ou seja, estes componentes não interferem na precisão da medida. Qualquer surto de tensão tem a sua corrente limitada pelo resistor e a tensão na entrada do Arduino permanece abaixo do valor máximo (6 Volts) devido à condução do diodo zener.

5.1.7 Sensor e Condicionador de Sinal para Tensão CC

Os condicionadores de tensão CC são apenas divisores de tensão com filtro. As medidas de tensão CC e as escalas são apresentadas na tabela 5.1:

Tabela 5.1 - Especificações dos Sensores/Condicionadores das Medidas de Tensão CC

Medida de Tensão CC	Escala Entrada	Escala Saída
Barramento Principal 24Vcc	0 a 30 Vcc	0 a 5Vcc

5.1.8 Sensor e Condicionador de Sinal para Corrente CC

Os condicionadores para medida de corrente contínua foram projetados usando os módulos sensores com circuito integrado ACS713 30A. Este CI opera usando o efeito Hall e tem fundo de escala unipolar de +30A. A saída do CI tem tensão 0,5V para entrada de 0A e saída de 4,5V para corrente de entrada de +30A.

Desta forma, o circuito de condicionamento deve fazer inicialmente uma subtração para retirar o offset de 0,5V e depois efetuar o ajuste de escala para a faixa de medida desejada. As medidas de corrente CC e as escalas são apresentadas na Tabela 5.2:

Tabela 5.2 - Especificações dos Sensores/Condicionadores das Medidas de Corrente CC

Medida de Corrente CC	Escala Entrada	Escala Saída
Entrada do Inversor 1	0 a 30,00A	0 a 5Vcc
Entrada do Inversor 2	0 a 30,00A	0 a 5Vcc
Saída da Fontes CA-CC	0 a 20,00A	0 a 5Vcc
Circuitos de Alimentados com 24Vcc	0 a 6,64A	0 a 5Vcc

A Figura 5.1 mostra a função de entrada/saída do CI sensor de corrente ACS713:

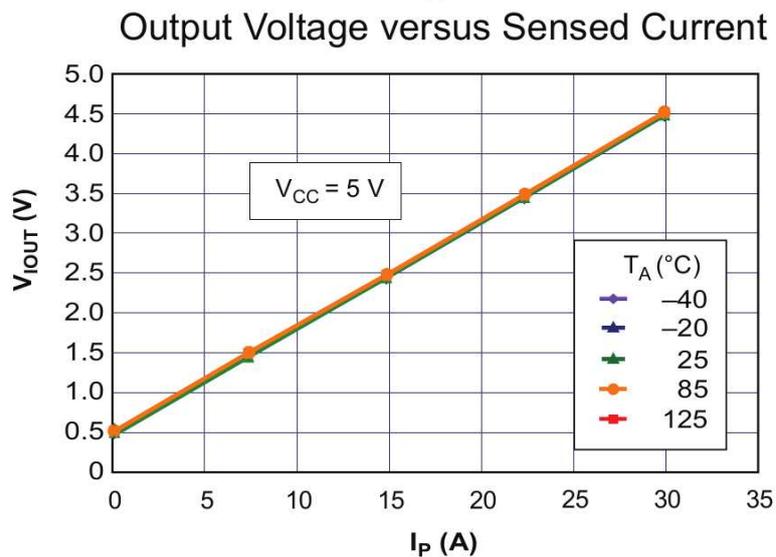


Figura 5.1 - Função Entrada/Saída do CI Sensor de Corrente ACS713

5.1.9 Sensor e Condicionador de Sinal para Tensão CA

A configuração de aterramento da instalação facilita o acoplamento dos sinais de tensão alternada, pois tanto os sinais de corrente contínua quanto os sinais de corrente alternada possuem a mesma referência, o terra de proteção. Assim, os circuitos condicionadores de corrente CA devem apenas transformar a entrada na faixa que abrange os 220VCA em saída que abrange os 5Vcc. Isto é feito com circuitos do tipo divisor de tensão, retificador de meia onda e filtro, não necessitando de isolamento com transformador. A Tabela 5.3 mostra os sinais de tensão CA e as escalas de entrada e saída:

Tabela 5.3 - Especificações dos Sensores/Condicionadores das Medidas de Tensão CA

Medida de Tensão CA	Escala Entrada	Escala Saída
Saída do Inversor 1	0 a 244 V RMS	0 a 5Vcc
Rede da Concessionária	0 a 244 V RMS	0 a 5Vcc
Saída do Inversor 2	0 a 244 V RMS	0 a 5Vcc

5.1.10 Sensor e Condicionador de Sinal para Corrente C. A.

Os condicionadores para medida de corrente alternada foram projetados usando o módulo sensor com circuito integrado ACS712. Este CI opera usando o efeito Hall e tem fundo de escala bipolar de $\pm 5A$. Alimentando o CI com +5Vcc, a saída do sensor obedece ao gráfico mostrado na Figura 5.2.

Desta forma, o circuito de condicionamento deve fazer inicialmente uma subtração para retirar o offset de 2,5V e depois efetuar o ajuste de escala e a filtragem. Ao fazer o ajuste de

offset, o semiciclo negativo é retirado do sinal, permanecendo apenas o semiciclo positivo, o que dispensa a retificação. As medidas de corrente CA e as escalas são apresentadas na Tabela 5.4:

Tabela 5.4 - Escalas de Medidas de Tensão CA para o Arduino

Medida de Corrente CA	Escala Entrada	Escala Saída
Saída do Inversor 1	0 a 1,894A RMS	0 a 5Vcc
Saída do Inversor 2	0 a 1,894A RMS	0 a 5Vcc
Alimentação da Carga 3	0 a 1,492A RMS	0 a 5Vcc

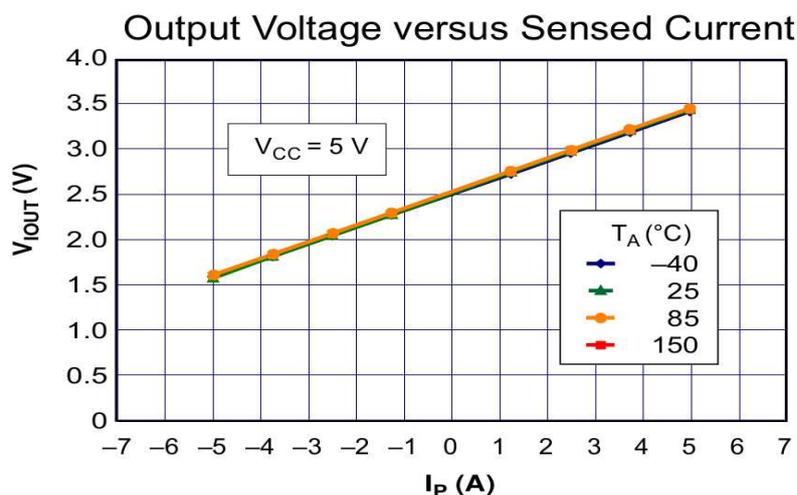


Figura 5.2 - Especificações dos Sensores/Condicionadores das Medidas de Corrente

5.1.11 Sensor e Condicionador de Sinal para Temperatura

As medidas de temperatura são feitas usando como sensor o Circuito Integrado LM35. Este CI possui encapsulamento TO92, igual a um transistor com três pinos, os quais recebem alimentação CC e fornecem a saída de medição de temperatura em tensão contínua.

Assim, para implementar um sensor de tensão, é preciso inicialmente montar o circuito integrado LM35 na extremidade de um cabo blindado com malha e duas vias. Em seguida fazer a conexão térmica do LM35 com o objeto ao qual se deseja fazer a medição e conectar os três terminais do cabo blindado à alimentação, à entrada do circuito condicionador e ao terra (negativo) da alimentação.

As especificações do circuito integrado sensor de temperatura LM35 são apresentadas na Tabela 5.5:

Tabela 5.5 - Especificações do CI Sensor de Temperatura LM35

Descrição	Valor
Tensão de Alimentação	4 a 20Vcc
Relação Temperatura / Tensão de Saída	10mV / °C
Faixa de Medição	2°C a 150°C

As medidas de temperatura e as escalas são apresentadas na Tabela 5.6:

Tabela 5.6 - Especificações dos Sensores/Condicionadores das Medidas de Temperatura

Medida de Temperatura	Escala Entrada	Escala Saída
Driver do Inversor 1	0 a 75°C	0 a 5Vcc
Transformador do Inversor 1	0 a 75°C	0 a 5Vcc
Driver do Inversor 2	0 a 75°C	0 a 5Vcc
Transformador do Inversor 2	0 a 75°C	0 a 5Vcc

5.1.12 Leitura das Medidas dos Controladores de Carga

A leitura das medidas fornecidas pelos controladores de carga é feita através de interface serial assíncrona padrão elétrico RS485 na taxa de 115200 bps. Isto é realizado usando-se a Interface Serial Assíncrona 2 do Equipamento Concentrador Arduino Mega 2560 (Sinais Tx2, Rx2) aplicados a uma placa conversora de sinal elétrico para o padrão RS485 com dois canais. As medidas obtidas de cada Controlador de Carga são mostradas na Tabela 5.7:

Tabela 5.7 - Medidas Obtidas de Cada Controlador de Carga

Medida	Descrição da Medida	Unidade
1	Tensão de Saída do Conjunto de Painéis Fotovoltaicos	Volts (V)
2	Corrente de Saída do Conjunto de Painéis Fotovoltaicos	Amperes (A)
3	Potência de Saída do Conjunto de Painéis Fotovoltaicos	Watts (W)
4	Tensão do Barramento Principal de 24Vcc (banco de baterias)	Volts (V)
5	Corrente fornecida ao Barramento Principal	Amperes (A)
6	Potência fornecida ao Barramento Principal	Watts (W)
7	Temperatura do Banco de Baterias	Graus Centígrados

Cada controlador de carga recebe em sua entrada a conexão de um conjunto de dois painéis fotovoltaicos conectados em série. A saída de cada controlador de carga é ligada no barramento principal de 24Vcc através de disjuntores termomagnéticos do tipo Quick-Lag de corrente nominal igual a 32 Amperes curva B.

5.2 Montagem das Mensagens Padrão CoAP-OSA-CBM

Neste trabalho, foi implementado um subconjunto das informações definido para as mensagens padrão OSA-CBM binárias, constituído basicamente de:

1. Mensagens de estados do equipamento
2. Mensagens de estados digitais,
3. Mensagens de medições analógicas,
4. Mensagens de ajuste de parâmetros;
5. Mensagens de comando digital.

Não foram implementadas mensagens de eventos, porque neste sistema, as informações de estado digital trafegam em protocolo CoAP sobre datagramas UDP em rede Ethernet, o que permite a rápida atualização dos estados.

5.2.1 Definições sobre o Formato da Mensagem de Requisição do Cliente

A montagem do primeiro byte da mensagem de requisição CoAP do cliente para o servidor usa os seguintes parâmetros:

1. O número da versão é sempre 1 (bits 7 e 6 = 01);
2. O tipo da mensagem é Confirmável (bits 5 e 4 = 00);
3. Não é usado o Token (bits 3, 2, 1 e 0 = 0000).

Portanto, o primeiro byte da mensagem de requisição do cliente para o servidor em protocolo CoAP é: 0x40 (hexa) = 0100 0000 (binário). A montagem do segundo byte da mensagem de requisição do cliente para o servidor usa os seguintes parâmetros:

1. A função é Requisição (bits 7, 6 e 5 = 000);
2. O método é GET (bits 4, 3, 2, 1 e 0 = 00001).

O segundo byte da mensagem de requisição do cliente para o servidor em protocolo CoAP é: 0x01 (hexa) = 0000 0001 (binário). Os dois bytes seguintes compõem o identificador de 16 bits da mensagem, na ordem: byte mais significativo (MSB), byte menos significativo (LSB). O cliente gera este valor incrementado-o a cada mensagem enviada e recebida do servidor. O valor do identificador inicia em 1 após o reset.

O campo Opção da mensagem de requisição possui duas Instâncias que definem uma URI e um byte de comando. A montagem dos bytes das duas instâncias do campo Opção é mostrada a seguir:

1. Primeira Instância número 11 (0x0B hexa) - URI-Path seguida dos caracteres ASCII correspondentes à URI (“estados”). O valor do campo Delta da Opção = 11 (0x0B hexa), que é igual ao da primeira instância. O número de bytes da opção é o mesmo número de caracteres da URI = “estados” = 7. Portanto, a sequência de bytes da primeira instância do campo Opção se inicia com o byte 0xB7 seguido pelos 7 códigos ASCII correspondentes aos caracteres da URI;
2. Segunda Instância número 12 (0x0C hexa) – Content-Format com o tipo de dados application/octet-stream número 42 (0x2A hexa). O valor do campo Delta da Opção = 1, que é igual ao da primeira instância mais 1 para gerar o valor da segunda instância. Esta instância de opção define o tipo de dados do Payload levado pela mensagem de requisição, o qual é de um byte correspondente ao comando. Portanto, a sequência de bytes da segunda instância do campo Opção se inicia com o byte 0x11 seguido por um byte com código correspondente ao tipo de dado application/octet-stream = 0x2A.

Após o campo Opção, é inserido o byte indicador de final deste campo e de início do campo Payload. Este byte sempre tem o valor 0xFF.

Após o byte 0xFF indicador de final do campo Opção, estão os bytes do campo Payload. O primeiro byte do campo Payload contém o código de comando, sendo seguido pelos seis bytes da hora e data do computador Raspberry PI 3. Quando não há comando solicitado, o valor deste byte é igual a um.

A Tabela 5.8 mostra os bytes da mensagem CoAP de requisição do cliente para o servidor.

Tabela 5.8 - Bytes do Início da Mensagem CoAP de Requisição do Cliente para o Servidor

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	GET	Identificador		Op1	e	s	t	a	d	o	s	Op2	OS	
40	01	MSB	LSB	B7	65	73	74	61	64	6F	73	11	2A	FF

A Tabela 5.9 mostra a posição dos bytes do relógio no Payload CoAP:

Tabela 5.9 - Bytes do Payload da Mensagem CoAP de Requisição do Cliente para o Servidor

15	16	17	18	19	20	21
Comando	Hora	Minuto	Segundo	Dia	Mês	Ano
00 - 17	00 - 23	00 - 59	00 - 59	01 - 31	01 - 12	00 - 99

5.2.2 Definições sobre o Formato da Mensagem de Resposta do Servidor

A montagem do primeiro byte da mensagem de resposta do servidor CoAP para o cliente usa os seguintes parâmetros:

1. O número da versão é sempre 1 (byte 0: bits 7 e 6 = 01);
2. Caso a mensagem de requisição enviada pelo cliente tenha sido recebida corretamente, o Tipo da Mensagem de Resposta = 2, que indica Reconhecimento - ACK => bits 5 e 4 = 10. Caso a mensagem de requisição enviada pelo cliente tenha sido recebida com erro de verificação de integridade (CRC), o Tipo da Mensagem de Resposta = 3, que indica Reset - RST => bits 5 e 4 = 11;
3. Não é usado o Token (byte 0: bits 3, 2, 1 e 0 = 0000);

Portanto, o primeiro byte da mensagem de resposta do servidor para o cliente em protocolo CoAP é: 0x60 (hexa) = 0110 0000 no caso de requisição correta, e 0x70 (hexa) = 0111 0000 no caso de requisição recebida com erro. Caso a mensagem de requisição tenha sido recebida corretamente, a montagem do segundo byte da mensagem de resposta usa os seguintes parâmetros:

1. A Classe = 2 indica Resposta com Sucesso. Byte 1: bits 7, 6 e 5 = 010;
2. O Detalhe = 5 indica Conteúdo (Content). Byte 1: bits 4, 3, 2, 1 e 0 = 00101.

O segundo byte da mensagem de resposta do servidor para o cliente em protocolo CoAP é: 0x45 (hexa) = 0100 0101 (binário). Os dois bytes seguintes são o identificador da mensagem, que é carregado com o mesmo valor lido da mensagem de requisição do cliente.

O campo Opção da mensagem de resposta possui somente uma Instância que define o tipo de informação do campo Payload que é application/octet-stream (binário). A montagem dos bytes da Instância é mostrada a seguir:

Primeira e única instância número 12 (0x0C hexa) – Content-Format com o tipo application/octet-stream número 42 (0x2A hexa). O valor do campo Delta da Opção = 12 (0x0X hexa), que é igual ao número da opção, por ser a primeira instância. Esta instância de opção define o tipo de dados do payload levado pela mensagem de resposta, o qual carrega as informações sobre todos os estados e medidas da usina. Portanto, a sequência de bytes da instância do campo Opção se inicia com o byte 0x0C1 seguido por um byte com código correspondente ao tipo de dado application/octet-stream = 0x2A.

Após o campo Opção, é inserido o byte indicador de final deste campo e de início do campo Payload. Este byte sempre tem o valor 0xFF. Após o byte 0xFF indicador de final do campo Opção, está o campo Payload. O campo Payload é de uso geral, e pode carregar informações em formato texto ou binário. Visando a compactação das mensagens, foi usado o formato binário, pois, o uso de uma mensagem menor diminui os tempos de tráfego e, conseqüentemente, aumenta a confiabilidade da comunicação.

Desta forma, para compor o protocolo CoAP-OSA-CBM, o payload da mensagem de resposta do servidor em protocolo CoAP deve carregar uma mensagem OSA-CBM em formato binário. Esta mensagem deve ter um cabeçalho padrão e uma área de dados com todas as informações de estados e medidas da usina. Esta última característica também contribui para a agilidade e confiabilidade da comunicação, pois, o envio de uma mensagem de atualização com todas as informações de supervisão, reduz o tráfego e os procedimentos de enlace, aumentando a confiabilidade dos procedimentos de comunicação.

5.2.3 Ordem dos Bytes para os Valores Numéricos no Campo de Dados

O padrão OSA-CBM, assim como os protocolos IP, definem a ordem crescente dos bytes no que se refere à inserção dos valores numéricos nas mensagens (“big-endian”). Isto significa, que o byte mais significativo do valor numérico entra primeiro, sendo seguido pelos outros bytes até a última posição ser ocupada pelo byte menos significativo.

5.2.4 Estrutura Geral da Mensagem Padrão OSA-CBM Binária

Todas as mensagens OSA-CBM binárias para troca de informações referentes ao monitoramento de condição obedecem a uma estrutura geral. Visando facilitar os procedimentos de montagem e decodificação, foi implementado somente um tipo de mensagem, que possui um bloco de dados e tem comprimento fixo. Assim, não foram usados os campos opcionais: Comprimento da Mensagem e Número de Blocos. Após o cabeçalho OSA-CBM foram usados somente dois campos: o campo de Data / Hora e o campo de Bloco de Conteúdo da Informação, onde é identificado o tipo de informação presente nos campos de dados. O campo Data/Hora usado é no formato compacto com oito bytes e o campo com o Conteúdo da Informação possui quatro bytes.

A Tabela 5.10 mostra os campos da mensagem e a sua descrição (MIMOSA, 2010):

Tabela 5.10 - Estrutura Geral da Mensagem Padrão OSA-CBM Binária

Campo da Mensagem	Bytes Usados	Descrição
Cabeçalho (Header)	1-N	Os bytes de cabeçalho são usados para indicar o tipo de mensagem que está sendo transmitida e vários usos de componentes de informação opcionais. O cabeçalho apresenta as bases para interpretar os bytes
Comprimento da Mensagem (Message Length)	0,4	(Opcional) Contém o número de bytes total da mensagem. Existem métodos alternativos para conhecer o comprimento. Portanto, este é um argumento opcional.
Número de Blocos (Block Count)	0-2	(Opcional) Contém a quantidade de blocos de dados enviados. Normalmente, apenas um bloco é necessário. O bit indicador “Último Bloco” no conteúdo do bloco de dados é um método alternativo para verificação do final esperado dos blocos.
Identificador da mensagem (Message Identifier)	1,2,4,8	Identifica um tipo de mensagem específica de uma plataforma. Não é usado em todos os tipos de mensagens.
Identificador da plataforma (Platform Identifier)	0,3+	(Opcional) Identifica a plataforma a partir da qual uma mensagem é gerada. Mensagens transmitidas internamente em uma plataforma podem excluir esse identificador para reduzir o tamanho da mensagem.
Data / Hora (Time)	2+	Este campo aceita vários formatos de data/hora, incluindo microssegundo POSIX e nanossegundo ISO 8601. Outros formatos são possíveis.
Bloco de Conteúdo da Informação (Information Content Block 1)	3+	Este bloco contém dados referentes ao equipamento e ao sistema, que inclui informações sobre verificação de erros ou sincronização. O bloco tem tamanho variável e tipo de conteúdo variável. O byte do cabeçalho possui o código que identifica o tipo de informação presente no bloco.
...Blocos de dados... (...Block N...)	...	Pode haver até 65.535 blocos de dados

5.2.5 Especificação do Campo Cabeçalho da Mensagem

O cabeçalho apresenta o uso dos parâmetros opcionais e estabelece as regras necessárias para decodificar o fluxo de dados da mensagem. O primeiro byte do cabeçalho identifica o tipo geral da mensagem. A norma define que o cabeçalho pode ser composto por somente um byte ou por mais bytes. Neste trabalho o cabeçalho tem somente um byte. Existem quatro tipos principais de mensagens. Os dois primeiros bits do primeiro byte do cabeçalho identificam o tipo de mensagem principal. O primeiro procedimento ao decodificar a mensagem é verificar esses dois bits.

A Tabela 5.11 mostra a função dos bits do byte do cabeçalho da Mensagem Padrão OSA-CBM Binária:

Tabela 5.11 - Especificação do Cabeçalho da Mensagem Padrão OSA-CBM Binária

Bit	Descrição
7,6	(00) Cabeçalho tipo 0. É usado para tráfego de blocos de informação de uso geral. (01) Cabeçalho tipo 1. É usado para chamadas de interface e outras funções de alto nível. Possui 2 bytes. (10) Cabeçalho tipo 2. É usado para outras funções. Possui 5 bytes. (11) Cabeçalho tipo 3. É usado para outras funções. Possui comprimento variável.
5	Indicador do uso do campo Comprimento da Mensagem 0 = O campo Comprimento da Mensagem não é usado 1 = O número total de bytes da mensagem é carregado no campo Comprimento da Mensagem com 4 bytes
4	Indicador do uso do campo Numero de Blocos 0 = O campo Numero de Blocos não é usado 1 = O número total de blocos da mensagem é carregado no campo Numero de Blocos com 2 bytes
3	Indicador do uso do campo Identificador da Plataforma 0 = O campo Identificador da Plataforma não é usado 1 = O campo Identificador da Plataforma é usado
2,1	Tamanho do Indicador Geral do Conteúdo da Mensagem (00) = 1 byte / (01) = 2 bytes / (10) = 4 bytes / (11) = 8 bytes
0	É determinado pelo tipo da mensagem

Neste trabalho, o cabeçalho é do tipo 0, composto por somente um byte (bit 7 = 0 e bit 6 = 0). O campo comprimento da mensagem é usado, portanto, o bit 5 = 1. O campo número de blocos da mensagem não é usado porque é sempre igual a um, logo o bit 4 = 0. O campo identificador da plataforma é usado, portanto o bit 3 = 1. O indicador geral do conteúdo da mensagem possui um byte, logo o bit 2 = 0 e o bit 1 = 0. O bit 0 = 0 é definido aqui. Assim, o byte do cabeçalho possui o formato binário mostrado na Tabela 5.12:

Tabela 5.12 - Byte do Cabeçalho da Mensagem Padrão OSA-CBM Binária já Configurado

Byte do Cabeçalho da Mensagem Padrão OSA-CBM Binária								
Bit →	7	6	5	4	3	2	1	0
Valor →	0	0	1	0	0	0	0	0

5.2.6 Montagem do Campo de Dados da Mensagem OSA-CBM Binária

Após o cabeçalho OSA-CBM está o campo com a informação de data, hora e fuso horário em um formato compacto de oito bytes. Assim, considerando que a norma permite o uso de padrões alternativos de data e hora, e visando a máxima compactação da mensagem binária, foi usado este formato compacto. A Tabela 5.13 mostra o formato do campo de data, hora e fuso horário:

Tabela 5.13 - Campo de Data, Hora e Fuso Horário

Byte	Função	Faixa
8	Hora	0 a 23
9	Minuto	0 a 59
10	Segundo	0 a 59
11	Dia	1 a 31
12	Mês	1 a 12
13	Ano (MSB)	0 a 65535
14	Ano (LSB)	
15	Fuso Horário	-24 a +24

Após o campo de data, hora e fuso horário está o campo de Conteúdo da Informação de quatro bytes, o qual carrega um código de identificação do tipo de dados, neste caso, definido com o valor 1. Após o campo de Conteúdo da Informação, está o campo de Estados do Equipamento com doze bytes. Na Tabela 5.14 estão os pontos de estado do equipamento:

Tabela 5.14 - Campo de Estados do Equipamento

Byte	Função
20	Estado de Comunicação da UTR (Normal / Falha)
21	Estado de Comunicação do Controlador de Carga 1 (Normal / Falha)
22	Estado de Comunicação do Controlador de Carga 2 (Normal / Falha)
23	Reserva
24	Reserva
25	Reserva
26	Reserva
27	Reserva
28	Reserva
29	Reserva
30	Reserva
31	Reserva

Após o Campo de Estados do Equipamento está o Campo de Estados Digitais. Um estado digital pode ser obtido diretamente de uma entrada digital física ou pode ser um valor resultante de uma operação lógica realizada pelo software da UTR. Foi definido que cada estado digital é carregado em um byte, permitindo, assim, valores de estado na faixa de 0 a 255. Na implementação do software do controlador, alguns estados, tais como o nível da caixa d'água pode ter 5 estados. O estado de saúde dos equipamentos também é considerado estado digital, e pode ter valores de 0 a 100%. A Tabela 5.15 mostra o Campo de Estados Digitais:

Tabela 5.15 - Campo de 80 Estados Digitais

Byte	Descrição do Estado Digital	Byte	Descrição do Estado Digital
32	Entrada Digital Chave Local Remoto	72	Estado da Caixa Azul
33	Entrada Digital Chave Economia/Normal	73	Fonte 1 24Vcc Ligada
34	Entrada Digital Chave Manual/Automático	74	Alarme Sobre Corrente Inversor 1
35	Entrada Digital Chave Habilita Carga 1	75	Alarme Sobre Corrente Inversor 2
36	Entrada Digital Chave Habilita Carga 2	76	Reserva
37	Entrada Digital Disjuntor de Entrada Inversor 1	77	Reserva
38	Entrada Digital Circuito Boia Nível Baixo	78	Reserva
39	Entrada Digital Boia Nível Baixo	79	Reserva
40	Entrada Digital Alimentação Bomba	80	Reserva
41	Entrada Digital Circuito Rede Bomba	81	Reserva
42	Estado da Rede	82	Reserva
43	Modo de Operação	83	Reserva
44	Modo de Comando	84	Reserva
45	Modo de Controle	85	Reserva
46	Carga 1 Habilitada	86	Reserva
47	Carga 2 Habilitada	87	Reserva
48	Carga 3 Habilitada	88	Reserva
49	Carga 4 Habilitada	89	Reserva
50	HabCom – Habilitação de Comando	90	Reserva
51	Estado do Inversor 1	91	Reserva
52	Estado do Inversor 2	92	Reserva
53	Estado da Carga 3	93	Reserva
54	Bomba Ligada	94	Reserva
55	Modo de Controle da Carga 1	95	Reserva
56	Falha do Inversor 1	96	Reserva
57	Alarme Sub Tensão do Inversor 1	97	Reserva
58	Alarme Sobre Tensão do Inversor 1	98	Reserva
59	Alarme Sobre Temp. do Driver do Inversor 1	99	Reserva
60	Alarme Sobre Temp. do Trafo do Inversor 1	100	Reserva
61	Alarme Disjuntor Aberto Inversor 1	101	Reserva
62	Falha do Inversor 2	102	Reserva
63	Alarme Sub Tensão do Inversor 2	103	Reserva
64	Alarme Sobre Tensão do Inversor 2	104	Saúde Inversor 1
65	Alarme Sobre Temp. do Driver do Inversor 2	105	Saúde Inversor 2
66	Alarme Sobre Temp. do Trafo do Inversor 2	106	Saúde Baterias
67	Alarme Disjuntor Aberto Inversor 2	107	Saúde Controlador de Carga 1
68	Reserva	108	Saúde Controlador de Carga 2
69	Entrada Digital Caixa Azul Precisa Encher	109	Reserva
70	Entrada Digital Caixa Azul Cheia	110	Reserva
71	Fonte 2 24Vcc Ligada	111	Reserva

A Tabela 5.16 mostra o estado de cada saída digital de comando e sua descrição. O estado de cada saída somente pode ter os valores zero e um.

Tabela 5.16 - Campo de 48 Estados das Saídas Digitais

Byte	Descrição do Estado da Saída Digital	Byte	Descrição do Estado da Saída Digital
112	SD00: Controle Chave de Transferência 1	136	Reserva
113	SD01: Controle Inv. 1 e Chave de Transferência 4	137	Reserva
114	SD02: Controle Chave de Transferência 3	138	Reserva
115	SD03: Controle Conexão do Inversor 2 na CT2	139	Reserva
116	SD04: Controle do LED 2 (Amarelo)	140	Reserva
117	SD05: Controle do LED 3 (Verde)	141	Reserva
118	SD06: Controle Conexão do Inversor 2 na CT1	142	Reserva
119	SD07: Controle Conexão do Inversor 2 na CT3	143	Reserva
120	SD08: Controle do LED 1 (Verde)	144	Reserva
121	SD09: Controle do LED 4 (Verde)	145	Reserva
122	SD10: Liga/Desliga Inversor 2	146	Reserva
123	SD11: Controle Conexão do Inversor 1 na CT4	147	Reserva
124	SD12: Controle do LED Azul 1	148	Reserva
125	SD13: Controle do LED Azul 2	149	Reserva
126	SD14: Controle do LED Azul 3	150	Reserva
127	SD15: Controle do LED Azul 4	151	Reserva
128	SD16: Habilitação da Bomba	152	Reserva
129	SD17:	153	Reserva
130	SD18:	154	Reserva
131	SD19:	155	Reserva
132	Reserva	156	Reserva
133	Reserva	157	Reserva
134	Reserva	158	Reserva
135	Reserva	159	Reserva

O controlador Arduino Mega 2560 que tem a função de UTR, recebe até dezesseis sinais analógicos, que são selecionados através de um multiplexador, e enviados a um conversor analógico digital unipolar de dez bits. Além das medidas geradas pelos sinais analógicos recebidos nas entradas, existem também as medidas geradas por cálculo do software da UTR e as medidas recebidas por interface de comunicação.

Os Controladores de Carga fornecem as medidas através de interface de comunicação serial assíncrona RS485 e protocolo MODBUS RTU. Essas medidas são fornecidas em registros de dezesseis bits. Portanto, foram definidos na mensagem, campos de dezesseis bits para carregar todas as medidas.

A Tabela 5.17 apresenta cada medida e sua descrição.

Tabela 5.17: Campo de 60 Medidas

Byte	Descrição da Medida	Byte	Descrição da Medida
160	Tensão do Barramento Principal 24Vcc	220	Cont. Carga 2 - Battery charging current 2
162	Reserva	222	Cont. Carga 2 - Battery charging power 2
164	Temperatura do Driver do Inversor 1	224	Cont. Carga 2 - Battery Temperature 2
166	Corrente CC: Circuitos de Corrente Continua	226	Corrente Total Gerada
168	Tensão CA: Saída do Inversor 1	228	Potencia Total Gerada
170	Tensão CA: Rede	230	Potencia Consumida pelos Circuitos de 24Vcc
172	Tensão CA: Saída do Inversor 1	232	Potencia Fornecida pela Fonte 24Vcc
174	Temperatura do Transformador do Inversor 1	234	Corrente de Carga ou Descarga do Banco de Baterias
176	Temperatura do Driver do Inversor 2	236	Potencia de Entrada do Inversor 2
178	Temperatura do Transformador do Inversor 2	238	Potencia de Saida do Inversor 2
180	Corrente CA: Saida do Inversor 2	240	Potencia de Entrada do Inversor 1
182	Corrente CC: Saida da Fonte 24Vcc	242	Potencia de Saida do Inversor 1
184	Corrente CC: Entrada do Inversor 2	244	Corrente Total Consumida pelas Cargas
186	Corrente CA: Saida do Inversor 1	246	Potencia Total Consumida pelas Cargas
188	Corrente CA:Carga 3	248	Tempo da Caixa Azul em Nível Baixo
190	Corrente CC: Entrada do Inversor 1	250	IFontesCC12
192	Media Estendida da Tensao 24Vcc Geral	252	Reserva
194	Temporizador de Bomba Ligada em segundos	254	Reserva
196	Cont. Carga 1 - PV array voltage 1	256	Reserva
198	Cont. Carga 1 - PV array current 1	258	Reserva
200	Cont. Carga 1 - PV array power 1	260	Reserva
202	Cont. Carga 1 - Battery voltage 1	262	Reserva
204	Cont. Carga 1 - Battery charging current 1	264	Reserva
206	Cont. Carga 1 - Battery charging power 1	266	Reserva
208	Cont. Carga 1 - Battery Temperature 1	268	Reserva
210	Reserva	270	Reserva
212	Cont. Carga 2 - PV array voltage 2	272	Reserva
214	Cont. Carga 2 - PV array current 2	274	Reserva
216	Cont. Carga 2 - PV array power 2	276	Reserva
218	Cont. Carga 2 - Battery voltage 2	278	Reserva

5.2.7 Estrutura Completa da Mensagem CoAP-OSA-CBM Binária

A Tabela 5.18 mostra a estrutura completa da mensagem CoAP-OSA-CBM de resposta do Concentrador na função de Servidor em Protocolo CoAP:

Tabela 5.18 - Estrutura Geral da Mensagem CoAP-OSA-CBM

Posição no Buffer		Número de Bytes	Descrição	
Inicial	Final			
0	6	7	Cabeçalho CoAP	
7	7	1	Cabeçalho OSA-CBM	Payload CoAP (Posições 7 a 319) Inclui a Mensagem OSA-CBM e os 2 Bytes do CRC16
8	15	8	Data/Hora em formato compacto (8 bytes)	
16	19	4	Conteúdo da Informação (4 bytes)	
20	31	12	12 Estados do Equipamento	
32	111	80	80 Estados Digitais	
112	159	48	48 Estados de Saídas Digitais	
160	279	120	60 Medidas (2 bytes por medida)	
280	317	38	Reserva	
318	319	2	Verificação de Integridade (CRC16)	

5.3 Montagem das Mensagens em Protocolo HTTP

5.3.1 Conversão das Informações e Montagem da Mensagem XML

O Software Servidor HTTP e Conversor Local recebe a mensagem binária em protocolo CoAP-OSA-CBM, e converte as informações de supervisão recebidas para uma mensagem do tipo texto em formato XML. A mensagem XML é uma sequência de caracteres com as etiquetas de identificação e os valores atuais das variáveis.

A mensagem XML possui no início uma etiqueta principal <LOCAL001> que identifica o local da usina. Ao final da mensagem, a etiqueta </LOCAL001> indica o fim do campo. Entre as etiquetas de início e fim referentes ao local, existem quatro campos delimitados por quatro etiquetas referentes aos conjuntos de sinais da usina. Dentro dos campos referentes aos conjuntos de sinais da usina, estão as etiquetas de identificação de cada variável de supervisão.

A seguir são apresentados os campos de conjuntos de sinais e os sinais correspondentes de cada campo:

GERAL: contém as informações de estados digitais e medidas referentes aos estados de comunicação, medidas de tensão do barramento e da rede, modo de operação, modo de comando, modo de controle, e posição das chaves de transferência (alimentação pela rede ou pelo inversor) para as três cargas. A Tabela 5.19 apresenta as variáveis contidas no campo GERAL da mensagem XML:

Tabela 5.19 - Variáveis do Campo Geral da Mensagem XML

Etiqueta (Tag)	Descrição da Variável	Estados ou Medida
COMCNV	Estado de Comunicação do Conversor Rasp. PI 3 com o Servidor em Nuvem	Normal/Falha
COMCNC	Estado de Comunicação do Servidor HTTP Raspberry PI 3 com o Concentrador	Normal/Falha
COMUTR	Estado de Comunicação do Concentrador com a UTR	Normal/Falha
COMCC1	Estado de Comunicação do Controlador de Carga 1 com o Concentrador	Normal/Falha
COMCC2	Estado de Comunicação do Controlador de Carga 2 com o Concentrador	Normal/Falha
CLK	Relógio da UTR	HH:MM:SS
DATA	Data da UTR	DD:MM:AA
CMDEX	Comando do browser enviado para a UTR	Comando (Texto)
MDOP	Modo de Operação	Economia/Normal
MDCOM	Modo de Comando	Local/Remoto
MDCT1	Modo de Controle da Carga 1	Manual/Automático
MDCT234	Modo de Controle das Cargas 2, 3 e 4	Manual/Automático
ENCG1	Energia da Carga 1	Rede/Inversor 2
ENCG2	Energia da Carga 2	Rede/Inversor 2
ENCG3	Energia da Carga 3	Rede/Inversor 2
ICG3	Corrente da Carga 3	Amperes
VBAT	Tensão do Barramento Principal de 24Vcc	Volts
VREDE	Tensão da Rede	Volts
ESTVRD	Estado da Tensão da Rede (Falha, Baixo, Normal, Alto)	Estado
TBAT	Temperatura do Banco de Baterias	°C
SDBAT	Saúde do Banco de Baterias	0 a 100%

AGUA: contém as informações de estados e medidas referentes à caixa d' água principal (Caixa Azul) e ao funcionamento da bomba d' água do poço tubular profundo. A tabela 5.20 apresenta uma amostra do campo AGUA da mensagem XML e as suas variáveis, onde os valores amostrados estão em negrito:

Tabela 5.20 - Variáveis do Campo AGUA da Mensagem XML

Etiqueta (Tag)	Descrição da Variável	Estados ou Medida
ESTCXAZ	Estado da Caixa d' Água (Caixa Azul)	Precisa Encher / Cheia
NIVCXAZ	Nível da Caixa Azul	Normal/Baixo
ESTBMB	Estado da Bomba do Poço	Ligada/Desligada
ESTDJB	Estado do Disjuntor da Boia da Caixa Azul	Ligada/Desligada
ESTDJRB	Estado do Disjuntor de Rede da Alimentação da Bomba do Poço	Ligada/Desligada
ENBMB	Energia da Bomba do Poço	Rede/Inversor 1
TMPBL	Tempo da Bomba do poço Ligada	HH:MM:SS

GERCONS : contém as informações de estados e medidas referentes à geração solar e ao consumo das cargas. A Tabela 5.21 apresenta uma amostra do campo GERCONS da mensagem XML e as suas variáveis, onde os valores amostrados estão em negrito:

Tabela 5.21 - Variáveis do Campo GERCONS da Mensagem XML

Etiqueta (Tag)	Descrição da Variável	Estados ou Medida
VP12	Tensão na Saída do Conjunto de Painéis 1 e 2 (ligados em série)	Volts
IS12	Corrente na Saída do Conjunto de Painéis 1 e 2 (ligados em série)	Amperes
ISCC1	Corrente na Saída do Controlador de Carga 1	Amperes
WSCC1	Potência na Saída do Controlador de Carga 1	Watts
SDCC1	Saúde do Controlador de Carga 1	0 a 100%
VP34	Tensão na Saída do Conjunto de Painéis 3 e 4 (ligados em série)	Volts
IS34	Corrente na Saída do Conjunto de Painéis 3 e 4 (ligados em série)	Amperes
ISCC2	Corrente na Saída do Controlador de Carga 2	Amperes
WSCC2	Potência na Saída do Controlador de Carga 1	Potência em Watts
SDCC2	Saúde do Controlador de Carga 2	0 a 100%
ITOTGER	Corrente Total Gerada	Amperes
WTOTGER	Potência Total Gerada	Watts
ITOTCG	Corrente Total Consumida pelas Cargas	Amperes
WTOTCG	Potência Total Consumida pelas Cargas	Watts
ESTFT1	Estado da Fonte CA/CC 1	Ligada/Desligada
ESTFT2	Estado da Fonte CA/CC 2	Ligada/Desligada
ICIRCC	Corrente Consumida pelos Circuitos Alimentados com 24Vcc	Amperes
WCIRCC	Potência Consumida pelos Circuitos Alimentados com 24Vcc	Watts

INV: contém as informações de estados e medidas referentes aos dois inversores SPWM, que convertem a tensão contínua do barramento de 24Vcc para 220Vca. A Tabela 5.22 apresenta uma amostra do campo INV da mensagem XML e as suas variáveis, onde os valores amostrados estão em negrito:

Tabela 5.22 - Variáveis do Campo INV da Mensagem XML

Etiqueta (Tag)	Descrição da Variável	Estados ou Medida
ESTIV2	Estado do Inversor 2	Ligado/Desligado
IEIV2	Corrente de Entrada do Inversor 2	Amperes
WEIV2	Potência de Entrada do Inversor 2	Watts
VSIV2	Tensão de Saída do Inversor 2	Volts
ISIV2	Corrente de Saída do Inversor 2	Amperes
WSIV2	Potência de Saída do Inversor 2	Watts
TDIV2	Temperatura dos Drivers do Inversor 2	°C
TTIV2	Temperatura do Transformador do Inversor 2	°C
EFIV2	Eficiência do Inversor 2	0 a 100%
SDIV2	Saúde do Inversor 2	0 a 100%
ESTIV1	Estado do Inversor 1	Ligado/Desligado
IEIV1	Corrente de Entrada do Inversor 1	Amperes
WEIV1	Potência de Entrada do Inversor 1	Watts
VSIV1	Tensão de Saída do Inversor 1	Volts
ISIV1	Corrente de Saída do Inversor 1	Amperes
WSIV1	Potência de Saída do Inversor 1	Watts
TDIV1	Temperatura dos Drivers do Inversor 1	°C
TTIV1	Temperatura do Transformador do Inversor 1	°C
EFIV1	Eficiência do Inversor 1	0 a 100%
SDIV1	Saúde do Inversor 1	0 a 100%

5.4 Montagem do Arquivo FCL (Fuzzy Control Logic)

O arquivo FCL (Fuzzy Control Logic) deve receber as instruções para implementar os mecanismos de monitoramento de condição dos equipamentos. Estas instruções referem-se a um conjunto de sistemas de controle fuzzy baseado em regras, onde cada um fornece em sua saída a variável que corresponde à saúde de um equipamento. Neste arquivo, devem ser definidas as variáveis de entrada com as suas funções de pertinência. Em seguida devem ser definidas as variáveis de saída e suas funções de pertinência. Por fim, são definidas as regras do tipo se então.

5.4.1 Definição das Variáveis de Entrada

As variáveis de entrada devem fornecer as informações necessárias para possibilitar a implementação dos procedimentos de monitoramento de condição do inversor. As informações de supervisão básicas são: tensão, corrente e temperatura. As medidas de potência de entrada e potência de saída são mais úteis nesta situação de monitoramento de condição, porque indicam a energia absorvida e fornecida pelo equipamento. Assim, as medidas de corrente de entrada e corrente de saída não foram usadas, sendo substituídas pelas medidas de potência. Portanto, a seção VAR_INPUT do arquivo FCL deve receber as seguintes variáveis referentes à supervisão do inversor que são mostradas no Algoritmo 5.1:

Algoritmo 5.1 declaração das variáveis de entrada do inversor em FCL

```
Vs : REAL;      // Vs = Tensão de Saída  
We : REAL;      // We = Potência de Entrada  
Ws : REAL;      // Ws = Potência de Saída  
Ei : REAL;      // Ei = Eficiência do Inversor  
Td : REAL;      // Td = Temperatura dos Dissipadores  
Tt : REAL;      // Tt = Temperatura do Transformador
```

5.4.2 Definição das Funções de Pertinência das Funções de Entrada

Os critérios para definição dos parâmetros das funções de pertinência são baseados em conhecimento especialista sobre as especificações técnicas e sobre as condições de operação do equipamento. Os conjuntos fuzzy das variáveis de entrada foram escolhidos de forma padronizada, ou seja, refletindo a intensidade ou a condição de normalidade / anormalidade. O tipo de função escolhido foi triangular ou trapezoidal, onde o extremo inferior é sempre zero e o superior é o valor máximo que a variável pode assumir.

Por exemplo, a potência fornecida por este inversor pode variar de zero a 500 Watts, portanto, os conjuntos fuzzy foram definidos somente em função da intensidade: Baixa, Media, Alta e MuitoAlta. Entretanto, em relação à tensão de saída, o valor nominal é 220 Volts. Assim, faz sentido definir a região em torno de 220 Volts como Normal e criar os outros conjuntos fuzzy para os outros valores de tensão: MuitoBaixa, Baixa e Alta indicando condições de anomalia.

5.4.3 Fuzzificação das Variáveis – Tensão de Saída

A variável Vs = Tensão de Saída recebeu quatro conjuntos fuzzy que indicam a condição da tensão de alimentação das cargas. A tensão deve normalmente estar próxima de 220 Volts. A

escala horizontal vai de zero a 242 Volts RMS. Os conjuntos fuzzy e a definição dos pontos das suas funções de pertinência são mostrados no Algoritmo 5.2:

Algoritmo 5.2 fuzzificação da variável tensão de saída do inversor em FCL

FUZZIFY Vs

TERM **MuitoBaixa** := (0, 1) (176, 1) (198, 0) ;

TERM **Baixa** := (176, 0) (198, 1) (220, 0) ;

TERM **Normal** := (198, 0) (220, 1) (242, 0) ;

TERM **Alta** := (220, 0) (242, 1);

END_FUZZIFY

A Figura 5.3 mostra graficamente as funções de pertinência associadas a cada conjunto fuzzy. A área mais à esquerda corresponde aos valores de tensão MuitoBaixa, fora da tolerância máxima permitida nas especificações das concessionárias de energia. A próxima área à direita corresponde à tensão Baixa. A próxima área à direita, centrada em 220V, corresponde à tensão normal. A última área à direita corresponde à tensão alta, a qual ainda está dentro da tolerância de 10%.

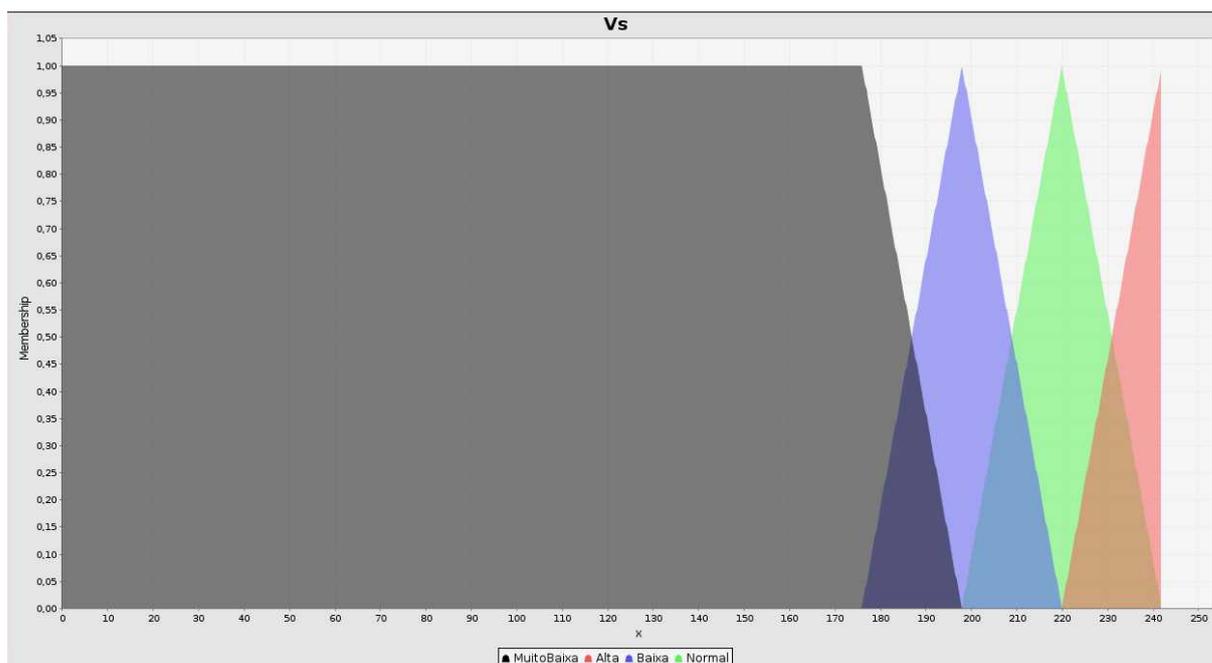


Figura 5.3 - Funções de Pertinência Associadas à Variável V_e = Tensão de Saída

5.4.4 Fuzzificação das Variáveis – Potência de Entrada

A variável W_e = Potência de Entrada recebeu quatro conjuntos fuzzy que indicam a intensidade da potência consumida pelo inversor usado neste projeto. A escala horizontal vai de zero a 600 Watts. Os conjuntos fuzzy e a definição dos pontos das suas funções de pertinência são mostrados no Algoritmo 5.3:

Algoritmo 5.3 fuzzificação da variável potência de entrada do inversor em FCL

```
FUZZIFY We  
  TERM Baixa := (0, 1) (75, 1) (225, 0);  
  TERM Media := (75, 0) (225, 1) (375, 0);  
  TERM Alta := (225, 0) (375, 1) (525, 0);  
  TERM MuitoAlta := (375, 0) (525, 1) (600, 1);  
END_FUZZIFY
```

A Figura 5.4 mostra graficamente as funções de pertinência associadas a cada conjunto fuzzy. A área mais à esquerda corresponde a valores baixos de potência consumida. A próxima área à direita em formato triangular corresponde à potência Média. A próxima área à direita, também em formato triangular, corresponde à potência alta consumida. A área mais à direita corresponde à potência muito alta, chegando ao limite de capacidade do equipamento.

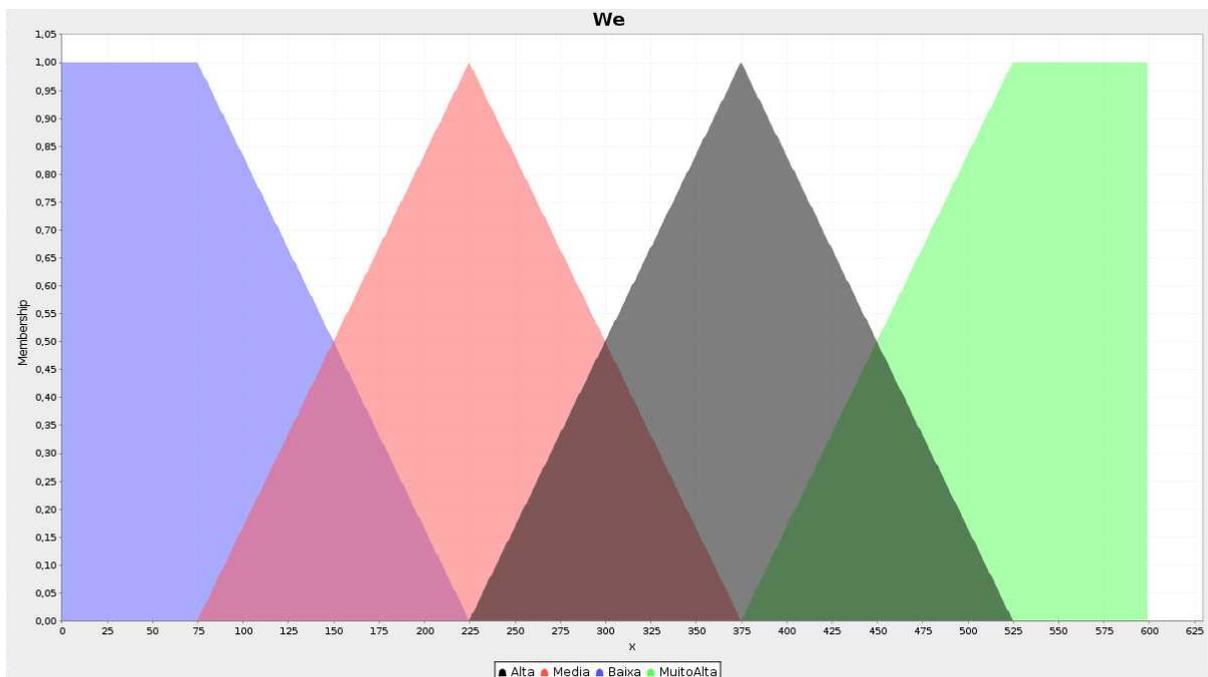


Figura 5.4 - Funções de Pertinência Associadas à Variável We = Potência de Entrada

5.4.5 Fuzzificação das Variáveis – Potência de Saída

A variável W_s = Potência de Saída recebeu quatro conjuntos fuzzy que indicam a intensidade da potência fornecida pelo inversor. A escala horizontal vai de zero a 440W. Os conjuntos fuzzy e a definição dos pontos das suas funções de pertinência são mostrados no Algoritmo 5.4:

Algoritmo 5.4 fuzzificação da variável potência de saída do inversor em FCL

```
FUZZIFY Ws  
TERM Baixa := (0, 1) (55, 1) (165, 0);  
TERM Media := (55, 0) (165, 1) (275, 0);  
TERM Alta := (165, 0) (275, 1) (385, 0);  
TERM MuitoAlta := (275, 0) (385, 1) (440, 1);  
END_FUZZIFY
```

A Figura 5.5 mostra graficamente as funções de pertinência associadas a cada conjunto fuzzy. A área mais à esquerda corresponde a valores baixos de potência fornecida. A próxima área à direita em formato triangular corresponde à potência Média. A próxima área à direita, também em formato triangular, corresponde à potência alta fornecida. A área mais à direita corresponde à potência muito alta, chegando ao limite de capacidade do equipamento.

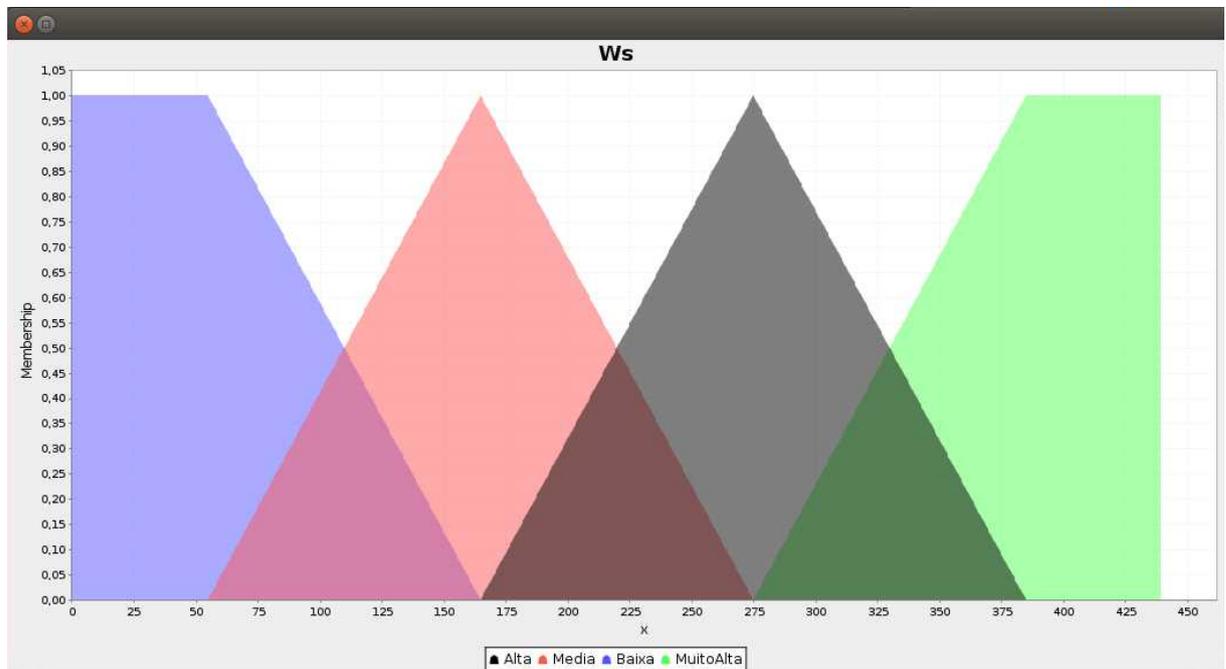


Figura 5.5 - Funções de Pertinência Associadas à Variável Ws = Potência de Saída

5.4.6 Fuzzificação das Variáveis – Eficiência do Inversor

De acordo com a Norma Internacional IEC/TS 61836, a eficiência de conversão de corrente contínua (CC) para corrente alternada (CA) de um equipamento inversor, é definida como a razão entre a potência fornecida na saída do inversor e a potência recebida na entrada do inversor (BASSAN, 2018). A eficiência do inversor em porcentagem é dada através da Expressão 5.1:

$$E_i = \frac{100 * W_s}{(W_e)} \quad (5.1)$$

Onde W_s = Potência de Saída, W_e = Potência de Entrada. A variável E_i recebeu três conjuntos fuzzy. A escala horizontal vai de zero a 100%. A eficiência de um inversor, em operação normal, varia pouco dentro de uma determinada faixa, sofrendo variação principalmente devido à temperatura. Uma grande variação na eficiência normalmente indica a ocorrência de falhas relativamente graves. Assim, foram definidos três conjuntos fuzzy, onde o conjunto Baixa indica gravidade alta. Os conjuntos fuzzy e a definição dos pontos das suas funções de pertinência são mostrados no Algoritmo 5.5:

Algoritmo 5.5 fuzzificação variável eficiência do inversor em FCL

```
FUZZIFY  $E_i$   
  TERM Baixa := (0, 1) (60, 1) (70, 0);  
  TERM Media := (60, 0) (70, 1) (80, 0);  
  TERM Normal := (70, 0) (80, 1) (100, 1);  
END_FUZZIFY
```

Neste trabalho de pesquisa, a eficiência é usada como informação útil no monitoramento de condição, já que um dos indicadores mais importantes de anomalia no funcionamento é a sua redução. A redução na eficiência de conversão de energia, devido à ocorrência de falha, se traduz por dissipação excessiva de calor e na redução da capacidade de potência.

O valor da eficiência normalmente é fornecido em porcentagem. Os inversores mais modernos de dois estágios tem eficiência maior que 90%. O inversor usado neste projeto, pelas razões que já foram expostas, tem eficiência menor, em torno de 80% devido à arquitetura de um estágio com transformador de baixa frequência. A maior parte da perda é devida ao transformador. Por esta razão, o conjunto fuzzy referente à eficiência Normal tem valor máximo igual a 80%. O conjunto Normal vai decrescendo até zerar em 70%, onde o conjunto Media tem seu valor máximo. O conjunto Baixa inicia na eficiência de 70% e tem valor máximo em 60% até zero. A eficiência tendendo a Baixa indica falha potencialmente grave provavelmente no transformador.

A Figura 5.6 mostra graficamente as funções de pertinência associadas a cada conjunto fuzzy. A região mais à esquerda corresponde a valores baixos de eficiência. A região central, em formato triangular, corresponde à eficiência Média. A região mais à direita corresponde à eficiência Normal do inversor.

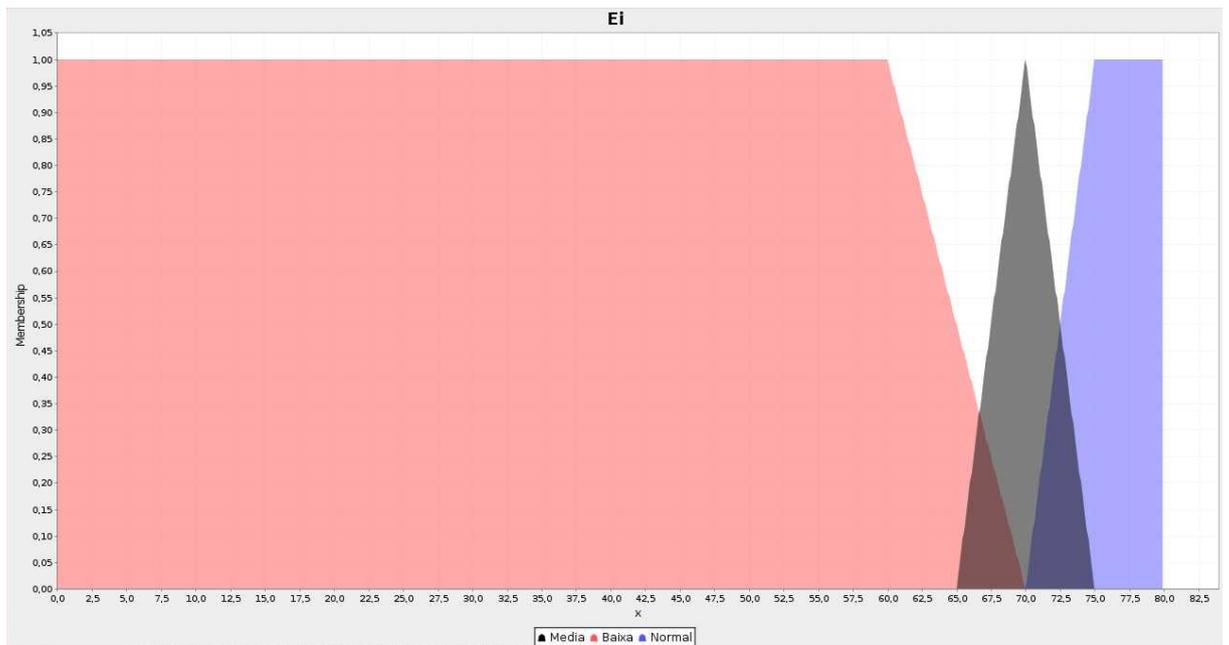


Figura 5.6 - Funções de Pertinência Associadas à Variável Ei = Eficiência do Inversor

5.4.7 Fuzzificação das Variáveis – Temperatura dos MOSFETs

A variável Td = Temperatura dos Dissipadores dos MOSFETs recebeu quatro qualificadores que indicam o valor da temperatura. A escala horizontal vai de zero a 75°C. Os qualificadores e a definição dos pontos das suas funções de pertinência são mostrados no Algoritmo 5.6:

Algoritmo 5.6 fuzzificação variável Temperatura dos MOSFETs em FCL

```

FUZZIFY Td
  TERM Baixa := (0, 1) (35, 1) (45, 0);
  TERM Normal := (35, 0) (45, 1) (55, 0);
  TERM Alta := (45, 0) (55, 1) (65, 0);
  TERM MuitoAlta := (55, 0) (65, 1) (75, 1);
END_FUZZIFY

```

A Figura 5.7 mostra graficamente as funções de pertinência associadas a cada qualificador. A área mais à esquerda corresponde aos valores baixos de temperatura. Isto significa que o inversor acabou de ser ligado, ou que está desligado, ou pode indicar falha no sensor, pois, em operação normal, a temperatura tende a estabilizar em torno de 40°C. Na mesma figura 5.7, a próxima área à direita, em formato triangular, corresponde aos valores normais de temperatura. A próxima área à direita, também em formato triangular, corresponde à temperatura alta. A área mais à direita corresponde à temperatura muito alta:

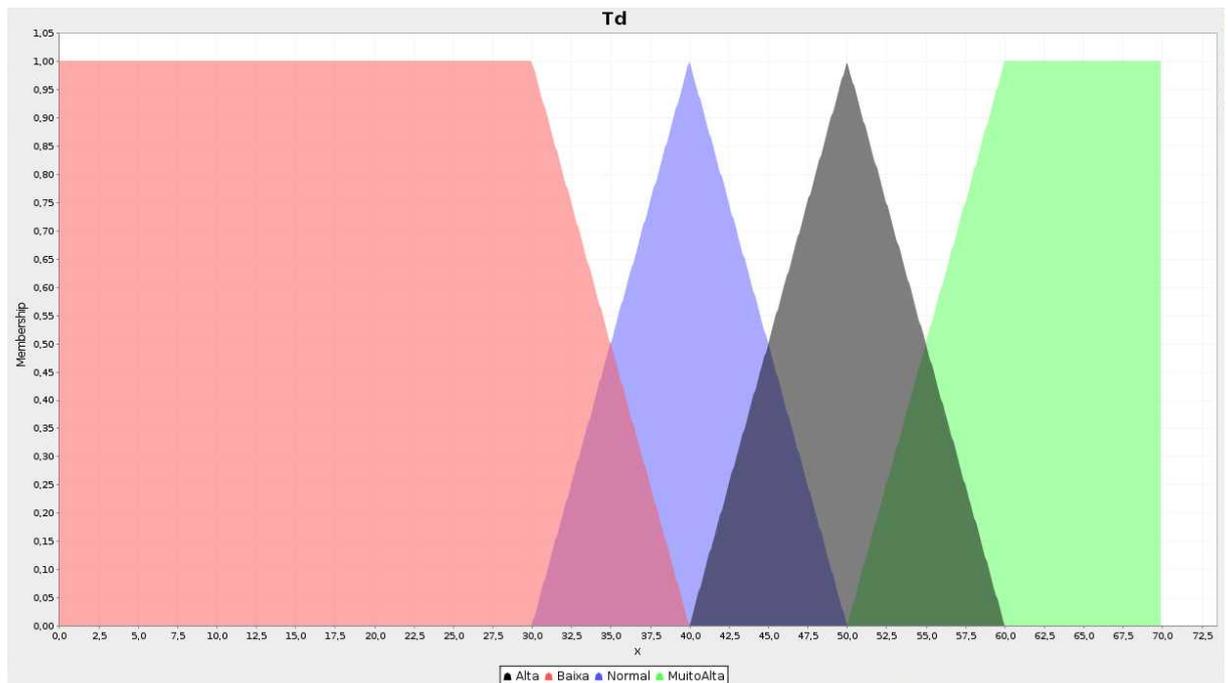


Figura 5.7 - Funções de Pertinência Associadas à Variável Td = Temp. dos MOSFETs

5.4.8 Fuzzificação das Variáveis – Temperatura do Transformador

A variável Tt = Temperatura do Transformador de Núcleo de Ferro recebeu quatro qualificadores que indicam a faixa referente ao valor da temperatura do componente. A escala horizontal da medida de temperatura vai de zero a 75°C, faixa que inclui os valores mínimos e máximos possíveis.

Os qualificadores e a definição dos pontos das suas funções de pertinência são mostrados no Algoritmo 5.7:

Algoritmo 5.7 fuzzificação variável Temperatura do Transformador em FCL

```

FUZZIFY Tt
  TERM Baixa := (0, 1) (35, 1) (45, 0);
  TERM Normal := (35, 0) (45, 1) (55, 0);
  TERM Alta := (45, 0) (55, 1) (65, 0);
  TERM MuitoAlta := (55, 0) (65, 1) (75, 1);
END_FUZZIFY

```

A Figura 5.8 mostra graficamente as funções de pertinência associadas a cada qualificador. A área mais à esquerda corresponde aos valores baixos de temperatura. Isto significa que o inversor acabou de ser ligado, ou que está desligado, ou pode indicar falha no sensor, pois, em operação normal, a temperatura tende a se estabilizar em torno de 42,5°C. A próxima área à direita, em formato triangular, corresponde aos valores normais de temperatura. A próxima

área à direita, também em formato triangular, corresponde à temperatura alta e a área mais à direita corresponde à temperatura muito alta:

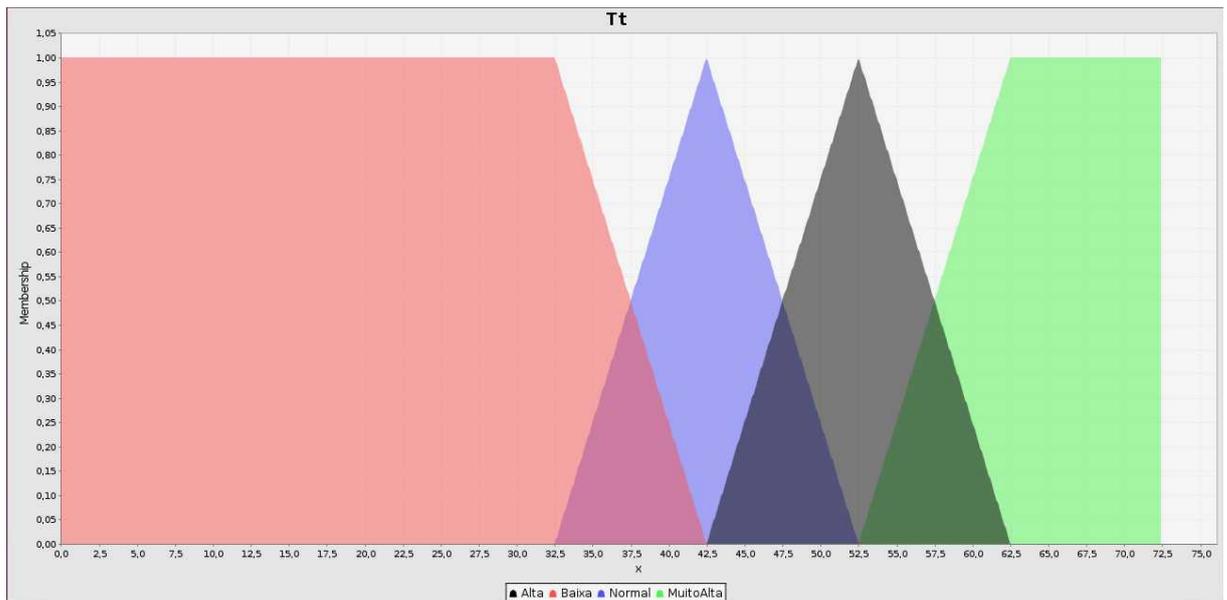


Figura 5.8 - Funções de Pertinência Associadas à Variável Tt = Temp. do Transformador

5.4.9 Definição das Variáveis de Saída

O critério para definição das variáveis de saída deve levar em conta os objetivos propostos dentro do conceito de manutenção baseada em condição. Desta forma, o primeiro passo é usar o conhecimento especialista para identificar o contexto de operação do equipamento.

Um inversor é um equipamento eletrônico de potência que recebe, converte e fornece energia para vários tipos de cargas. A potência fornecida pode variar muito dentro da faixa de operação, e devido a questões econômicas, os inversores são projetados levando-se em consideração que a probabilidade de ocorrer operação com alto carregamento durante tempo prolongado é baixa. A operação do inversor em situação próxima do limite máximo de potência leva ao aquecimento dos dissipadores dos MOSFETs e do transformador devido à dissipação de energia calculada pela potência dissipada Pd dada pela Expressão 5.2:

$$Pd = \frac{We (100 - Ei)}{100} \quad (5.2)$$

Onde We é a potência de entrada do inversor em Watts e Ei é a eficiência do inversor em porcentagem. Por exemplo, no caso do inversor estar fornecendo potência máxima em condições normais, We = 570W, a eficiência Ei = 80%, a potência transformada em calor pelo inversor é igual a 134W. A variável de saída deve refletir a “saúde” do equipamento através

da aplicação das regras de inferência fuzzy. Algumas condições presentes nas medidas podem indicar anomalias no funcionamento, principalmente a queda na eficiência e o aquecimento excessivo. Portanto, foi definida a seguinte variável de saída para o monitoramento de condição do inversor que é mostrada no Algoritmo 5.8:

Algoritmo 5.8 fuzzificação variável Temperatura do Transformador em FCL

```
VAR_OUTPUT
  SaudeInversor : REAL;
END_VAR
```

5.4.10 Defuzzificação - Variáveis de Saída - SaudeInversor

A variável de saída SaudeInversor recebeu quatro qualificadores que indicam o nível de saúde do inversor, ou seja, uma variável composta pelas condições de operação com o objetivo de detectar anomalias. Os valores para SaudeInversor são definidos no intervalo 0 a 100% conforme é mostrado no Algoritmo 5.9:

Algoritmo 5.9 defuzzificação da variável de saída SaudeInversor em FCL

```
DEFUZZIFY SaudeInversor
  TERM Critica := (0, 1) (20, 1) (40, 0);
  TERM Alerta := (20, 0) (40, 1) (60, 0);
  TERM Atencao := (40, 0) (60, 1) (80, 0);
  TERM Normal := (60, 0) (80, 1) (100,1);
END_DEFUZZIFY
```

A Figura 5.9 mostra graficamente as funções de pertinência associadas a cada qualificador. A área mais à esquerda corresponde a valores muito baixos ou críticos de saúde do inversor. A próxima área à direita, em formato triangular, corresponde ao estado de Alerta, ou seja, estados baixos de saúde. A próxima área à direita, também em formato triangular, corresponde ao estado de Atenção, logo abaixo do estado Normal. A região mais à direita corresponde ao estado de saúde normal do inversor.

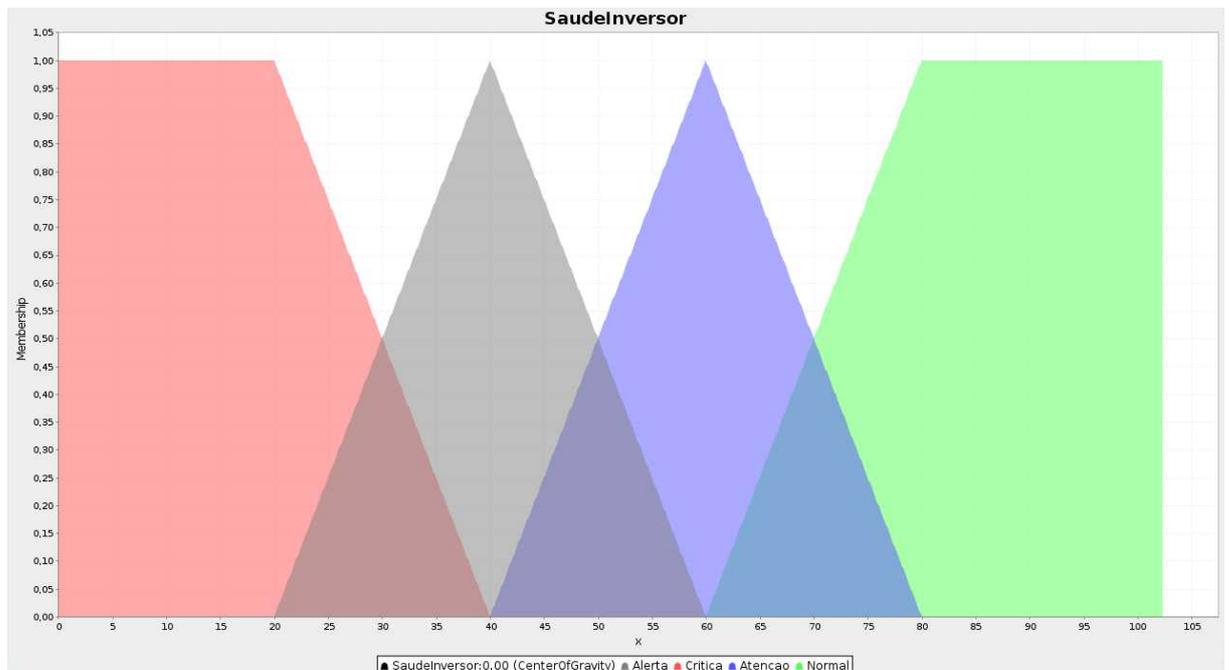


Figura 5.9 - Funções de Pertinência Associadas à Variável de Saída SaudeInversor

5.4.11 Definição das Regras de Inferência Fuzzy

Neste projeto foram definidas duas variáveis de saída para o monitoramento de condição do inversor: EstresseInversor e SaudeInversor. Portanto, deve-se ter dois conjuntos de regras para ativação de cada variável. Estes dois conjuntos de regras podem ser colocados dentro do mesmo bloco. A seguir, é apresentado no Algoritmo 5.10 um conjunto simplificado de regras:

Algoritmo 5.10 regras para a lógica de controle fuzzy em FCL

```

RULEBLOCK Regras1
// Use 'min' for 'and' (also implicit use 'max' for 'or' to fulfill DeMorgan's Law)
  AND : MIN;
  ACT : MIN; // Use 'min' activation method
  ACCU : MAX; // Use 'max' accumulation method

  RULE 1 : IF Td IS Baixa OR Tt IS Baixa OR Ws IS Baixa THEN EstresseInversor IS Baixo;
  RULE 2 : IF Td IS Normal OR Tt IS Normal OR Ws IS Media THEN EstresseInversor IS Medio;
  RULE 3 : IF Td IS Alta OR Tt IS Alta OR Ws IS Alta THEN EstresseInversor IS Alto;
  RULE 4 : IF Td IS MuitoAlta OR Tt IS MuitoAlta OR Ws IS MuitoAlta THEN
    EstresseInversor IS MuitoAlto;

// Regras para SaudeInversor = Critica
  RULE 5 : IF ((Ei IS Baixa) OR (Td IS MuitoAlta) OR (Tt IS MuitoAlta)) THEN SaudeInversor IS
    Critica;

// Regras para SaudeInversor = Alerta
  RULE 6 : IF ((Ws IS Baixa) OR (Ws IS Media)) AND (Ei IS Normal) AND ((Td IS Alta) OR (Tt
  IS
    Alta)) THEN SaudeInversor IS Alerta;

// Regras para SaudeInversor = Atencao
  RULE 3 : IF (Ws IS Media) AND (Ei IS Normal) AND ((Td IS Alta) OR (Tt IS Alta))
    THEN SaudeInversor IS Atencao;
  RULE 3 : IF (Ws IS MuitoAlta) THEN SaudeInversor IS Atencao;

```

```
// Regras para SaudeInversor = Normal
RULE 7 : IF (Ws IS Baixa) AND (Ei IS Normal) AND ((Td IS Normal) OR (Tt IS Normal))
        THEN SaudeInversor IS Normal;
RULE 8 : IF (Ws IS Media) AND (Ei IS Normal) AND ((Td IS Normal) OR (Tt IS Normal)) THEN
        SaudeInversor IS Normal;
RULE 9 : IF (Ws IS Alta) AND (Ei IS Normal) AND ((Td IS Normal) OR (Tt IS Normal) OR (Td
IS
        Alta) OR (Tt IS Alta)) THEN SaudeInversor IS Normal;

RULE 10 : IF (Ws IS MuitoAlta) AND (Ei IS Normal) AND ((Td IS Normal) AND (Tt IS Normal))
        THEN SaudeInversor IS Normal;

END_RULEBLOCK
```

6 - Resultados Obtidos

Visando comprovar as funcionalidades de supervisão, controle e monitoramento de condição que foram implementadas, foram criados um conjunto de testes. Assim, neste capítulo, são apresentadas a descrição dos testes, os resultados obtidos e as análises correspondentes. Os testes foram divididos basicamente em dois temas principais:

1. Testes dos Recursos de comunicação;
2. Testes dos Recursos de monitoramento de condição;

6.1 Testes dos Recursos de Comunicação

6.1.1 Considerações Gerais

Os testes durante o desenvolvimento e a análise das mensagens trocadas entre os dispositivos foram feitas usando-se o programa Wireshark. O Wireshark é um analisador de protocolos de rede amplamente usado nos meios acadêmico e empresarial. Este analisador mostra de forma bem detalhada as mensagens, permitindo verificar de modo bem objetivo a compatibilidade com os protocolos usados. Assim, é testada a compatibilidade das mensagens implementadas com as regras definidas pelos protocolos. Portanto, neste capítulo serão apresentadas as análises de todas as mensagens desenvolvidas durante esse projeto usando as capturas de telas das análises feitas através deste programa.

As informações de supervisão e controle sobre a usina serão apresentadas usando o navegador Google Chrome, o qual mostra uma tabela com todos os estados e medidas, além dos botões de comando. Este navegador foi escolhido por ser amplamente usado pelos usuários dos serviços Web inclusive para dispositivos móveis.

O Computador Raspberry PI 3 instalado na usina realiza as funções de conversão de protocolos CoAP para HTTP. Este computador roda simultaneamente dois softwares Java (jar) com as seguintes funções:

1. Servidor HTTP para a Intranet. Este software tem por função receber as solicitações dos clientes HTTP na rede local, fornecer os códigos HTML e Javascript, efetuar a comunicação com o Concentrador em Protocolo CoAP-OSA-CBM e responder às requisições de atualização com os valores das variáveis em formato XML. Ao receber a solicitação CoAP-OSA-CBM, o Concentrador envia uma mensagem de requisição MODBUS RTU a cada um dos dispositivos do processo e espera a resposta;

2. Cliente HTTP Conversor de Protocolos para Atualização do Servidor em Nuvem. Este software tem por função periodicamente efetuar a comunicação com o Concentrador em protocolo CoAP-OSA-CBM para obter as informações dos dispositivos do processo. Em seguida, o software envia uma mensagem HTTP com método POST e área de dados binária (Type = Octet-Stream) para atualizar os valores das variáveis.

O computador virtual em nuvem rodando Ubuntu 16.04 também roda um software em Java (jar), o qual recebe requisições de clientes em protocolo HTTP, permitindo o acesso às informações da usina por navegadores remotos. A Figura 6.1 mostra o diagrama simplificado para ilustrar a plataforma de testes das mensagens de comunicação:

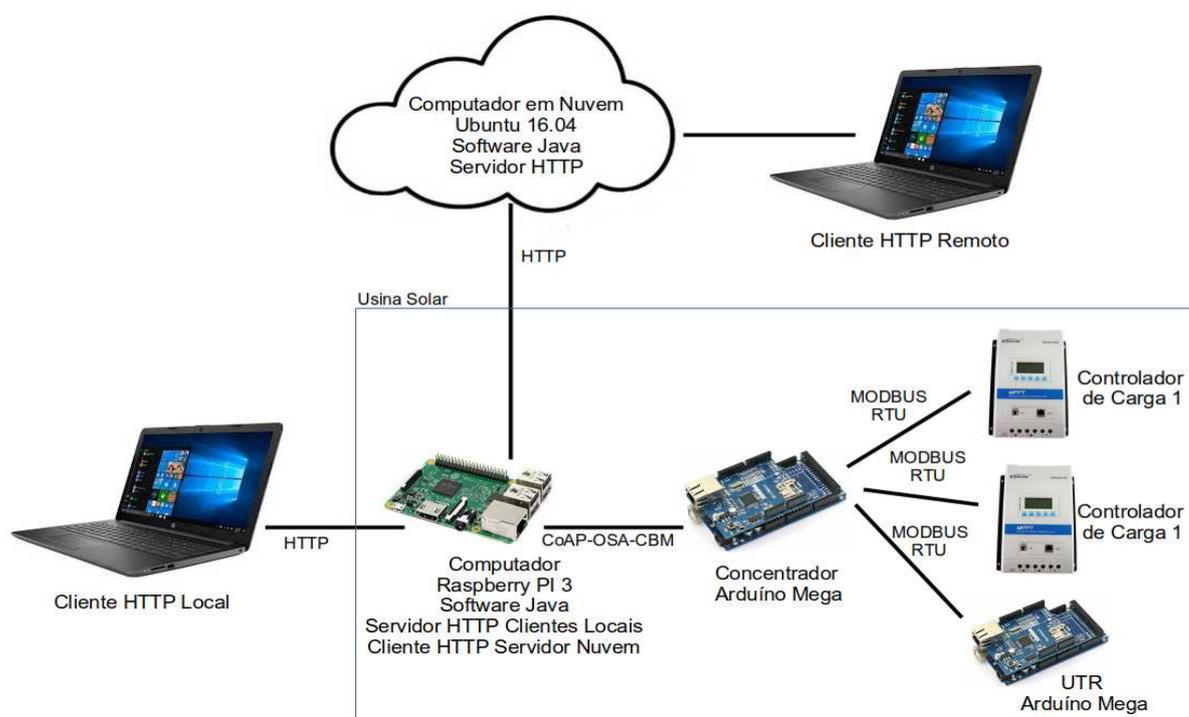


Figura 6.1 - Diagrama de Comunicação Simplificado

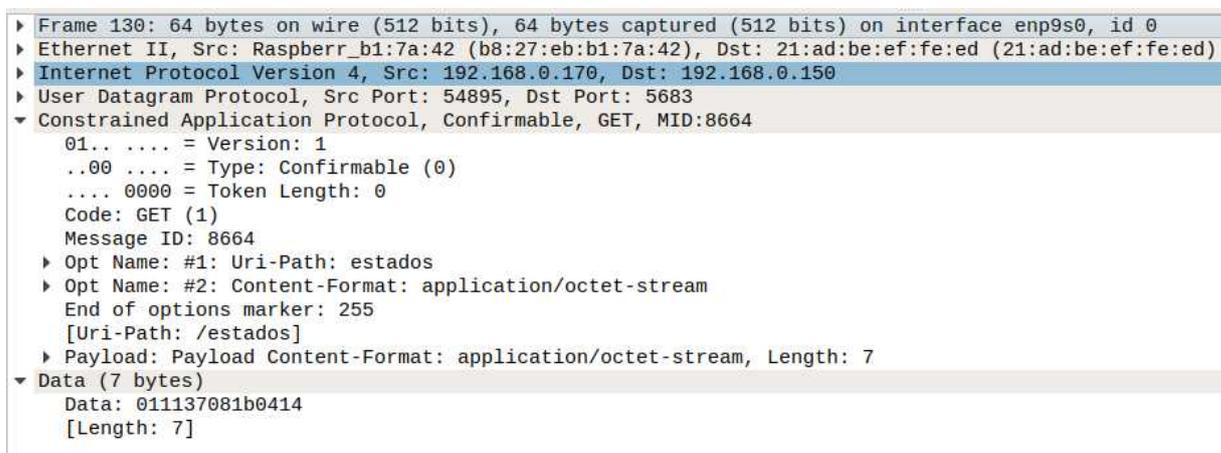
6.2 Análise da Comunicação - Protocolo CoAP-OSA-CBM

6.2.1 Considerações Gerais

A comunicação entre o Computador Raspberry PI 3 e o Concentrador Arduino Mega 2560 é feita usando o protocolo CoAP sobre datagramas UDP em rede Ethernet. O Servidor Raspberry PI 3, no endereço IP = 192.168.0.170, tem a função de Cliente e o Concentrador, no endereço IP = 192.168.0.150, opera como Servidor CoAP.

6.2.2 Análise da Requisição CoAP-OSA-CBM do Cliente para o Servidor

A Figura 6.2, obtida com o programa Wireshark, mostra a descrição dos campos da mensagem de requisição em protocolo CoAP do cliente para o servidor:



```
▶ Frame 130: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enp9s0, id 0
▶ Ethernet II, Src: Raspberr_b1:7a:42 (b8:27:eb:b1:7a:42), Dst: 21:ad:be:ef:fe:ed (21:ad:be:ef:fe:ed)
▶ Internet Protocol Version 4, Src: 192.168.0.170, Dst: 192.168.0.150
▶ User Datagram Protocol, Src Port: 54895, Dst Port: 5683
▼ Constrained Application Protocol, Confirmable, GET, MID:8664
    01.. .... = Version: 1
    ..00 .... = Type: Confirmable (0)
    .... 0000 = Token Length: 0
    Code: GET (1)
    Message ID: 8664
    ▶ Opt Name: #1: Uri-Path: estados
    ▶ Opt Name: #2: Content-Format: application/octet-stream
    End of options marker: 255
    [Uri-Path: /estados]
    ▶ Payload: Payload Content-Format: application/octet-stream, Length: 7
▼ Data (7 bytes)
    Data: 011137081b0414
    [Length: 7]
```

Figura 6.2 - Mensagem de Requisição CoAP apresentada pelo programa Wireshark

Na Linha 1 (mais escura), pode-se ver a origem da requisição no endereço 192.168.0.170 como cliente em protocolo CoAP, fazendo a solicitação ao servidor no endereço 192.168.0.150. Na Linha 2 aparece a identificação do protocolo UDP e os endereços das portas de conexão do equipamento de origem e destino. O valor 5683 para a porta de conexão é padronizado e identifica o uso das mensagens do protocolo CoAP sobre datagramas UDP.

A Linha 3 apresenta a identificação do uso do protocolo CoAP, com o tipo de mensagem Confirmable (Confirmável), o método GET, e o número sequencial de identificação de mensagem MID = 8664. O número sequencial de identificação é incrementado a cada mensagem de requisição. As Linhas 4,5,6,7 e 8 mostram os valores dos campos binários do início da mensagem, onde estão as informações de versão do protocolo CoAP, tipo de mensagem, método, número identificador sequencial e o uso ou não do Token. Nesta implementação, o atributo Token não é usado, porque se trata de requisições periódicas onde as informações sempre estão disponíveis, portanto, dispensando o uso deste recurso.

A Linha 9 mostra o primeiro campo Opt Name (Opção), que é do tipo URI-Path com o valor “estados”. Este tipo de opção fornece um indicador de recurso (URI) para instruir o servidor CoAP sobre qual informação foi requisitada pelo cliente. No caso, a palavra “estados” solicita o envio de todos os estados e medidas da usina. A Linha 10 mostra o segundo campo Opt name (Opção) do tipo Content-Format: application/octet-stream (formato binário). Esta opção informa que o conteúdo do campo Payload da mensagem de requisição é do tipo binário.

A Linha 11 mostra a identificação do byte de valor 255 decimal, que marca o fim do campo de opções e o início do Payload. A Linha 12 mostra o valor da URI e a Linha 13 apresenta o tipo de dado presente no Payload e o seu comprimento (7 bytes). A Linha 14 define o conteúdo do Payload como Data, por ser binário, e a Linha 15 mostra os valores dos bytes deste campo em formato hexadecimal. A Tabela 6.1 mostra o conteúdo do payload:

Tabela 6.1 - Bytes do Payload da Mensagem de Requisição CoAP

Byte	Valor Hexa	Valor Decimal	Descrição
0	01	01	Comando de Leitura de Estados e Medidas para a UTR
1	11	17	Hora do Relógio do Computador Raspberry PI 3
2	37	55	Minuto do Relógio do Computador Raspberry PI 3
3	08	08	Segundo do Relógio do Computador Raspberry PI 3
4	1b	27	Dia do Relógio do Computador Raspberry PI 3
5	04	04	Mês do Relógio do Computador Raspberry PI 3
6	14	20	Ano do Relógio do Computador Raspberry PI 3

6.2.3 Análise da Resposta CoAP-OSA-CBM do Servidor para o Cliente

A Figura 6.3 mostra a análise da mensagem de resposta CoAP-OSA-CBM do tipo ACK (Acknowledgement) enviada pelo Concentrador. Esta mensagem, faz o reconhecimento da requisição e carrega as informações de supervisão da usina (piggy-back response).

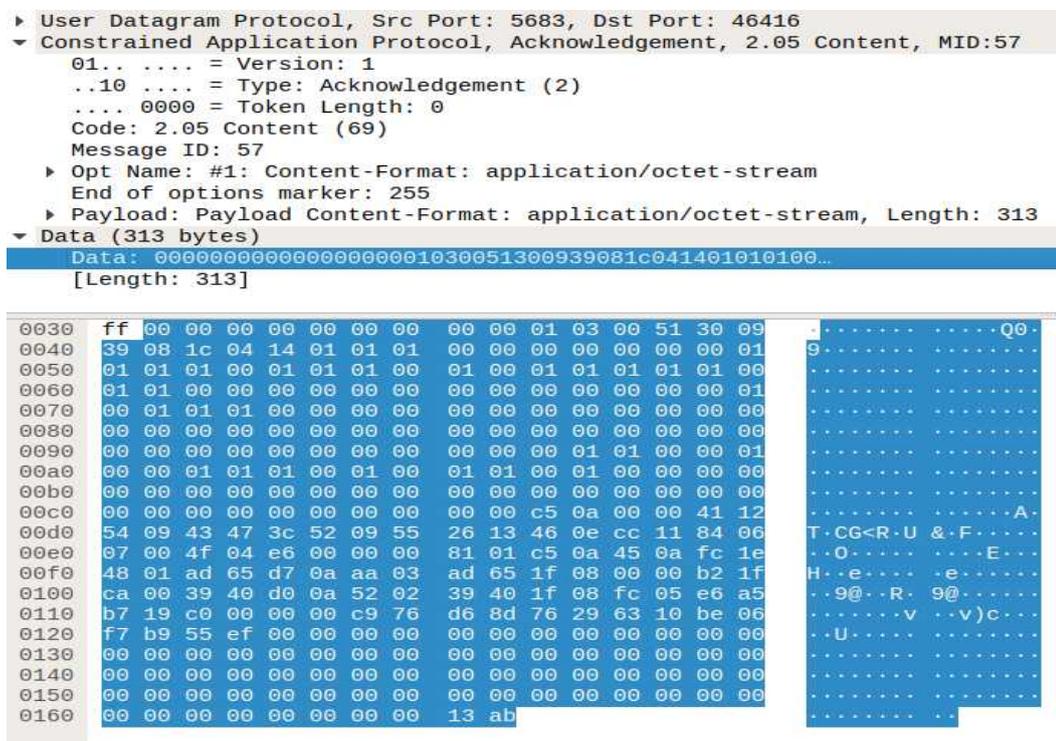


Figura 6.3 - Mensagem de Resposta CoAP-OSA-CBM do Concentrador

A primeira linha identifica o uso do Protocolo UDP com porta de origem do Concentrador igual a 5683. A segunda linha identifica o tipo da mensagem CoAP: Acknowledgement (Reconhecimento). O Código de Resposta: 2.05 indica Resposta com Sucesso do tipo Conteúdo (Content). O Identificador Sequencial da Mensagem MID = 1. Existe somente um campo de opção: Opt Name #1: Content-Format: application/octet-stream, que indica que o campo Payload carrega dados binários.

O cabeçalho da mensagem CoAP possui 7 bytes, e a mensagem completa tem 320 bytes, portanto, o Payload do tipo application/octet-stream tem 313 bytes (Data (313 bytes)). O payload CoAP carrega a mensagem OSA-CBM binária que contém todas as informações de estados e medidas da usina. Na análise, o payload aparece somente no formato octet-stream (binário) com 313 bytes. Na Figura 6.3, os bytes do Payload estão marcados após o byte 0xff (255 decimal) que sinaliza o final do campo de opções e o início do Payload.

6.3 Análise da Comunicação - Protocolo HTTP na Intranet

A seguir são apresentadas as análises das mensagens em protocolo HTTP trocadas entre o Servidor HTTP Local e os clientes.

6.3.1 Análise da Resposta do Servidor HTTP Local à Requisição GET /

A Figura 6.4 mostra a resposta do servidor HTTP local ao receber a requisição GET / do navegador Google Chrome:

```
▶ Internet Protocol Version 4, Src: 192.168.0.170, Dst: 192.168.0.158
▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 38082, Seq
▶ [24 Reassembled TCP Segments (33370 bytes): #539(1448), #541(1448),
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\n
    Server: Java HTTP Server from PraxServer : 1.0\n
    Date: Sun Feb 02 18:30:43 BRST 2020\n
    Content-type: text/html\n
  ▼ Content-length: 33224\n
    [Content length: 33224]
    \n
    [HTTP response 1/1]
    [Time since request: 0.194552537 seconds]
    [Request in frame: 532]
    [Request URI: http://192.168.0.170:8080/]
    File Data: 33224 bytes
▶ Line-based text data: text/html (629 lines)
```

Figura 6.4 - Análise da Resposta do Servidor Local à Requisição GET /

A primeira linha mostra o endereço IP do computador Raspberry PI 3 onde está instalado o software Servidor HTTP (IP = 192.168.0.170) e do computador onde está instalado o navegador (IP = 192.168.0.158). A linha seguinte mostra os valores das portas de conexão: a porta de origem no servidor HTTP com valor 8080 é padrão para este tipo de aplicação.

A quarta linha mostra a identificação do Protocolo HTTP, sendo seguida pelas informações do cabeçalho recebido: HTTP/1.1 200 OK indicando a correta resposta à requisição GET / feita pelo navegador. Em seguida são apresentadas as características do servidor HTTP: *Java HTTP Server from PraxServer 1.0*. A mensagem de resposta é do tipo text/html com 33224 caracteres. O tipo text/html significa que é uma mensagem de texto em formato HTML. As últimas duas linhas mostram que a mensagem do tipo texto tem 33224 bytes e 629 linhas.

Esta mensagem é lida de um arquivo que contém os códigos HTML e Javascript. O código HTML refere-se à tabela com os campos para os identificadores, estados, medidas e botões de comando a ser apresentada na tela do navegador. O código Javascript tem por função fazer a requisição periódica de mensagem XML com os valores de todas as variáveis para atualização dos campos da tabela e encaminhar os comandos.

6.3.2 Análise da Mensagem de Atualização XML para o Navegador

A Figura 6.5 mostra a mensagem de resposta do Servidor HTTP Local, contendo uma mensagem XML requisitada pelo navegador usando o método GET / local001.xml. Esta requisição é feita periodicamente pelo programa Javascript.

```
▶ Internet Protocol Version 4, Src: 192.168.0.170, Dst: 192.168.0.158
▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 38258, Seq:
▶ [2 Reassembled TCP Segments (2685 bytes): #16(1448), #18(1237)]
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\n
    Server: Java HTTP Server from PraxServer : 1.0\n
    Date: Tue Apr 28 14:42:30 BRT 2020\n
    Content-type: text/xml\n
  ▶ Content-length: 2542\n
    \n
    [HTTP response 1/1]
    [Time since request: 0.528152281 seconds]
    [Request in frame: 13]
    [Request URI: http://192.168.0.170:8080/local001.xml]
    File Data: 2542 bytes
  ▶ eXtensible Markup Language
```

Figura 6.5 - Análise da Resposta do Servidor Local à Requisição GET / atualiza

A primeira linha mostra os endereços IP do Servidor HTTP e do computador onde está o navegador. A linha seguinte mostra as portas de conexão com valor 8080 (padrão HTTP).

A quarta linha mostra a identificação do Protocolo HTTP, seguida pelas informações do cabeçalho: HTTP/1.1 200 OK indicando a correta resposta à requisição GET / local001.xml. Em seguida estão as características do servidor HTTP: *Java HTTP Server from PraxServer 1.0*. A resposta é do tipo texto em formato XML mostrada na linha 8 (*Content-type: text/xml*). O número de caracteres da mensagem XML é mostrada na linha 9 (*Content-length: 2542*).

6.3.3 Supervisão e Controle através de Navegador na Rede Local

A Figura 6.6 apresenta a tela do navegador Google Chrome ao carregar a página de Supervisão e Controle do Servidor HTTP Local com seus valores atualizados:

Supervisão e Controle Água, Energia e Usina Solar						
Conexão Servidor	Normal	Conexão Concentrador	Normal	Conexão UTR	Normal	Normal
Conexão C. Carga 1	Normal	Conexão C. Carga 2	Normal	Hora UTR:	24/04/20 11:40:36	Economia
Comando Apaga Indicadores de Falha		Temperatura Baterias	21,53°C	Estado do Inversor 2	Ligado	Manual 1
Tensão Barramento	27,50 V	Saúde Baterias	95 %	Corrente Entrada Inv2	7,91 A	Auto 1
Tensão Rede	Normal 209,14 V	Tensão Painéis 1 e 2	79,03 V	Potência Entrada Inv2	217,52 W	Manual 234
Modo de Operação	Normal	Corrente Painéis 1 e 2	3,58 A	Tensão Saida Inv2	216,42 V	Auto 234
Modo de Comando	Remoto	Corrente Saida CC1	10,26 A	Corrente Saida Inv2	0,885 A	Acerta
Modo Controle 1	Automatico	Potência Saida CC1	283,68 W	Potência Saida Inv2	191,53 W	Apaga
Modo Controle 234	Automatico	Saúde CC1	100 %	Temperatura Driver Inv2	44,38 C	Habilita 1
Energia Carga 1	Inversor 2	Tensão Painéis 3 e 4	79,84 V	Temperatura Trafo Inv2	47,88 C	Desabilita 1
Energia Carga 2	Rede	Corrente Painéis 3 e 4	3,19 A	Eficiência Inversor 2	88 %	Habilita 2
Energia Carga 3	Inversor 2	Corrente Saida CC2	9,13 A	Saúde Inversor 2	90 %	Desabilita 2
Corrente Carga 3	0,637 A	Potência Saida CC2	255,36 W	Estado do Inversor 1	Ligado	Habilita 3
Estado da Caixa Azul	Precisa Encher	Saúde CC2	100 %	Corrente Entrada Inv1	11,05 A	Desabilita 3
Nível da Caixa Azul	Normal	Corrente Total Gerada	19,39 A	Potência Entrada Inv1	303,87 W	Habilita 4
Estado da Bomba	Ligada	Potência Total Gerada	539,04 W	Tensão Saida Inv1	181,46 V	Desabilita 4
Disjuntor da Bóia	Ligado	Corrente Total Cargas	21,33 A	Corrente Saida Inv1	1,634 A	Atualiza
Disjuntor Rede Bomba	Desligado	Potência Total Cargas	586,97 W	Potência Saida Inv1	296,50 W	AutoRefresh
Energia Bomba	Inversor 1	Estado Fonte CC1	Ligada	Temperatura Driver Inv1	40,38 C	
Tempo Bomba Ligada	01:52:47	Estado Fonte CC2	Ligada	Temperatura Trafo Inv1	50,15 C	
Hora Bomba Ligou	-----	Corrente Cargas 24Vcc	2,358 A	Eficiência Inversor 1	97 %	
Hora Bomba Desligou	-----	Potência Cargas 24Vcc	64,84 W	Saúde Inversor 1	80 %	

Figura 6.6 - Tela do Navegador ao carregar a página de Supervisão e Controle Local

O código HTML mostra esta tabela na tela do navegador. As colunas 1, 3 e 5 a partir da esquerda possuem campos de descrição, sendo seguidas pelas colunas 2, 4 e 6 com os campos de valores dos estados e medidas. A coluna 7 contém os botões de comando. O campo Conexão Servidor mostra o estado de comunicação do navegador com o Servidor HTTP Local. O campo Conexão Concentrador mostra o estado de comunicação entre o Concentrador e o computador Raspberry PI 3. Os campos Conexão UTR, Conexão Controlador de Carga 1 e Conexão Controlador de Carga 2 mostram os estados de comunicação destes dispositivos com o Concentrador.

6.4 Análise usando o Protocolo HTTP para Acesso Remoto

A seguir são apresentadas as análises das mensagens em protocolo HTTP trocadas entre o Servidor HTTP em Nuvem e os clientes.

6.4.1 Envio dos Valores das Variáveis para o Servidor HTTP em Nuvem

O Servidor HTTP em Nuvem tem as suas variáveis atualizadas através de uma mensagem com método POST que carrega todos os estados e medidas em formato binário. Esta mensagem, gerada e enviada a cada quatro segundos, é feita pelo computador Raspberry PI 3 rodando o Software Cliente HTTP para Atualização do Servidor em Nuvem. A mensagem é mostrada na Figura 6.7:

```
Transmission Control Protocol, Src Port: 52692, Dst Port: 8080, Seq: 158, Ack: 1, Len: 320
[2 Reassembled TCP Segments (477 bytes): #43(157), #45(320)]
Hypertext Transfer Protocol
  ▶ POST /atualiza HTTP/1.1\r\n
    Host: 192.168.0.158:8080\r\n
  ▶ Content-Length: 320\r\n
    Content-Type: application/octet-stream\r\n
    User-Agent: (Linux x86_64) PraxClient/1.0\r\n
    \r\n
    [Full request URI: http://192.168.0.158:8080/atualiza]
    [HTTP request 1/1]
    [Response in frame: 47]
    File Data: 320 bytes
Data (320 bytes)
```

Figura 6.7 - Análise da Mensagem de Atualização do Conversor HTTP ao Servidor em Nuvem

A primeira linha mostra os endereços das portas de conexão, onde a porta destino do Servidor em Nuvem tem o valor padrão 8080. As linhas 3, 4, 5, 6 e 7 apresentam o método de requisição POST com a URI atualiza. Isto indica que o cliente solicita atualização de base de dados no Servidor em Nuvem usando a URI atualiza.

A linha 6 mostra o número de bytes da área de dados igual a 320 (Content-Length: 320) e a linha 7 contém o tipo de dados binário (Content-Type: application/octet-stream). A área de dados da mensagem HTTP, com 320 bytes, carrega o conteúdo binário CoAP-OSA-CBM.

O Software Cliente HTTP rodando no Computador Raspberry PI 3 efetua comunicação com o Concentrador a cada quatro segundos, armazena a mensagem recebida e a envia ao Servidor em Nuvem, que converte as informações recebidas em formato binário CoAP-OSA-CBM para o formato de mensagem XML.

6.4.2 Resposta do Servidor em Nuvem à Mensagem de Atualização

O Servidor HTTP em Nuvem, após receber corretamente a mensagem de atualização dos valores das variáveis, responde com uma mensagem HTTP do tipo XML com um código de comando. A função principal desta mensagem de resposta é enviar um comando para a UTR, que pode ter sido solicitado por um cliente HTTP remoto. Quando um cliente remoto pressiona um botão de comando na tela do navegador, o código do comando é recebido e armazenado no Servidor em Nuvem, sendo enviado para o Computador Raspberry PI 3 da usina como resposta da próxima mensagem de atualização dos valores das variáveis. A mensagem é mostrada na Figura 6.8:

```
▶ Internet Protocol Version 4, Src: 200.98.140.180, Dst: 192.168.0.158
▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 53798, Seq: 1,
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\n
    Server: Java HTTP Server from PraxServer : 1.0\n
    Date: Tue Apr 28 16:44:28 BRT 2020\n
    Content-type: text/xml\n
  ▶ Content-length: 50\n
    \n
    [HTTP response 1/1]
    [Time since request: 0.036114626 seconds]
    [Request in frame: 595]
    [Request URI: http://192.168.0.158:8080/atualiza]
    File Data: 50 bytes
▼ extensible Markup Language
  ▼ <?xml
    version="1.0"
    encoding="UTF-8"
    ?>
  ▼ <CMD>
    </CMD>
```

Figura 6.8 - Resposta à Mensagem de Atualização do Servidor em Nuvem

A mensagem de resposta em protocolo HTTP do Servidor em Nuvem possui tipo de conteúdo text/xml e contém cinquenta caracteres. Esta mensagem traz o código do comando e também sinaliza ao software cliente HTTP no Computador Raspberry PI 3 o correto recebimento da mensagem de atualização dos valores das variáveis. Caso não exista solicitação de comando pendente, o campo <CMD> é enviado vazio no formato <CMD></CMD>. Esta situação é mostrada na mensagem de resposta analisada na Figura 6.8.

6.4.3 Análise da Resposta do Servidor em Nuvem à Requisição GET /

A Figura 6.9, obtida com o programa Wireshark, mostra a resposta do servidor HTTP em Nuvem ao receber a requisição GET / do Navegador de um cliente HTTP remoto:

```
▶ Internet Protocol Version 4, Src: 200.98.140.180, Dst: 192.168.0.158
▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 58884, Seq: 36401, Ack: 477, Len: 1339
▶ [27 Reassembled TCP Segments (37739 bytes): #19(1400), #21(1400), #23(1400), #25(1400), #27(1400)
▶ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\n
    Server: Java HTTP Server from PraxServer : 1.0\n
    Date: Wed Apr 29 17:37:25 BRT 2020\n
    Content-type: text/html\n
  ▶ Content-length: 37594\n
    \n
    [HTTP response 1/1]
    [Time since request: 0.197786781 seconds]
    [Request in frame: 17]
    [Request URI: http://200.98.140.180:8080/]
    File Data: 37594 bytes
▶ Line-based text data: text/html (686 lines)
```

Figura 6.9 - Análise da Resposta do Servidor em Nuvem à Requisição GET /

Na Figura 6.9, a primeira linha mostra dois endereços: 1. O endereço IP do computador virtual em nuvem onde está instalado o software Servidor HTTP (IP = 200.98.140.180); 2. O endereço IP do computador cliente HTTP onde está instalado o Navegador Google Chrome (IP = 192.168.0.158).

A segunda linha apresenta os valores das portas de conexão: o valor 8080 é padrão. O cliente HTTP efetua a conexão ao servidor HTTP usando a porta com valor 58884. A quarta linha mostra a identificação do Protocolo HTTP, sendo seguida pelas informações do cabeçalho: HTTP/1.1 200 OK indicando a correta resposta à requisição GET / feita pelo navegador.

A sexta linha apresenta as informações referentes ao Software do Servidor HTTP: *Java HTTP Server from PraxServer 1.0*, seguida pelas informações de data, hora, o tipo da mensagem: text/html e o número de caracteres desta mensagem: 37594.

A mensagem de resposta é do tipo text/html com 37594 caracteres. O tipo text/html significa que é uma mensagem de texto em formato HTML. A mensagem de texto é composta por caracteres normalmente no padrão Unicode RTF 8.

A mensagem enviada como resposta pelo servidor em nuvem, é lida de um arquivo em formato texto, o qual contém os códigos HTML e Javascript. O código HTML refere-se à tabela com os campos para os identificadores, estados, medidas e botões de comando a ser apresentada na tela do navegador. O programa Javascript tem por função fazer a requisição periódica de mensagem XML com os valores de todas as variáveis para atualização dos campos da tabela.

6.4.4 Análise da Mensagem XML para o Navegador Remoto

A Figura 6.10, obtida com o programa Wireshark, mostra a mensagem de resposta do Servidor HTTP em Nuvem. É uma mensagem XML, em resposta à requisição GET / local001.xml. Esta requisição é feita automaticamente e periodicamente pelo programa Javascript executado no Navegador. A mensagem de resposta XML tem por objetivo atualizar automaticamente os valores das variáveis de supervisão da usina.

```
▶ Internet Protocol Version 4, Src: 200.98.140.180, Dst: 192.168.0.158
▶ Transmission Control Protocol, Src Port: 8080, Dst Port: 58886, Seq: 1401, Ack: 356, Len: 684
▶ [2 Reassembled TCP Segments (2083 bytes): #78(1400), #80(683)]
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\n
    Server: Java HTTP Server from PraxServer : 1.0\n
    Date: Wed Apr 29 17:37:26 BRT 2020\n
    Content-type: text/xml\n
  ▶ Content-length: 1940\n
    \n
    [HTTP response 1/1]
    [Time since request: 0.079423098 seconds]
    [Request in frame: 76]
    [Request URI: http://200.98.140.180:8080/local001.xml]
    File Data: 1940 bytes
  ▶ eXtensible Markup Language
```

Figura 6.10 - Análise da Resposta do Servidor em Nuvem à Requisição GET/atualiza

A primeira linha mostra o endereço do computador virtual em nuvem onde está instalado o software Servidor HTTP (IP = 200.98.140.180) e do computador onde está instalado o Navegador Google Chrome (IP = 192.168.0.158). A linha seguinte mostra os valores das portas de conexão: a porta de origem com valor 8080 é padrão para servidor HTTP.

A quarta linha mostra a identificação do Protocolo HTTP, sendo seguida pelas informações do cabeçalho recebido: HTTP/1.1 200 OK indicando a correta resposta à requisição GET / local001.xml feita pelo navegador. A próxima linha apresenta as características do Software

Servidor HTTP: *Java HTTP Server from PraxServer 1.0*. O tipo da mensagem de resposta é mostrada na linha 8 (*Content-type: text/xml*). O número de caracteres da mensagem XML é mostrada na linha 9 (*Content-length: 1940*). A última linha mostra que o conteúdo da mensagem está no formato Extensible Markup Language (XML) com 1940 bytes.

6.4.5 Supervisão e Controle através de Browser Remoto

A Figura 6.11 apresenta a tela do navegador Google Chrome ao carregar a página de Supervisão e Controle fornecida pelo Servidor em Nuvem:

Supervisao em Tempo Real - Água, Energia e Usina Solar - Servidor na Nuvem - Rev 1.0					
Conexão Servidor	Normal	Conexão Concentrador	Normal	Conexão UTR	Normal
Conexão Conversor	Normal	Conexão C. Carga 2	Normal	Hora UTR:	24/04/20 11:07:58
Conexão C. Carga 1	Normal	Temperatura Baterias	21,47°C	Estado do Inversor 2	Ligado
Modo Supervisão		Saúde Baterias	95 %	Corrente Entrada Inv2	7,58 A
Tensão Barramento	27,41 V	Tensão Painéis 1 e 2	67,21 V	Potência Entrada Inv2	207,76 W
Tensão Rede	Normal 227,43 V	Corrente Painéis 1 e 2	3,94 A	Tensão Saida Inv2	216,13 V
Modo de Operação	Normal	Corrente Saida CC1	9,37 A	Corrente Saida Inv2	0,844 A
Modo de Comando	Remoto	Potência Saida CC1	265,35 W	Potência Saida Inv2	182,41 W
Modo Controle 1	Automatico	Saúde CC1	100 %	Temperatura Driver Inv2	45,25 C
Modo Controle 234	Automatico	Tensão Painéis 3 e 4	72,05 V	Temperatura Trafo Inv2	48,53 C
Energia Carga 1	Inversor 2	Corrente Painéis 3 e 4	3,38 A	Eficiencia Inversor 2	87 %
Energia Carga 2	Rede	Corrente Saida CC2	8,85 A	Saúde Inversor 2	90 %
Energia Carga 3	Inversor 2	Potência Saida CC2	243,90 W	Estado do Inversor 1	Ligado
Corrente Carga 3	0,627 A	Saúde CC2	100 %	Corrente Entrada Inv1	11,05 A
Estado da Caixa Azul	Precisa Encher	Corrente Total Gerada	18,22 A	Potência Entrada Inv1	302,88 W
Nível da Caixa Azul	Normal	Potência Total Gerada	509,25 W	Tensão Saida Inv1	181,59 V
Estado da Bomba	Ligada	Corrente Total Cargas	20,91 A	Corrente Saida Inv1	1,619 A
Disjuntor da Bóia	Ligado	Potência Total Cargas	571,01 W	Potência Saida Inv1	293,99 W
Disjuntor Rede Bomba	Desligado	Estado Fonte CC1	Ligada	Temperatura Driver Inv1	40,65 C
Energia Bomba	Inversor 1	Estado Fonte CC2	Ligada	Temperatura Trafo Inv1	50,69 C
Tempo Bomba Ligada	01:20:10	Corrente Cargas 24Vcc	2,324 A	Eficiência Inversor 1	97 %
-----	-----	Potência Cargas 24Vcc	63,70 W	Saúde Inversor 1	90 %

Figura 6.11 - Tela do Navegador ao carregar a página de Supervisão do Servidor em Nuvem

A página HTML mostra esta tabela na tela do Navegador, onde as colunas 1, 3 e 5 a partir da esquerda possuem campos de descrição, sendo seguidas pelas colunas 2, 4 e 6 com os campos de valores dos estados e medidas. O campo Conexão Servidor mostra o estado de

comunicação do Navegador com o Servidor HTTP em Nuvem. O campo Conexão Conversor mostra o estado de comunicação do Computador Raspberry PI 3 da Usina com o Servidor em Nuvem. Os demais campos tem a mesma função apresentada na página de supervisão e controle local.

6.5 Análise do Monitoramento de Condição dos Inversores

6.5.1 Considerações Gerais

Os inversores são os equipamentos mais complexos e sujeitos a falhas da usina, principalmente por estarem conectados diretamente à rede elétrica de baixa tensão. Assim, a eficiência dos recursos de proteção e monitoramento de condição foram analisados através da simulação de um conjunto de falhas prováveis e as ações correspondentes tomadas pelos softwares. O equipamento Inversor 1 alimenta a bomba d'água instalada em um poço profundo. O equipamento Inversor 2 alimenta as demais cargas.

O software da UTR possui um conjunto de indicadores que recebem o valor 1 na ocorrência de falhas graves. Desta forma, as falhas graves são registradas através destes indicadores e podem ser lidas posteriormente. No caso de um ou mais indicadores de falha grave de um inversor estiver acionado, o inversor correspondente não é habilitado a ligar até que o indicador de falha seja apagado. Isto pode ser feito por ativação de comando na tela do navegador por um usuário ou reiniciando a UTR. Desta forma, o inversor permanece desligado e as cargas alimentadas pela rede sempre que ocorre uma falha grave, impedindo a propagação da falha, tornando a operação da usina mais segura e facilitando os procedimentos de manutenção.

6.5.2 Considerações sobre os Testes dos Inversores

Na usina solar estão instalados dois inversores SPWM de um estágio. Este tipo de inversor possui um transformador de baixa frequência com núcleo de ferro que possui quatro funções:

1. Funciona como indutor no filtro de saída juntamente com um capacitor para filtrar a portadora de 24KHz;
2. Eleva a tensão fornecida pela ponte H a partir da alimentação em corrente contínua de 24Vcc;
3. Proporciona isolamento elétrico também conhecida como isolamento galvânica entre a alimentação em corrente contínua de 24Vcc e a alimentação em corrente alternada de 220VCA para as cargas;

4. Funciona como elemento de proteção contra surtos de corrente para os elementos eletrônicos de chaveamento (MOSFETs).

Esta última função permite que se use um disjuntor termomagnético do tipo quick-lag como proteção contra curto-circuito na saída. Isto ocorre porque a indutância do transformador não permite que a corrente do curto atinja valores muito altos que possam danificar os elementos de chaveamento no tempo de abertura do disjuntor.

A potência máxima fornecida por cada inversor não é maior que 400 Watts. Desta forma, foi usado disjuntor termomagnético Siemens Quick-Lag B2. Este disjuntor atua a proteção térmica a partir da corrente RMS maior que dois Amperes, e atua a proteção magnética (instantânea) a partir da corrente instantânea maior que seis Amperes. Um curto circuito, portanto, gera uma corrente maior que seis Amperes em um tempo de até oito milissegundos, que é igual ao meio ciclo da frequência de saída de 60Hz, o que causa o desarme da proteção instantânea do disjuntor.

Além do disjuntor termomagnético de saída, a UTR fornece mais cinco proteções contra falhas graves para cada inversor apresentadas a seguir:

1. Subtensão na Saída. Valor Nominal = 220VCA. Limite Inferior = 198VCA (-10%);
2. Sobreensão na Saída. Valor Nominal = 220VCA. Limite Superior = 231VCA (+5%);
3. Sobretemperatura nos Drivers da Ponte H. Limite Superior = 55 Graus Centígrados;
4. Sobretemperatura no Transformador. Limite Superior = 55 Graus Centígrados;
5. Sobrecorrente na Entrada 24Vcc. Limite Superior = 20 Amperes;

O módulo de controle dos inversores permite o ajuste da tensão de saída, pois, possui uma entrada de realimentação. Este ajuste é feito através de um trim-pot na placa principal, o qual permite variar a tensão de menos vinte por cento a mais dez por cento da tensão nominal de saída.

A potência contínua máxima fornecida por cada inversor não é maior que 440 Watts. Desta forma, foi usado disjuntor termomagnético Siemens Quick-Lag B2. Este disjuntor atua a proteção térmica a partir da corrente RMS maior que dois Amperes, e atua a proteção magnética (instantânea) a partir da corrente instantânea maior que seis Amperes (disjuntor curva B).

6.5.3 Teste de Subtensão na Saída do Inversor

Este teste foi feito reduzindo-se a tensão de saída do inversor usando-se o trim-pot de ajuste até a tensão ficar um pouco abaixo da tensão limite para detecção de subtensão em torno de 195VCA. O módulo de controle SPWM EGS002 permite o ajuste da tensão RMS da saída do inversor através de um trim-pot de ajuste.

Como esperado, o software de proteção da UTR detectou a ocorrência de subtensão na saída do inversor, e após dez segundos, tempo definido pelo procedimento de proteção, o inversor foi desligado e as cargas foram transferidas para a rede da concessionária. O indicador correspondente foi acionado e o inversor permaneceu desligado até que este indicador fosse apagado.

O tempo de espera de dez segundos entre a detecção e o desligamento do inversor e o acionamento do indicador tem por finalidade evitar que reduções rápidas da tensão de saída do inversor causadas pelo acionamento de cargas com alta corrente de partida acionem a proteção. Estas cargas são normalmente motores de indução monofásicos.

6.5.4 Teste de Sobretensão na Saída do Inversor

Este teste foi feito aumentando-se a tensão de saída do inversor usando-se o trim-pot de ajuste até a tensão ficar um pouco acima da tensão limite para detecção de sobretensão em torno de 235VCA. O módulo de controle SPWM EGS002 permite o ajuste da tensão RMS da saída do inversor através de um trim-pot de ajuste.

O software de proteção da UTR detectou a ocorrência de sobretensão na saída do inversor, e após dez segundos, tempo definido pelo procedimento de proteção, o inversor foi desligado e as cargas foram transferidas para a rede da concessionária. O indicador correspondente foi acionado e o inversor permaneceu desligado até que este indicador fosse apagado.

6.5.5 Teste de Curto Circuito na Saída do Inversor

Um curto-circuito na saída CA do inversor gera uma corrente maior que 6 Amperes, que é a corrente de desarme da proteção magnética (instantânea) do disjuntor quick-lag. Este valor de corrente pode ocorrer em um tempo que varia de menos de um milissegundo até quatro milissegundos após o evento de curto circuito. Isto é ilustrado na Figura 6.12, onde:

O tempo mínimo de abertura do disjuntor acontece no caso da onda de tensão estar passando pelo máximo. Neste caso, a onda de subida da corrente aumenta com a taxa de subida máxima. O tempo máximo de abertura do disjuntor acontece no caso da onda de tensão estar passando pelo zero. Neste caso, a onda de subida da corrente aumenta junto com o aumento da onda de tensão.

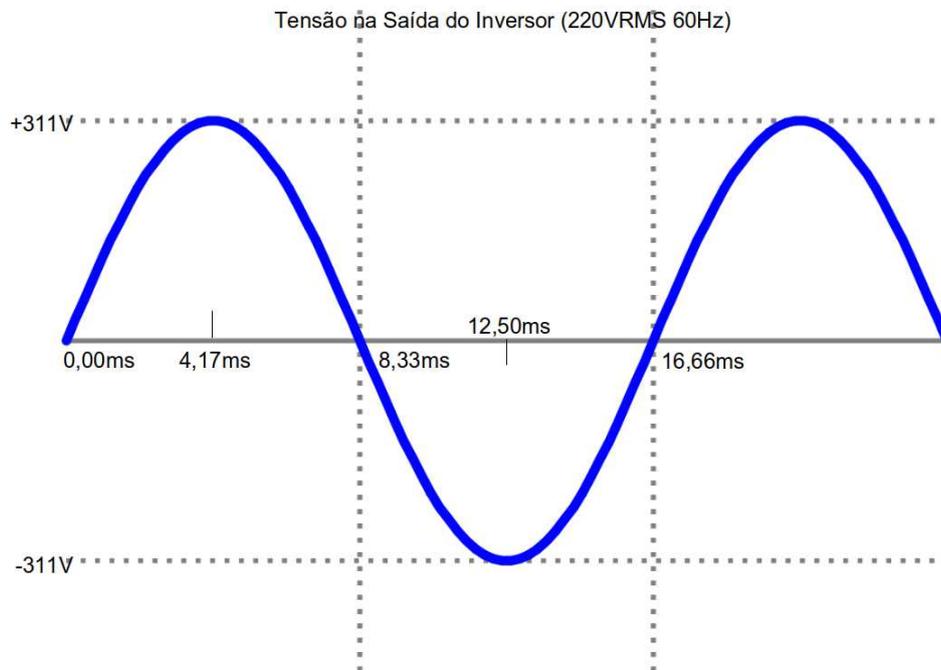


Figura 6.12 - Onda de Tensão na Saída do Inversor SPWM

Resultado do Teste: um curto-circuito na saída do inversor provocou o desarme do disjuntor B2 quick-lag de saída, desconectando as cargas. O desarme do disjuntor de saída também provocou a desconexão do sensor de tensão CA da UTR. Em seguida, o software de proteção da UTR detectou a ocorrência de subtensão na saída do inversor, e após dez segundos, o inversor foi desligado e as cargas foram transferidas para a rede da concessionária conforme esperado. O indicador correspondente foi acionado e o inversor permaneceu desligado até que este indicador foi apagado por comando.

6.5.6 Teste de Sobretemperatura dos Drivers (MOSFETs)

Este teste foi feito conectando-se uma carga de 350 Watts na saída do inversor e desligando-se o conjunto de dois ventiladores dos drivers da ponte H (MOSFETs). Estes ventiladores são instalados na parte superior da caixa do inversor, e servem para jogar o ar quente para fora da caixa. A parte inferior da caixa possui dois orifícios com o mesmo diâmetro dos ventiladores,

o que permite a circulação de um fluxo de ar de baixo para cima através dos dissipadores dos MOSFETs como mostra a Figura 6.13:

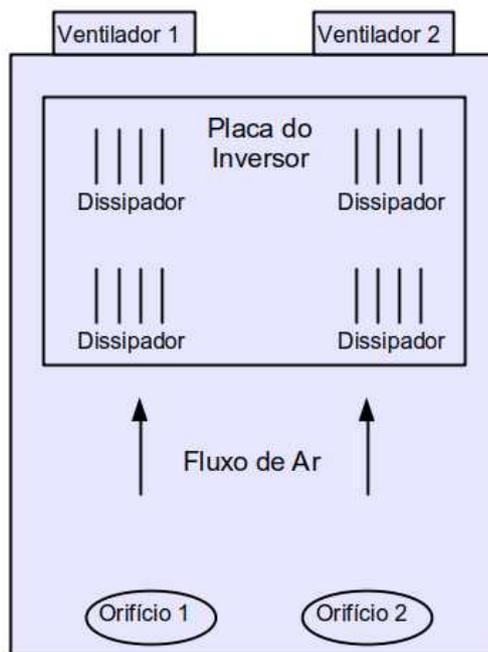


Figura 6.13 - Esquema de Ventilação Forçada dos Drivers do Inversor SPWM

Resultado do Teste: a temperatura nos dissipadores dos MOSFETs subiu até ultrapassar o valor limite. Como esperado, o software da UTR desligou o inversor correspondente, e as cargas em corrente alternada foram transferidas para a alimentação da rede da concessionária. O indicador de falha de sobretemperatura nos MOSFETs foi acionado e o inversor permaneceu desligado até que este indicador fosse apagado.

6.5.7 Teste de Sobretemperatura do Transformador do Inversor

Este teste foi feito conectando-se uma carga de 350 Watts na saída do inversor e desligando-se a ventilação forçada que faz o resfriamento do transformador de núcleo de ferro. Este transformador é instalado abaixo da caixa mostrada na figura 6.13. O ventilador é instalado sob o transformador do inversor, jogando ar diretamente sobre o componente.

Resultado do Teste: a temperatura do transformador subiu até ultrapassar o valor limite e o software desligou o inversor e as cargas foram transferidas para a rede da concessionária. O indicador correspondente foi acionado e o inversor permaneceu desligado até que este indicador fosse apagado.

6.5.8 Teste de Sobrecarga na Saída e Sobrecorrente de Entrada

O teste foi realizado solicitando-se uma corrente maior que dois amperes na saída do inversor com o objetivo de causar sobrecarga na saída e a consequente sobrecorrente na entrada de corrente contínua do inversor. Para isso, foi conectada uma carga de 600W na saída de corrente alternada 220V do inversor, o que causa a demanda de uma corrente maior que 20 Amperes na entrada de corrente contínua.

Resultado do Teste: o software da UTR, como esperado, detectou a sobrecorrente na entrada de alimentação, desligou o inversor, e as cargas em corrente alternada foram transferidas para a alimentação pela rede da concessionária. O indicador de falha correspondente foi acionado e o software da UTR manteve o inversor desligado até que o indicador de falha de sobrecorrente de entrada ser apagado por comando.

Observação 1: existem duas proteções de sobrecarga para os inversores: a proteção térmica do disjuntor quick-lag na saída de corrente alternada, e a proteção de sobrecorrente na entrada de corrente contínua feita pelo software da UTR. Quando ocorre sobrecarga na saída, normalmente é a proteção de sobrecorrente na entrada de corrente contínua que atua, por ser mais rápida que a atuação da proteção térmica do disjuntor. De qualquer maneira, a proteção térmica do disjuntor de saída opera como proteção de retaguarda no caso de sobrecarga, aumentando significativamente a confiabilidade dos mecanismos de proteção da usina.

Observação 2: além da ocorrência de sobrecarga na saída, podem ocorrer falhas internas no inversor que levem à atuação da proteção de sobrecorrente na entrada, tal como um componente de chaveamento em curto-circuito ou um problema nos sinais de controle destes componentes. Entretanto, esta ocorrência é muito rara, pois, é preciso que ocorra extremo sobreaquecimento ou surto de tensão na alimentação do inversor que cause danos a componentes. O sobreaquecimento possui proteção própria e existem diversas proteções contra surtos de tensão no barramento principal de corrente contínua.

6.5.9 Detecção de Falha Real no Inversor 1

Além dos testes usando os procedimentos de simulação de falhas, foi identificada uma falha real em um componente eletrônico (MOSFET) do Inversor 1. A constatação da redução do valor do indicador de saúde do Inversor 1 levou a uma inspeção detalhada do equipamento. Esta inspeção detectou o aquecimento anormal de um conjunto de dois MOSFETs da ponte H. Isto ocorreu mesmo com a ventilação forçada operando normalmente.

Assim, a placa principal foi retirada para manutenção e foi identificado um MOSFET aberto por causa de um rompimento mecânico do terminal. Este MOSFET fazia parte de um conjunto de dois MOSFETs ligados em paralelo, ou seja, durante a operação do inversor, somente um MOSFET estava conduzindo toda a corrente em um dos pontos de chaveamento da ponte H. Isto aumenta a dissipação de calor neste conjunto causando aquecimento anormal do equipamento.

A Figura 6.14 mostra o diagrama esquemático simplificado da ponte H que transforma a corrente contínua de 24Vcc em corrente alternada na frequência 24KHz com modulação SPWM na frequência de 60Hz. Na ponte existem quatro conjuntos de elementos de chaveamento, cada um com dois MOSFETs de potência conectados em paralelo. Para aplicar o sinal na saída, o módulo de controle aciona cada conjunto de dois MOSFETs simultaneamente da seguinte forma: O acionamento do Conjunto 1 com o Conjunto 4 gera um sinal positivo na saída; O acionamento do Conjunto 3 com o Conjunto 2 gera um sinal negativo na saída.

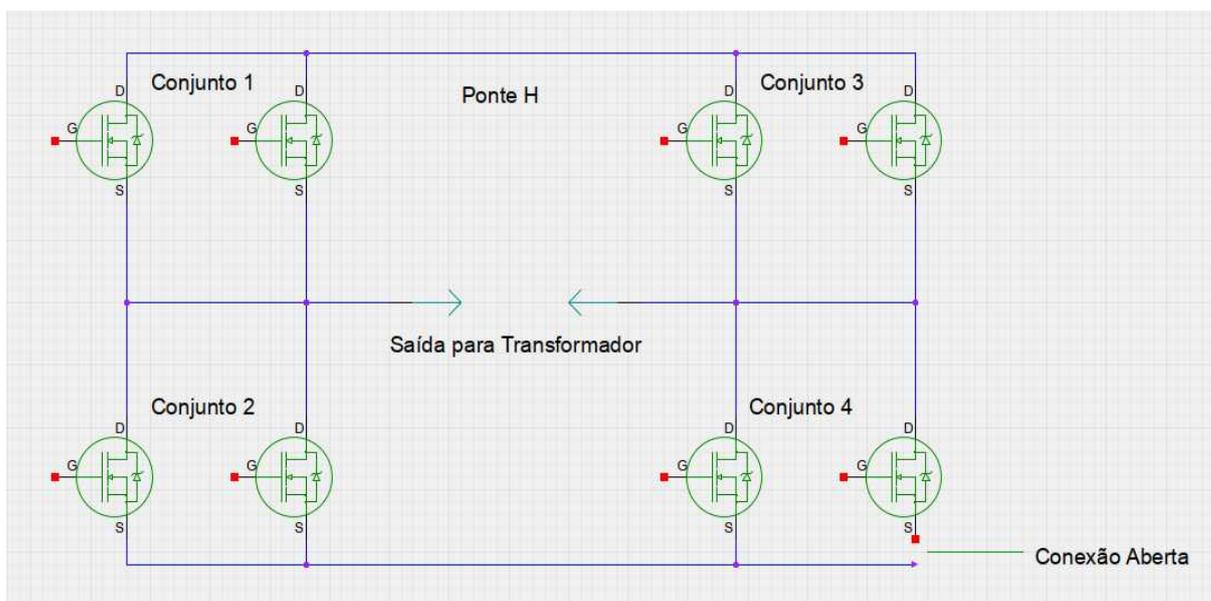


Figura 6.14 - Diagrama Esquemático Simplificado da Ponte H com Falha de Conexão

Cada elemento de conexão da ponte H é composto por dois MOSFETs ligados em paralelo para reduzir pela metade a resistência ôhmica. Isto tem por finalidade reduzir pela metade a dissipação de calor causada pela corrente elétrica, e conseqüentemente, aumentar a eficiência e melhorar a confiabilidade.

No caso aqui analisado, como pode ser visto no esquemático, um dos MOSFETs do Conjunto 4 está com o terminal de supridor aberto. Nesta condição, o inversor funciona, fornecendo o

sinal de 220V 60Hz em sua saída e alimentando as cargas. Entretanto, ocorre um aquecimento anormal do dissipador de calor do Conjunto 4 de MOSFETs, pois, o valor da resistência ôhmica deste conjunto é o dobro dos outros conjuntos, causando, desta forma, o dobro da dissipação de calor.

Em cargas baixas e médias, que é a condição normal de operação do inversor, isto não causa a atuação da proteção térmica. Entretanto, este aquecimento anormal causa uma diminuição do valor da variável SaudeInversor, o que foi efetivamente verificado na observação diária feita através da supervisão da usina.

6.6 Análise do Monitoramento de Condição dos Painéis

6.6.1 Considerações Gerais

Em uma usina solar do tipo off-grid, somente é solicitada a potência máxima dos painéis fotovoltaicos eventualmente, quando a usina está funcionando próxima da sua capacidade máxima de potência. Assim, para viabilizar a análise, o teste foi feito em um dia claro em um horário em que a energia solar máxima gerada é menor que a energia total consumida pelas cargas. Esta situação faz com que os equipamentos Controladores de Carga solicitem potência máxima dos painéis, viabilizando a realização do teste.

É bom lembrar que a diferença de energia, no caso do consumo ser maior que a geração solar, é fornecida pela rede da concessionária através das fontes de alimentação CA/CC, as quais injetam corrente no barramento principal de corrente contínua da usina.

Na usina, estão instalados dois conjuntos de dois painéis fotovoltaicos de 72 células e potência máxima de 320 Watts. Um conjunto de painéis está instalado na direção Noroeste e outro conjunto está instalado na direção Nordeste. Existem dois pequenos painéis fotovoltaicos de 5Wp instalados em local de fácil acesso, onde cada painel é posicionado na mesma direção o conjunto de dois painéis fotovoltaicos de 320Wp correspondente. Assim, é possível calcular em tempo real a Irradiância (I_r) aproximada que incide em cada conjunto de dois painéis da usina instalados no telhado, e desta forma, calcular a Potência Máxima (P_m) que esse conjunto de painéis pode fornecer mostrada na Expressão 6.1:

$$P_m = 0,32 * I_r \quad (6.1)$$

6.6.2 Histórico

Em Brasília durante a época da seca, que coincide com o inverno, chove muito pouco ou praticamente nada como aconteceu no ano de 2019. Neste ano, a última chuva mais expressiva aconteceu em meados de maio, e choveu somente no início de outubro. Ocorreu uma redução na capacidade de geração no final do período da seca de 15 a 20%, fato verificado através da comparação com as irradiâncias fornecidas pelos pequenos painéis fotovoltaicos de 5Wp.

Foi constatado, durante uma inspeção feita no final de setembro de 2019, que a poeira acumulada sobre os painéis no período da seca causou esta diminuição de geração. A inspeção é feita somente a cada 3 ou 4 meses, já que os painéis fotovoltaicos estão instalados no telhado da casa, local de difícil acesso. A comprovação desse fato foi feita logo após as primeiras chuvas expressivas no início de outubro que lavaram os painéis, fazendo a geração fotovoltaica voltar ao normal.

6.7 Análise do Monitoramento de Condição das Baterias

6.7.1 Introdução

A estimativa da saúde do banco de baterias somente pode ser calculada após o fim de um período de descarga. Assim, foi realizado um procedimento de teste, onde a energia da rede da concessionária foi desligada por um determinado tempo na falta de geração solar (à noite), simulando uma interrupção na rede.

O tempo de interrupção foi determinado usando-se o maior período registrado no último ano, de aproximadamente de dez horas. Os períodos de interrupção são longos porque a usina está instalada em uma propriedade rural, onde o serviço de manutenção da concessionária é normalmente mais demorado.

6.7.2 Cálculo da Saúde do Banco de Baterias

O gráfico apresentado na Figura 6.15 mostra a variação da tensão do banco de baterias durante o procedimento de teste de descarga. A Figura 6.16 apresenta a Ilustração Gráfica do Método de Estimativa do Tempo Máximo Projetado de Descarga das Baterias.

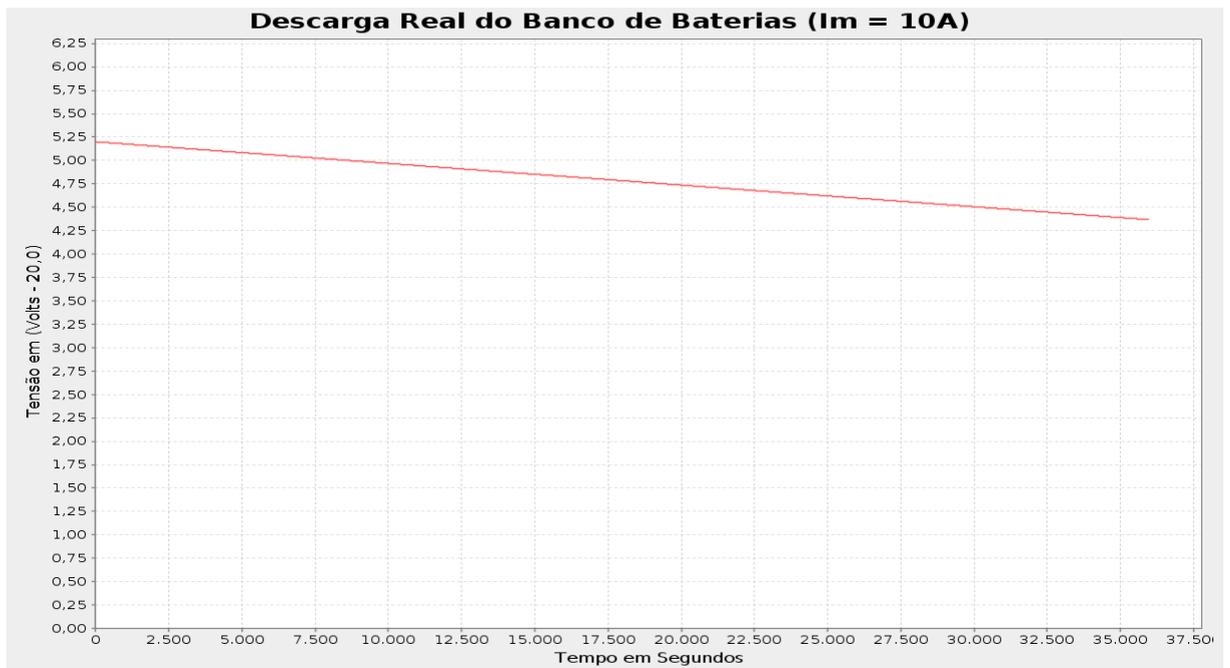


Figura 6.15 - Gráfico de Descarga Real do Banco de Baterias de 10 horas

Na Figura 6.15, o tempo total de descarga foi de 10 horas = 36000 segundos. A tensão inicial do banco de baterias $V_i = 25,20V$. A tensão final do banco de baterias após a descarga $V_f = 24.368V$.

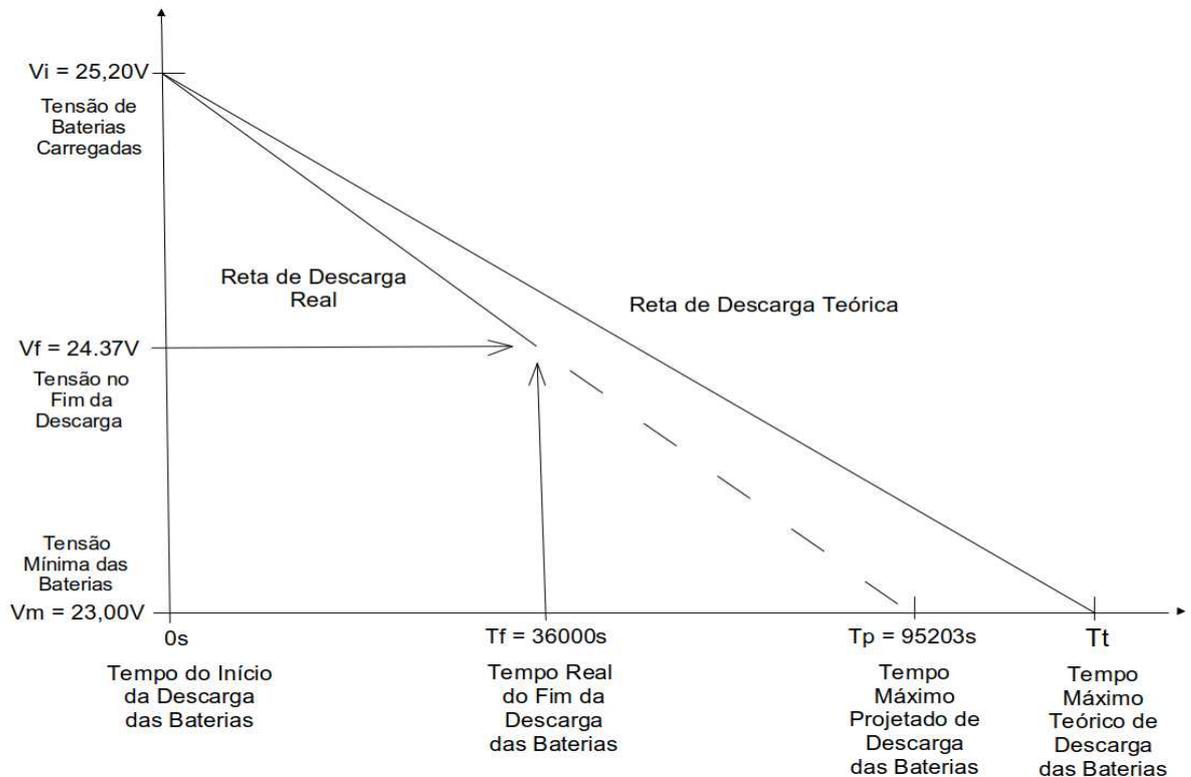


Figura 6.16 - Ilustração do Método de Estimativa do Tempo Máximo Projetado de Descarga

O Cálculo do Tempo Máximo Projetado e da Saúde do Banco de Baterias neste caso real foi realizado através das expressões mostradas a seguir:

Através da Expressão 4.4 foi calculado o tempo máximo de descarga projetado $T_p = 95203$ s.

Através da Expressão 4.5 foi calculada saúde do banco de baterias $S_{bb} = 95\%$.

7 - Conclusões

Na data da conclusão desta dissertação, já fazem 20 meses que a usina está em operação, em constante aperfeiçoamento e testes dentro deste trabalho de pesquisa. A funcionalidade de supervisão e controle facilitou muito a operação e manutenção da usina, contribuindo também para o desenvolvimento dos procedimentos de gerenciamento de cargas e monitoramento dos equipamentos.

Os benefícios ao usuário incluem principalmente a antecipação e resolução de falhas, e a economia de tempo, pois, com apenas uma visualização é possível verificar as condições de todo o sistema e tomar as ações necessárias caso seja necessário. Especificamente neste estudo de caso, foi implementado também a supervisão e controle do sistema de fornecimento de água, obtida através do bombeamento de um poço tubular profundo.

A seguir são apresentadas análise e conclusões dos principais tópicos apresentados no Capítulo 6.

7.1 Arquitetura de Controladores da Usina

Neste projeto, foram usados dois controladores do tipo Arduino Mega 2560 para implementar as funções de aquisição de dados de pontos físicos, processamento, controle e comunicação. Um controlador recebeu a função de Unidade Terminal Remota (UTR), realizando as tarefas de aquisição de dados de pontos físicos, processamento e controle. O outro controlador recebeu a função de Concentrador, efetuando a comunicação com os equipamentos do processo e conversão de protocolos de MODBUS RTU para CoAP-OSA-CBM. Inicialmente, foi constatado que somente um controlador Arduino Mega 2560 tem capacidade de processamento, memória, canais de comunicação e pontos físicos suficientes para implementar as funções de Unidade Terminal Remota e de Concentrador. Entretanto, durante o desenvolvimento do projeto, verificou-se o seguinte:

A UTR efetua o controle das chaves de transferência, o controle dos inversores e os algoritmos de proteção primária destes equipamentos. Assim, deve-se evitar ao máximo a interrupção na operação deste equipamento, primeiramente para que não ocorra interrupções no fornecimento de energia às cargas, e também para evitar interrupção no fornecimento de energia solar. Desta forma, estando o software da UTR estável, os ajustes no software de

comunicação que sejam necessários, são realizados no equipamento Concentrador, sem necessidade de interrupção na operação da UTR.

Esta arquitetura foi bastante adequada durante o desenvolvimento dos recursos de comunicação, pois, o software da UTR foi concluído antes e colocado em operação para máximo aproveitamento da geração solar. O uso do protocolo MODBUS RTU em interface serial assíncrona na UTR também tornou mais rápido o desenvolvimento, porque é muito usado em sistemas de supervisão, controle e automação na área elétrica, sendo de fácil implementação. A conclusão é que, devido ao baixo custo desses controladores, é altamente recomendável o uso deste tipo de arquitetura para aumentar a confiabilidade de sistemas de automação, supervisão e controle que tem melhores recursos de comunicação.

A conexão dos pontos físicos do controlador Arduíno Mega 2560 é feita através de base de pinos nas laterais da placa. A função de concentrador requer o uso de uma placa de interface Ethernet do tipo “piggy-back” denominada “shield Ethernet”, a qual é instalada sobre a placa do controlador Arduíno Mega 2560. O uso desta placa shield faz com que a conexão de diversos pinos de interface do controlador seja transferida para a base de pinos da placa shield. Esta característica causa dois problemas:

1. O primeiro problema é a redução da confiabilidade, pois é inserida uma conexão adicional nestes pinos de interface, a qual é causada pela instalação da placa shield Ethernet sobre a placa principal do controlador;
2. O segundo problema ocorre caso seja necessária a substituição da placa shield Ethernet, o que resulta em desconexão e conexão de cabos de interface. Esta dificuldade foi verificada na prática.

Assim, para obter a máxima confiabilidade de conexão e facilitar os procedimentos de manutenção, o controlador Arduíno Mega 2560 que tem a função de UTR não usa nenhum tipo de placa piggy-back. Isto é possível também porque o controlador Arduíno Mega 2560 possui mais três interfaces seriais assíncronas além da interface serial principal. Desta forma, a serial principal é usada exclusivamente para carga de programas e outra interface serial é usada para comunicação com o equipamento Concentrador.

Outro fato verificado durante o desenvolvimento deste projeto foi a fragilidade em relação a surtos de tensão das entradas e saídas de sinal do controlador Arduíno Mega. Estes surtos de tensão, causados principalmente por descargas atmosféricas, quando chegam em qualquer pino do controlador, causam sérios danos ao circuito integrado principal, inutilizando o

equipamento. Desta forma, a placa do controlador Arduino Mega 2560 que tem a função de UTR, foi montada em uma placa de circuito impresso que possui elementos de proteção contra surtos em todos os pinos. Cada elemento de proteção é composto por um resistor em série e um diodo zener de 5,1V em paralelo. O resistor limita a corrente e o diodo zener limita a tensão em cada entrada, evitando danos ao controlador. A conclusão é que, em sistemas que recebem sinais conectados à rede elétrica, deve-se sempre montar controladores Arduino usando-se esses elementos de proteção.

7.2 Uso do Computador Raspberry PI 3

Este computador atendeu todas as especificações e está operando sem falhas há 22 meses. O Raspberry PI 3 foi instalado na usina solar junto com os controladores Arduino, o que permite, usando o Ambiente de Desenvolvimento Integrado instalado, fazer a edição, compilação e carga de programas na UTR e no Concentrador. Estes equipamentos são conectados ao computador Raspberry PI 3 através de interface USB, o que permite a carga de programa compilado e monitoramento através da serial. Concluindo, o Raspberry PI 3 foi uma solução excelente, de baixo custo e alta confiabilidade, se mostrando muito adequado a aplicações que demandam disponibilidade 24/7.

7.3 Implementação do Protocolo CoAP

Inicialmente foram consideradas duas opções e realizados alguns testes para definição do protocolo a ser usado dentro do conceito de Internet das Coisas: MQTT e CoAP. Estes testes foram feitos usando um controlador Arduino na função de servidor, recebendo a conexão de um cliente, a requisição e a resposta com as informações. Foi constatado que o protocolo CoAP que opera sobre telegramas UDP tem uma performance significativamente melhor, pois, a implementação do protocolo UDP no Arduino é bem eficiente, usando pouco processamento e tornando a comunicação mais rápida, eficiente e confiável neste controlador especificamente. Aliado a esse fato, o tráfego dos pacotes UDP na rede local também é mais rápido, usando menos banda. Entretanto, em algumas redes, o tráfego de pacotes UDP é limitado. Por isso, o envio das informações para o servidor em nuvem, pelo computador Raspberry PI 3 é feito usando protocolo HTTP carregando a mesma mensagem binária usada pelo protocolo CoAP.

A conclusão é que, além da excelente performance do protocolo CoAP em controladores Arduino, a implementação das mensagens é bem facilitada, pois se trata de mensagens

compactas em formato binário. Outro ponto a favor é que os métodos CoAP e os códigos de resposta são similares aos métodos HTTP, facilitando a integração entre os dois protocolos.

7.4 Implementação dos Recursos de Acesso Web

Os recursos de acesso Web foram desenvolvidos com o objetivo de torná-los compatíveis com todos os dispositivos que podem rodar um navegador, incluindo computadores, tablets e smartphones. Estes recursos são basicamente mensagens em protocolo HTTP, reconhecidas por todos os navegadores e aplicativos que porventura sejam desenvolvidos.

O procedimento de atualização dos valores das variáveis na tabela usando o objeto Javascript XMLHttpRequest é muito mais rápido e eficiente do que a atualização dos valores através da carga do código HTML inteiro, parecendo transparente ao usuário. Isto ocorre principalmente porque a atualização da código HTML requer o tráfego de mais de 30 KBytes em contraste com a mensagem XML de atualização que tem em torno de 2 KBytes.

Concluindo, esta compatibilidade com os navegadores dispensa a aquisição ou o desenvolvimento de aplicativos específicos tais como softwares SCADA, facilitando os procedimentos de monitoramento ao permitir usar qualquer navegador padrão.

7.5 Recursos de Detecção de Falhas Graves

Durante estes dois anos de operação da usina, foram detectadas falhas graves em algumas situações, tais como subtensão, sobrecorrente e sobretemperatura, com o desligamento do equipamento e ativação do indicador corretamente pelo software da UTR. Em nenhuma situação o equipamento foi danificado pela falha.

Concluindo: os recursos de detecção de falhas graves, implementados na UTR, permitiram o rápido desligamento dos equipamentos sob condições de operação fora dos limites, impedindo que estas ocorrências causassem a propagação das falhas. Os equipamentos, neste caso, são os inversores, por serem mais sujeitos a situações de sobrecarga.

7.6 Monitoramento de Condição dos Inversores

O monitoramento de condição dos inversores feito por aplicação de lógica de controle Fuzzy, usando a variável de saída como indicador de saúde do equipamento, permitiu detectar anormalidade na operação de um dos inversores pelo valor mais baixo de saúde apresentado.

Esta informação levou ao procedimento de desmontagem do equipamento para uma inspeção mais detalhada. Esta inspeção identificou que a ponte H do driver, composta por quatro grupos de dois MOSFETs ligados em paralelo, tinha um MOSFET aberto, o que causava aquecimento anormal e queda da eficiência.

Este tipo de defeito é muito difícil de detectar, pois devido a critérios de projeto que objetivam a robustez do equipamento, cada MOSFET é capaz de suportar toda a corrente que circula na ponte H, mantendo a operação. Entretanto, a corrente circulando por apenas um MOSFET em um ponto de alimentação da ponte, gera o dobro da dissipação de calor causando aquecimento anormal no dissipador.

Concluindo: o recurso de monitoramento de condição baseado na aplicação de lógica de controle fuzzy, apesar de ser bem simples, foi capaz de informar corretamente a ocorrência da anomalia. Isto foi feito através de uma diminuição no valor percentual da variável de saída indicadora de saúde do equipamento.

7.7 Monitoramento de Condição dos Painéis Fotovoltaicos

A diminuição da capacidade do conjunto de painéis fotovoltaicos, detectada através da redução do valor percentual da saúde, foi devida ao acúmulo de poeira durante a época de seca, fato verificado em uma inspeção no telhado. Este recurso em usinas off-grid como o estudo de caso deste trabalho, somente tem como fornecer o valor de saúde quando o consumo é igual ou maior que a energia solar gerada. Esta situação ocorre normalmente às 9h da manhã quando a bomba de água é ligada com as demais cargas da casa e a capacidade de geração solar ainda é menor que o consumo.

Concluindo: o recurso de monitoramento de condição dos painéis funcionou corretamente, indicando uma situação real de queda na capacidade de geração fotovoltaica. A correta operação desta funcionalidade depende da manutenção dos pequenos painéis para que estes forneçam o valor correto de potência em relação à irradiância. Para isto, deve ser possível a limpeza periódica destes pequenos painéis, os quais devem ser instalados em local de fácil acesso na mesma direção do conjunto de painéis fotovoltaicos principais.

7.8 Monitoramento de Condição das Baterias

O banco de baterias é mantido sempre em tensão de flutuação, sendo usado somente na falta simultânea de geração solar e da tensão da rede da concessionária. Assim, a condição das

baterias normalmente é ótima. É importante observar que a vida útil das baterias é dada em ciclos de carga e descarga, portanto, procedimentos de descarga de alta corrente, que fornecem um estado de saúde mais preciso diminuem a vida útil. Assim, visando o mínimo desgaste das baterias, foi desenvolvido o procedimento de verificação de saúde usado neste trabalho, o qual é feito quando há necessidade de descarga.

Neste trabalho de pesquisa, com o objetivo de avaliar o procedimento, foi feita uma descarga de dez horas com corrente média de dez amperes na falta de geração solar, fato que resultou em um valor percentual de saúde de 95%. As baterias foram adquiridas em setembro de 2018, ou seja, estão instaladas há 20 meses aproximadamente. Consultando o gráfico do tempo de vida das baterias, usando o valor de profundidade de descarga mínimo de 10%, obtém-se uma vida útil de 2500 ciclos. Usando o valor de 95% de saúde, e considerando que a saúde representa aproximadamente a vida útil com profundidade de descarga mínima, estima-se que foram feitos 5% de ciclos de descarga, ou seja, ocorreram 125 ciclos de descarga nestes 20 meses.

Concluindo: o valor calculado da saúde do banco de baterias pode ser usado para estimar a vida útil de forma aproximada com mínimo desgaste. Outra conclusão obtida deste trabalho foi que, no caso de baterias de chumbo ácido, a energia armazenada custa mais caro que a fornecida pela concessionária. Portanto, à noite, quando não há geração fotovoltaica, as cargas são alimentadas pela rede, mantendo as baterias em flutuação. Desta forma, as baterias são usadas para fornecer energia somente no caso da falta de geração fotovoltaica e da rede simultaneamente.

7.9 Custo da Usina Solar

O custo da usina solar usada como estudo de caso, em setembro de 2018, foi de aproximadamente R\$ 21.000,00, incluindo os equipamentos, material e custos de instalação. O custo do sistema de supervisão, controle e MBC foi de aproximadamente R\$ 4.000,00, o qual não inclui o computador usado para desenvolvimento. A conclusão é que o custo do sistema de supervisão, controle e monitoramento de condição é viável, mesmo para uma pequena usina como esta, com apenas quatro painéis fotovoltaicos de 320Wp.

8 - Sugestões para Trabalhos Futuros

Neste trabalho foi feita a integração ao sistema do equipamento Controlador de Carga EPEVER 3210 em protocolo MODBUS RTU Serial. Futuramente, espera-se fazer a integração de dispositivos de outros fabricantes, incluindo Controladores de Carga, Inversores Off-Grid, Inversores String e Micro Inversores.

Os inversores são os equipamentos mais complexos e mais sujeitos a falhas em uma usina solar. Por isso, é importante aumentar o conjunto de regras fuzzy destinadas ao monitoramento de condição desses equipamentos. Isto deve permitir a identificação mais precisa destas falhas.

A pesquisa de novos tipos de sensores que possam trazer informações úteis aos procedimentos de monitoramento de condição, por exemplo, sensores de distorção da forma de onda e sensores de vibração.

Realizar a integração do servidor em nuvem com um Banco de Dados Histórico. Estas informações podem ser usadas para implementar um sistema de prognóstico de falhas usando recursos adicionais de Inteligência Artificial.

9 - Referências Bibliográficas

ABADIA, M. R. **Desenvolvimento de uma Base de Conhecimento com Regras Fuzzy e Produção Visando Monitoramento Baseado em Condição de uma Usina Hidrelétrica.** Dissertação de Mestrado em Sistemas Mecatrônicos. Departamento de Engenharia Mecânica, Universidade de Brasília, 2015.

AGUENTA, L., IQBAL, T. - **Development of an IoT Based Open Source SCADA System for PV System Monitoring.** 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). 5-8 May 2019. ISBN:978-1-7281-0320-4. IEEE Xplore. 2019.

AGUENTA, L., IQBAL, T. - **Design and implementation of a low-cost, open source IoT-based SCADA System using ESP32 with OLED, ThingsBoard and MQTT protocol.** AIMS Electronics and Electrical Engineering, 4(1): 57–86. 12 December 2019.

BARAKAT, E., SINNOB, N. - **Renewable Energy Controller System.** Elsevier – ScienceDirect Energy Procedia 6 353–360. 2011.

BASUVAIYAN, S., PREETHI, J. - **Internet of Things Towards Smart Solar Grid Systems for Home Appliances.** International Conference on Intelligent Data Communication Technologies and Internet of Things 2019. ICICI 2019: pp 771-778. November 2019.

BEDI, G. et al. - **Review of Internet of Things (IoT) in Electric Power and Energy Systems.** IEEE Internet of Things Journal, Vol. 5, No. 2, April 2018.

BORMANN, C., Ersue, M., and Keranen, A. - **RFC 7228 - Terminology for Constrained-Node Networks.** Internet Engineering Task Force (IETF). ISSN: 2070-1721. 2014.

BUDIMAN et al. - **IoT Technology Monitoring, Controlling and Data Logging for ATS on Grid Connected Solar-Wind Hybrid System.** 2nd Forum in Research, Science, and Technology. IOP Conf. Series: Journal of Physics: Conf. Series 1167 (2019) 012021 doi:10.1088/1742-6596/1167/1/012021. 2019.

EGMICRO CORP. - **EG8010 Datasheets - ASIC for Single-Phase SPWM Control.** EG Microelectronics Corporation 2014.

EGMICRO CORP. - **EGS002 Datasheets - EG8010 Demo Board.** EG Microelectronics Corporation 2014.

- ENERTEC. **Manual Técnico das Baterias Estacionárias Freedom**. ENERTEC do BRASIL Ltda, 2008.
- EPSOLAR. **TRIRON N Series - Modular MPPT Solar Charge Controller User Manual**. Beijing Epsolar Technology Co. Ltd, 2016.
- FIELDING, R. et al. - **Hypertext Transfer Protocol – HTTP/1.1**. Network Working Group. 1999.
- GILLESPIE, A. - **Condition Based Maintenance: Theory, Methodology, & Application**. Conference: Reliability and Maintainability Symposium At: Tarpon Springs, FL. January 2015.
- GOLPIRA, H., BAHRAMARA, S. - **Internet-of-things-based optimal smart city energy management considering shiftable loads and energy storage**. Journal of Cleaner Production 264 (2020) 121620. ScienceDirect, ELSEVIER, 2020.
- GREEN, J., OTTERDAHL, B. - **Resource-constrained Industrial Things – Proposal for the Adaptation of CoAP to EtherNet/IP/TM**. Conference: Presented at the ODVA 2017 Industry Conference & 18th Annual Meeting. February 21-23, 2017. Palm Harbor, Florida, USA. 2017.
- IEC - **International Standard IEC 61131-7. Programmable Controllers – Part 7: Fuzzy Control Programming**. IEC - International Electrotechnical Commission, 2010.
- KEKRE, A., GAWRE, S. - **Solar Photovoltaic Remote Monitoring System Using IOT**. Proceeding International conference on Recent Innovations in Signal Processing and Embedded Systems (RISE -2017) 27-29 October, 2017.
- KRISHNAN, R. et al. - **Fuzzy Logic based Smart Irrigation System using Internet of Things**. Journal of Cleaner Production 252 (2020) 119902. ScienceDirect, ELSEVIER, 2020.
- LEE, I., LEE, K. - **The Internet of Things (IoT): Applications, investments, and challenges for enterprises**. Business Horizons (2015) 58, 431-440. ScienceDirect, ELSEVIER, 2015.
- MAJHI, B. - **Analysis of Single-Phase SPWM Inverter**. A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Technology in Electrical

Engineering. Department of Electrical Engineering. National Institute of Technology. Rourkela. 2012.

MARCHILDON, J. et al. - **SOC and SOH Characterisation of Lead Acid Batteries**. Université du Québec à Trois-Rivières. Institut de recherche sur l'hydrogène. Département de génie électrique et génie informatique, 2015.

MARKVART, T. and CASTAFIER, L. - **Practical Handbook of Photovoltaics**. 2003 Elsevier Ltd.

MACHINERY INFORMATION MANAGEMENT OPEN SYSTEMS ALLIANCE (MIMOSA). - **Specification for Binary OSA-CBM Messages Release V1 (Mapping OSA-CBM into a binary format)**. MIMOSA - Open Standards for Physical Asset Management. 2010.

NEVES, E. P., DUARTE, M. A. Q. - **Sistema baseado em regras fuzzy para avaliação da qualidade da água**. Revista Eletrônica Paulista de Matemática. ISSN 2316-9664 Volume 14, fev. 2019. Edição Ermac. 2019.

OASIS - **MQTT Version 5.0**. Committee Specification 01. 2017.

PATIL, S., et al. - **Solar Energy Monitoring System Using IoT**. Indian Journal of Scientific Research, 2017, p.149 (7). ISSN: 2250-0138 (Online). 2017.

PINHO, J. T. et al. - **Manual de Engenharia para Sistemas Fotovoltaicos**. Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito – CRESESB - Cepel – Eletrobrás 2014.

POOJA, K., RUBYMEENA, A. - **Bidirectional DC-DC Converter for Solar Battery Backup Applications**. International Journal for Modern Trends in Science and Technology. ISSN: 2455-3778 :: Volume: 06, Issue No: 02, February 2020.

RAHMAN, A. ET AL., - **IoT based Hybrid Green Energy Driven Highway Lighting System**. 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech). Date of Conference: 5-8 Aug. 2019. IEEE Xplore. ISBN:978-1-7281-3025-5

ROSS, T. J. - **Fuzzy Logic with Engineering Applications**. John Wiley & Sons, Ltd. ISBN: 978-0-470-74376-8. 2010.

SEITZ, L., Gerdes, S., Selander, G., Mani, M., and Kumar, S. (2016). **RFC 7744 - Use Cases for Authentication and Authorization in Constrained Environments**. Internet Engineering Task Force (IETF). ISSN: 2070-1721. 2016.

SHELBY, Z. et al. - **RFC 7252 - The Constrained Application Protocol (CoAP)**. Internet Engineering Task Force (IETF). ISSN: 2070-1721. 2014.

SOETEDJO, A., et al. - **Web-SCADA for Monitoring and Controlling Hybrid Wind-PV Power System**. TELKOMNIKA (Telecommunication Computing Electronics and Control). June 2014.

TEKLEMARIAM, G. K. et al. - **Facilitating the creation of IoT applications through conditional observations in CoAP**. Article in EURASIP Journal on Wireless Communications and Networking. December 2013.

TELRANDHE, B. et al. - **IoT Based Digital Charge Controller**. International Research Journal of Engineering and Technology (IRJET). Volume: 06 Issue: 04. Apr 2019.

VACHTSEVANOS, G. et. al. - **Intelligent Fault Diagnosis and Prognosis for Engineering Systems**. John Wiley & Sons, Inc. ISBN: 978-0-471-72999-0. 2006.

VENKATRAMANAN D., VINOD J. - **A Reconfigurable Solar Photovoltaic Grid-Tied Inverter Architecture for Enhanced Energy Access in Backup Power Applications**. IEEE Transactions on Industrial Electronics. 2019.

VARGAS-SALGADO, C. et al. - **Low-cost web-based Supervisory Control and Data Acquisition system for a microgrid testbed: A case study in design and implementation for academic and research applications**. Heliyon open access journal. Elsevier 2019.

ZANELLA, A. et al. - **Internet of Things for Smart Cities**. IEEE Internet of Things Journal, Vol. 1, No. 1, February 2014.

Apêndice A - Descrição dos Softwares

Este Apêndice contém a descrição dos cinco softwares desenvolvidos neste projeto. Os softwares da UTR e do Concentrador, que rodam no controlador Arduino Mega 2560, foram desenvolvidos em linguagem C usando o Ambiente Integrado de Desenvolvimento do controlador Arduino² na versão 1.8.7 (Arduino IDE 1.8.7). Os demais softwares, destinados a rodar no computador Raspberry PI 3, foram desenvolvidos em linguagem Java usando o Ambiente Integrado de Desenvolvimento IDE Eclipse³ 2019-3.

Os arquivos fonte de todos os softwares estão disponíveis na pasta pública do armazenamento em nuvem acessível através do link mostrado a seguir:

<https://drive.google.com/drive/folders/1KQVWek3Psq0dbCT8Z69pjvIAhOaVBoWC?usp=sharing>

A.1 Considerações Gerais

O Arduino IDE 1.8.7 permite a edição de somente um arquivo fonte em linguagem C do tipo texto. Após a edição do arquivo fonte, este deve ser compilado e carregado na memória do controlador através de interface serial USB. O IDE Arduino foi instalado no computador Raspberry PI 3. Este computador é instalado na usina, próximo aos controladores Arduino Mega 2560, permitindo a conexão por interface serial USB. O acesso ao computador Raspberry PI 3 é feito usando-se o utilitário VNC (Virtual Network Computing). Este utilitário permite o acesso ao vídeo, mouse e teclado do Raspberry PI 3 por outro computador conectado na rede Ethernet.

O IDE Eclipse 2019-3 foi instalado em um computador Dell Inspiron com processador Intel Core I5, 4GB RAM, 500GB HD e sistema operacional Ubuntu 18.04.5 LTS. Os programas desenvolvidos em linguagem Java deste trabalho estão contidos em dois Projetos Java criados no IDE Eclipse. O Projeto 1 contém os dois programas que rodam no Raspberry PI 3. O Projeto 2 contém o programa que roda no computador virtual em nuvem.

Após a compilação dos módulos em Java, é gerado um arquivo de programa do tipo jar, o qual é carregado na memória do computador Raspberry PI 3 usando-se FTP (File Transfer Protocol) através do utilitário FileZilla. Por fim, o arquivo de programa Java do tipo jar é executado usando-se os recursos do JRE (Java Runtime Environment). O JRE já vem no

2 <https://www.arduino.cc/en/main/software>

3 <https://www.eclipse.org/downloads/packages/release/2019-03>

mesmo pacote do sistema operacional do computador Raspberry PI 3, denominado Raspberry PI OS⁴ (anteriormente chamado de Raspbian).

A.2 Software da Unidade Terminal Remota (UTR)

A.2.1 Requisitos do Software da UTR

Neste trabalho, o requisito mais importante do software que roda na UTR é a execução simultânea⁵ de todas as suas rotinas principais. Este requisito é fundamental, porque a UTR deve executar funções de alta prioridade, como aquisição dos pontos físicos e proteção primária dos inversores. Devido à simplicidade do ambiente de desenvolvimento do Arduíno, que não possui recursos específicos para implementação de funções concorrentes, fez-se necessário a definição de um conjunto de diretrizes que permitam a sua execução de modo independente. A execução das funções de modo independente tem por finalidade simular a execução concorrente.

Visando atender o requisito de execução independente, a execução de uma função não deve atrasar ou impedir a execução de outras funções. Por este motivo, o software da UTR foi desenvolvido levando-se em consideração alguns princípios básicos das técnicas de programação em tempo real. Estes princípios foram incorporados nas diretrizes mostradas a seguir:

1. O arquivo texto único, que contém o programa fonte, foi dividido em três campos principais na ordem a seguir: a) Procedimentos de Inicialização (Setup); b) Programa Principal (loop infinito); c) Rotinas;
2. As Rotinas não podem ter instruções de atraso (delay) nem loops de espera. Todos os procedimentos de temporização devem ser feitos através do incremento ou decremento de contadores, onde cada chamada da Rotina corresponde a um período igual à base de tempo;
3. O Programa Principal deve conter apenas chamadas às Rotinas e terminar com uma instrução de espera (delay(t)), onde t é o tempo de espera em milissegundos. Esta última instrução é usada para gerar o tempo base de espera para todas as Rotinas. Neste software foi usada a base de tempo de cinquenta milissegundos (50ms);

4 <https://www.raspberrypi.org/downloads/raspberry-pi-os/>

5 Na verdade, as instruções e as rotinas são executadas pelo processador do controlador uma após a outra. Entretanto, os tempos de processamento são muito menores que os tempos do processo, significando na prática, que pode-se considerar a execução simultânea das funções.

4. A Rotina de Comunicação Serial deve ler a mensagem completa recebida em protocolo MODBUS RTU. Isto é feito para evitar o uso de procedimentos de espera na rotina de recepção de mensagens.

A Rotina de Comunicação Serial deve carregar a mensagem completa a transmitir em protocolo MODBUS RTU na memória de transmissão. Isto é feito para evitar o uso de procedimentos de espera na rotina de transmissão das mensagens.

Os itens a seguir apresentam a descrição de cada rotina principal e cada rotina secundária do software da UTR.

A.2.2 Rotina ComunicaçãoSerial()

A rotina ComunicaçãoSerial() recebe as mensagens provenientes do Concentrador (dispositivo mestre) em protocolo MODBUS RTU, por interface serial assíncrona. Após a recepção e a verificação de integridade da mensagem através da comparação dos dois bytes CRC16, executa o comando recebido e envia a mensagem de resposta ao Concentrador. A rotina ComunicaçãoSerial() chama as rotinas secundárias MontaEDs() e ExecutaComando().

Cada estado de supervisão digital é armazenado em um byte na memória da UTR. Estados de supervisão digital podem ser pontos lidos de entradas digitais físicas, indicadores⁶ de estado dos equipamentos da usina ou indicadores de estado da própria UTR. Visando compactar a mensagem de resposta da UTR em protocolo MODBUS RTU, a rotina MontaEDs() carrega até oito estados digitais em cada byte dos registros MODBUS correspondentes.

A rotina ExecutaComando() tem por função executar um comando. A mensagem de requisição, enviada pelo Concentrador e recebida pela UTR, sempre tem um comando identificado por um código, que são números de 1 a 17. O código número 1, ao ser interpretado pela rotina, faz com que a UTR envie somente a mensagem de resposta com todos os estados digitais e medidas. Os outros números de identificação de comando causam a execução de uma determinada função.

Algumas mensagens de requisição de comando vem acompanhada de registros contendo informações complementares. Pode-se citar como exemplo, o comando de acerto de relógio, que vem com a informação atualizada de data e hora. Após a execução de qualquer comando,

6 Os Indicadores de Estado Digital binários, com valores zero ou um, são comumente chamados de “Flags” pelos desenvolvedores de software.

o software da UTR sempre envia a mensagem de resposta contendo todos os estados digitais, todas as medidas e o código de comando recebido na mensagem de requisição.

A.2.3 Rotina CarregaMedidas()

Esta rotina faz a leitura e condicionamento de sinal de todos os pontos físicos de supervisão analógica e digital. Cada ponto analógico é lido vinte vezes a cada cinquenta milissegundos e o resultado acumulado. Após os vinte ciclos de leitura, que corresponde a um tempo de atualização de um segundo, é calculada a média para cada ponto. Em seguida é feita a conversão do valor da média de cada medida para a unidade própria de engenharia (Volts, Amperes ou Graus Celsius). O valor já em formato de engenharia é armazenado na variável correspondente e fica disponível para envio ao Concentrador.

A medida da tensão do banco de baterias, que é a mesma do barramento principal de corrente contínua, recebe um tratamento diferente. Nesta medida específica é calculada a média estendida de dois minutos, ou cento e vinte ciclos de aquisição analógica de um segundo. Este procedimento tem por objetivo evitar o chaveamento de cargas CA na operação automática quando ocorrem variações rápidas na tensão CC. Isto ocorre principalmente durante a partida de um motor alimentado pelo Inversor 2, ocorrência que provoca temporariamente alta demanda de corrente do barramento principal de 24Vcc.

A.2.4 Rotina VerificaTensaoRede()

A rotina VerificaTensaoRede() tem por função identificar duas condições referentes à tensão 220Vca proveniente da rede da concessionária: a primeira condição é a ocorrência de falta de tensão; a segunda condição é o retorno da tensão aos níveis normais. A ocorrência de falta é detectada em alguns segundos quando a tensão cai abaixo de um limite mínimo inferior pré estabelecido.

A detecção da ocorrência do retorno da tensão da rede é feita também por comparação com um limite mínimo dentro da faixa de variação normal. Nesse caso, a comparação é feita quarenta vezes, a cada ciclo de aquisição. A tensão CA da rede só é considerada normal quando permanece durante quarenta ciclos, ou quarenta segundos, acima do valor mínimo normal. Este procedimento evita chaveamentos desnecessários das cargas entre a rede e o inversor no caso de instabilidade da tensão da rede. Desta forma, caso ocorra interrupção da rede, e posterior retorno, deseja-se que o software da UTR somente transfira a alimentação das cargas para a rede quando esta permanecer estável.

A.2.5 Rotina VerificaChaves()

Na usina solar, existe uma determinada caixa elétrica plástica que contém circuitos de condicionamento de sinal e fontes de alimentação. Na lateral esquerda desta caixa, estão instaladas quatro chaves de comutação, que permitem que um operador local possa enviar comandos para a UTR. A rotina VerificaChaves() lê o estado de cada chave em cinco entradas digitais, e através de comparações, ativa ou desativa os indicadores correspondentes. A chave inferior Local/Remoto, na posição Local, habilita os comandos na UTR feitos através das chaves.

Esta rotina não atua diretamente ligando ou desligando os inversores e comutando as cargas, apenas ativa ou desativa o indicador de modo de operação, o indicador de modo de controle e os quatro indicadores de habilitação das cargas em corrente alternada. A única rotina que liga e desliga inversores e comuta cargas é a rotina VerificaModoOp().

A.2.6 Rotina VerificaModoOp()

A Rotina Verifica Modo de Operação realiza os procedimentos principais de ativação dos inversores e chaveamento de cargas CA. Esta rotina ativa ou desativa os inversores através das rotinas secundárias LigaInversor1(), LigaInversor2(), DesligaInversor1() e DesligaInversor2(). Os procedimentos são realizados levando-se em consideração um conjunto de condições: 1) Valor da tensão 24 Vcc; 2) Falta de tensão CA da rede da concessionária; 3) Hora do dia e dia do ano.

A condição prioritária é o valor da tensão 24Vcc do barramento principal. No caso da tensão 24Vcc estar abaixo do limite mínimo, os inversores são desligados para preservar as baterias independentemente de outras condições, por ser uma ação de alta prioridade. Foi definido o limite mínimo de 23,0Vcc para o banco de baterias visando a sua máxima durabilidade.

No caso dos procedimentos referentes à falta de tensão CA da rede, existem duas condições principais: período com ou sem geração solar. Em ambas as condições, se está habilitado o modo de operação Normal, as cargas prioritárias 1 e 2 são alimentadas pelo Inversor 2, mesmo se não estejam com seus indicadores habilitados. Durante o período de geração solar, normalmente as cargas 1, 2 e 3 estão conectadas ao Inversor 2.

A última condição refere-se à hora do dia e dia do ano. A rotina faz a leitura do relógio calendário interno da UTR, habilitando ou desabilitando a ativação dos inversores de acordo

com a hora do dia e o dia do ano. O dia do ano é usado para referir-se a uma tabela que contém a hora do nascer e do pôr do Sol. Assim, a rotina habilita a ativação de cada carga CA um certo período depois do nascer do Sol, e desabilita cada carga um determinado período antes de o pôr do Sol. As cargas mais prioritárias são habilitadas mais cedo e desabilitadas mais tarde. Esta condição é a de menor prioridade.

Todos os procedimentos da rotina `VerificaModoOp()` usam três tipos de informação: estados digitais fornecidos por indicadores de estado digital; média estendida da tensão do barramento principal 24 Vcc; hora do dia e dia do ano. As duas últimas são lidas de um circuito integrado relógio de tempo real com bateria própria. As informações de estado digital são: o indicador de Tensão CC, o indicador de Tensão da Rede, o indicador de Modo de Operação, o indicador de Modo de Controle e os quatro indicadores de habilitação das cargas CA.

A.2.7 Rotinas `VerificaInversor1()` e `VerificaInversor2()`

As rotinas `VerificaInversor1()` e `VerificaInversor2()` tem por função realizar os procedimentos de proteção primária dos inversores, sendo idênticas em seus códigos e em sua operação. A primeira tarefa dessas rotinas é fazer a verificação de três indicadores digitais que informam se: o inversor está ligado; a saída do disjuntor da tensão CC de entrada está energizada; a saída do inversor está conectada na chave de transferência correspondente. Se estas verificações iniciais estão corretas, então é confirmado que o inversor está ligado, energizado e conectado. Em seguida, o software inicia os procedimentos de verificação das medições de tensão de saída, temperatura dos drivers, temperatura do transformador e corrente de entrada.

Os procedimentos de verificação das medições operam através da comparação dos valores medidos com variáveis contendo limites pré estabelecidos. Quando um limite é ultrapassado, são ligados dois indicadores: falha geral do inversor e identificador. O procedimento de verificação da tensão de saída do inversor possui duas comparações, limites inferior e superior. Isto é necessário porque a tensão de saída deve permanecer dentro de limites padronizados para a alimentação de corrente alternada. Os demais procedimentos somente possuem comparação com limite superior.

O procedimento de detecção de tensão baixa na saída do inversor possui temporização. Isto foi necessário porque o acionamento de cargas que tem alta corrente de partida, como motores, causam queda de tensão temporária na saída do inversor. Assim, valor da tensão pode ficar abaixo do limite mínimo na saída do inversor por um tempo curto, ocorrência que

poderia causar a detecção de falha. Após a partida normal do motor, a tensão deve voltar rapidamente ao seu valor normal.

A.2.8 Rotinas LigaInversor1() e LigaInversor2()

As rotinas LigaInversor1() e LigaInversor2() são idênticas em sua operação, e tem por função realizar os procedimentos de ativação dos inversores e controle do chaveamento das cargas em corrente alternada. Estes procedimentos incluem as verificações e temporizações. Caso ocorra a habilitação para a ativação de um inversor, é feita a verificação do indicador de falha correspondente. Caso o indicador esteja ativado, a rotina não liga o inversor.

O procedimento de temporização tem por objetivo desligar as cargas em corrente alternada da rede durante dois segundos antes de conectá-las à saída do inversor. Isto evita descontinuidades na onda senoidal de alimentação fornecida para as cargas (ver item A.2.10 Considerações sobre o Chaveamento de Cargas CA).

A.2.9 Rotinas DesligaInversor1() e DesligaInversor2()

As rotinas DesligaInversor1() e DesligaInversor2() são idênticas em sua operação, e tem por função realizar os procedimentos de desativação dos inversores e controle do chaveamento das cargas 220Vca. Estes procedimentos incluem as temporizações e possuem a verificação de somente uma condição: a informação indicando que inversor está ligado fornecida por indicador interno.

O procedimento de temporização tem por objetivo desligar as cargas em corrente alternada do inversor durante dois segundos antes de conectá-las à rede. Isto evita descontinuidade na onda senoidal de alimentação fornecida para as cargas.

As rotinas destinadas a desligar os inversores são procedimentos de segurança de alta prioridade, pois, na ocorrência de falha grave em um inversor, este deve ser imediatamente desligado. Por esse motivo, não são feitas verificações de condições lidas por entradas digitais, apenas é lido o indicador de inversor ligado armazenado internamente.

A.2.10 Considerações sobre o Chaveamento de Cargas CA

O procedimento de temporização desliga as cargas CA da rede durante dois segundos antes de conectá-las na saída do inversor, o que evita descontinuidade rápidas na onda senoidal de 220V 60Hz. Estas descontinuidades podem causar danos aos equipamentos alimentados. Este

procedimento foi necessário porque, neste trabalho, ainda não foi implementado o recurso de sincronismo para conexão.

O recurso de sincronismo tem por objetivo esperar até que os sinais da rede e da saída do inversor estejam em fase e fazer a transferência. Neste caso, a transferência das cargas deve ser feita em poucos milissegundos. Isto deve ser feito para evitar que uma possível queda rápida de tensão possa afetar o funcionamento de algum equipamento conectado. Quando a transferência de carga CA é feita deste modo, torna-se imperceptível para os usuários.

A.2.11 Rotina VerifCxAzBomba()

Esta rotina verifica se a bomba d'água instalada no poço profundo foi ligada. Caso a bomba esteja energizada, zera e incrementa um contador de segundos que mostra o tempo em que a bomba permaneceu ligada até a caixa d'água encher. Esta informação de tempo é útil para detecção de anomalias no conjunto da bomba e da caixa d'água, tais como consumo excessivo, problemas na bomba e nível da água do poço.

A.2.12 Rotina VerifEstCxAz()

Na caixa d'água principal existem duas boias elétricas de nível: a superior que serve para controlar a ativação e desativação da bomba em operação normal; a inferior, destinada a indicar nível baixo de água na caixa. Os sinais de saída das duas boias, após serem condicionados, são aplicados em duas entradas digitais do Arduíno. Assim, esta rotina determina o estado da caixa d'água principal através da leitura dessas duas entradas digitais. O estado da caixa é fornecido por um indicador digital de seis estados (números de zero a cinco).

Desta forma, foram definidos três estados principais para a caixa d'água: Cheia; Precisa Encher Nível Normal; Precisa Encher Nível Baixo. Existem também mais três estados de falha: Indefinido; Falha de Sinalização 1; Falha de Sinalização 2. Os estados de falha tem por finalidade tornar mais seguro o procedimento de geração da informação de estado, ao mostrar a ocorrência de inconsistências nas informações fornecidas pelas entradas digitais. Estas inconsistências podem ser causadas principalmente por falhas nos contatos das boias, cabos abertos ou em curto circuito.

A.3 Software do Concentrador

O Concentrador é um equipamento, baseado no controlador Arduíno Mega 2560, que opera entre o Nível de Rede e o Nível de Processo. No Nível de Rede, dois softwares instalados no computador Raspberry PI 3 fazem requisições ao Concentrador em protocolo CoAP pela rede Ethernet. Esses softwares são: o Servidor HTTP Local e o Cliente HTTP para Atualização do Servidor em Nuvem. No Nível de Processo, existem três equipamentos que usam protocolo MODBUS RTU em modo Escravo, em interface serial assíncrona: a UTR, o Controlador de Carga 1 e o Controlador de Carga 2. Este software possui somente duas rotinas principais: `ComunicacaoEthernet()` e `ComunicacaoSerial()`. O programa principal possui somente chamada à rotina `ComunicacaoEthernet()`, pois esta chama a rotina `ComunicacaoSerial()` sempre que recebe uma mensagem de requisição em protocolo CoAP do Concentrador pela rede Ethernet.

A.3.1 Rotina `ComunicacaoEthernet()`

Esta rotina espera a recepção de uma mensagem de requisição em protocolo CoAP pela rede Ethernet. Caso ocorra a recepção de uma mensagem, é feita a verificação de sua integridade e a leitura do comando e as informações correspondentes, caso existam. Em seguida, esta rotina chama a rotina `ComunicacaoSerial()` que deve retornar com todos os estados e medidas dos três equipamentos do Processo. Por fim, é montada e enviada a mensagem de resposta em protocolo CoAP pela rotina `MontaMsgEstados()`. Assim, a leitura das informações de supervisão dos equipamentos do processo e o envio de comando para a UTR, caso exista, é feito somente se ocorrer uma requisição ao Concentrador em protocolo CoAP pela rede Ethernet.

A.3.2 Rotina `ComunicacaoSerial()`

Esta rotina realiza a Comunicação com os Equipamentos do Nível de Processo em três passos básicos realizados sequencialmente para cada equipamento no Nível de Processo. Estes passos são mostrados a seguir:

Passo 1: Envia a mensagem de solicitação para o equipamento, em protocolo MODBUS RTU, em modo Mestre;

Passo 2: Após o envio de cada mensagem de solicitação, espera a mensagem de resposta do equipamento chamado por um tempo máximo de um segundo;

Passo 3: Ao receber a mensagem de resposta, verificar a sua integridade através da comparação dos dois bytes CRC16. Caso a mensagem tenha sido recebida corretamente, primeiro descompacta as informações digitais através da rotina DesmontaEDs(). Em seguida armazena as informações recebidas nas variáveis correspondentes. Caso não seja recebida a mensagem de resposta no tempo de espera ou a verificação de integridade indicar erro, ligar o indicador de falha de comunicação do equipamento correspondente. Não foi implementado procedimento de re tentativa de comunicação. No caso do equipamento chamado não responder no tempo de um segundo, o programa chama o próximo equipamento. Isto evita atrasos desnecessários nos procedimentos de atualização de variáveis e envio de comandos no caso de falha em um dispositivo.

A.4 Considerações sobre os Softwares Java

Foram criados dois Projetos Java porque o software servidor HTTP a rodar na nuvem usa um conjunto bem menor de bibliotecas que os softwares que rodam no computador Raspberry PI 3. Assim, visando reduzir o tamanho do programa compilado e reduzir o tempo de carga no computador em nuvem, este o programa foi colocado em outro projeto.

A.5 Softwares Servidores HTTP (Local e na Nuvem)

A classe principal do programa Servidor HTTP Local (HTTPServer01), e também a classe principal do programa Servidor em Nuvem (HTTPCloudSrv01) possuem a interface “Runnable”. Esta característica permite a criação de Threads, que correspondem a determinados trechos de código que podem ser acionados várias vezes e rodar de modo simultâneo. Isto é necessário para que os softwares Servidores HTTP possam atender a várias solicitações de conexões de clientes HTTP ao mesmo tempo.

Dentro da classe do tipo “Runnable”, existe um método do tipo “run” onde está o código a ser executado de modo simultâneo. Assim, este código é executado usando a classe Thread, cada vez que uma conexão é aberta com um cliente HTTP. Quando a conexão com o cliente é terminada, a Thread correspondente é interrompida, ou seja, a rotina que trata de atender às solicitações específicas daquele cliente HTTP pára a sua execução.

Os procedimentos de atendimento aos clientes realizados pelos dois softwares servidores HTTP, local e na nuvem, são basicamente idênticos. O software permanece em modo de espera até receber uma requisição de um cliente. Ao receber a requisição do cliente com URI

vazia, os dois softwares servidores fornecem uma página HTML lida de um arquivo texto que contém os comandos HTML e o programa Javascript. A requisição URI = “atualiza” gera o envio da resposta HTTP em formato XML com os valores de todas as variáveis de supervisão e controle.

Comparando os dois softwares, a principal diferença está nos procedimentos de atualização da base de dados que contém os valores atualizados das variáveis de supervisão e controle. A outra diferença é na tabela de apresentação, onde aparece o indicador de Comunicação com o Software Cliente de Atualização do Servidor em Nuvem. Este indicador é ativado por um procedimento de espera pela mensagem de atualização. Se o servidor em nuvem recebe uma mensagem de atualização, este indicador é colocado na posição Normal. Se a mensagem de atualização das variáveis deixar de ser recebida por mais de oito segundos, este indicador é colocado em modo de falha de comunicação.

A atualização da base de dados do software servidor HTTP local é realizada através da comunicação direta com o equipamento Concentrador. Esta comunicação é feita cada vez que um cliente HTTP local solicita a atualização das variáveis de supervisão e controle. Ao receber a mensagem CoAP em formato binário, o software efetua a conversão dos valores e carrega em variáveis no formato long (32 bits). A mensagem CoAP contém os valores das variáveis em registros de 16 bits e informações de estado digital em bytes, onde há um byte para cada estado. No caso do Servidor HTTP instalado no computador em nuvem, a atualização das variáveis da base de dados é feita por iniciativa do software Cliente HTTP de Atualização do Servidor em Nuvem.

A.6 Software Cliente de Atualização do Servidor em Nuvem

O Software Cliente de Atualização do Servidor em Nuvem opera em um loop infinito, efetuando periodicamente, a cada quatro segundos, as seguintes funções em sequência:

1. Efetua a comunicação com o Concentrador. Envia a mensagem de requisição em protocolo CoAP e espera a resposta. Caso a mensagem de resposta CoAP esteja correta, armazena o payload da mensagem em formato binário;
2. Solicita conexão TCP/IP com o servidor HTTP em nuvem. Caso seja bem-sucedido, envia a mensagem de atualização para este servidor. Isto é feito usando uma mensagem HTTP do tipo “Octet-stream” com o payload recebido na mensagem CoAP do Concentrador. Em seguida fecha a conexão TCP/IP.

A mensagem de atualização é uma mensagem HTTP com cabeçalho padrão e o campo Payload da mensagem binária CoAP carregado na área de dados. Assim, o software Cliente de Atualização do Servidor em Nuvem não armazena informações de supervisão e controle, ficando a cargo dos servidores HTTP efetuarem o armazenamento e a conversão das informações. Isto tem por objetivo padronizar ambos os softwares servidores HTTP, local e na nuvem.

A.7 Programa Javascript

O programa Javascript é enviado pelos servidores HTTP para os navegadores ao final do mesmo arquivo texto que contém os comandos HTML. Este programa possui duas funções principais:

1. Atualizar automaticamente e periodicamente os valores das variáveis de estado e das medidas na tela do navegador da maneira mais rápida possível;
2. Armazenar um comando solicitado pela tela do navegador através de um botão e enviá-lo para o servidor HTTP.

A.7.1 Rotina AutoRefresh()

Esta rotina tem por finalidade iniciar o procedimento de chamada periódica da rotina principal loadXMLDoc(). Para isto é usada a função setInterval passando como parâmetros a rotina principal e o tempo do intervalo entre as chamadas (três segundos).

A.7.2 Rotina loadXMLDoc(recurso)

Esta rotina possui somente um parâmetro ou variável de entrada formada por uma caracteres (“string”) chamado recurso, o qual é usado como URI da mensagem de requisição ao servidor HTTP. A rotina, primeiramente, verifica o conteúdo da variável recurso. Caso seja uma string vazia, é usada a URI = “local001.xml” na mensagem de requisição. Este valor da URI significa que trata-se somente de uma mensagem de requisição de atualização de valores. Outra razão para que a rotina loadXMLDoc() seja chamada sem parâmetros é devido à função setInterval, que não permite o seu uso.

Caso o valor da variável recurso não seja vazio, a rotina loadXMLDoc(recurso) envia o conteúdo como URI na mensagem de requisição HTTP. A rotina loadXMLDoc(recurso) é chamada também cada vez que um botão de comando na tela do navegador é pressionado.

Quando isso ocorre, é carregado na variável recurso o valor do código de comando correspondente. Este código obedece ao formato cmd=XXXX, onde XXXX é um número de comando iniciando em 0001. Foram implementados dezessete tipos de comando, identificados com as URIs “cmd=0001” até “cmd=0017”.

Assim, a rotina loadXMLDoc(recurso) envia a requisição para o software servidor HTTP e espera a resposta. O software servidor HTTP, ao receber esta requisição, envia como resposta para o navegador uma mensagem do tipo XML contendo os valores atualizados de todas as variáveis de supervisão e controle. O software servidor HTTP local, quando recebe um comando do navegador, o repassa diretamente para o Concentrador. O software servidor HTTP em nuvem, quando recebe um comando do navegador, o envia na mensagem de resposta para o software cliente HTTP de atualização.

A rotina loadXMLDoc(recurso) usa a função getElementByTagName(“etiqueta”) para localizar os campos da mensagem XML. As funções childNodes e nodeValue são usadas para obtenção do valor de cada variável. Após o valor de uma variável ser lido do arquivo XML e carregado em uma variável do tipo string, a rotina localiza no arquivo HTML o local onde este valor deve ser carregado. Em cada local do arquivo HTML destinado a apresentar uma variável de supervisão existe um localizador. Este localizador é encontrado usando-se a função getElementById(“localizador”).innerHTML. Assim, é lido o valor de cada variável do arquivo XML e carregado em sua posição na tabela montada pelo arquivo HTML. Esta tabela é apresentada na tela do navegador para visualização pelo usuário.

Anexo A - Arquitetura OSA-CBM

A arquitetura OSA-CBM define uma estrutura baseada em seis blocos funcionais. Cada bloco é descrito bem como a interface entre esses blocos, especificando as entradas e saídas, o que facilita a integração de diversos componentes (MIMOSA, 2010). Em resumo, o padrão OSA-CBM descreve um sistema padronizado de troca de informações para os procedimentos de MBC, compondo um método sistemático para tratar os sinais dos sensores, efetuar o processamento desses sinais para obter as informações, gerar o relatório de integridade do sistema e as orientações para manutenção. O padrão OSA-CBM foi definido usando a UML (Unified Modeling Language) em um formato próprio.

Este padrão pode ser implementado de acordo com as necessidades dos sistemas, tais como uso de protocolos de alta velocidade em formato binário para uso em tempo real, ou ferramentas da internet. A especificação das mensagens OSA-CBM que mapeiam a UML abstrata podem ser montadas na forma de uma mensagem binária, a qual é bem mais eficiente e compacta que uma mensagem do tipo XML. A eficiência na composição da mensagem binária OSA-CBM encapsulada em protocolo UDP torna este conjunto possível de ser usado eficientemente por pequenos controladores como o Arduíno.

A.1 Blocos Funcionais da Arquitetura OSA-CBM

Os blocos funcionais da arquitetura OSA-CBM são mostrados na Figura A.1 e a descrição de cada um está a seguir:

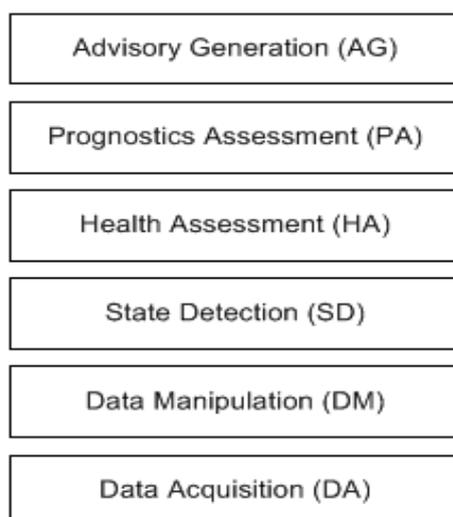


Figura A.1 - Blocos Funcionais do Padrão OSA-CBM

A.1.1 Aquisição de Dados (Data Acquisition - DA)

O procedimento de aquisição de dados tem por objetivo disponibilizar para o controlador responsável pelo processamento, as informações relevantes sobre o processo a ser monitorado. O componente principal da aquisição de dados é o sensor, através do qual uma grandeza de interesse é convertida em um sinal elétrico. As grandezas de interesse usadas nesse trabalho são: temperatura, tensão e corrente. Monitorar cada tipo requer um sensor específico para a função. Os sensores podem ser chamados também de transdutores.

O segundo passo da aquisição de dados é o condicionamento, já que muitas vezes o sinal elétrico de saída do sensor não é compatível com a entrada do controlador. Um exemplo típico de controlador usado no meio acadêmico é o Arduíno, que possui entradas analógicas e digitais que devem receber sinais em tensão na faixa de zero a cinco volts positivos. Assim, os sinais de todos os sensores devem ser condicionados para este padrão.

Atualmente, diversos equipamentos disponibilizam as informações de medição e estado através de interface de comunicação serial ou ethernet através de protocolo próprio. Assim, o procedimento de aquisição de sinal é feito, nesses equipamentos, diretamente através da comunicação com os mesmos.

A.1.2 Processamento de Sinal (Data Manipulation - DM)

Normalmente é preciso efetuar um procedimento de software para adequar a informação fornecida pelos sensores. Um exemplo simples de processamento de sinal é o cálculo da média das medidas provenientes dos sensores analógicos. Este procedimento estabiliza os valores, evitando instabilidade nos procedimentos de decisão.

Além da média, os valores devem ter a escala ajustada e o valor fixo, chamado de offset, deve ser subtraído. Ao final do processamento do sinal, a informação proveniente de aquisição é armazenada em formato de engenharia, ou seja, no valor e na unidade usada nos procedimentos de monitoramento de condição.

A.1.3 Monitoração de Condição (State Detection - SD)

A camada de Monitoramento de Condição ou Detecção de Estado recebe as informações fornecidas pela camada de processamento de sinal em conjunto com parâmetros de referência do sistema monitorado. A função principal desta camada é efetuar a comparação das

informações fornecidas pela camada de processamento de sinal com valores ou condições referentes às especificações do equipamento.

Os valores das faixas operacionais que determinam uma condição de funcionamento podem ser estáticos ou dinâmicos, dependendo da necessidade de monitoramento dos equipamentos. Esta camada deve gerar alerta baseado em faixas operacionais estabelecidas (ABADIA, 2015). A monitoração de condição pode ser realizada em dois níveis dependendo da gravidade da anomalia. No caso de equipamentos de potência, normalmente se define um limite máximo para os parâmetros operacionais mais importantes como tensão, corrente e temperatura, os quais são monitorados em dois níveis: grave e não grave. O nível grave requer ação imediata para evitar danos ao equipamento, sendo feito usando lógica simples pelo próprio controlador. As falhas graves são sobrecorrente, sobretensão e sobretemperatura acima de limites máximos. As informações referentes ao procedimento de monitoramento de condição, são ajustadas e enviadas para a camada de diagnóstico.

A.1.4 Diagnóstico (Health Assessment - HA)

A camada de diagnóstico tem por função principal receber as informações provenientes da camada de monitoramento da condição, efetuar a avaliação da condição de operação do equipamento, e gerar diagnósticos caso sejam detectadas anomalias no funcionamento. O diagnóstico de um equipamento pode ser realizado através de três métodos que são: 1) Regras de produção; 2) Casos; 3) Modelos, podendo gerar registros baseados em tendências de históricos de diagnóstico, estado operacional e histórico de ações de manutenção.

A.1.5 Prognósticos (Prognostics Assessment - PA)

Esta camada, como seu nome diz, pode prever a condição futura de um equipamento ou sistema, prognosticando o tempo que pode fazer suas funções dentro dos limites normais de operação antes de realizar uma ação de manutenção prévia a uma falha.

A.1.6 Tomada de Decisão (Advisory Generation -AG)

A camada tem como função integrar as informações recebidas dos módulos de diagnóstico e prognóstico, com o fim de entregar a melhor recomendação para a realização de uma ação de manutenção em uma eventualidade ocorrida no equipamento ou sistema. Assim, deve ser restabelecido o funcionamento ou realizada uma intervenção que permita completar a sequência atual em seu processo de produção sem ocorrência de falha.

Em sistemas mais simples, a tomada de decisão realizada pelo usuário ou operador do sistema tem por objetivo evitar maiores danos, desligar e substituir um equipamento que esteja com o seu nível de saúde baixo ou realizar algum procedimento de manutenção que seja necessário. A tomada de decisão pelo usuário é feita através da consulta às informações fornecidas pelas camadas de diagnóstico e prognóstico.

A.1.7 Apresentação

Esta camada não está representada no diagrama fornecido pela documentação, mas é igualmente importante porque representa a apresentação das informações geradas pelas camadas anteriores para o operador. As informações geradas para auxílio à manutenção destinam-se ao operador, o qual deve tomar a decisão mais acertada para a situação. Este operador pode ser leigo ou técnico, mas a informação gerada deve atender a ambos. A apresentação é feita normalmente por sinalizadores e principalmente por um software específico denominado Software Supervisório. Este software apresenta as informações do sistema como estados e medidas e permite o envio de comandos. O exemplo mais simples é um indicador de saúde de um equipamento, obtido através do processamento de diversos parâmetros, o qual é apresentado na forma de gráfico. Esta informação é clara para qualquer tipo de usuário.

Anexo B – Hardware de Processamento

Neste Anexo são apresentados os dois principais equipamentos de processamento usados neste trabalho. Ambos foram escolhidos por terem as seguintes características principais: baixo custo, fácil aquisição, tamanho reduzido, ótima performance, ótima confiabilidade, ampla aceitação no meio acadêmico, baixo consumo e tamanho reduzido.

B.1 Características do Computador Raspberry PI 3

A escolha do computador que tem a função principal de servidor HTTP, rodar os algoritmos de monitoramento de condição usando lógica de controle Fuzzy, e atualizar as variáveis do servidor em nuvem, considerou os seguintes fatores principais:

1. Baixo custo;
2. Alta confiabilidade;
3. Baixo consumo;
4. Alimentação única em corrente contínua;
5. Processador Quad Core;
6. Memória RAM mínima de 1GB;
7. Memória Flash mínima de 16GB;
8. Interface Ethernet 10/100BaseT;
9. Disponibilidade do acesso por VNC (Virtual Network Computing);
10. Sistema operacional semelhante ao Linux;
11. Capacidade de rodar o JRE (Java Runtime Environment);
12. Capacidade de rodar o Arduíno IDE (Integrated Development Environment);
13. Pelo menos uma interface USB 2.0 para carga de programa no Arduíno.

O computador disponível no mercado que atende a essas características, na época da implantação no fim de 2018, era o Raspberry PI 3. O Raspberry PI 3 é um computador completo, com memória RAM, interface de vídeo HDMI, processador Quad-Core, interface USB e Slot para memória não volátil flash do tipo Micro SD. O Raspberry PI 3 normalmente usa o sistema operacional Raspbian, que é um tipo de Linux e vem acompanhado de aplicativos de uso geral tais como navegador internet (Google Chromium) e pacote LibreOffice.

Assim, se forem conectados monitor, teclado e mouse, pode ser usado como um computador de uso geral para acesso à internet, edição de textos e planilhas, elaboração de apresentações e desenvolvimento de software. Outra aplicação comum é a de servidor embutido. Por causa disso o sistema operacional Raspbian já vem com o aplicativo VNC para acesso remoto pela rede Ethernet. O VNC (Virtual Network Computing) é um software que permite que um computador acesse remotamente outro computador através da rede. É um recurso usado para acessar servidores, que normalmente não tem monitor, mouse e teclado instalados. A Figura B.1⁷ mostra o computador Raspberry PI 3:



Figura B.1 - Computador Raspberry PI 3

A placa do computador Raspberry PI 3 foi montada em uma caixa elétrica com uma fonte de alimentação conversora de 24Vcc para 5Vcc e instalada na usina solar. Foi instalado um módulo de memória micro SD SanDisk de 64GB. Este computador é alimentado diretamente pelo serviço de corrente contínua da usina solar, proporcionando a máxima segurança contra interrupções no funcionamento.

B.2 Características do Controlador Arduino Mega 2560

A escolha do controlador que deve ter as funções de Unidade Terminal Remota e Concentrador de comunicações deve considerar as seguintes características:

1. Baixo custo, Alta confiabilidade e Baixo consumo;
2. Alimentação única em corrente contínua;
3. Mínimo de 16 Entradas Analógicas;
4. Conversor Analógico/Digital com resolução de pelo menos 10 bits;
5. Mínimo de 48 Entradas/Saídas Digitais configuráveis individualmente;
6. Processador RISC de alta performance;

⁷ <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

7. Memória FLASH mínima de 128KBytes e Memória RAM mínima de 4KBytes;
8. Disponibilidade de Relógio de Tempo Real (RTC) com bateria própria;
9. Pelo menos três interfaces seriais assíncronas de uso geral;
10. Opção de Interface Ethernet 10/100BaseT;
11. Programação em Linguagem de Alto Nível C/C++;
12. Ambiente de Desenvolvimento Integrado para Desenvolvimento de Software.

Atualmente, a disponibilidade de controladores e computadores de baixo custo e alto desempenho, com seus recursos de comunicação e conectividade, permitem integrar os mais diversos dispositivos e sistemas às redes locais e à Internet. O Arduino é um controlador com processador de oito bits que possui capacidade de aquisição de dados analógicos e digitais, contando também com interface de comunicação serial e interface Ethernet. A programação do Arduino é feita em linguagem C/C++ usando o Ambiente Integrado de Desenvolvimento de Software fornecido gratuitamente (Arduino IDE). O Arduino IDE pode ser instalado em qualquer computador com sistema operacional Linux ou Windows.

Devido à facilidade de aquisição de componentes eletrônicos e instrumentos de medida pela internet, é possível projetar, montar, testar e calibrar circuitos sensores e condicionadores com precisão e confiabilidade razoáveis para a maioria das aplicações. Assim, usando o Arduino como controlador e montando os circuitos sensores, condicionadores e fontes de alimentação, é possível compor um sistema de supervisão, controle e automação com custo realmente baixo. A Figura B.2⁸ mostra o controlador Arduino Mega 2560. A Figura B.3⁹ mostra o Arduino Mega 2560 com placa “shield” Ethernet 10/100BaseT instalada.

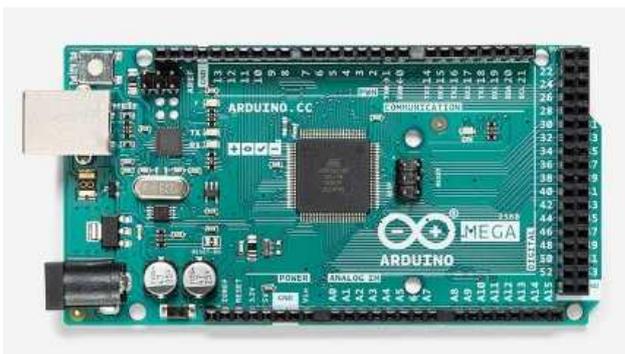


Figura B.2 - Controlador Arduino Mega 2560



Figura B.3 - Controlador Arduino Mega 2560 com Shield Ethernet instalado

8 <https://store.arduino.cc/usa/mega-2560-r3>

9 <http://www.lillyelectronics.com/ethernet-shield-w5100-for-arduino-uno-r3-mega-2560-1280-a057>

Anexo C - Protocolos de Comunicação

Neste Anexo são apresentadas as características dos principais protocolos de comunicação usados neste trabalho. O protocolo usado na comunicação do servidor Raspberry PI 3 com o controlador Arduíno Mega 2560 é o CoAP. Por ser um protocolo novo, e também por ser um dos principais protocolos usados no conceito de Internet das Coisas, o protocolo CoAP foi descrito em detalhes. O segundo protocolo principal é o HTTP, usado na comunicação do computador Raspberry PI 3 com os clientes, foi descrito de forma genérica, por fazer parte há vários anos das comunicações pela Internet.

C.1 Características Detalhadas do Protocolo CoAP

O protocolo CoAP, ou Constrained Applications Protocol – Protocolo para Aplicações Restritas, é baseado na troca de mensagens compactas que, por padrão, são encapsuladas em pacotes UDP, ou seja, cada mensagem CoAP ocupa a seção de dados de um datagrama. Caso seja necessário mais segurança no tráfego das informações, o protocolo CoAP pode ser usado sobre a Camada de Segurança do Datagrama de Transporte (Datagram Transport Layer Security - DTLS). O protocolo CoAP também pode ser usado em conjunto com outros protocolos de transporte, como SMS, TCP ou SCTP. A mensagem do protocolo CoAP é do tipo binário, composta por um cabeçalho de tamanho fixo de 4 bytes seguido por três campos:

1. Token;
2. Opções;
3. Payload.

O uso do protocolo UDP para carregar as mensagens, requer a implementação de procedimentos para tornar confiável a entrega dos pacotes. Assim, o tipo de mensagem Confirmável com os procedimentos de verificação e re-tentativas de transmissão no caso da detecção de erros, proporciona a confiabilidade na entrega das mensagens corretamente. O campo Token é opcional, e tem por finalidade identificar cada par de mensagens de solicitação de informações e a resposta correspondente. Neste trabalho, o campo Payload carrega as mensagens binárias OSA-CBM com os dados de supervisão e controle, compondo o novo protocolo denominado CoAP-OSA-CBM. A Tabela C.1 mostra os quatro campos de uma mensagem do protocolo CoAP. Apenas o campo Cabeçalho é obrigatório em todas as mensagens:

Tabela C.1 - Formato das Mensagens do Protocolo CoAP

Tamanho	Campo
4 bytes	Cabeçalho (Header)
Até 8 bytes	Token – Sincroniza chamadas e respostas
Até 12 bytes	Opções -
Até 65000 bytes	Payload – Carrega as Mensagens Binárias OSA-CBM

O protocolo CoAP define quatro tipos de mensagens em relação aos procedimentos de solicitação e resposta: Confirmável, Não-Confirmável, Reconhecimento, Reset. Os códigos referentes aos tipos de mensagens podem fazê-las carregar solicitações ou respostas. Uma mensagem de reconhecimento, enviada em resposta a uma solicitação feita por uma mensagem confirmável, pode carregar os dados solicitados para agilizar os procedimentos de comunicação. Desta forma, o protocolo CoAP tem por finalidade permitir a troca de informações da maneira mais eficiente possível com pequenos controladores e integrá-los aos recursos da Internet. A tabela C.2 mostra as camadas do protocolo CoAP:

Tabela C.2 - Camadas do Protocolo CoAP

Informações e Procedimentos da Aplicação
Procedimentos de Solicitações e Respostas
Composição das Mensagens CoAP
Encapsulamento das Mensagens CoAP no Protocolo UDP

C.1.2 Uso do Protocolo UDP para carregar as mensagens CoAP

O protocolo UDP (User Datagram Protocol) é um tipo de protocolo simples não orientado à conexão que possui alta performance em velocidade ao carregar as mensagens da aplicação. Na verdade, o protocolo UDP é uma pequena camada de transporte projetada sobre IP, com o mínimo de sobrecarga por informações que não pertencem ao conteúdo da aplicação. O protocolo UDP define apenas o cabeçalho da mensagem e o tamanho máximo do campo de dados do pacote. Assim, fica por conta do usuário inserir o conteúdo de acordo com a sua aplicação e controlar os procedimentos de verificação da integridade dos dados e solicitação de retransmissão no caso de detecção de erros. O uso de UDP é ideal para comunicação local com os pequenos controladores, pois, exige processamento significativamente menor que as conexões TCP/IP.

C.1.3 Protocolo UDP – User Datagram Protocol

O protocolo UDP é um tipo de protocolo simples não orientado à conexão que possui alta performance no requisito velocidade ao carregar as mensagens da aplicação. Na verdade, o

protocolo UDP é uma pequena camada de transporte projetada sobre IP, com o mínimo de sobrecarga por informações que não pertencem ao conteúdo da aplicação. A palavra “carregar” descreve literalmente o procedimento, porque este protocolo define apenas o cabeçalho da mensagem e o tamanho máximo, ficando por conta do usuário inserir o conteúdo de acordo com a sua aplicação e controlar os procedimentos de transmissão e recepção. Devido às suas características, o protocolo UDP é muito usado em aplicações de “stream” de vídeo e áudio, onde a velocidade é muito mais importante que a integridade dos dados. Ao usar o protocolo UDP, o usuário deve considerar as seguintes características:

1. O procedimento de transmissão e recepção não possui procedimentos definidos para verificação de erros, retransmissão ou tempo limite. Assim, caso sejam necessários tais procedimentos, estes devem ser implementados na camada de aplicação. Portanto, as mensagens UDP, chamadas Datagramas, são enviados individualmente e verificados quanto à integridade somente se chegarem ao receptor;
2. Da mesma forma, não são definidos procedimentos de ordenamento das mensagens UDP caso estas componham uma mensagem maior. Assim, caso a mensagem final seja composta por várias mensagens UDP, o ordenamento deve ser feito na camada de aplicação;
3. O protocolo UDP não possui controle de congestionamento, ou seja, quando são enviadas mais mensagens que podem ser tratadas pelo receptor. As medidas de controle de congestionamento devem ser implementadas na camada de aplicação;
4. O UDP, por não ser orientado à conexão, pode enviar mensagens do tipo “Broadcast”, ou seja, os datagramas podem ter seu endereço configurado para serem recebidos por todos os dispositivos na sub-rede.

O cabeçalho de um Datagrama do protocolo UDP é apresentado na Figura C.1¹¹:

		IPv4 Pseudo Header Format																															
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source IPv4 Address																															
4	32	Destination IPv4 Address																															
8	64	Zeroes								Protocol								UDP Length															
12	96	Source Port																Destination Port															
16	128	Length																Checksum															
20	160+	Data																															

Figura C.1 - Cabeçalho da Mensagem do Protocolo UDP

¹¹ https://en.wikipedia.org/wiki/User_Datagram_Protocol

O cabeçalho possui apenas vinte bytes, onde os doze primeiros contém o cabeçalho IP e os oito bytes restantes contém o cabeçalho UDP. As descrições dos campos do cabeçalho são mostradas a seguir:

1. **Endereço IP:** endereço IP de origem e endereço IP de destino da mensagem (IPv4);
2. **Source Port** (Número de Porta de Origem): número que identifica a aplicação (processo) que envia os dados;
3. **Destination Port** (Número de Porta de Destino): número que identifica a aplicação (processo) que deve receber os dados;
4. **Length** (Comprimento): Este campo contém o número de bytes do cabeçalho UDP mais número de bytes da área de dados. O cabeçalho UDP possui oito bytes, portanto, o número mínimo de bytes é oito. Um Datagrama UDP usando endereçamento IPv4 pode ter até 65.507 bytes (65.535 – 8 bytes do cabeçalho UDP – 20 bytes do cabeçalho IP). Usando endereçamento IPv6 é possível datagramas com mais de 65.535 bytes;
5. **Checksum:** Este campo pode conter o resultado de uma operação sobre os bytes do Datagrama com o objetivo de verificação de erros pelo receptor. O uso deste campo é opcional quando se usa o endereçamento IPv4, e obrigatório quando se usa endereçamento IPv6. Caso não seja usado deve ser preenchido com zeros.

A implementação do protocolo CoAP baseia-se no documento RFC7252 publicado pela Internet Engineering Task Force (IETF) em junho de 2014. Este protocolo define quatro tipos de mensagens: Confirmável, Não Confirmável, Reconhecimento e Reset; o tipo de mensagem, se é de solicitação ou de resposta é definido através de códigos de método e códigos de resposta. As solicitações de informações podem ser feitas usando-se mensagens Confirmáveis ou Não-Confirmáveis. As informações solicitadas podem ser carregadas nas mensagens de resposta ou nas mensagens de Confirmação.

O modelo abstrato do protocolo CoAP pode ser feito usando-se duas camadas:

1. Uma camada de mensagens no formato do protocolo CoAP encapsulada em datagrama UDP;
2. Os procedimentos de transmissão e recepção de mensagens em modo assíncrono usando os códigos Método e Resposta conforme mostrado na Tabela C.3.

Tabela C.3 - Modelo de Camadas do Protocolo CoAP

Aplicação
Solicitações e Respostas – Protocolo CoAP
Mensagens - Protocolo CoAP
UDP

As semânticas de solicitação e resposta CoAP são transportadas em mensagens que incluem um código de método ou de resposta, respectivamente. Informações opcionais (ou padrão) de solicitação e resposta, como URI e tipo de mídia de carga útil, são transportadas como opções. Um Token é usado para corresponder respostas a solicitações independentemente das mensagens subjacentes.

O protocolo suporta o armazenamento em cache de respostas, a fim de atender eficientemente às solicitações. Um recurso de armazenamento do tipo cache normalmente está localizado em um dispositivo cliente, servidor ou em um dispositivo intermediário de rede. O recurso de Proxying é útil em redes restritas por vários motivos, incluindo limitação de tráfego de rede, para melhorar o desempenho, para acessar recursos de dispositivos que possuem mecanismos de economia de energia por suspensão ou por motivos de segurança.

Quando um dispositivo envia uma mensagem através de um proxy, o URI do recurso a ser solicitado é incluído na solicitação, enquanto o endereço IP de destino é definido como o endereço do proxy. Como o CoAP foi projetado de acordo com a arquitetura REST, exibe uma funcionalidade semelhante à do protocolo HTTP. Assim, é bastante simples mapear de CoAP para HTTP e de HTTP para CoAP. Esse mapeamento pode ser usado para realizar uma interface HTTP REST usando CoAP ou para converter entre HTTP e CoAP. Esta função converte o método ou o código de resposta, o tipo de mídia e as opções no recurso HTTP correspondente.

C.1.4 Uso do Protocolo TCP/IP para carregar as mensagens CoAP

A norma aceita o uso de protocolo TCP/IP para envio das mensagens do protocolo CoAP [RFC0793], principalmente quando há bloqueio do tráfego de datagramas UDP. Estima-se que entre 2% a 4% das redes possuem algum tipo de restrição ao tráfego de UDP, mais notadamente em redes corporativas. Assim, torna-se interessante usar as mensagens do protocolo CoAP em conexão TCP/IP para envio e recebimento de dados remotamente, já que essa tarefa é realizada por equipamentos gateway com maior capacidade de processamento. O TCP utiliza mecanismos mais sofisticados de controle de congestionamento e fluxo do que os

mecanismos fornecidos pelo CoAP sobre UDP, o que é útil para a transferência de cargas úteis maiores.

C.1.5 Terminologia usada no Protocolo CoAP

- a) **Host:** Um equipamento na rede que usa o protocolo CoAP. O nome Host é mais consistente com o uso de padrões da Internet, sendo identificado por um endereço IP, um número de porta UDP e uma associação de segurança. Sempre um Host pode enviar e receber mensagens.
- b) **Emissor (Sender):** é o Host emissor da mensagem naquele momento.
- c) **Receptor (Recipient):** é o Host receptor da mensagem naquele momento. Durante o procedimento de comunicação, um Host envia e recebe mensagens, intercalando as funções de emissor e receptor.
- d) **Cliente (Client):** é o Host que faz a solicitação de dados.
- e) **Servidor (Server):** é o Host que recebe a solicitação de dados.
- f) **Servidor de Origem (Origin Server):** é o Servidor que fornece a informação originalmente.
- g) **Host Intermediário (Intermediary):** é o Host que atua como cliente e servidor entre o cliente final e o servidor de origem. Um tipo comum de Host Intermediário é denominado Proxy.
- h) **Proxy:** é um tipo de Host intermediário que tem por função principal encaminhar mensagens de solicitação e resposta entre cliente e servidor, podendo também realizar armazenamento em cache, conversão de nome e conversão de protocolo. Neste trabalho, o equipamento Concentrador, composto pelo hardware Raspberry PI 3 e pelo software Java opera como Proxy, intermediando as mensagens entre os clientes e os controladores Arduíno.
- i) **Mensagem Confirmável (Confirmable Message):** As mensagens do tipo Confirmável (CON) quando enviadas, requerem uma mensagem de resposta indicando se foi recebida corretamente ou não. A integridade da mensagem é verificada, e caso tenha sido recebida corretamente, o destinatário envia uma mensagem de Reconhecimento (ACK). Caso a mensagem tenha sido recebida com erros, o destinatário envia uma mensagem de Reset (RST). No caso de não recebimento, o emissor espera por um tempo determinado e tenta novamente enviar a mensagem.

- j) **Mensagem Não-Confirmável (Non-Confirmable Message):** As mensagens do tipo Não-Confirmável (NON), quando enviadas, não requerem uma mensagem de resposta. Este tipo de mensagem é usada em mensagens cíclicas de requisição de medição de sensores, por exemplo. Caso ocorra erro na recepção, o destinatário descarta a mensagem.
- k) **Mensagem de Reconhecimento (Acknowledgement Message):** uma mensagem de reconhecimento (ACK) é enviada em resposta a uma mensagem de solicitação informando que a mensagem foi recebida.
- l) **Mensagem de Reset (Reset Message):** Uma mensagem de Reset (RST) indica que a mensagem foi recebida, mas o procedimento de integridade detectou erro ou falta algum contexto para processá-la adequadamente. Uma mensagem RST pode ser enviada em resposta a uma mensagem Confirmável ou Não-Confirmável dependendo da situação. Caso seja enviada uma mensagem de solicitação do tipo vazia (EMPTY), causa como resposta uma mensagem de Reset. Este procedimento é usado para teste de enlace, sendo denominado "CoAP ping".
- m) **Resposta Incorporada (Piggy-backed Response):** uma mensagem de Reconhecimento (ACK) pode carregar as informações solicitadas pela mensagem Confirmável (CON), caso as informações estejam disponíveis. Este tipo de resposta de Reconhecimento com Informações é denominada Resposta Incorporada.
- n) **Resposta Separada (Separate Response):** caso as informações não estejam disponíveis quando do recebimento de uma mensagem Confirmável (CON) de solicitação, é enviada uma mensagem de Reconhecimento (ACK) vazia, e posteriormente as informações são enviadas através de outra mensagem Confirmável. Este tipo de mensagem é denominada Resposta Separada.
- o) **Opção Crítica (Critical Option):** Uma opção que precisa ser entendida pelo terminal que recebe a mensagem para processar adequadamente a mensagem. Observe que a implementação de opções críticas é, como o nome "Opção" implica, geralmente opcional: opções críticas sem suporte levam a uma resposta de erro ou rejeição sumária da mensagem.
- p) **Opção Eletiva (Elective Option):** Uma opção que deve ser ignorada por um nó de extremidade que não o entende. Processar a mensagem mesmo sem entender a opção é aceitável.

- q) **Opção Não-Segura (Unsafe Option):** Uma opção que precisa ser entendida por um proxy que recebe a mensagem para poder encaminhar a mensagem com segurança.
- r) **Opção Segura (Safe Option):** Uma opção que se destina a ser segura para encaminhamento por um proxy que não a entende. Encaminhar a mensagem mesmo sem entender a opção é aceitável.
- s) **Descoberta de Recursos (Resource Discovery):** O processo em que um cliente CoAP consulta um servidor por sua lista de recursos hospedados.
- t) **Formato do Conteúdo (Content-Format):** este campo identifica o formato da informação contida no Payload da mensagem. Os tipos mais usados são texto: text/plain e binário: application/octet-stream.

C.1.6 Uso do Protocolo UDP para Carregar o Protocolo CoAP

Devido à sua simplicidade e rapidez, o protocolo UDP foi escolhido preferencialmente na norma para carregar as mensagens do protocolo CoAP. A rapidez das redes permite que as informações de estado sejam enviadas em somente uma mensagem, agilizando todos os procedimentos de transmissão, recepção e verificação de integridade. Devido ao fato do UDP não ser orientado à conexão, é preciso implementar os procedimentos de verificação de erros e re-tentativas de transmissão das mensagens no próprio software aplicativo.

Outra razão decisiva para o uso do UDP é o tipo de acesso aos dados pelo Arduíno disponibilizado pela biblioteca de funções. As rotinas da biblioteca de funções EthernetUDP do Arduíno IDE efetuam a transmissão e recepção dos bytes de dados entre o processador e a placa Ethernet através da interface SPI. A SPI - Serial Peripheral Interface - é um tipo de interface de comunicação serial síncrona full-duplex usada para comunicação de curta distância usada principalmente em sistemas embarcados. Esta interface proporciona taxa de transmissão/recepção com excelente velocidade, o que permite aproveitar ao máximo a interface Ethernet.

C.1.7 Formato das Mensagens no Protocolo CoAP

Uma mensagem do protocolo CoAP é composta por uma sequência de bytes em formato binário, ou seja, cada byte pode assumir valores de 0 a 255. Isto a torna mais compacta e eficiente, se comparada a mensagens compostas por sequências de caracteres. Além do

formato binário, as mensagens são montadas de forma compacta, com um pequeno cabeçalho (header) e mais três campos: Token, Opções e Payload.

Para evitar a fragmentação, e conseqüentemente os procedimentos de ordenamento de pacotes, uma mensagem do protocolo CoAP sempre é carregada na seção de dados de somente um datagrama UDP, fazendo o envio e recebimento o mais eficiente possível. O protocolo CoAP define quatro tipos de mensagens:

1. Confirmadas;
2. Não Confirmadas;
3. Confirmação;
4. Reset.

A Tabela C.4 mostra o formato de uma mensagem do protocolo CoAP. A linha superior, onde estão os subcampos: Ver, T, TKL, Code e Message ID, correspondem ao Campo Cabeçalho (Header):

Tabela C.4 - Formato Básico das Mensagens do Protocolo CoAP

Cabeçalho – 4 Bytes
Token (até 8 bytes)
Opções (até 16 bytes)
Payload (até 65000 bytes)

Uma mensagem CoAP é composta basicamente por quatro campos que são os seguintes:

1. Um cabeçalho (header) binário de 4 bytes;
2. Um campo opcional para Identificador (Token) de tamanho variável que pode conter de zero a oito bytes. Os valores deste campo são usados para correlacionar mensagens de requisição e resposta. As regras para definir os valores são apresentadas na seção de procedimentos de requisição e resposta;
3. Um campo de opções de tamanho variável
4. Um campo de dados da aplicação, normalmente binários (Payload).

C.1.8 Cabeçalho das Mensagens do Protocolo CoAP

Todas as mensagens do protocolo CoAP se iniciam com um cabeçalho de 4 bytes de tamanho fixo. Os campos do cabeçalho são mostrados na Tabela C.5.

Tabela C.5 - Formato do Cabeçalho das Mensagens do Protocolo CoAP

Byte	0				1				2				3																		
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
	Ver		Tipo		N. B. Token				Código (c.dd)				Identificador da Mensagem																		

Ver - Versão: inteiro de 2 bits sem sinal. Identifica a versão da implementação do protocolo CoAP. Atualmente, este campo deve ter o valor 1 (01 binário). Valores diferentes são reservados para futuras aplicações.

Tipo: inteiro de 2 bits sem sinal. Identifica o tipo da mensagem em relação aos procedimentos de envio e resposta:

- 00 (0) => Confirmável (CON);
- 01 (1) => Não-Confirmável (NON);
- 10 (2) => Reconhecimento (ACK);
- 11 (3) => Reset (RST).

N. B. Token – Número de Bytes do Token: inteiro de 4 bits sem sinal. É o número de bytes do campo Token, que pode conter de 0 a 8 bytes (valores binários de 0000 a 1000 no campo de 4 bits). Quando a funcionalidade de identificação da mensagem por Token não é usado, o valor deste campo deve ser zero (0000). Os valores de 9 a 15 (1001 a 1111 binários) são reservados e não devem ser usados.

Código: inteiro de 8 bits sem sinal, dividido em dois subcampos: classe (c) = 3 bits mais significativos e detalhe (dd) = 5 bits menos significativos, documentado como "c.dd". O primeiro subcampo, denominado Classe "c" é um dígito que pode ter os valores de 0 a 7. O segundo subcampo, denominado Detalhe "dd", são dois dígitos que podem ter os valores de 00 a 31. A Tabela C.6 mostra os códigos das mensagens de solicitação:

Tabela C.6 - Códigos dos Tipos de Mensagens de Solicitação

Classe		Detalhe		Código	Método
decimal	binário	decimal	binário		
0	000	0	00000	0	EMPTY
0	000	1	00001	1	GET
0	000	2	00010	2	POST
0	000	3	00011	3	PUT
0	000	4	00100	4	DELETE
0	000	5	00101	5	FETCH
0	000	6	00110	6	PATCH
0	000	7	00111	7	iPATCH

A classe c = 0 indica uma solicitação, a classe c = 2 indica uma resposta de sucesso, a classe c = 4 indica uma resposta de erro do cliente e uma classe c = 5 indica uma resposta de erro do

servidor. O código 0.00 indica uma mensagem vazia. No caso de uma solicitação com a classe $c = 0$, o campo detalhe indica o método de solicitação. Na resposta, o detalhe indica um código de resposta. Os valores possíveis são mantidos nos registros de códigos CoAP. Os códigos de resposta 96-127 estão reservados para uso futuro. Todos os outros códigos de resposta não são atribuídos. As Tabelas C.7, C.8 e C.9 mostram os códigos das mensagens do protocolo CoAP com todas as combinações possíveis de classe (c) e detalhe (d):

Tabela C.7 - Códigos dos Tipos de Mensagens de Resposta de Sucesso

Classe		Detalhe		Código		Método
decimal	binário	decimal	binário	inteiro	c.dd	
2	010	1	00001	65	2.01	Created
2	010	2	00010	66	2.02	Deleted
2	010	3	00011	67	2.03	Valid
2	010	4	00100	68	2.04	Changed
2	010	5	00101	69	2.05	Content

Tabela C.8 - Códigos dos Tipos de Mensagens de Resposta com Erro de Cliente

Classe		Detalhe		Código		Método
decimal	binário	decimal	binário	inteiro	c.dd	
4	100	0	00000	128	4.00	Bad Request
4	100	1	00001	129	4.01	Unauthorized
4	100	2	00010	130	4.02	Bad Option
4	100	3	00011	131	4.03	Forbidden
4	100	4	00100	132	4.04	Not Found
4	100	5	00101	133	4.05	Method Not Allowed
4	100	6	00110	134	4.06	Not Acceptable
4	100	12	01100	140	4.12	Request Entity Incomplete
4	100	13	01101	141	4.13	Conflict
4	100	15	01111	143	4.15	Precondition Failed

Tabela C.9 - Códigos dos Tipos de Mensagens de Resposta com Erro de Servidor

Classe		Detalhe		Código		Método
decimal	binário	decimal	binário	inteiro	c.dd	
5	101	0	00000	160	5.00	Internal Server Error
5	101	1	00001	161	5.01	Not Implemented
5	101	2	00010	162	5.02	Bad Gateway
5	101	3	00011	163	5.03	Service Unavailable
5	101	4	00100	164	5.04	Gateway Timeout
5	101	5	00101	165	5.05	Proxying Not Supported

Observação: O código do protocolo CoAP gera identificadores similares aos usados no protocolo HTTP. Por exemplo, uma resposta de sucesso que leva uma informação é $c = 2$ e d

= 5 (código 2.05 Resposta OK e Conteúdo), ou uma resposta de um recurso não encontrado c = 4 e d = 4 (código 4.04 Não Encontrado).

(Message ID) - Identificação da mensagem: inteiro de 16 bits sem sinal, onde o primeiro byte é o mais significativo e o segundo byte o menos significativo. Este campo possui um número que identifica a mensagem, sendo usado principalmente para evitar a ocorrência de mensagens duplicadas. É usado para fazer a referência entre uma mensagem de solicitação do tipo Confirmável (CON) e a mensagem de resposta, que pode ser de dois tipos: 1) Mensagem de Reconhecimento (ACK) no caso do correto recebimento; 2) Mensagem de Reset (RST) no caso de erro na recepção. Assim, o campo Message ID da mensagem de solicitação e da mensagem de resposta deve conter o mesmo valor. Este campo também é usado no caso do envio de mensagem do tipo Não Confirmável (NON).

C.1.9 Campo Token das Mensagens do Protocolo CoAP

O Campo Cabeçalho é seguido pelo campo Token, que pode ser de 0 a 8 bytes, conforme determinado pelo campo Comprimento do Token no Cabeçalho. O campo Token tem por finalidade efetuar a correspondência entre uma solicitação de uma determinada informação e a resposta correspondente. O valor do Token é um número sequencial que pode ter de 0 a 8 bytes. O significado de um Token de comprimento zero é a ausência deste mecanismo de identificação, mas normalmente existe outro recurso para esta finalidade tal como o endereço IP, por exemplo. Uma mensagem de solicitação enviada por um cliente, possui um Token que o servidor deve repetir sem modificação, em qualquer resposta.

O valor do Token é gerado pelo cliente de acordo com regras próprias. Um servidor pode ser questionado por mais de um cliente. Desta forma, o valor do campo Token presente na mensagem de solicitação é usado pelo servidor para identificar o cliente, e assim, direcionar corretamente a mensagem de resposta. Normalmente o cliente deve gerar um valor único para o Token em uso para um determinado par de origem / destino. As exceções são quando o cliente acessa servidores diferentes em endereços IP distintos, ou acessa o mesmo servidor com múltiplas conexões em portas diferentes. Nesses casos, duas mensagens podem ter o mesmo valor de Token, já que o endereço IP ou o número da porta podem identificar o par emissor / receptor. Um valor de Token vazio é apropriado, por exemplo, quando nenhum outro Token está em uso para um destino ou quando as solicitações são feitas em série por destino e recebem respostas em encadeamento. Existem, no entanto, várias estratégias de implementação possíveis para cumprir isso.

C.1.10 Campo Opções das Mensagens do Protocolo CoAP

Os campos Cabeçalho e Token podem ser seguidos pelo campo de Opções. O protocolo CoAP define um número de opções que podem ser incluídas em uma mensagem, onde cada instância de uma determinada opção em uma mensagem especifica o Número da Opção, o Comprimento do Valor da Opção e o Valor da Opção. O número da opção não é fornecido diretamente, sendo calculado como a soma do parâmetro Delta e o número de opção da instância anterior. No caso da primeira instância, assume-se que a instância anterior tem número de opção zero. Uma opção pode ter várias instâncias. Por exemplo, uma mensagem de requisição CoAP pode conter uma URI e um payload. Assim, o campo Opção deve ter pelo menos duas instâncias, Uri-Path e outra do tipo Content-Format com a identificação do tipo de dados do payload. A Tabela C.10 mostra o byte de definição de uma instância de uma opção :

Tabela C.10 - Formato do Campo Opção

7	6	5	4	3	2	1	0	Bit
Delta para Compor o Número da Opção				Número de Bytes da Opção				1 Byte

C.1.11 Descrição dos Subcampos do Campo Opções

Delta da Opção (Option Delta): Inteiro sem sinal de 4 bits. Um valor entre 0 e 12 indica o parâmetro Delta da Opção. O parâmetro Delta define o número da opção. O primeiro Delta é igual ao próprio número da opção. Para definir o número da próxima instância o parâmetro delta é igual ao número da instância menos o número da instância. Por exemplo, caso o número da primeira instância seja 11 (URI-Path) e o número da segunda instância seja 14 (Max-Age), o número do primeiro parâmetro delta = 11 e o número do segundo delta = 3 (11 + 3 = 14) que é o número da segunda instância.

Comprimento da Opção (Option Length): Inteiro sem sinal de 4 bits. Um valor entre 0 e 12 indica o comprimento do Valor da Instância da Opção, em bytes.

Valor (Value): Uma sequência de bytes referentes à opção. O tipo de opção determina o número de bytes e o formato do valor. A tabela C.11 mostrada no item C.1.12 a seguir apresenta os formatos de valor para o campo opção.

C.1.12 Tabela de Tipos de Opções das Mensagens do Protocolo CoAP

A Tabela C.11 mostra os tipos de opções de mensagens definidos para o protocolo CoAP:

Tabela C.11 - Tipos de Opções para as Mensagens CoAP

Número	Nome da Opção	Referência
0	(Reserved)	RFC7252
1	If-Match	RFC7252
3	Uri-Host	RFC7252
4	ETag	RFC7252
5	If-None-Match	RFC7252
7	Uri-Port	RFC7252
8	Location-Path	RFC7252
11	Uri-Path	RFC7252
12	Content-Format	RFC7252
14	Max-Age	RFC7252
15	Uri-Query	RFC7252
16	Accept	RFC7252
20	Location-Query	RFC7252
35	Proxy-Uri	RFC7252
39	Proxy-Scheme	RFC7252

C.1.13 Campo Payload das Mensagens do Protocolo CoAP

Após os campos Token e Opções, caso existam, o protocolo CoAP pode carregar informações no campo denominado Payload. A existência do campo Payload deve ser sinalizada por um byte de valor 0xFF, após o qual se inicia a parte da mensagem correspondente ao payload. O tipo de dados que é levado no campo payload deve ser identificado pela opção do tipo “Content-Format” (Opção Número 12) seguida pelo byte identificador do tipo.

Os tipos de dados mais usados são do tipo texto (text/plain) e binário (application/octet-stream). O tipo de dados binário proporciona a máxima compactação. Esta característica é interessante no caso de envio de uma quantidade maior de bytes em uma mensagem. O tipo texto é muito usado no envio de dados de medição em caracteres, tais como, por exemplo, “10.5”. A Tabela C.12 mostra os tipos de dados definidos para o protocolo CoAP.

Tabela C.12 - Tipos de Dados definidos para a Opção 12: Content-format

Tipo de Dados (Media Type)	Identificador	Referência
text/plain; charset=utf-8	0 (0x00)	[RFC2046] [RFC3676] [RFC5147]
application / link-format	40 (0x28)	[RFC6690]
application / xml	41 (0x29)	[RFC3023]
application / octet-stream	42 (0x2A)	[RFC2045] [RFC2046]
application / exi	47 (0x2F)	[EXIMIME]
application / json	50 (0x32)	[RFC4627]

C.1.14 Procedimentos de Troca de Mensagens no Protocolo CoAP

As mensagens CoAP são trocadas de forma assíncrona entre os pontos finais do CoAP. Eles são usados para transportar solicitações e respostas do CoAP. Como o CoAP é vinculado a transportes não confiáveis, como o UDP, as mensagens do CoAP podem chegar fora de ordem, aparecer duplicadas ou desaparecer sem aviso. Por esse motivo, o CoAP implementa um mecanismo de confiabilidade leve, sem tentar recriar o conjunto completo de recursos de um transporte como o TCP. Tem os seguintes recursos:

1. Confiabilidade simples de retransmissão de parada e espera com backoff exponencial para mensagens de confirmação.
2. Detecção duplicada para mensagens confirmadas e não confirmadas.

Um ponto final CoAP é a origem ou o destino de uma mensagem CoAP. É identificado dependendo do modo de segurança usado. Sem segurança, o Host é identificado apenas por um endereço IP e um número de porta UDP. Existem diferentes tipos de mensagens, onde o tipo de mensagem é especificado pelo campo Tipo do Cabeçalho CoAP. Uma mensagem pode conter uma solicitação, uma resposta ou estar vazia. Isso é sinalizado pelo campo Request / Response Code no cabeçalho do CoAP. Uma mensagem vazia tem o campo Código definido como 0 e o campo Comprimento do Sinal deve ser igual a 0, e nenhum byte deve estar presente após o campo ID da Mensagem.

O CoAP faz uso dos métodos GET, PUT, POST e DELETE de maneira similar ao HTTP. O suporte a URI em um servidor é simplificado, pois o cliente já analisa o URI e o divide em componentes de host, porta, caminho e consulta, utilizando os valores padronizados para obter eficiência. Os códigos de resposta correspondem a um pequeno subconjunto de códigos de resposta HTTP com alguns códigos específicos de CoAP adicionados. A camada de solicitação e resposta do protocolo CoAP refere-se aos tipos básicos de mensagem e aos procedimentos de envio e resposta.

A confiabilidade é fornecida marcando uma mensagem como Confirmable (CON). Uma mensagem de confirmação é retransmitida usando um tempo limite padrão e recuo exponencial entre retransmissões, até que o destinatário envie uma mensagem de confirmação (ACK) com o mesmo ID de mensagem (por exemplo, 0x7d34) do terminal correspondente. Quando o destinatário não é capaz de processar uma mensagem de confirmação (ou seja, nem mesmo é capaz de fornecer uma resposta de erro adequada), ele responde com uma mensagem de reset (RST) em vez de um reconhecimento (ACK).

As mensagens do protocolo CoAP são encapsuladas em datagramas do protocolo UDP, o qual não possui procedimentos de verificação de erro e re-tentativas, para enviar as mensagens. Assim, faz-se necessário a implementação de um mecanismo para realizar estas funções. A mensagem do tipo Confirmável (CON), quando é enviada de um Dispositivo 1 para um Dispositivo 2, sempre requer uma mensagem de resposta do Dispositivo 2 para o Dispositivo 1. Neste caso podem ocorrer três situações:

1. A mensagem enviada pelo Dispositivo 1 foi bem recebida pelo Dispositivo 2. Isto significa que o procedimento de verificação de integridade, realizado no Dispositivo 2, não detectou erro na mensagem recebida. Neste caso, o Dispositivo 2 deve enviar uma mensagem de resposta do tipo Reconhecimento (ACK) para o Dispositivo 1, informando o fato da mensagem ter sido bem recebida. Caso a mensagem recebida pelo Dispositivo 2 solicite alguma informação, a mensagem de resposta do tipo ACK, enviada pelo Dispositivo 2, pode carregar a informação solicitada, se estiver disponível. Quando a mensagem de reconhecimento (ACK) carrega um Payload de informação, é denominada “Piggy-Back” response;
2. A mensagem enviada pelo Dispositivo 1 não foi bem recebida pelo Dispositivo 2. Isto significa que o procedimento de verificação de integridade, realizado no Dispositivo 2, detectou erro na mensagem recebida. Neste caso, o Dispositivo 2 envia uma mensagem de Reset (RST) para o Dispositivo 1 informando a falha no recebimento da mensagem. Assim, o Dispositivo 1 pode efetuar uma nova tentativa de envio da mensagem para o Dispositivo 2.
3. A mensagem foi enviada pelo Dispositivo 1 e não foi recebida pelo Dispositivo 2. Isto ocorre quando há falha no canal de comunicação. Neste caso, o emissor deve esperar a resposta por um tempo determinado e enviar novamente a mensagem para o receptor.

A Figura C.2 apresenta a mensagem de solicitação confirmável (CON) enviada pelo cliente com o campo identificador: Message ID = 0xAA51, sendo corretamente recebida pelo servidor. Neste caso, o servidor responde com a mensagem de reconhecimento (ACK) que tem a mesma identificação, indicando o correto recebimento da mensagem.

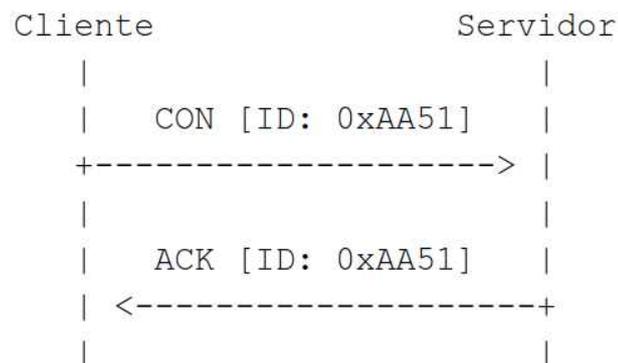


Figura C.2 - Fluxo de Mensagem do tipo CON: sem erro

A Figura C.3 mostra a mensagem de solicitação do tipo confirmável (CON) que não foi corretamente recebida pelo servidor. O servidor responde com a mensagem de Reset (RST), informando o erro na recepção. Assim, o cliente pode tentar novamente enviar a mensagem:

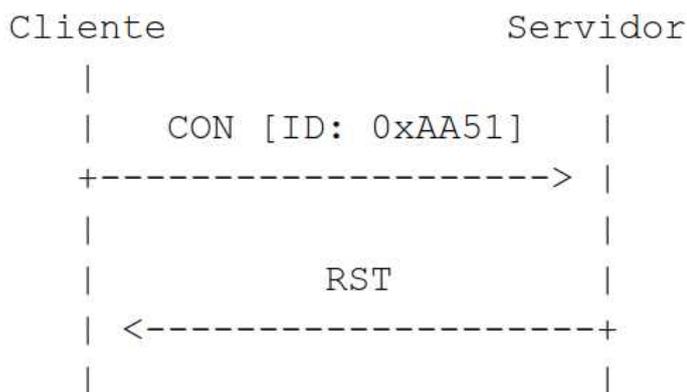


Figura C.3 - Fluxo de Mensagem do tipo CON: com erro

A Figura C.4 apresenta um exemplo onde a informação solicitada em uma mensagem Confirmável (CON) não está disponível. Neste caso, o servidor responde com a mensagem de reconhecimento (ACK) vazia. A mensagem de solicitação do valor de corrente (GET/corrente) carrega o Token de valor 0x14 que a identifica.

Assim, que a informação solicitada estiver disponível, o servidor envia para o cliente uma nova mensagem do tipo Confirmável (CON) com a informação solicitada anteriormente, e espera a mensagem de reconhecimento do servidor (ACK). O valor do Token identifica a solicitação da informação para o servidor, permitindo que este realize o gerenciamento do envio das informações para os clientes corretamente, mesmo que seja necessário o uso de mais de uma mensagem para realizar este procedimento.

Observação: o uso do campo Token é opcional na implementação do protocolo. Por exemplo, quando as informações de supervisão sempre estão disponíveis no servidor, não é necessário identificar uma solicitação para posterior envio.



Figura C.4 - Fluxo de Mensagens do tipo CON com informação não disponível

O tipo de mensagem Confirmável (CON) sempre é usada no caso do envio de comandos, onde é necessária a confirmação do correto recebimento do comando e a sua correta execução. Em sistemas que exigem maior confiabilidade, além da mensagem de resposta ser do tipo reconhecimento (ACK), indicando o correto recebimento, também é enviado na mensagem o estado do dispositivo após o comando, para indicação da correta execução.

Uma solicitação é transportada em uma mensagem Confirmável (CON) ou Não Confirmável (NON) e, se imediatamente disponível, a resposta a uma solicitação transportada em uma mensagem de confirmação é transportada na mensagem de Confirmação (ACK) resultante. Dois exemplos para uma solicitação básica GET com resposta em “piggy-back” são apresentados na Figura C.5, onde a primeira recebe a resposta com sucesso, identificada pelo código 2.05 Success – Content (Sucesso – Conteúdo), e a segunda recebe a resposta com código de erro 4.04 Not Found (Recurso não encontrado). É importante observar que os códigos de solicitação e resposta do Protocolo CoAP são idênticos ao do Protocolo HTTP, sendo que a diferença está no formato compactado do fluxo de bytes da mensagem CoAP.

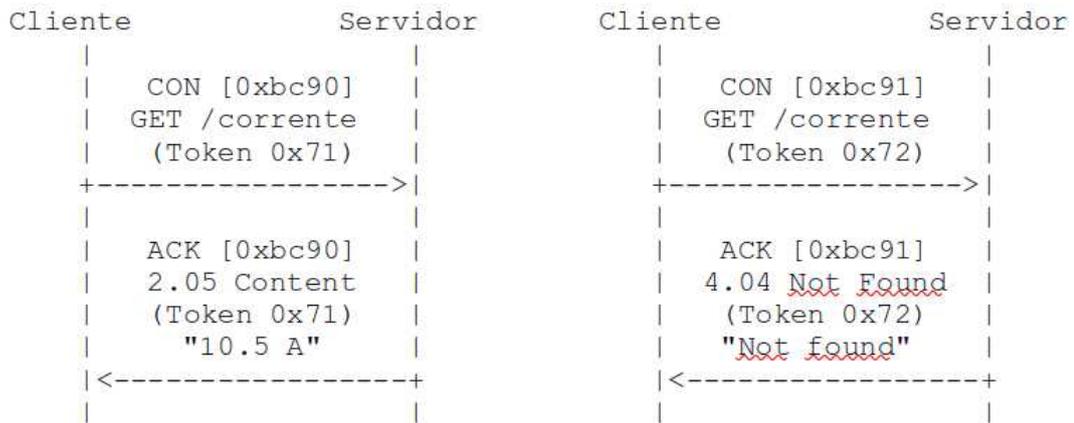


Figura C.5 - Transmissão e Recepção de Mensagem em Modo Confiável com Resposta

Se um servidor não puder responder imediatamente a uma solicitação realizada em uma mensagem que solicita confirmação, é usada uma mensagem de confirmação vazia para que o cliente possa parar de retransmitir a solicitação. Quando a resposta está pronta, o servidor a envia em uma nova mensagem que solicita confirmação. Isso é chamado de resposta separada, conforme mostrado na Figura C.6.

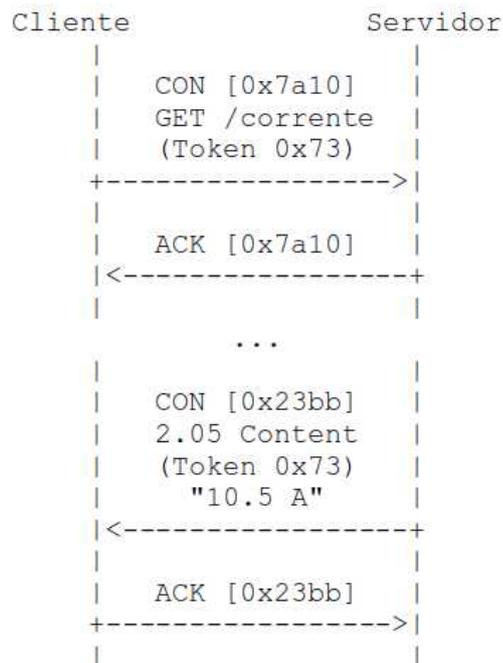


Figura C.6 - Requisição do tipo GET com Resposta Incorporada

Da mesma forma, se uma solicitação for enviada em uma mensagem não-confirmada, a resposta geralmente será enviada por meio de uma nova mensagem não confirmada, embora o servidor possa enviar uma mensagem de confirmação.

Uma mensagem que não requer transmissão confiável, por exemplo, cada medição única de um fluxo de dados do sensor, pode ser enviada como uma mensagem não confirmável (NON). Este tipo de mensagem, mesmo sem ter a necessidade de resposta possuem o campo de identificação (ID) de Mensagem, o qual é usado para detecção de duplicidade. Caso o receptor não consiga processar uma mensagem do tipo Não confirmável, ele pode responder com uma mensagem Redefinir (RST).

Como regra geral, podemos assumir que sempre uma solicitação de um cliente a um servidor usa uma mensagem do tipo Confirmável, pois é esperada uma resposta com a informação solicitada. A resposta do servidor pode ser do tipo Confirmável ou Não-Confirmável. No caso em que os clientes fazem atualização periódica de valores, a resposta do servidor normalmente é do tipo Não-Confirmável. Isto é feito principalmente para evitar excesso de tráfego. A Figura C.7 mostra o envio e recebimento de mensagens no modo não confiável.



Figura C.7 - Mensagens do tipo Não-Confíável (NON)

C.2 Características Gerais do Protocolo HTTP

As informações deste item foram obtidas da norma RFC2616 - Hypertext Transfer Protocol – HTTP/1.1. O desenvolvimento do HTTP foi iniciado por Tim Berners-Lee no CERN em 1989. Atualmente, o HTTP está na versão 1.1. O protocolo HTTP é implementado na arquitetura Cliente e Servidor. Esta arquitetura distribui as tarefas entre dois tipos de entidades na rede denominados Servidores e Clientes. A função dos servidores é rodar um ou mais programas que disponibilizam recursos para os clientes. Esses recursos são normalmente informações na forma de páginas HTML, arquivos de dados no formato da linguagem XML,

arquivos de dados no formato JSON arquivos de dados no formato binário (octet-stream) e programas (scripts). Os clientes são as entidades usuárias desses recursos.

O tipo de cliente mais difundido é do tipo Navegador (Browser), com opções de diversos fornecedores tais como Google Chrome, Mozilla Firefox, Internet Explorer. Atualmente, os clientes do tipo aplicativos para dispositivos móveis estão se tornando também muito difundidos. Esses aplicativos rodam nos sistemas operacionais Android e iOS, usados em celulares e tablets. Na verdade, qualquer programa que implemente uma API (Application Programming Interface) que possa enviar requisições, receber e tratar mensagens no protocolo HTTP é um cliente HTTP. Atualmente existem muitos mecanismos automáticos de consulta na Internet, daí a necessidade do uso de recursos de identificação de usuário humano.

C.2.1 Formato da Mensagem HTTP

Existem dois tipos de mensagem HTTP: requisição e resposta. Uma mensagem de requisição é formada por:

1. Uma linha de requisição;
2. As linhas de cabeçalho
3. O corpo da mensagem.

A linha de requisição é formada pelo método, a URL e a versão http, todos separados por um espaço. O método é o tipo de ação que a mensagem requer. Exemplos de métodos que são muito usados em mensagens http são GET, POST e HEAD. A URL é o objeto sobre o qual a mensagem quer realizar a ação(método) requisitada. A versão HTTP se refere à versão requisitada pela mensagem. As linhas de cabeçalho contém os detalhes referentes à requisição feita do cliente para o servidor. O cabeçalho das mensagens de requisição HTTP podem ser classificados em três tipos:

- Gerais: contêm informações referentes principalmente à própria mensagem, e são usadas para controlar seu processamento e prover o receptor com informações extras.
- Requisição: fornecem para o servidor mais informações sobre a natureza da requisição do cliente, e dão ao cliente mais controle sobre como a requisição é gerenciada. Podem também contar quais formatos ou códigos o cliente consegue processar.
- Entidade: descrevem a entidade contida no corpo da mensagem, se existir alguma.

Normalmente a mensagem de requisição não irá possuir cabeçalhos de entidade, pois, dificilmente uma mensagem de requisição trará consigo um corpo de mensagem. No corpo de

mensagem, quando o mesmo existir numa mensagem de requisição, haverá uma entidade, que pode ser um arquivo de música, uma imagem, uma página HTML, etc.

Uma mensagem de resposta é formada por uma linha de estado, as linhas de cabeçalho e o corpo da mensagem. A versão HTTP, o método da resposta, e uma mensagem associada ao código denominada URI são informados na primeira linha do cabeçalho. A versão HTTP se refere à versão da mensagem de resposta. O método da resposta e a mensagem associada a ele devem trazer a informação sobre os resultados do processamento da requisição do cliente. O código de resposta é um número de três dígitos que indica o resultado formal que o servidor está comunicando ao cliente. Já a mensagem associada é opcional, e é um texto descritivo que pode ser mostrado para o usuário humano do cliente http, que poderá então saber o que o servidor respondeu.

C.2.2 Características da Linguagem HTML

A Linguagem de Marcação de Hipertexto (HTML - Hypertext Markup Language) é um tipo de linguagem em formato texto que carrega basicamente conteúdo de informação e formatação para apresentação por navegadores (browsers). A primeira versão foi de 1991 e atualmente está na versão HTML5.2 de 2017. Esta linguagem usa caracteres texto sendo baseada em etiquetas ou marcadores (em inglês: tags), as quais definem atributos, valores e atributos filhos. Assim, é possível formatar telas de apresentação com texto, figuras, tabelas e hipertextos, os quais permitem carregar outra página a partir de um link. Também existem outros elementos como caixas de texto para entrada de dados, botões de seleção e botões para execução de funções.

O fato da linguagem HTML ser composta de caracteres, permite a composição de arquivos por qualquer editor de texto, o que facilita os procedimentos de desenvolvimento. Atualmente, a disponibilidade de codificação de caracteres como a UTF-8 (8-bit Unicode Transformation Format), permite a composição de páginas de apresentação usando caracteres de praticamente todas as línguas.

A codificação de caracteres UTF-8 é de tamanho variável, sendo compatível com o padrão mais antigo e universal ASCII de um byte por caracter, usando dois bytes para codificar os caracteres latinos, três bytes para as outras línguas e quatro bytes de codificação para outros tipos de caracteres. A codificação de caracteres ASCII (American Standard Code for Information Interchange; "Código Padrão Americano para o Intercâmbio de Informação") foi usada desde o início do advento dos computadores, possuindo 128 códigos disponíveis. Esta

codificação foi usada para a língua inglesa, a língua oficial da computação, e atualmente é muito usada em arquivos de dados do tipo XML e JSON que carregam dados numéricos e demais informações que são compatíveis com a codificação ASCII.

C.2.3 Características da Linguagem XML

A linguagem XML (Extensible Markup Language) é formatada como uma sequência de caracteres texto, suportando o uso de codificação Unicode. A XML permite a elaboração de arquivos contendo informações estruturadas hierarquicamente. Este tipo de estrutura hierárquica pode ser usada para a criação de muitos tipos de arquivos de dados, contendo, por exemplo: registros de bancos de dados; documentos de texto formatados usados por programas de edição de texto como o MS Word e LibreOffice; imagens em formato vetorial; e qualquer tipo de informação tal como configuração e dados de supervisão e controle de sistemas elétricos como descrito na norma IEC61850.

O formato da linguagem XML não depende de hardware ou software, portanto, pode ser usado para intercâmbio de informações entre diferentes sistemas de processamento de dados. Mais especificamente este recurso é útil no caso de informações referentes a banco de dados, que podem ser armazenados por um sistema e lidos por outro. Mesmo arquivos com dados usados por aplicativos tais como editores de texto podem ser facilmente compartilhados entre plataformas diferentes, por terem o mesmo formato. A própria estrutura do arquivo em formato XML facilita o uso por banco de dados relacionais, pois já tem uma estrutura interna com campos de dados relacionados a campos de tipos de dados.

Um arquivo em formato XML possui etiquetas (tags) que identificam os dados, sendo, portanto, fáceis de ler em qualquer editor de texto ou em um terminal com saída texto. Assim, a depuração se torna mais fácil, pois, a leitura dos dados é imediata pelo operador humano. A maior desvantagem do formato XML é o excesso de caracteres devido às etiquetas de início e final de cada campo de dados, o que causa o aumento do tráfego e maior demora no envio das mensagens. Por exemplo, comparando mensagens usadas para envio de dados numéricos, uma mensagem codificada no formato XML pode ser de cinco a dez vezes maior que uma mensagem codificada em formato binário (octet-stream) em número de bytes.

Atualmente, o formato XML é muito usado para troca de informações em aplicações Web, principalmente para envio de dados de formulários e o recebimento de dados para atualização de campos para o usuário. Neste tipo de aplicação, inicialmente é carregado o arquivo com a página HTML junto com um programa (Script) no navegador ou aplicativo. Assim, o

programa (Script) rodando no navegador ou no aplicativo, gerencia a entrada de dados pelo usuário e a troca de informações com o servidor. Este tipo de serviço Web se torna muito mais confiável e estável, pois o próprio navegador armazena as informações inseridas pelo usuário e as envia ao servidor em um arquivo XML que é muito menor que a página HTML completa. A resposta do servidor e a atualização dos campos de dados na página também é feita por arquivo XML, tornando o procedimento mais rápido e seguro, parecendo transparente ao usuário.

Por exemplo, mesmo que ocorra quebra de conexão do cliente com o servidor durante um procedimento de preenchimento de formulário, como as informações são armazenadas em variáveis pelo programa no navegador. Ao ser restabelecida a conexão, o procedimento pode continuar normalmente. No caso de aplicações Web que requerem a atualização periódica de campos de dados para apresentação, a solicitação e recebimento dos dados em formato XML pelo programa torna esse procedimento muito mais eficiente que a atualização de toda a página HTML. A eficiência indica um tráfego de dados muito menor. Por exemplo, a atualização de uma página inteira de um sistema de supervisão pode demandar o tráfego de 30.000 a 40.000 caracteres, enquanto o tráfego apenas dos valores das variáveis pode demandar o tráfego de apenas 2.000 caracteres em uma mensagem em formato XML.

C.2.4 Características dos Programas (Scripts) que rodam nos Clientes

Os programas (scripts) rodam nos clientes executando diversas funções que tornam a operação mais segura e confiável, reduzindo o tráfego de informações entre o cliente e servidor. Os scripts mais difundidos são o Javascript e o PHP, os quais são normalmente carregados junto com a página HTML em formato texto.

Na arquitetura cliente servidor e respeitando o modelo REST, o HTTP é considerado um protocolo sem estado, o que significa que: se um cliente fizer uma requisição, e em seguida fizer a mesma requisição, a resposta é idêntica. Os scripts viabilizam o uso da arquitetura REST, pois, permitem um ótimo nível de automatismo por parte da aplicação rodando no navegador. Assim, por exemplo, o preenchimento de um formulário pelo usuário é feito fundamentalmente no programa (Script), usando variáveis e regras definidas no próprio cliente. Quando um conjunto de informações já está pronto e verificado, através de um comando do usuário, as informações são enviadas para o servidor normalmente em um formato padrão como XML. Da mesma maneira, a resposta do servidor também pode ser um conjunto necessário de informações para atualizar a apresentação ao usuário.

C.2.5 Características da Sessão HTTP

Uma sessão HTTP pode ser definida como uma sequência de envios e recebimentos de mensagens de requisição e resposta. Assim, um determinado cliente HTTP inicia o procedimento solicitando o estabelecimento de uma conexão do tipo TCP/IP em uma determinada porta padrão de um servidor HTTP. O software do servidor, ao detectar o estabelecimento da conexão, recebe os bytes da mensagem de requisição do cliente e executa os procedimentos de verificação e armazenamento das informações. Caso a mensagem recebida seja uma mensagem de requisição válida, o servidor envia para cliente uma mensagem HTTP padrão de resposta. Esta mensagem de resposta do servidor normalmente carrega as informações solicitadas pelo cliente, tais como uma página HTML de apresentação ou um conjunto de dados para atualização. Pode ocorrer também, que a mensagem de requisição do cliente carregue informações, tais como os dados de um formulário preenchido pelo usuário, ou dados para atualização da base de dados do servidor. Neste último caso, o servidor normalmente responde com uma mensagem de reconhecimento dos dados recebidos.

C.2.6 Características dos Métodos HTTP

O protocolo HTTP define um conjunto de métodos, os quais são identificados por palavras em formato texto. Desta forma, um cliente HTTP envia um método a um servidor HTTP com o objetivo de definir qual a ação vai ser executada pelo receptor da requisição, ou seja, o servidor HTTP. O nome de cada método já é auto explicativo, tal como o método GET, usado quando o cliente solicita um recurso ao servidor.

Um recurso é identificado por uma sequência de caracteres que vem após o método, denominada URI. Quando o método GET é solicitado sem o recurso, normalmente indica ao servidor para enviar a página raiz ou índice principal (index). Um servidor HTTP deve implementar ao menos os métodos GET e HEAD. Os métodos mais usados nas trocas de mensagens no protocolo HTTP são o método GET e o método POST. Os métodos definidos para o protocolo HTTP são os seguintes: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS e CONNECT.

O método GET tem por função requisitar uma informação ou um conjunto de informações referentes ao recurso especificado. Em princípio, requisições usando o método GET devem apenas recuperar dados e não devem ter qualquer outro efeito. Por exemplo, respostas comuns à determinadas requisições com método GET podem ter o formato de uma página HTML, um

programa (script) Javascript ou atualizações de variáveis em formato XML. O W3C publicou um conjunto de orientações sobre as características da requisição GET." O método HEAD é uma variação do GET em que o recurso não é retornado. É usado para obter metainformações por meio do cabeçalho da resposta, sem ter que recuperar todo o conteúdo.

O método POST tem por finalidade enviar informações para serem processadas pelo servidor. Estas informações, normalmente, são dados inseridos pelo usuário em um formulário HTML, armazenados por um script no navegador e enviados por comando na tela. Em sistemas de supervisão e controle em tempo real, um cliente tem por função atualizar periodicamente e automaticamente os valores das variáveis em um servidor.

A utilização do método POST em uma requisição ocorre quando é necessário enviar dados ao servidor para serem processados, geralmente por um programa script identificado no Request-URI. Uma requisição por meio desse método sempre requer que as informações submetidas sejam incluídas no corpo da mensagem e formatadas basicamente de duas formas: como uma query string ou em formato binário (Octet-Stream). Todas as mensagens contém cabeçalhos adicionais especificando seu tamanho (Content-Length) e seu formato (Content-Type). Desta forma, o método POST oferece uma maior segurança em relação aos dados transferidos, ao contrário do método GET que os dados são anexados a URL, ficando visíveis ao usuário.

O método PUT tem por função enviar informações para o servidor de forma semelhante ao POST, através do corpo do HTTP a diferença entre os dois métodos é semântica. Por exemplo: Caso você necessite atualizar os dados de um usuário, utilizando o método PUT você pode os atualizar diversas vezes, pois, o PUT vai sobrescrever os dados com isso ficará somente com um único registro atualizado. Se você executasse este mesmo procedimento utilizando o método POST, você criaria diversos registros para cada requisição realizada.

C.2.7 Códigos de status de respostas HTTP

Os códigos de status das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída. As respostas são agrupadas em cinco classes:

1. Respostas de informação (100-199);
2. Respostas de sucesso (200-299);
3. Redirecionamentos (300-399);
4. Erros do cliente (400-499);
5. Erros do servidor (500-599).

C.3 Características do Modelo Arquitetural REST

O termo transferência de estado representacional foi introduzido e definido em 2000 por Roy Fielding em sua tese de doutoramento. A tese de Fielding explicou os princípios REST que eram conhecidos como "modelo de objeto HTTP", a partir de 1994, e foram usados no projeto dos padrões HTTP 1.1 e URI (Uniform Resource Identifiers). O termo destina-se a evocar uma imagem de como um aplicativo da Web bem projetado se comporta: é uma rede de recursos da Web (uma máquina de estados virtuais) na qual o usuário avança pelo aplicativo selecionando identificadores de recursos, como <http://www.exemplo.com/artigos/21>, e operações de recursos, como GET ou POST (transições de estado do aplicativo), resultando na próxima representação do recurso (o próximo estado do aplicativo) sendo transferida para o usuário final para seu uso.

Os Recursos da Web foram definidos pela primeira vez na World Wide Web como documentos ou arquivos identificados por suas URLs. No entanto, hoje, eles têm uma definição muito mais genérica e abstrata que engloba qualquer coisa ou entidade que pode ser identificada, nomeada, endereçada ou manipulada, da forma que for, na Web. Em um Web service RESTful, as solicitações feitas ao URI de um recurso provocará uma resposta com uma carga útil formatada em HTML, XML, JSON ou algum outro formato. A resposta pode confirmar que alguma alteração foi feita no recurso armazenado e a resposta pode fornecer links de hipertexto para outros recursos ou conjuntos de recursos relacionados. Quando o HTTP é usado, como é o mais comum, as operações (métodos HTTP) disponíveis são GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS e TRACE..

Usando um protocolo sem estado e operações padrão, os sistemas RESTful buscam desempenho, confiabilidade e capacidade de crescimento rápido, reutilizando componentes que podem ser gerenciados e atualizados sem afetar o sistema como um todo, mesmo enquanto estiver em execução. O termo Transferência Representacional de Estado (REST - Representational State Transfer) pode ser considerado como um modelo de arquitetura de software que define um conjunto de restrições a serem usados para a criação de web services (serviços Web). Os aplicativos Web que atendem a essas regras são denominados de RESTful, e possuem interoperabilidade com os demais recursos da Internet. Os serviços do tipo RESTful permitem que os sistemas solicitantes acessem e manipulem representações textuais de recursos da Web usando um conjunto uniforme e predefinido de operações sem estado. Mais especificamente, dentro do contexto desse trabalho, a arquitetura RESTful Environments (CoRE) tem por objetivo adequar a arquitetura REST para os dispositivos com

restrição de processamento e memória, assim como também para redes com restrição de velocidade. Assim, o protocolo CoAP atende aos requisitos REST, sendo ideal para interfaceamento com o protocolo HTTP.

C.3.1 Princípios do Modelo Arquitetural REST

Um protocolo cliente/servidor sem estado: cada mensagem HTTP contém toda a informação necessária para compreender o pedido. Como resultado, nem o cliente e, nem o servidor necessitam gravar nenhum estado das comunicações entre mensagens. Na prática, muitas aplicações baseadas em HTTP utilizam cookies e outros mecanismos para manter o estado da sessão (algumas destas práticas, como a reescrita de URLs, não são permitidas pela regra do REST).

Um conjunto de operações bem definidas que se aplicam a todos os recursos de informação: HTTP em si define um pequeno conjunto de operações, as mais importantes são POST, GET, PUT e DELETE. Com frequência estas operações são combinadas com operações CRUD para a persistência de dados, onde POST não se encaixa exatamente neste esquema. Uma sintaxe universal para identificar os recursos. No sistema REST, cada recurso é unicamente direcionado através da sua URI. O uso de hipermídia, tanto para a informação da aplicação como para as transições de estado da aplicação: a representação deste estado em um sistema REST são tipicamente HTML ou XML. Como resultado disto, é possível navegar com um recurso REST a muitos outros, simplesmente seguindo ligações sem requerer o uso de registros ou outra infraestrutura adicional.

C.3.2 Recursos do Modelo Arquitetural REST

Um conceito importante em REST é a existência de recursos (elementos de informação), que podem ser usados utilizando um identificador global (um Identificador Uniforme de Recurso) para manipular estes recursos, os componentes da rede (clientes e servidores) se comunicam através de uma interface padrão (HTTP) e trocam representações de recursos (os arquivos ou ficheiros são recebidos e enviados) – é uma questão polêmica e gera grande discussão, sem a distinção entre recursos e suas representações é demasiado utópico o seu uso prático na rede, onde é popular na comunidade RDF. O pedido pode ser transmitido por qualquer número de conectores (por exemplo clientes, servidores, caches, etc) mas não poderá ver mais nada do seu próprio pedido (conhecido com separação de capas, outra restrição do REST, que é um princípio comum com muitas outras partes da arquitetura de redes e da informação). Assim,

uma aplicação pode interagir com um recurso conhecendo o identificador do recurso e a ação requerida, não necessitando conhecer se existem caches, proxys, ou outra, entre ela e o servidor que guarda a informação. A aplicação deve compreender o formato da informação de volta (a representação), que é geralmente um documento em formato HTML ou XML, onde também pode ser uma imagem ou qualquer outro conteúdo.

Anexo D: Características dos Inversores

Um inversor é basicamente um dispositivo que converte energia elétrica em corrente contínua para energia elétrica em corrente alternada (MAHJI, 2012). Por exemplo, um inversor de um sistema de geração solar típico recebe entrada em corrente contínua com tensão nominal de 24 Volts e converte para 220 Volts em corrente alternada com frequência de 60Hz e forma de onda senoidal pura. Antes da popularização da energia solar, os inversores já eram amplamente utilizados para diversas aplicações, como fonte de alimentação ininterrupta em equipamentos do tipo “No-Break”, e para alimentação e controle de velocidade de motores. Os inversores de potência senoidais trifásicos permitem obter a máxima eficiência e controle de motores de indução trifásicos.

O inversor é o equipamento individual mais caro e mais sujeito a defeitos em uma usina solar, justamente por estar diretamente envolvido com os usuários e com as condições da instalação elétrica do local. A saída de tensão alternada de um inversor pode ser submetida às condições mais comuns de falha que são: sobrecarga, curto-circuito e surtos de tensão que podem causar danos ao equipamento. Um tipo de defeito comum é a falha nos ventiladores, que por possuírem partes móveis, são sujeitos a problemas mecânicos. A não ser que ocorra uma falha catastrófica, um equipamento não pára de funcionar quando ocorrem diversos tipos de danos. Normalmente, a partir de uma condição de falha, ocorre uma degradação progressiva no funcionamento que termina por agravar os danos e causar a pane. Esse é o princípio da Manutenção Baseada em Condição.

Existem basicamente dois tipos de inversores usados em sistemas de geração solar: os inversores com saída em onda modificada e os inversores com saída em onda senoidal que usam tecnologia de Modulação por Largura de Pulso Senoidal (SPWM – Sinusoidal Pulse Width Modulation).

D.1 Inversores com Saída em Onda Modificada

D.1.1 Características Gerais e Restrições

Os inversores com saída em onda modificada basicamente geram uma onda de saída alternada em formato retangular, que possui a mesma frequência da rede e aproximadamente o mesmo valor RMS que uma onda senoidal de mesma frequência. Este tipo de inversor possui um

conjunto de circuitos eletrônicos mais simples, e conseqüentemente tem custo mais baixo. Entretanto, possuem restrições em sua aplicação, já que a onda de saída em corrente alternada que alimenta as cargas não é pura, ou seja, a onda de saída possui componentes de frequência múltiplos de 60Hz.

Estes componentes de frequência mais altos, presentes no sinal de saída dos inversores de onda modificada, causam sobreaquecimento em motores e transformadores de fontes de alimentação lineares, que foram projetados para alimentação com onda senoidal pura. Portanto, este tipo de inversor não deve ser usado nessas aplicações, que incluem refrigeradores, freezers, ar condicionado e equipamentos alimentados com fontes lineares. Desta forma, os inversores com saída em onda modificada são mais adequados para alimentar cargas que possuem fontes de alimentação chaveadas ou cargas resistivas, tais como computadores, monitores, televisores LCD e LED, lâmpadas eletrônicas, lâmpadas incandescentes, switches, roteadores, carregadores de celular, etc. A figura D.1 mostra a forma de onda de saída real, mostrada na tela de um osciloscópio, gerada por um inversor com saída em onda modificada:

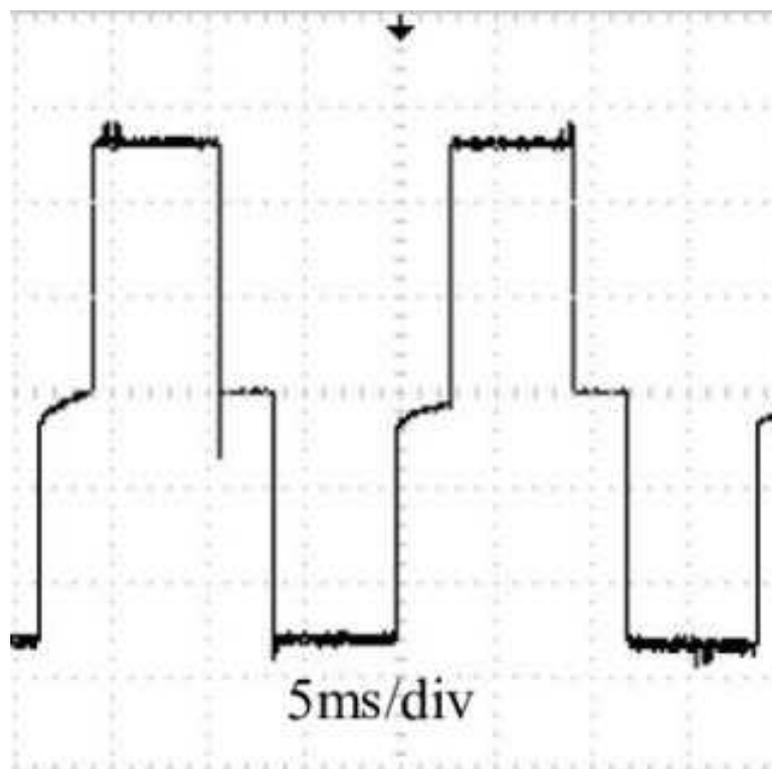


Figura D.1 - Forma de Onda de Saída de um Inversor de Onda Modificada

D.1.2 Arquitetura do Inversor com Saída em Onda Modificada

Este tipo de inversor normalmente possui dois estágios:

1. O primeiro estágio elevador de tensão que transforma a tensão contínua de entrada em uma tensão contínua de amplitude mais alta, com valor aproximadamente igual ao valor de pico de uma onda senoidal equivalente. A eletrônica desse estágio é basicamente uma fonte de alimentação chaveada CC-CC com transformador de alta frequência e circuito de controle PWM;
2. O segundo estágio é composto por um circuito de controle PWM e uma ponte H, ou ponte completa que transforma essa alta tensão contínua gerada pelo primeiro estágio em uma tensão alternada de formato retangular. Esta tensão alternada é fornecida pela ponte H diretamente para as cargas.

D.2 Inversores SPWM com Saída em Onda Senoidal

D.2.1 Características Gerais dos Inversores SPWM

Os inversores que usam a tecnologia de Modulação por Largura de Pulso Senoidal (SPWM – Sinusoidal Pulse Width Modulation), possuem saída senoidal pura e podem alimentar qualquer tipo de dispositivo da rede de corrente alternada. Atualmente, esta é a tecnologia mais usada para gerar sinal senoidal de potência em inversores com alta eficiência. Principalmente nos últimos vinte e cinco anos, a implementação deste recurso de conversão de energia tem sido aperfeiçoado até chegar aos níveis de eficiência atuais maiores que 95% para os inversores “String” integrados com conversores MPPT.

Esse tipo de inversor possui um sistema de controle bem mais complexo, resultando em custo mais alto do equipamento. Entretanto, tem sido cada vez mais usado em aplicações de geração solar, pois, permitem o uso para alimentar qualquer dispositivo projetado para a rede de corrente alternada, resultando em melhor aproveitamento da energia solar. Por esses motivos, tem sido usado em larga escala, o que está reduzindo significativamente o custo deste tipo de inversor, tornando-o acessível ao uso em usinas de qualquer capacidade. Atualmente, existem alguns pequenos fabricantes nacionais de inversores SPWM de um estágio, que são mais simples, robustos, possuem boa qualidade e custo acessível.

A Modulação SPWM usada em um equipamento de conversão de energia, no caso um inversor, tem por objetivo gerar uma saída senoidal de potência a partir de uma entrada de

corrente contínua com a máxima eficiência possível. Visando compreender a técnica de modulação SPWM, inicialmente serão apresentadas as características básicas da tecnologia de modulação de largura de pulso (PWM – Pulse Width Modulation), a qual é amplamente usada em todas as fontes de alimentação chaveadas que convertem corrente contínua para corrente contínua.

D.2.2 Modulação em Largura de Pulso (PWM)

A modulação por largura de pulso (PWM) é um método de controle destinado a proporcionar a transferência de energia com a maior eficiência possível em conversores de corrente contínua para corrente contínua. Neste método, um sinal retangular de alta frequência tem a relação entre o estado ativo e inativo controlada de modo a ajustar a tensão da saída do conversor, dependendo do modo de operação. O circuito de controle desse tipo de conversor é de segunda ordem, ou seja, efetua a operação de diferença entre uma tensão de referência e um sinal de realimentação, o qual é proporcional à saída. Esta função tem por objetivo efetuar a manutenção do valor correto da tensão na saída do conversor.

Um sinal exemplo de modulação PWM é mostrado na figura D.2, onde pode-se ver que os pulsos têm a mesma amplitude no eixo vertical. Isso significa que esses pulsos são gerados por dispositivos de chaveamento que operam somente em dois estados: aberto e fechado. Esta característica fornece a máxima eficiência de conversão de energia, pois, os dispositivos de chaveamento usados atualmente possuem baixíssima resistência no estado fechado (R_{on}), reduzindo muito as perdas de energia por dissipação causadas pela passagem de corrente (Potência dissipada em Watts: $P = R_{on}I^2$).

Como pode ser visto na figura D.2, os pulsos se repetem na frequência de operação do conversor, onde os três pulsos mais à esquerda são mais estreitos e os três pulsos mais à direita são mais largos. Na figura D.2 também são mostradas duas linhas tracejadas: 1) Uma linha tracejada mais baixa para os três pulsos estreitos à esquerda, indicando que esses pulsos geram uma tensão média mais baixa; 2) Uma linha tracejada mais alta para os três pulsos largos à direita indicando que esses pulsos geram uma tensão média mais alta.

Portanto, quanto maior a largura do pulso dentro do período, maior a tensão média na saída e maior a energia transferida. De modo inverso, quanto mais estreito é o pulso dentro do período, menor é a tensão de saída e menor a energia transferida. A conversão dos pulsos para a tensão média contínua na saída é feita através de filtro LC (Indutor e Capacitor), dimensionados de acordo com a frequência de operação e corrente de saída.

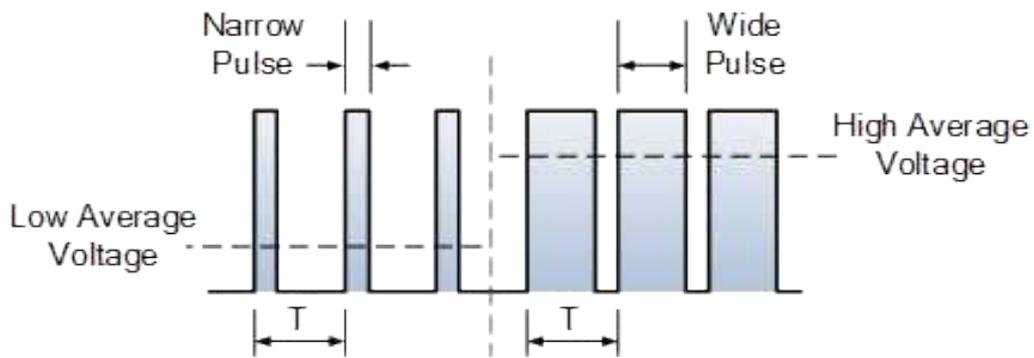


Figura D.2 - Forma de Onda da Modulação de Largura de Pulso

D.2.3 Modulação em Largura de Pulso Senoidal (SPWM)

A tecnologia de modulação por largura de pulso senoidal é um tipo de modulação de largura de pulso onde a referência de tensão é um sinal senoidal ao invés de uma tensão contínua. Isto resulta em uma tensão média de saída que varia de modo senoidal em uma frequência determinada. Nesse tipo de sistema, a frequência principal do sinal SPWM deve ser um múltiplo da frequência fundamental da onda senoidal. Assim, quanto maior for a relação entre a frequência principal e a frequência da onda senoidal, mais eficiente e menos dispendioso é o procedimento de filtragem da portadora e mais puro o sinal fundamental senoidal de saída.

Os circuitos integrados dedicados usados atualmente, trabalham com relação de quatrocentos para um (24KHz : 60Hz), o que permite o uso de indutores e capacitores menores para implementar um filtro que proporciona alta rejeição da frequência principal do sinal SPWM. A maior rejeição e conseqüente maior atenuação da frequência principal, fornece um sinal senoidal mais puro na saída de potência.

A Figura D.3 apresenta um exemplo de sinal bipolar que usa modulação SPWM. O termo bipolar significa que o sinal assume valores positivos e negativos em torno de uma referência, como é usado em um sinal de alimentação de corrente alternada. Neste exemplo, a frequência principal do sinal SPWM é apenas seis vezes a frequência fundamental para ilustração.

Pode-se observar na Figura D.3: quando a largura do pulso da frequência fundamental é menor, corresponde a uma tensão média menor, que implica em menor amplitude da onda senoidal da frequência fundamental.

De forma inversa, quando a largura do pulso é maior, corresponde a uma tensão média maior, que implica em maior amplitude da onda senoidal da frequência fundamental. Pode-se ver também na figura D.3, que a amplitude de tensão do pulso da frequência principal (V) é maior

que a amplitude máxima da onda senoidal. Isto é feito para evitar o uso de 100% da largura do pulso com o objetivo de manter uma margem de segurança para compensar quedas de tensão, as quais são causadas principalmente por aumento no fluxo de corrente.

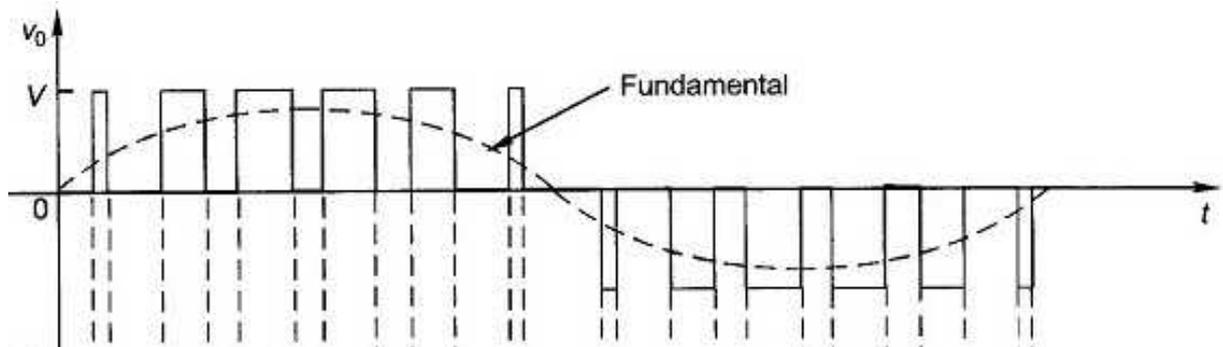


Figura D.3 - Forma de Onda da Modulação de Largura de Pulso Senoidal

O diagrama de inversor SPWM mostrado na figura D.4 a seguir, é chamado de básico ou direto, porque faz a conversão da tensão contínua de entrada diretamente para o sinal SPWM, o qual é filtrado para fornecer a saída senoidal de potência. O inversor SPWM básico ou direto possui a maior eficiência possível pela sua simplicidade. Na verdade, este circuito faz parte de todos os tipos de inversor SPWM, que são:

1. O Inversor direto;
2. O Inversor de um estágio com transformador de núcleo de ferro;
3. O Inversor de dois estágios.

Observação: nas aplicações on-grid, onde a saída do inversor é conectada diretamente à rede do consumidor, existem circuitos de controle de injeção de corrente e de mecanismos de segurança entre a saída do inversor e a rede que não estão mostrados na figura D.4.

No diagrama mostrado na figura D.4, os valores mínimos da tensão CC que é chaveada pela ponte H para gerar o sinal SPWM, são mostrados acima do barramento positivo na parte de cima do diagrama. Pode-se ver que este valor é aproximadamente dez por cento maior que o valor de pico da onda senoidal. O valor máximo desta tensão é definido pelo projetista de acordo com a mínima largura de pulso permitida. Este valor depende do valor da indutância do indutor de filtro. A Figura D.4 apresenta um diagrama de um inversor SPWM básico:

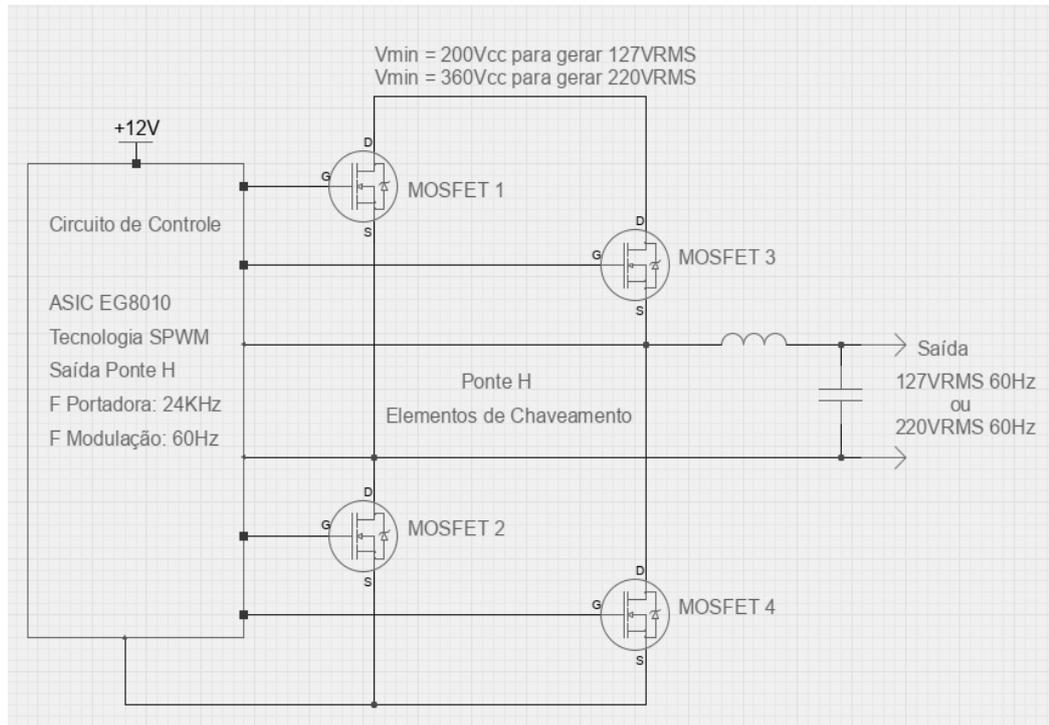


Figura D.4 - Diagrama de um Inversor SPWM Básico

O inversor direto normalmente é usado em conjunto com um conversor MPPT, sendo conectado diretamente em sua saída. Um conversor MPPT recebe em sua entrada um conjunto de painéis fotovoltaicos e procura o ponto de máxima potência ao demandar corrente destes painéis. O conjunto Conversor MPPT + Inversor Direto é chamado de “Inversor String”, pois é preciso conectar uma certa quantidade de painéis fotovoltaicos em série na entrada do conversor MPPT para que seja possível a este gerar a tensão contínua mínima a ser aplicada na entrada do inversor básico.

D.2.4 Características do Módulo EGS002

A implementação de inversores com tecnologia SPWM até alguns anos atrás era um processo trabalhoso e dispendioso, pois dependia do uso de um microcontrolador de boa performance e desenvolvimento de software. Isto era necessário para gerar os sinais de ativação dos elementos de chaveamento da ponte H e fazer a variação da largura de pulso com modulação senoidal.

Mais recentemente, com o crescente uso dos sistemas geração solar, ocorreu um significativo aumento na demanda de inversores com saída senoidal, fato que viabilizou o desenvolvimento de um Circuito Integrado de Aplicação Específica (ASIC – Application Specific Integrated Circuit), o EG8010 pela empresa EG Microelectronics Corporation. Este circuito integrado demanda alguns componentes auxiliares para implementar um circuito de controle completo

de um inversor de tecnologia SPWM. O fabricante fornece um módulo específico com o EG8010 mais os componentes auxiliares com o objetivo de facilitar o projeto e a montagem de inversores SPWM por integradores. A Figura D.5 apresenta o Módulo EGS002¹²:

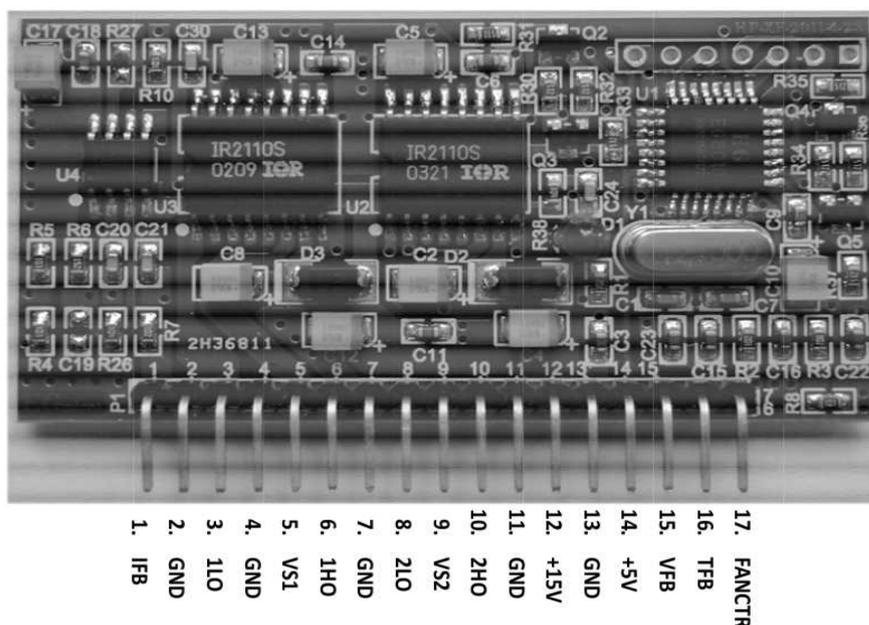


Figura D.5 - Módulo Controlador SPWM EGS002 com o CI EG8010

O circuito integrado EG8010 tem como função principal gerar todos os sinais necessários para controlar os elementos de chaveamento da ponte H. O ajuste da largura de pulso da frequência fundamental, a qual implica no ajuste da tensão de saída, é feito através da comparação de uma tensão de referência com o sinal de realimentação obtido da tensão CA após o filtro de saída. Além das funções principais, este circuito integrado permite configurar a geração de sinais senoidais com diversas frequências padronizadas e possui as proteções de temperatura, sobrecorrente, subtensão e sobretensão.

Objetivando facilitar a implementação dos equipamentos, foi desenvolvido o módulo EGS002, mostrado na figura D.5, que contém o circuito integrado EG8010 e componentes auxiliares. Este módulo tem os circuitos de interface que permitem a conexão direta aos terminais Gate dos MOSFETs da ponte H. Além disso, este módulo foi projetado para ser instalado em uma base de pinos, facilitando muito o desenvolvimento das placas de potência dos inversores SPWM de baixo custo, já que o encapsulamento do circuito integrado EG8010 é do tipo de montagem em superfície (SMD).

¹² EGMicro, 2014

D.2.5 Inversor de um Estágio com Transformador de Núcleo de Ferro

O Inversor SPWM de um estágio com transformador de núcleo de ferro é usado em aplicações onde é necessária robustez aliada ao máximo de simplicidade e economia. Esta arquitetura é usada atualmente por pequenos fabricantes e por usuários que fazem montagens artesanais. A Figura D.6 apresenta o diagrama simplificado de um inversor SPWM de um estágio:

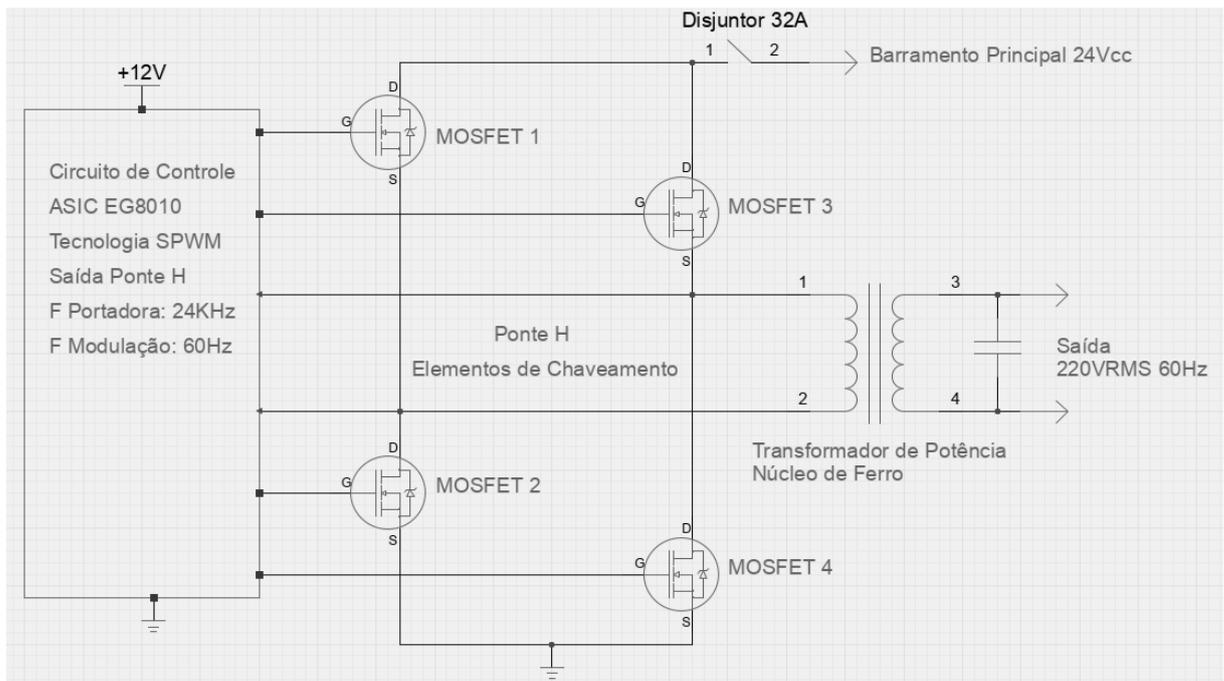


Figura D.6 - Diagrama Simplificado de um Inversor SPWM de um Estágio

A montagem desse tipo de inversor por pequenos fabricantes e por montadores artesanais é possível porque os procedimentos de montagem, verificação e teste são bem simplificados neste tipo de equipamento. Além disso, uma montagem cuidadosa, respeitando principalmente considerações de compatibilidade eletromagnética, circulação de corrente na entrada de corrente contínua, dissipação de calor, e proteções permitem obter um equipamento robusto e confiável de baixo custo.

Como pode ser visto na Figura D.6, neste tipo de equipamento, o sinal SPWM de potência, gerado pela ponte H, é aplicado no enrolamento primário do transformador de núcleo de ferro. Desta forma, o sinal de saída de potência senoidal é obtido no enrolamento secundário do transformador, o qual tem um capacitor conectado em paralelo para compor o filtro de alta frequência. A seguir são apresentadas as características do inversor com transformador de núcleo de ferro:

1. O transformador com núcleo de ferro proporciona elevação da tensão contínua de entrada para a tensão alternada de saída. A maioria dos sistemas de geração solar off-grid de baixo custo, que usa banco de baterias, tem barramento com tensão nominal de 24Vcc. Alguns sistemas de maior potência usam tensão de barramento de 48Vcc. Assim, a elevação da tensão é necessária para fornecer a tensão alternada de 127Vca ou 220Vca;
2. O transformador com núcleo de ferro fornece a isolação elétrica entre a entrada de corrente contínua e a saída de corrente alternada. Esta característica é essencial, mesmo em sistemas de baixo custo, pois, permite o uso eficiente dos recursos de proteção contra surtos de tensão. A isolação elétrica permite a conexão do negativo da tensão contínua do barramento principal ao terra de proteção, recurso necessário para fornecer proteção para os controladores de carga e para os painéis fotovoltaicos contra descargas atmosféricas;
3. O transformador com núcleo de ferro fornece a indutância necessária para compor o filtro LC de saída, dispensando a instalação deste componente. Desta forma, basta conectar um capacitor na saída do transformador para realizar a filtragem da alta frequência do sinal SPWM. Esta característica é importante, porque o indutor de potência é um componente de difícil aquisição por pequenos fabricantes e montadores artesanais;
4. O transformador com núcleo de ferro não permite que surtos de tensão, que porventura ocorram na rede de corrente alternada conectada na saída, danifiquem os elementos de chaveamento (MOSFETs) ou o Módulo de Controle SPWM. Estes surtos de tensão, normalmente causados por descargas atmosféricas, são absorvidos pelos elementos de proteção da rede, normalmente varistores de óxido metálico, não se propagando para o enrolamento de entrada e componentes
5. A indutância do transformador com núcleo de ferro limita o tempo de subida da corrente que circula nos elementos de chaveamento caso ocorra curto-circuito na saída. Esta característica é muito importante, porque permite que seja usado disjuntor termomagnético do tipo quick-lag como proteção contra curto-circuito. Este tipo de proteção tem custo bem menor que uma proteção eletrônica equivalente.

D.2.6 Inversor SPWM de dois Estágios

Atualmente, todos os principais fabricantes de inversores SPWM para aplicações off-grid e também de micro inversores usam a arquitetura de dois estágios, pois proporcionam eficiência de conversão mais alta, menor tamanho e peso do equipamento em comparação com o inversor SPWM de um estágio.

Este tipo de inversor possui um conversor que eleva a tensão contínua de entrada para o valor necessário de tensão contínua para gerar o sinal de corrente alternada de saída. Essa alta tensão contínua, é então, aplicada em uma ponte H que gera o sinal SPWM de saída. Por fim, o sinal SPWM é aplicado a um filtro composto por um indutor (L) e um capacitor (C) para retirar o sinal de alta frequência e restar somente o sinal de 60Hz na tensão da rede (127VCA ou 220VCA).

O bloco mais à esquerda na figura D.6 corresponde a um conversor de tensão contínua para tensão contínua, o qual fornece a tensão necessária para que a ponte H possa gerar o sinal de saída SPWM. Como pode ser visto na figura D.6, o valor mínimo para gerar a tensão de saída padrão de 220VCA é de 360Vcc. No caso de usinas off-grid, esta tensão é gerada a partir da tensão contínua do barramento CC, nos valores nominais de 12Vcc, 24Vcc ou 48Vcc. Em inversores de melhor qualidade, o conversor CC-CC fornece a tensão contínua estabilizada para a ponte H do gerador do sinal SPWM. Esta característica melhora a regulação da tensão CA de saída do inversor. A Figura D.7 apresenta o diagrama simplificado de um inversor SPWM de dois estágios:

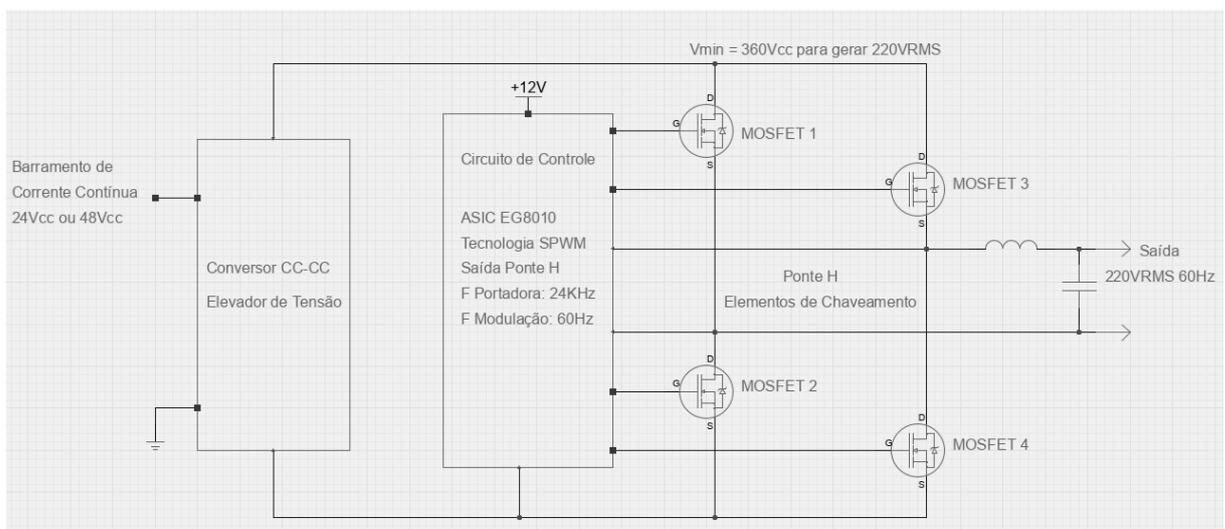


Figura D.7 - Diagrama Simplificado de um Inversor SPWM de dois Estágios

Este tipo de inversor é usado também nos equipamentos denominados “Micro-Inversores” usados em sistemas on-grid, os quais possuem um conversor MPPT integrado, destinado a receber até dois painéis fotovoltaicos. Neste caso, dois painéis fotovoltaicos em série fornecem um valor de tensão que varia de 70 a 92V_{cc}, valor insuficiente para alimentar diretamente a ponte H do gerador de sinal SPWM. Por isso, é preciso o conversor CC-CC para elevar a tensão de saída do conversor MPPT para o valor de tensão necessária para viabilizar o gerador SPWM.

Anexo E – Sistemas de Controle Fuzzy

E.1 Características Gerais

A chamada Lógica Fuzzy apareceu nos meios científicos através do artigo Fuzzy Sets publicado no Journal Information and Control em 1965 por Lofti Asker Zadeh. O nome Fuzzy que possui os significados em português: difuso ou impreciso foi dado a um tipo de lógica, a qual é derivada dos conjuntos de mesmo nome. A lógica fuzzy é destinada ao tratamento de informações ambíguas, incompletas ou imprecisas, fornecendo resultados úteis para a tomada de decisões em diversas áreas (ABADIA 2015). Em contraste com a lógica convencional, onde só existem os estados verdadeiro e falso para uma proposição, a lógica fuzzy permite valores entre 0 e 1, proporcionando um tratamento aproximado dos dados. Este tipo de tratamento de dados permite reduzir a complexidade dos algoritmos de controle de processos, o que resulta na viabilização do uso de controladores de baixo custo.

Nos sistemas lógicos binários, o resultado de uma proposição somente pode ter dois valores: verdadeiro (1) ou falso (0). Entretanto, na lógica Fuzzy, uma proposição varia em grau de verdade de 0 a 1, o que leva a ser parcialmente verdadeira ou parcialmente falsa. A outra característica fundamental da lógica fuzzy é a expressão em linguagem humana dos valores verdade em relação à proposição. Neste caso, cada termo linguístico é interpretado como um subconjunto fuzzy do intervalo unitário ou completo.

Os predicados da lógica clássica são exatos, o que significa que não admitem interpretações. Pode-se citar como exemplo o resultado da proposição ($a > b$). Neste caso o resultado sempre é verdadeiro para valores de a maiores que b e falso para valores de a menores ou iguais a b . No caso da lógica fuzzy, pode-se ter as expressões linguísticas alto e baixo para o resultado de uma proposição, ou seja, os predicados são fuzzy ou difusos. A lógica fuzzy admite diversos qualificadores tais como muito, pouco, mais, menos. Assim, exemplos de proposições podem ser:

- **A temperatura está alta;**
- **A temperatura está muito alta;**
- **A temperatura está baixa.**

O mesmo caso se aplica aos quantificadores. Podem ser usados diversos tais como: poucos, muitos, frequentemente, etc.

E.1.2 Características dos Conjuntos Fuzzy

Na lógica fuzzy, o grau de verdade ou a intensidade de uma proposição é dado pela chamada Função de Pertinência, a qual caracteriza totalmente um conjunto fuzzy. Formalmente, considerando o Universo U e um conjunto F , uma função de pertinência μ_F mapeia os elementos do conjunto F para um intervalo de valores de zero a um $[0,1]$: $\mu_{F:U} \rightarrow [0,1]$. Um conjunto fuzzy F em U é expresso como um conjunto de pares ordenados como mostra a expressão E.1 a seguir:

$$F = \{ (y, \mu_F(y)) \mid y \in U \} \quad (E.1)$$

Onde: F = Conjunto Fuzzy, $\mu_F(y)$ = Função de Pertinência, U = Universo, y = elemento de U . Assim, a função de pertinência fornece como resultado um número real no intervalo de zero a um que representa o grau de pertinência de um elemento y do universo ao conjunto fuzzy F . O grau de pertinência fornece, desta forma, um valor que indica o quanto o elemento y pertence ao conjunto F .

- $\mu_F(y) = 0$ indica que y não é compatível com o conjunto F ;
- $0 < \mu_F(y) < 1$ indica que y é parcialmente compatível com o conjunto F ;
- $\mu_F(y) = 1$ indica que x é totalmente compatível com o conjunto F .

E.1.3 Características das Funções de Pertinência

A função de pertinência fornece um valor numérico que indica a intensidade com que um elemento pertence a um conjunto fuzzy. Normalmente são usadas funções não probabilísticas monotonicamente crescentes, decrescentes ou subdivididas em parte crescente e parte decrescente. Os parâmetros da função são definidos por conhecimento especialista. As funções mais usadas são:

1. Função Triangular;
2. Função Trapezoidal;
3. Função Gaussiana;
4. Função Sino Generalizada.

Estas funções são apresentadas na Figura E.1:

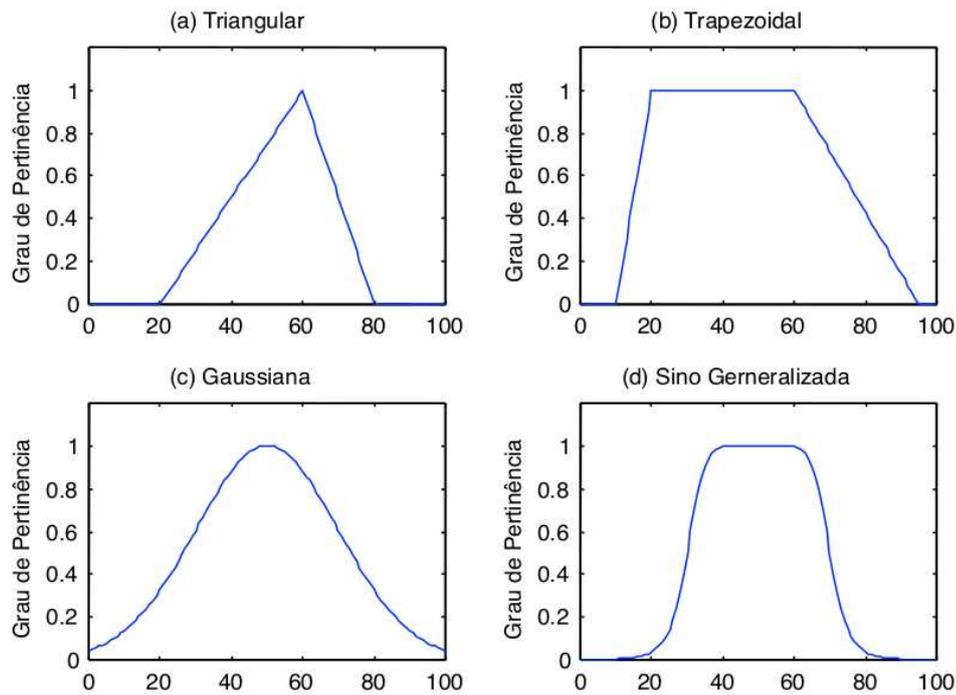


Figura E.1: Funções de Pertinência Contínuas

A função de pertinência pode ser definida também através do mapeamento individual de cada ponto.

E.1.4 Sistema de Controle Fuzzy Baseado em Regras

Um sistema de controle em tempo real pode ser modelado através da lógica fuzzy, mas ao invés de somente modelar o sistema através das suas próprias características, a definição das regras para as ações de controle é feita baseada em conhecimento especialista. Isto significa que é possível montar um controlador usando os procedimentos de atuação de um operador ou técnico sobre o processo. Assim, é possível implementar um sistema de controle usando regras do tipo se - então que são aplicadas em variáveis fuzzy, no “mundo fuzzy”, e depois retornando ao mundo dos sinais através das técnicas de defuzificação (ROSS, 2010).

O primeiro passo na montagem de um sistema de controle fuzzy é a definição dos parâmetros do sistema a ser modelado. Alguns parâmetros são fixos e outros podem ser modificados para melhor performance. Os seguintes parâmetros podem ser considerados: 1. Quantidade de Variáveis de Saída; 2. Quantidade de Variáveis de Entrada; 3. Ajustes sobre os dados de entrada (somas, multiplicações, etc); 4. Variáveis Linguísticas; 5. Funções de Pertinência; 6. Contexto das Variáveis; 7. Parâmetros das Funções de Pertinência; 8. Parâmetros de condicionamento e ajuste das entradas e saídas (ganho e offset).

E.1.5 Fuzzificação

A fuzzificação é o processo de tornar difusa (fuzzy) uma grandeza exata (crisp). Para fazer isso, constata-se que grandezas aparentemente exatas carregam uma incerteza considerável quando analisadas do ponto de vista do raciocínio humano. Assim, os conjuntos fuzzy permitem propriedades imprecisas de pertinência ao invés dos conjuntos clássicos que contém objetos que satisfazem propriedades precisas de pertinência (ROSS, 2010). Os passos principais do processo de fuzzificação são:

1. Análise do Problema;
2. Definição das Variáveis;
3. Definição das Funções de Pertinência.;
4. Criação das Regiões.

E.1.6 Base de Conhecimento

A base de conhecimento contém o conhecimento especialista a respeito do processo. A base de conhecimento possui dois componentes principais:

1. Uma base de dados que armazena as definições necessárias sobre os parâmetros de conversão analógica/digital, os parâmetros referentes às normalizações, os qualificadores de cada variável fuzzy de entrada e saída e as definições de cada função de pertinência associada a cada qualificador ou conjunto;
2. Um conjunto de regras de controle fuzzy do tipo se – então que caracterizam os objetivos e a política de controle e suas metas denominado Base de Regras.

A base de regras é um conjunto de regras fuzzy que efetua o relacionamento entre os termos linguísticos das variáveis de entrada e saída definidas para o sistema. Cada regra presente na base deve satisfazer a seguinte estrutura: SE a está A_i ENTÃO b está B_i , onde a e b são as variáveis fuzzy, tais como Temperatura e Tensão, e A_i e B_i são conjuntos fuzzy tais como Alta e Baixa. As variáveis e os conjuntos são termos linguísticos correspondentes às variáveis de entrada e saída do sistema. Assim, a definição da base de regras é o procedimento que reúne as informações sobre o processo, as quais devem resultar nas ações desejadas. Quando mais regras forem inseridas na base respeitando a consistência das informações, mais precisa se torna a ação de saída. As ações de saída, neste caso, são os valores das variáveis, que podem ser numéricos e traduzidos nos conjuntos fuzzy de saída (NEVES 2019).

A definição das regras de controle fuzzy do tipo se – então não é simples como no caso da lógica clássica, porque a lógica fuzzy permite gradações na intensidade das assertivas. Assim, o resultado da aplicação de uma regra se – então fuzzy são valores de pertinência a alguns conjuntos. O valor da intensidade ou força da regra é obtido através de uma combinação das intensidades das variáveis fuzzy de entrada.

E.1.7 Sistema de Inferência Fuzzy

O Sistema de Inferência Fuzzy tem por função efetuar o processamento das variáveis fuzzy de entrada usando as regras se – então armazenadas na base de conhecimento, com o objetivo de inferir as ações de controle fuzzy. Este processamento inclui as operações fuzzy de *união* (*operador or*), *intersecção* (*operador and*) e *negação*. A condição SE constitui uma região fuzzy no espaço das variáveis de entrada, enquanto a ação ENTAO descreve uma conclusão, cuja inferência conduz a algum resultado. A figura E.2 mostra o diagrama de um Sistema de Inferência Fuzzy:

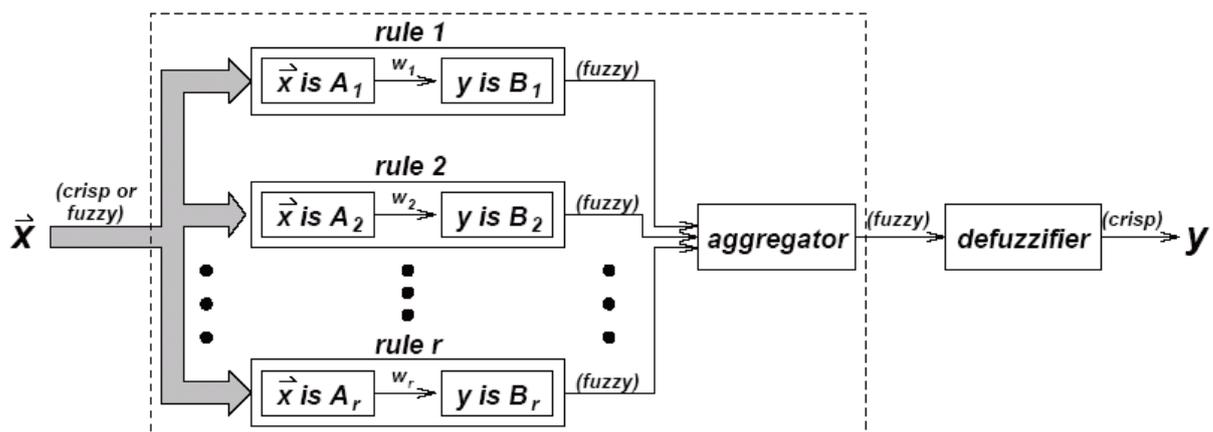


Figura E.2 - Diagrama do Sistema de Inferência Fuzzy

O Sistema de Inferência Fuzzy que segue o Modelo de Mamdani aplicado em um controle em tempo real realiza os seguintes procedimentos:

As variáveis de entrada medidas e calculadas são fuzzificadas, ou seja, são transformadas em valores normalizados no intervalo [0,1]. Estes valores representam a pertinência do valor da variável em conjuntos fuzzy específicos e se traduzem em expressões linguísticas;

Em cada regra fuzzy, uma ou mais entradas fuzzificadas são processadas para gerar um resultado chamado de Intensidade da Regra (Rule Strength);

1. A Intensidade da Regra é então combinada com a Função de Pertinência da variável fuzzy de saída gerando o Resultado da Regra;

2. Em seguida é feita a combinação todos os Resultados das Regras para obtenção da Distribuição de Saída Fuzzy;
3. O passo final é a defuzzificação da Distribuição de Saída Fuzzy.

E.1.8 Defuzzificação

A interface de defuzzificação transforma as ações de controle fuzzy resultantes da aplicação dos passos da Máquina de Inferência Fuzzy em ações de controle não fuzzy. O passo seguinte da defuzzificação é a transformação dos valores normalizados para a escala usada pelo sistema de controle.

Após os procedimentos do Sistema de Inferência Fuzzy, é necessária a determinação de uma ação de controle não fuzzy que melhor represente a decisão fuzzy, para ser efetivamente enviada ao controle. O procedimento de defuzzificação não possui um critério definido, ficando a cargo do projetista escolher um procedimento que seja mais adequado ao processo em questão.

E.1.9 Apresentação e Tratamento das Informações

Os resultados da aplicação das regras de inferência fuzzy para uma variável de saída, contém basicamente o valor resultante da defuzzificação, e os valores de intensidade ou de pertinência da variável aos conjuntos fuzzy. No exemplo apresentado, o resultado da defuzzificação fornece um valor em porcentagem para a variável de saída ventilação forçada. Este valor em porcentagem pode alimentar um sistema de controle proporcional de velocidade dos ventiladores para ajustar o nível de ventilação continuamente.

O valor resultante da defuzzificação de uma variável de saída pode ser apresentado diretamente na tela de um software supervisorio ao usuário informando as condições de operação. Esta apresentação pode ser feita mostrando o valor numérico, ou pode ser feita graficamente através de ponteiros, ou barras de intensidade. A principal utilidade dos sistemas de inferência fuzzy está no fato dos resultados serem aproximados à linguagem humana, facilitando a análise pelo usuário do sistema. Portanto, é muito mais fácil a um operador tomar uma decisão ao ver o resultado de uma variável chamada Saude do que avaliar a condição observando diversas medidas e estados. Se o valor desta variável é apresentado de forma gráfica pelo software supervisorio, a tarefa é facilitada mais ainda.

Outro tipo de tratamento e uso das informações referentes a uma variável de saída, é feito através da aplicação de regras do tipo se-então aos resultados do procedimento de inferência fuzzy. Um exemplo simples é a comparação dos valores de pertinência aos conjuntos fuzzy para realização de alguma ação. No exemplo mostrado no item 2.15.7 o valor de pertinência da variável de saída Ventilacao ao conjunto fuzzy Media é maior que os valores e pertinência aos conjuntos Alta e Baixa. Assim, a simples comparação de valores de pertinência pode gerar uma informação digital para acender um indicador mostrando que o nível da intensidade da ventilação é Media, por exemplo.

E.2 Características da Linguagem FCL

Neste trabalho, as ferramentas de análise destinadas aos procedimentos de monitoramento de condição são desenvolvidos na linguagem FCL (Fuzzy Control Language – Linguagem de Controle Fuzzy), linguagem própria para programação de controle fuzzy baseado em regras. As especificações para a sintaxe FCL são descritas na Norma IEC 61131-7 publicada pela International Electrotechnical Commission (IEC).

O pacote jFuzzyLogic_v3.0 instalado no Ambiente de Desenvolvimento Eclipse Java permite fazer programas em linguagem Java que efetuam a interpretação e a execução dos comandos presentes em um arquivo do tipo FCL. Assim, a declaração das variáveis, os procedimentos de fuzzificação, defuzzificação e os procedimentos da inferência fuzzy podem ser escritos em linguagem própria e executados por um programa Java padrão através do uso dos objetos da classe FIS (Fuzzy Inference System).

A linguagem Java possui os recursos para realizar as seguintes funções principais: 1) Efetuar a comunicação com os controladores para receber as informações de supervisão e enviar os comandos; 2) Efetuar a comunicação com o usuário, apresentando de modo organizado as informações necessárias para a supervisão e controle local; 3) Mostrar graficamente as funções de pertinência das variáveis de entrada e dos resultados das operações de inferência fuzzy, facilitando a análise e a otimização dos parâmetros.

Os comandos da linguagem FCL devem ser escritos em um arquivo tipo texto, que pode ser editado por qualquer editor de texto comum. Por uma questão de padronização a extensão do arquivo deve ter a extensão fcl.

E.2.2 Bloco Funcional

Um conjunto de comandos referente a um processo específico é denominado Bloco Funcional. Cada bloco funcional deve começar com o comando `FUNCTION_BLOCK` seguido pelo nome do bloco, e terminar com o comando `END_FUNCTION_BLOCK`. Pode existir mais de um bloco funcional em um mesmo arquivo FCL.

Dentro de cada Bloco Funcional devem ser declaradas as variáveis de entrada, as variáveis de saída, os parâmetros de fuzzificação das variáveis de entrada, os parâmetros de defuzzificação das variáveis de saída, o método de defuzzificação e as regras fuzzy.

E.2.3 Declaração das Variáveis de Entrada

As variáveis de entrada são declaradas dentro da seção que inicia com o comando `VAR_INPUT` e termina com o comando `END_VAR`. O nome destas variáveis será usado na fuzzificação. No exemplo mostrado no Algoritmo E.1, é definida uma variável do tipo `REAL`, ou seja, número ponto flutuante:

Algoritmo E.1 Declaração de uma Variável de Entrada em FCL

```
VAR_INPUT  
  TemperaturaDriver : REAL;  
END_VAR
```

E.2.4 Declaração das Variáveis de Saída

As variáveis de saída são declaradas dentro da seção que inicia com o comando `VAR_OUTPUT` e termina com o comando `END_VAR`. O nome destas variáveis será usado na defuzzificação. No exemplo apresentado no Algoritmo E.2, é declarada uma variável do tipo `REAL`, ou seja, número ponto flutuante, Ventilacao:

Algoritmo E.2 Declaração de uma Variável de Saída em FCL

```
VAR_OUTPUT  
  Ventilacao : REAL;  
END_VAR
```

E.2.5 Fuzzificação das Variáveis de Entrada

Deve ser feita uma seção de fuzzificação para cada variável de entrada. Cada seção inicia com o comando `FUZZIFY` seguido pelo nome da variável e termina com o comando `END_FUZZIFY`. Dentro da seção `FUZZIFY` devem ser declarados os conjunto fuzzy para cada variável após o comando `TERM`. Após a declaração de cada qualificador deve ser

definida a função de pertinência. Neste exemplo as funções de pertinência dos tipos triangular e trapezoidal são definidas através dos seus pontos, que devem ser declarados por pares x,y entre parênteses. O valor de y, que reflete a intensidade da pertinência, é normalizado, ou seja, deve estar sempre na faixa de 0 a 1.

No exemplo apresentado no Algoritmo E.3, a variável Temperatura é fuzzificada com três conjuntos fuzzy (Normal, Alta, MuitoAlta) e três funções de pertinência do tipo triangular/trapezoidal associadas. O valor de x refere-se ao valor da temperatura em graus Celsius.

Algoritmo E.3 Fuzzificação de uma Variável de Entrada em FCL

FUZZIFY Temperatura

TERM Normal := (30, 1) (40, 1) (45, 0);

TERM Alta := (40, 0) (45, 1) (50, 1) (55, 0);

TERM MuitoAlta := (50, 0) (55, 1) (60, 1);

END_FUZZIFY

A Figura E.3 apresenta mostra graficamente as três funções de pertinência associadas à variável Temperatura definidas na seção FUZZIFY Temperatura. Este gráfico é gerado pelo programa Java com o comando `fis.getVariable("Temperatura").chart(true)`. A área correspondente ao qualificador ou conjunto **Normal** está mais à esquerda, em formato trapezoidal. A área central corresponde ao conjunto fuzzy **Alta**. A área correspondente ao qualificador ou conjunto **MuitoAlta** está mais à direita.

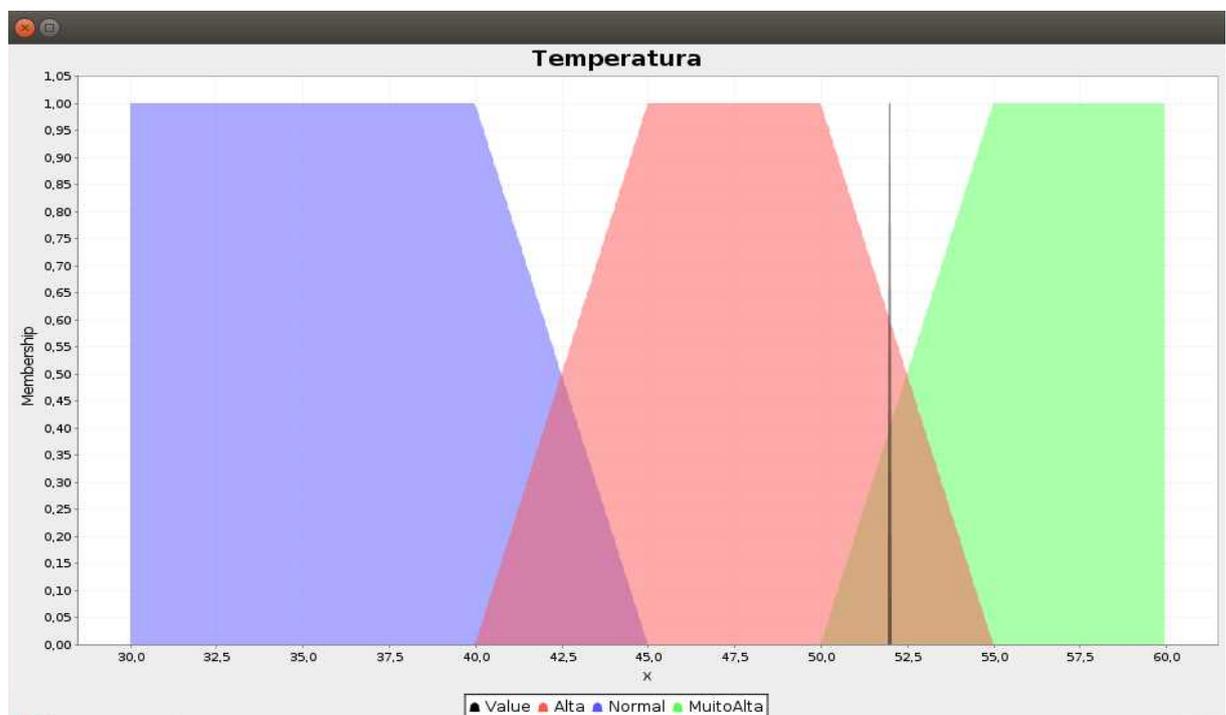


Figura E.3 - Funções de Pertinência associadas à variável Temperatura

O traço vertical preto na figura E.3 indica o valor atual que foi atribuído à variável Temperatura (52°C) através do comando `fis.setVariable("Temperatura", 52.0)` no programa Java. A seguir são apresentados os atributos referentes à variável Temperatura obtidos com o comando Java `System.out.println(fis.getVariable(Temperatura));`

Temperatura :

Value: 52.0

Term: **Alta** 0.6 PieceWiseLinear : (40.0, 0.0), (45.0, 1.0), (50.0, 1.0), (55.0, 0.0);

Term: **Normal** 0.0 PieceWiseLinear : (30.0, 1.0), (40.0, 1.0), (45.0, 0.0);

Term: **MuitoAlta** 0.4 PieceWiseLinear : (50.0, 0.0), (55.0, 1.0), (60.0, 1.0) ;

Pode-se ver que o valor definido para a variável Temperatura = 52°C, corresponde a um valor de pertinência de 0,6 ao conjunto **Alta**, um valor de pertinência de 0,0 ao conjunto Normal, e um valor de pertinência de 0,4 ao conjunto **MuitoAlta**.

E.2.6 Defuzzificação das Variáveis de Saída

Deve ser feita uma seção de defuzzificação para cada variável de saída. Cada seção inicia com o comando DEFUZZIFY seguido pelo nome da variável e termina com o comando END_DEFUZZIFY. Dentro da seção DEFUZZIFY devem ser declarados cada qualificador da variável de saída após o comando TERM.

Após a declaração de cada qualificador deve ser definida a função de pertinência. Neste caso, as funções de pertinência são definidas através dos seus pontos, que devem ser declarados por pares x,y entre parênteses. O valor de y, que reflete a intensidade da pertinência, é normalizado, ou seja, deve estar sempre na faixa de 0 a 1.

Após a declaração dos qualificadores da variável e as funções de pertinência, deve ser definido o método de defuzzificação. O método mais comum é o Centro de Gravidade (COG). A seguir no algoritmo E.4 é apresentado um exemplo de defuzzificação da variável de saída Ventilacao. O comando DEFAULT faz a variável de saída Ventilacao = 0 caso nenhuma regra seja ativada.

No exemplo apresentado no Algoritmo E.4, a variável Ventilacao é fuzzificada com três qualificadores ou conjuntos (Baixa, Media, Alta) e três funções de pertinência do tipo triangular/trapezoidal associadas. O valor de x refere-se à intensidade da ventilação na escala de 0 a 100%.

Algoritmo E.4 Defuzzificação de uma Variável de Saída em FCL

```
DEFUZZIFY Ventilacao
  TERM Baixa := (0, 1) (25, 1) (50, 0);
  TERM Media := (25, 0) (50, 1) (75, 0);
  TERM Alta := (50, 0) (50, 1) (100, 0);
  METHOD : COG;
  DEFAULT := 0;
END_DEFUZZIFY
```

A Figura E.4 mostra graficamente as três funções de pertinência associadas à variável de saída Ventilacao definida na seção DEFUZZIFY Ventilacao. Este gráfico é gerado pelo programa Java com o comando `fis.getVariable("Ventilacao").chart(true)`. A área correspondente ao conjunto fuzzy **Baixa** está mais à esquerda. A área central corresponde ao conjunto fuzzy **Media**. A área correspondente ao conjunto **Alta** está mais à direita.

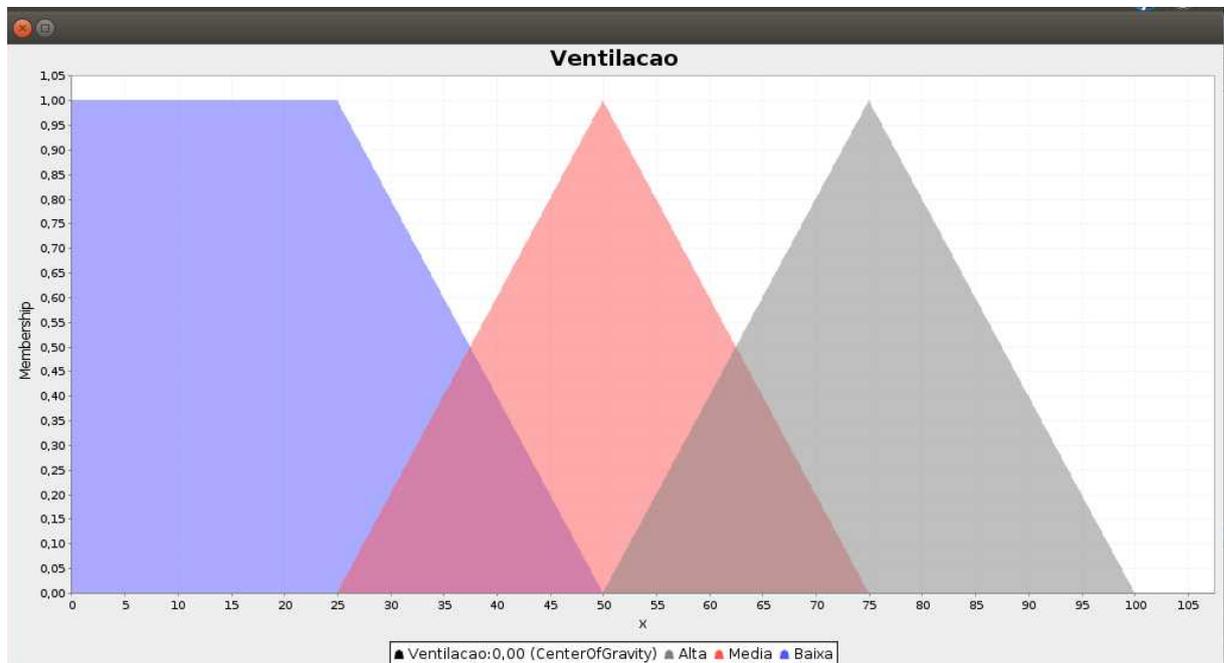


Figura E.4 - Funções de Pertinência associadas à variável Ventilacao

Observação: O gráfico mostrado na Figura E.4 foi gerado apenas para mostrar as funções de pertinência associadas à variável de saída Ventilacao. as regras da inferência fuzzy ainda não foram ativadas. Por isso, o valor da variável Ventilacao = 0,0.

E.2.7 Definição das Regras

Deve ser criada uma seção de regras iniciada com o comando RULEBLOCK seguido pelo nome da seção, e terminada com o comando END_RULEBLOCK. A precisão da operação da inferência fuzzy depende das regras, então é importante definir o máximo de situações

possíveis para que as variáveis de saída respondam de maneira satisfatória às condições das entradas.

No início da seção devem ser definidas as operações AND (operação lógica e), OR (operação lógica OU), ACT (ativação) e ACCU (acumulação).

Após definidos os modos de operação, devem ser definidas as regras do procedimento de inferência no formato IF (condição fuzzy) THEN (ação fuzzy). O exemplo do algoritmo E.5, mostrado a seguir, mostra a definição das operações AND para MIN (mínimo), ACT também para MIN (mínimo) e ACCU para MAX (máximo) e algumas regras fuzzy:

Algoritmo E.5 Exemplo de Regras em FCL

```
RULEBLOCK No1
  AND : MIN;
  ACT : MIN;
  ACCU : MAX;
  RULE 1 : IF Temperatura IS Normal THEN Ventilacao IS Baixa;
  RULE 2 : IF Temperatura IS Alta THEN Ventilacao IS Media;
  RULE 3 : IF Temperatura IS MuitoAlta THEN Ventilacao IS Alta;
END_RULEBLOCK
```

E.2.8 Resultado da Aplicação das Regras

A aplicação das regras de inferência fuzzy à variável de saída **Ventilacao**, após a defuzzificação, gerou uma “string” com o seguinte formato:

```
Ventilacao :
Defuzzifier : CenterOfGravity
Latest defuzzified value: 60.484
Default value: 0.0
Term: Alta 0.419 PieceWiseLinear: (50.0, 0.0), (75.0, 1.0), (100.0, 0.0);
Term: Media 0.581 PieceWiseLinear: (25.0, 0.0), (50.0, 1.0), (75.0, 0.0);
Term: Baixa 0.0 PieceWiseLinear: (0.0, 1.0), (25.0, 1.0), (50.0, 0.0);
```

A primeira linha apresenta o método usado para defuzzificação (CenterOfGravity ou Centro de Gravidade). A linha seguinte mostra o valor resultante da defuzzificação da variável. Este valor pertence à faixa de valores na escala horizontal onde foram definidos os conjuntos fuzzy. Neste caso, a faixa refere-se à intensidade da ventilação forçada que vai de 0 a 100%. Assim, o valor resultante para a intensidade de ventilação é igual a 60,484%. Os valores de pertinência aos conjuntos fuzzy Alta, Media e Baixa resultantes foram 0,419 , 0,581 e 0 respectivamente.

E.2.9 Uso de Arquivos FCL no Programa Java

O programa Java deve usar o pacote JFuzzyLogic3.0 ou equivalente para interpretar os comandos de um arquivo FCL. Esta biblioteca deve estar disponível para o projeto no Eclipse IDE. Para isso, no menu principal na opção Project – Properties – Java Build Path, deve-se adicionar o arquivo JfuzzyLogic_3.0.jar. Após a biblioteca estar disponível no Eclipse IDE, o programa Java deve ter o comando inicial na seção de importação de bibliotecas conforme mostra o Algoritmo E.6:

Algoritmo E.6 Comando para Importação de Bibliotecas JfuzzyLogic em Java

```
import net.sourceforge.jFuzzyLogic.FIS;
```

Já dentro do programa Java, na inicialização é preciso fazer a leitura do arquivo FCL. O procedimento padrão considerando a condição de erro é mostrado no algoritmo E.7 a seguir:

Algoritmo E.7 Leitura do Arquivo FCL em Java

```
String fileName = "/home/java/Fuzzy/Exemplo.fcl";
FIS fis = FIS.load(fileName,true);
    if( fis == null ) {
        System.err.println("Não foi possível ler o arquivo: " + fileName +
        "");
        return;
    }
```

A atribuição do valor das variáveis de entrada é feita usando os comandos mostrados abaixo. No exemplo mostrado no algoritmo E.8 a seguir, foi usado valor fixo para as variáveis:

Algoritmo E.8 Atribuição do valor das variáveis de entrada em Java

```
fis.setVariable("Temperatura", 52.0);
```

A execução dos comandos da linguagem FCL é feita através do comando mostrado no algoritmo E.9 a seguir:

Algoritmo E.9 Comando para execução dos comandos de um arquivo FCL em Java

```
fis.evaluate();
```

O resultado da aplicação da inferência Fuzzy na variável de saída **Ventilacao**, é obtido através do comando mostrado no algoritmo E.10 a seguir que retorna uma string:

Algoritmo E.10 Leitura de uma Variável de Saída em Java

```
fis.getVariable("Ventilacao").
```

Anexo F – Tipos Básicos de Usinas Solares

F.1 Usinas Solares do tipo “On Grid”

Os denominados Inversores “String” para uso em usinas “on-grid”, recebem diretamente a energia dos painéis fotovoltaicos e fornecem a corrente alternada para a rede do usuário. Um equipamento Inversor On-Grid, na verdade, é composto por dois equipamentos operando em conjunto:

1. O conversor MPPT, que recebe a saída do conjunto de painéis fotovoltaicos, tem por função obter a máxima potência através da leitura da tensão e o controle da corrente demandada dos painéis. O Conversor MPPT fornece em sua saída uma tensão contínua com valor determinado para ser usado pelo Equipamento Inversor;
2. O equipamento Inversor propriamente dito, converte a tensão contínua fornecida pelo conversor MPPT em corrente alternada a ser injetada na rede do usuário. A máxima potência que pode ser fornecida pelo inversor é informada pelo conversor MPPT.

Os equipamentos inversores usados nas usinas solares do tipo “on-grid” são projetados para ter a sua saída de corrente alternada ligada diretamente na rede elétrica do local. Este tipo de inversor funciona somente se houver tensão fornecida pela rede elétrica da concessionária, e possui as seguintes características fundamentais:

1. Ao ser energizado e receber a tensão da rede da concessionária, o equipamento inversor on-grid efetua a sincronização da frequência e fase da sua saída de corrente alternada com a frequência e a fase da energia fornecida pela rede da concessionária;
2. O equipamento inversor on-grid efetua o controle de fornecimento de corrente para a rede elétrica. Este mecanismo ajusta automaticamente o valor da corrente injetada na rede elétrica de acordo com a energia fornecida pelos painéis fotovoltaicos;
3. O controle de desconexão à rede caso ocorra subtensão. Assim, o inversor on-grid somente injeta corrente na rede se a tensão da rede da concessionária esteja presente e com o seu valor dentro dos limites estabelecidos.

Este tipo de inversor necessita de níveis de tensão contínua de entrada maiores que 200 Volts de corrente contínua para que possam gerar a tensão alternada padrão na saída de 127 Volts, ou de 360 Volts de corrente contínua para gerar a tensão alternada de 220 Volts. Portanto,

considerando que os maiores painéis disponíveis atualmente, que possuem setenta e duas células e fornecem tensão na faixa de 38 Volts com potência máxima, torna-se necessário a associação de vários painéis em série para que a tensão de entrada do inversor seja fornecida dentro dos valores permitidos.

Os sistemas de geração solar on-grid convencionais possuem pelo menos um conjunto de painéis fotovoltaicos conectados em série conectados a um inversor on-grid do tipo “string”. O termo String, em português, significa linha/corda, que simboliza uma linha de módulos fotovoltaicos ligados em série que se comportam como um único painel com tensão de saída mais alta. Alguns inversores permitem conectar apenas uma única linha e outros inversores permitem que duas ou mais linhas de painéis fotovoltaicos independentes sejam conectadas num mesmo inversor (PINHO, 2014). Os painéis conectados em série, nesse caso, devem estar todos posicionados da mesma maneira com relação ao Sol. Isto ocorre porque devido à ligação série, se um painel receber menos luz que os outros, a corrente fornecida por todos é limitada à corrente fornecida por este, reduzindo significativamente a geração total de energia.

Devido à necessidade da instalação de vários painéis juntamente com o inversor string, o custo inicial das instalações de geração fotovoltaica se torna relativamente alto para aplicações “on-grid” de pequena potência convencionais. Assim, com o objetivo de aumentar a modularidade e reduzir os custos iniciais, foi desenvolvido o equipamento denominado “Micro Inversor”, o qual pode operar somente com um ou dois painéis fotovoltaicos.

O equipamento Micro Inversor possui conversor CC-CC que permite alimentá-lo com a tensão de um painel (30 a 40Vcc) e gerar a tensão contínua de 200 ou 360V para obtenção da tensão alternada de 127 ou 220VCA. Devido às suas pequenas dimensões e proteção contra chuva, o equipamento pode ser fixado atrás dos painéis fotovoltaicos, barateando e facilitando a instalação. Uma das principais vantagens do Micro Inversor é a possibilidade de cada painel poder operar de forma independente.

Assim, não é necessário que todos os painéis estejam posicionados da mesma maneira em relação ao Sol. Isto possibilita a aplicação dos painéis em telhados com diversas inclinações e/ou pontos de sombreamentos críticos. A figura F.1 mostra um diagrama simplificado de uma usina do tipo on-grid:

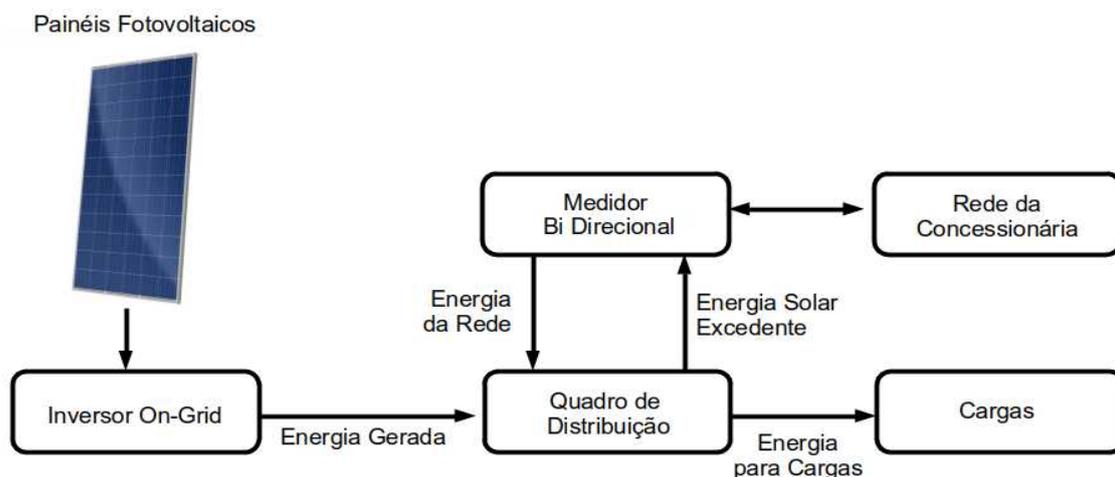


Figura F.1 - Diagrama Simplificado de uma Usina Solar On Grid

F.2 Usinas Solares do tipo “Off Grid” e Híbridas

Os Sistemas Off-Grid são sistemas de geração solar fotovoltaica que operam de modo independente da rede elétrica da concessionária. A figura F.2 mostra um diagrama simplificado de uma usina solar off-grid. A usina pode trabalhar totalmente independente da rede ou pode ter os recursos de chaveamento das cargas entre a rede da concessionária e os inversores. Estes recursos de chaveamento são denominados de Chaves de Transferência, que permitem selecionar entre as fontes de energia para as cargas. As usinas “Off-Grid” que tem recursos de chaveamento com a energias da rede da concessionária são denominadas Usinas Híbridas.

No sistema “Off-Grid” devem ser instalados equipamentos denominados Controladores de Carga. Estes equipamentos recebem as saídas dos painéis fotovoltaicos e convertem a tensão para aplicar a corrente gerada em um barramento de corrente contínua. Neste barramento são conectadas também as baterias e as cargas. As cargas principais são os inversores, mas dependendo da instalação podem existir cargas alimentadas em corrente contínua. A faixa de tensão utilizada no barramento geral de corrente contínua depende do arranjo e do tipo das baterias. O arranjo das baterias fornece a tensão padronizada do barramento. Os valores mais utilizados são 24 Volts e 48 Volts, sendo que o padrão 24 Volts é o mais amplamente usado devido ao menor custo dos equipamentos.

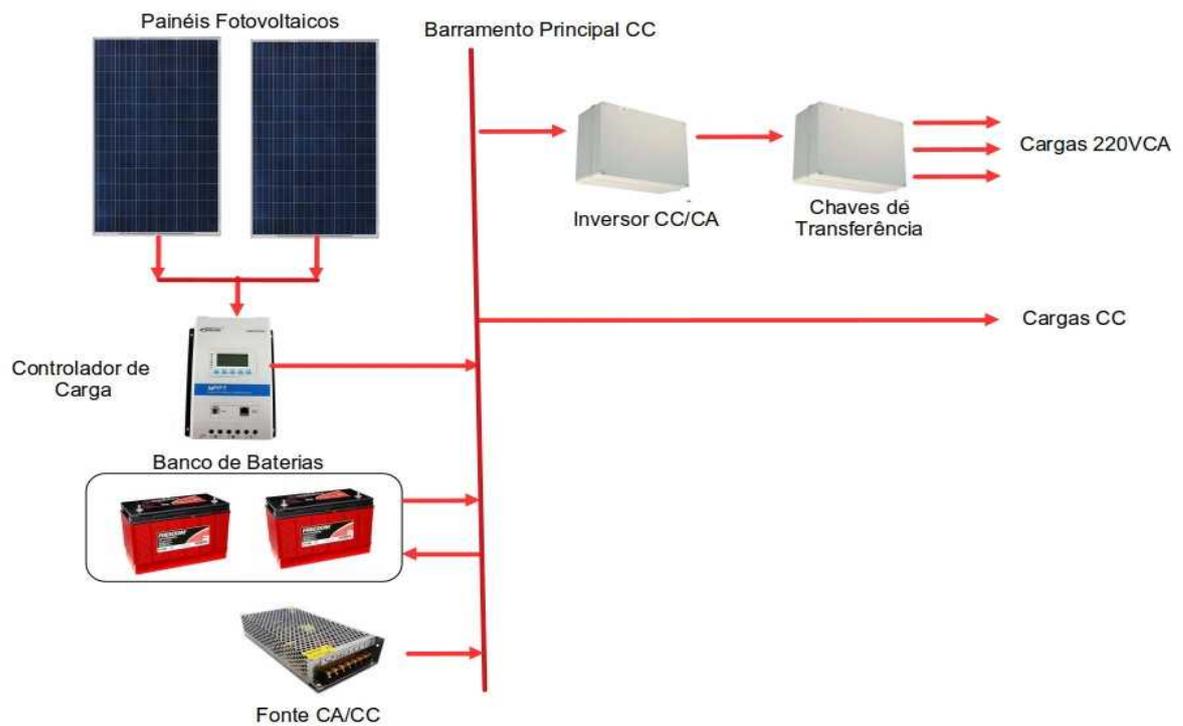


Figura F.2 - Diagrama Simplificado de uma Usina Solar Off Grid

F.3 Características do Controlador de Carga

Em sistemas de geração solar do tipo “off-grid”, que utilizam baterias, é necessário o uso de um equipamento denominado Controlador de Carga. O Controlador de Carga ou Regulador de Carga é basicamente um equipamento que regula a tensão e a corrente que são fornecidas pelos painéis fotovoltaicos com os objetivos de carregar de modo correto as baterias e fornecer energia para as cargas (PINHO, 2014). A função de regular a tensão e corrente fornecidas pelos painéis fotovoltaicos nos controladores mais modernos é feita de modo a encontrar o ponto de máxima potência, e assim, converter a energia solar da maneira mais eficiente. Este tipo de controlador é denominado Controlador MPPT (Maximum Power Point Tracking – Rastreio do Ponto de Máxima Potência). Como o próprio nome já diz, este tipo de equipamento microprocessado ajusta a corrente demandada dos painéis conectados em sua entrada para sempre obter a máxima potência. Além de otimizar a geração nas condições de boa incidência de luz solar, esta característica permite gerar energia mesmo com níveis de insolação mais baixos, ao amanhecer, ao entardecer e com céu nublado.

Um painel fotovoltaico em circuito aberto gera uma determinada tensão quando recebe a incidência da luz solar. Ao ser solicitada corrente do painel fotovoltaico pelo controlador de carga, a tensão cai em seus terminais. Assim, o controlador de carga executa um

procedimento em que a corrente solicitada é aumentada gradualmente, a tensão é medida, e a potência fornecida é calculada periodicamente através do produto da tensão x corrente. Ao ser aumentada a corrente, a potência fornecida aumenta até certo ponto, após o qual começa a diminuir. Esse é o ponto de máxima potência fornecida pelos painéis. Para cada nível de potência incidente de luz por área (Irradiância), existe um ponto de tensão e corrente onde se encontra a máxima potência. Este comportamento dos painéis fotovoltaicos é causado pelas suas características intrínsecas de tensão e corrente em função da potência de luz incidente por área (Irradiância).

Por exemplo, a tensão de saída dos painéis fotovoltaicos de setenta e duas células podem variar de 30 a 46 volts em corrente contínua. Este tipo de painel fornece uma corrente máxima de aproximadamente 9 amperes em tensão de 38 Volts para potência máxima. Portanto, para alimentar um sistema de geração “off-grid” com banco de baterias de tensão nominal de 24 volts, é necessário fazer a regulação da tensão e corrente para as faixas de tensão e corrente permitidas.

Os controladores de carga mais modernos são microprocessados e efetuam todos os procedimentos necessários para carregar corretamente as baterias e também alimentar as cargas. Estes controladores permitem selecionar o tipo de bateria usada para ajuste preciso das condições de carga, e efetuam diversos ciclos de equalização das baterias e mantêm a tensão de flutuação entre os ciclos de modo automático. Estes procedimentos mantêm sempre o banco de baterias em sua melhor condição.

Anexo G - Baterias de Chumbo Ácido

G.1 Características Gerais

Existem basicamente dois tipos de baterias de chumbo ácido: com eletrólito fluido e com eletrólito em forma de gel. As baterias com eletrólito fluido podem ser seladas ou não, mas sempre devem ser instaladas em local nivelado na posição correta. As baterias com eletrólito gel sempre são seladas e podem ser instaladas em locais não nivelados. As baterias seladas também são chamadas de baterias sem manutenção, porque não necessitam de verificação do nível do eletrólito. Entretanto, caso ocorra alta corrente de carga ou alta temperatura de operação que provoque evaporação excessiva, e o nível do eletrólito fique baixo, a bateria deve ser substituída.

Normalmente são usados conjuntos denominados bancos de baterias, onde são feitas as associações série-paralelo necessárias para obtenção da tensão e capacidade desejadas. Os sistemas mais populares e de menor custo operam com tensão nominal de 24Vcc. Considerando que os fabricantes normalmente produzem baterias de chumbo ácido estacionárias com tensão nominal de 12 Volts, um banco de baterias de um sistema padrão de 24 Volts precisa no mínimo de duas baterias ligadas em série.

Os dois parâmetros mais importantes para qualquer tipo de bateria são: a tensão nominal em Volts (V) e a capacidade de armazenamento de energia em Amperes-Hora (Ah). Uma bateria, ou acumulador é composto por um conjunto de células, ou elementos em série. Cada elemento produz um valor de tensão que é determinado pelas características químicas. Um elemento de uma bateria de chumbo ácido tem tensão nominal de 2 Volts. As baterias de chumbo ácido de 12 Volts possuem seis elementos em série (ENERTEC, 2008).

G.2 Cálculo da Capacidade de Carga

O cálculo da capacidade do banco de baterias não é simples, e depende do modo de operação, que inclui o consumo das cargas. O modo de operação define em que condições as baterias são usadas, que basicamente se resume na quantidade de energia que é retirada a cada ciclo de descarga. O tempo de vida útil das baterias depende do modo de operação, principalmente no caso das baterias de chumbo ácido. Assim, o tempo de vida diminui significativamente caso as baterias sejam submetidas sempre a ciclos de descarga profunda e carga. A descarga

profunda ocorre quando uma bateria é descarregada até próximo de sua capacidade mínima. No caso de baterias de chumbo ácido, isto ocorre quando a tensão de cada elemento se aproxima de 1,75 Volts, que corresponde à tensão de 10,5 Volts em uma bateria que possui tensão nominal de 12 Volts.

G.3 Informações Fornecidas pelo Fabricante

O fabricante fornece um gráfico com a curva de descarga para cada tipo de bateria em função da corrente consumida, e uma tabela com diversas correntes de descarga e o tempo total de descarga. Por exemplo, para uma determinada bateria de chumbo ácido, o fabricante fornece o tempo de descarga de 20 Horas para uma corrente de descarga de 5,3 Amperes (A). Nessas condições de descarga, esta bateria possui capacidade de $20 \times 5,3 = 106$ Amperes / Hora (Ah). Nas baterias de chumbo ácido, quanto menor a corrente de descarga, maior a capacidade. Neste mesmo exemplo, para uma corrente de descarga de 9,4A, o tempo de descarga é 10h, portanto nestas condições, a bateria possui capacidade de 94Ah.

O fabricante fornece uma tabela que possui uma linha para cada tipo de bateria, a relação entre o tempo e os valores de corrente de descarga (A) em diferentes regimes na temperatura de 25°C. A tensão inicial é a tensão da bateria de 12 Volts carregada, entre 12,6 e 13Vcc, e a tensão final = 10,5Vcc, é a tensão da bateria completamente descarregada. Através desta tabela, constata-se que a capacidade da bateria é máxima com a corrente mínima e diminui conforme a corrente de descarga aumenta. Por exemplo, uma bateria de 12V 115Ah, pode fornecer uma corrente de 1,2A durante 100 horas, mas fornece uma corrente de apenas 60A durante somente 1 hora. A tabela é fornecida com 19 valores de corrente na faixa de 10% a 133% da capacidade em Ah.

O fabricante fornece uma tabela que contém a capacidade em ampere-hora (Ah) em diferentes regimes de descarga na temperatura de 25°C. Esta tabela mostra a variação da capacidade em Amperes Hora de cada bateria, em relação à corrente de descarga. Por exemplo, uma bateria que tem capacidade de 120Ah em regime de descarga com corrente de 1,2A, reduz a sua capacidade para 60Ah em regime de descarga com corrente de 60A.

O fabricante fornece uma tabela com a capacidade de fornecimento de energia em Watts-Hora em diferentes regimes de descarga na temperatura de 25°C. Esta tabela mostra o tempo que um determinado valor de energia é fornecido até a descarga total da bateria. Por exemplo, uma bateria de 12V 115Ah pode fornecer um valor de energia de 13Wh durante 100 horas. A

mesma bateria é capaz de fornecer 310Wh durante duas horas. Esta unidade é usada em medição de energia elétrica pelas concessionárias, mais comumente em KWh (quilowatt hora). Um Wh = 3600 Joules. Um KWh = 3.600.000 Joules.

O fabricante fornece um conjunto de gráficos, que mostram para cada tipo de bateria, a variação da tensão da bateria em relação ao tempo para diversas correntes constantes de descarga. No tempo zero, a bateria está completamente carregada com a tensão normalmente entre 12,6 e 13,0Vcc, e no tempo final a bateria está com a tensão de 10,5Vcc indicando que está completamente descarregada. São fornecidos dois gráficos para cada bateria, onde o primeiro gráfico mostra regimes de descarga com correntes mais baixas e tempos maiores, e o segundo gráfico mostra regimes de descarga com correntes mais altas e tempos menores. Estes gráficos em conjunto com as tabelas servem de referência para o projetista da usina solar efetuar o dimensionamento do banco de baterias. O dimensionamento deve levar em consideração diversos fatores tais como o modo principal de utilização. Nesta tarefa entram questões técnicas e econômicas, já que o tempo de vida das baterias depende do modo de uso.

O fabricante fornece duas informações essenciais para o correto carregamento deste tipo de bateria: tensão de flutuação e tensão de equalização e carga. A tensão de flutuação aplica-se quando a bateria está carregada em modo de espera. Modo de espera significa que a bateria está pronta para fornecer energia caso seja solicitado, no caso de uma interrupção no fornecimento de energia da concessionária ou na redução da geração de energia solar. Isto é feito para manter a bateria sempre carregada em bom estado de conservação, já que as baterias deste tipo tem uma taxa de auto descarga relativamente alta. Em sistemas de energia ininterrupta, o banco de baterias normalmente está em estado de espera, sendo mantido em estado de flutuação, com as baterias recebendo um determinado valor de tensão especificado pelo fabricante. Entretanto, após uma descarga, a carga total da bateria só é completada quando a tensão é mantida por um determinado tempo em seu valor de equalização e carga até que a corrente se reduza, indicando que a bateria está carregada. As baterias de chumbo ácido tem tensão de flutuação na faixa de 13,2 a 13,8Vcc e tensão de equalização e carga na faixa de 14,4 a 15,5Vcc na temperatura de 25°C.

Esses valores de tensão mudam ligeiramente com a variação da temperatura, o que é chamado de compensação de temperatura, desta forma, para cada 1°C acima de 25°C, deve-se subtrair 0,033 V e para cada 1°C abaixo de 25°C, deve-se adicionar 0,033 V aos valores de tensão de flutuação, equalização e carga. A compensação de temperatura é feita automaticamente pelos

controladores de carga microprocessados, os quais são fornecidos com sensores de temperatura que são instalados junto ao banco de baterias.

O tempo de vida útil das baterias de chumbo ácido estacionárias é definido em número de ciclos de carga e descarga. O tempo de vida útil ou o número de ciclos depende fundamentalmente de dois fatores: A profundidade da descarga e a temperatura de operação. A profundidade da descarga é calculada em porcentagem sobre a faixa de tensão entre o estado carregada e o estado totalmente descarregada da bateria. A Figura G.1 mostra o gráfico do número de ciclos x profundidade da descarga:

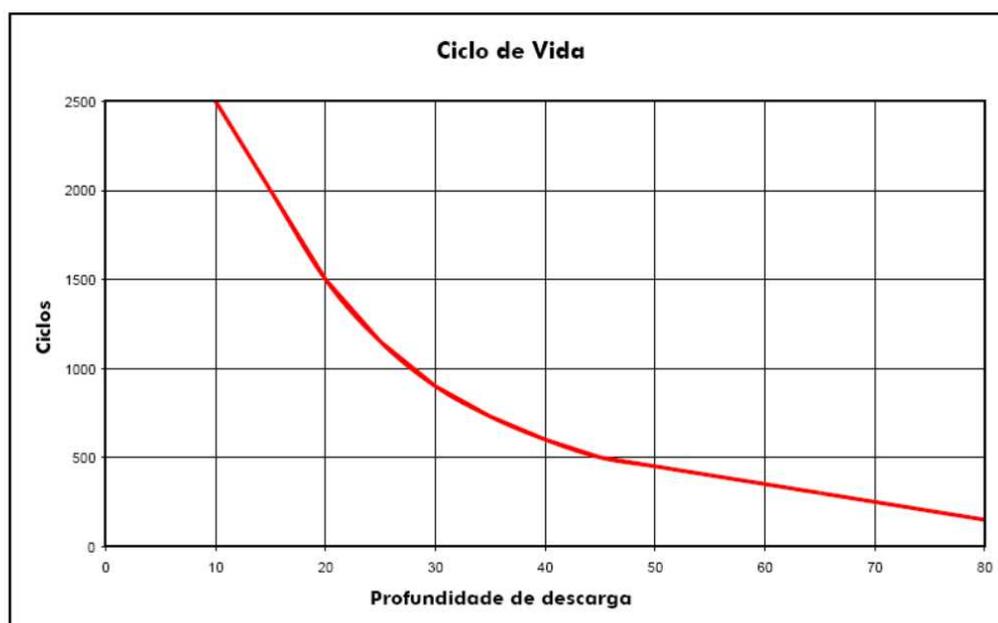


Figura G.1 - Gráfico do Número de Ciclos x Profundidade da Descarga

Considerando um sistema de geração solar fotovoltaica, um ciclo de carga e descarga tem a duração de um dia, quando as baterias são carregadas durante o dia e se descarregam à noite. Portanto, este gráfico mostra que operar com uma profundidade de descarga de 20%, leva a uma vida útil de 1500 ciclos, que corresponde a um pouco mais de 4 anos. Conforme a descarga se torna mais profunda, a vida útil da bateria se reduz drasticamente a valores próximos de 200 ciclos. Portanto, o dimensionamento e o modo de uso do banco de baterias requer a definição de critérios principalmente econômicos, já que o uso intensivo das baterias, submetendo-as a descargas mais profundas, com o objetivo de economia de energia, pode se tornar mais caro que a energia economizada. O custo principal de manutenção de uma usina solar off-grid é igual ao custo do banco de baterias dividido pela sua vida útil em meses.