

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**



ISRAEL CARLOS DA SILVA JUNIOR

**SISTEMA DE DETECÇÃO DE CARGAS PELA PRÉ-
DETERMINAÇÃO DOS VALORES DE PARÂMETROS DE
REGIME PERMANENTE**

VITÓRIA – ES
MARÇO/2017

ISRAEL CARLOS DA SILVA JUNIOR

**SISTEMA DE DETECÇÃO DE CARGAS PELA PRÉ-
DETERMINAÇÃO DOS VALORES DE PARÂMETROS DE
REGIME PERMANENTE**

Parte manuscrita do Projeto de Graduação do aluno **Israel Carlos da Silva Junior**, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof^ª. Dr^ª Jussara Farias Fardin

Coorientador: MSc. Hader Aguiar Azzini

VITÓRIA – ES
MARÇO/2017

ISRAEL CARLOS DA SILVA JUNIOR

**SISTEMA DE DETECÇÃO DE CARGAS PELA PRÉ-
DETERMINAÇÃO DOS VALORES DE PARÂMETROS DE
REGIME PERMANENTE**

Parte manuscrita do projeto de graduação do aluno **Israel Carlos da Silva Junior**, apresentada ao departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista

Aprovado em XX de XX de XX

COMISSÃO EXAMINADORA

Prof^a. Dr^a. Jussara Farias Fardin
Universidade Federal do Espírito Santo - UFES
Orientadora

Prof. Dr. Wagner Teixeira da Costa
Instituto Federal do Espírito Santo - IFES
Examinador

Eng. Ronaldo Borges de Alvarenga Neto
Examinador

DECLARAÇÃO DO AUTOR

Declaro, para fins de pesquisa acadêmica, didática e técnico-científica, que este Trabalho de Conclusão de Curso pode ser parcialmente utilizado, desde que se faça referência à fonte e ao autor.

Vitória, 18 de março de 2017.

Israel Carlos da Silva Junior

AGRADECIMENTO

Agradeço primeiramente a Deus, por me amparar nos momentos de angústia e pela forma como a minha vida mudou.

À minha família e amigos, pelo apoio incondicional. Em especial, à minha amada mãe, fundamental nessa vitória, agradeço por cada valioso ensinamento que me concedeu e que carregarei comigo por toda vida. Obrigado, mãe, por se dedicar tanto por mim.

Agradeço à minha querida namorada, que acompanhou cada passo meu nessa árdua caminhada... sempre me apoiando e não me permitindo desistir.

A minha professora e orientadora, Jussara Farias Fardin, pelo conhecimento passado durante os últimos anos, e pela paciência e incentivo nesse processo para que a realização deste trabalho se tornasse possível. Obrigado, Professora.

Agradeço também ao meu querido amigo Willian Garcia Viega dos Santos, pelo apoio técnico e moral para o desenvolvimento deste projeto. Valeu, Mano.

A todos os meus colegas de curso, que dividiram comigo os momentos de esforço e insônia, as alegrias e as tristezas intrínsecas ao curso de Engenharia. Desejo sucesso profissional e pessoal a cada um de vocês.

Agradeço ainda a cada professor e servidor da UFES que, direta ou indiretamente, contribuíram para essa vitória iluminando meu caminho com conhecimento e dedicação.

“...então cerra os punhos, sorria. E jamais volte para sua quebrada de mãos e mente vazias.”

Levanta e Anda – Emicida.

RESUMO

A necessidade de desenvolver sistemas capazes de auxiliar na preservação do planeta tem sido cada vez mais discutida. Logicamente, preservar remete à ideia de utilizar de forma mais eficiente os recursos energéticos que dispomos. Pensando em termos de energia elétrica, uma estratégia já utilizada na busca pela eficiência energética do sistema elétrico é o monitoramento do uso dessa energia nos lares brasileiros.

Existem várias vantagens no monitoramento do consumo de energia elétrica. Conhecendo o perfil de consumo dos usuários do sistema elétrico, podemos criar dispositivos capazes de otimizar o consumo individual ou de grupos específicos, planejar a geração e distribuição de energia por setores aumentando a eficiência do sistema e ainda traçar perfis socioeconômicos que servem como estudo de mercado pra empresas fornecedoras de aparelhos elétricos.

Baseado na importância do tema, o trabalho aqui apresentado objetiva a criação de um sistema de monitoramento não intrusivo, capaz de executar a detecção de vários tipos de cargas partindo apenas dos sinais de tensão e corrente no ramal de entrada de uma unidade consumidora. A ideia, basicamente, é criar um perfil para cada tipo de carga através de testes para pré-determinação dos valores de parâmetros como tensão e corrente, e então, realizar a comparação desses perfis com os valores calculados em tempo real por um microcontrolador. Para o processamento dos sinais foi utilizado um microcontrolador Arduino com um processador Atmel, e para a apresentação dos resultados, foi criada uma interface utilizando-se a plataforma de criação de supervisórios ScadaBR.

A grande vantagem deste método na detecção de cargas é que não é necessário acesso à carga para o monitoramento da mesma.

LISTA DE FIGURAS

Figura 1 - Posicionamento do Monitor na unidade consumidora.....	17
Figura 2 - Modo de coleta de dados do sistema descrito.....	18
Figura 3 - Triângulo de potências.....	27
Figura 4 - Esquema de obtenção dos sinais de tensão e corrente.....	28
Figura 5 - Arquitetura da plataforma Arduino.....	29
Figura 6 - Sensor de corrente ACS 712.....	30
Figura 7 - Transformador de tensão Hayonik com relação 10:1.....	30
Figura 8 – Circuito de adequação da tensão e os sinais esperados.....	31
Figura 9 - Simulação do circuito de adequação no Multisim.....	32
Figura 10 - resultado da simulação em Multisim do circuito de adequação.....	32
Figura 11 - Resultados obtidos em laboratório para o circuito de adequação da tensão.....	33
Figura 12 - Sinal de referência da tensão próximo de 2.5 V.....	33
Figura 13 - Sinal senoidal de entrada da tensão no microcontrolador.....	34
Figura 14 – Fluxograma do algoritmo de cálculo dos parâmetros de regime permanente.....	36
Figura 15 - Display de LCD como possibilidade de apresentação de resultados no Arduino.....	41
Figura 16 - Configuração dos dados Serial Modbus.....	43
Figura 17 - Imagem da watchlist em funcionamento.....	44
Figura 18 - Configuração dos dados do tipo Data Source Meta.....	45
Figura 19 - Interface de detecção de cargas.....	46
Figura 20 - Representação visual das cargas na interface de detecção.....	47
Figura 21 - representação gráfica das grandezas envolvidas no projeto.....	47
Figura 22 – Diagrama de definição do intervalo de detecção de um parâmetro X.....	49
Figura 23 - Lâmpada incandescente utilizada nos testes de detecção.....	50
Figura 24 – Dados do Serial Monitor durante a ligação da lâmpada incandescente.....	51
Figura 25 - Lâmpada fluorescente compacta usada nos testes de detecção.....	54
Figura 26 – Dados do serial Monitor durante a ligação da lâmpada fluorescente.....	55
Figura 27 - Ventilador utilizado nos testes de detecção.....	58
Figura 28 – Dados do Serial Monitor durante a ligação do ventilador em velocidade 1.....	59
Figura 29 - dados do Serial Monitor durante a ligação do ventilador em velocidade 2.....	60
Figura 30 - Dados do Serial Monitor durante a ligação do ventilador em velocidade 3.....	61
Figura 31 - Ferro de passar utilizado nos testes de detecção de cargas.....	68

Figura 32 - Dados do Serial Monitor durante a ligação do ferro de passar em potência mínima	69
Figura 33 - Dados do Serial Monitor durante a ligação do ferro de passar em potência máxima	70
Figura 34 - Refrigerador utilizado nos testes de detecção	73
Figura 35 - Dados do serial monitor durante a ligação da geladeira	74
Figura 36 - Forno de microondas utilizado nos testes de detecção de cargas	77
Figura 37 - Dados do Serial Monitor para a operação do micro-ondas em modo 'aquecimento'	78
Figura 38 - Dados do Serial Monitor para a operação do micro-ondas em modo 'descongelamento'	79
Figura 39 - Dados do <i>serial monitor</i> para a ligação simultânea das lâmpadas incandescente e fluorescente.....	88
Figura 40 - gráficos obtidos durante a ligação da lâmpada incandescente.....	92
Figura 41 - Interface de detecção durante a ligação da lâmpada incandescente	93
Figura 42 - Gráficos obtidos durante a ligação da lâmpada fluorescente.....	94
Figura 43 - Interface de detecção durante a ligação da lâmpada incandescente	94
Figura 44 - Gráficos obtidos durante a ligação do ventilador em potência mínima.....	95
Figura 45 - Gráficos obtidos durante a ligação do ventilador em potência média	96
Figura 46 - Gráficos obtidos durante a ligação do ventilador em potência máxima.....	97
Figura 47 - Interface de detecção durante a ligação da lâmpada incandescente	97
Figura 48 - Gráficos obtidos durante a ligação do ferro de passar em potência mínima	98
Figura 49 - Gráficos obtidos durante a ligação do ferro de passar em potência máxima.....	99
Figura 50 -Interface de detecção durante a ligação do ferro de passar.....	100
Figura 51 - Gráficos obtidos durante a ligação do refrigerador.....	100
Figura 52 - Interface de detecção durante a ligação do refrigerador	101
Figura 53 - Gráficos obtidos durante a ligação do forno de micro-ondas em modo 'aquecimento'	102
Figura 54 - gráficos obtidos durante a ligação do forno de micro-ondas em modo 'descongelamento'	103
Figura 55 - Interface de detecção durante a ligação do forno de micro-ondas.....	104
Figura 56 - gráficos obtidos durante a ligação simultânea das lâmpadas incandescente e fluorescente.....	104

Figura 57 - Interface de detecção durante a ligação de duas cargas simultaneamente (lâmpadas).....	105
--	-----

LISTA DE TABELAS

Tabela 1 - Relação entre o tipo de carga e valor da variável <i>identificador</i>	37
Tabela 2 - Erro associado a cada grandeza devido à imprecisão dos sensores.....	40
Tabela 3 – Valores mínimos e máximos dos parâmetros medidos na lâmpada incandescente.....	51
Tabela 4 - Valores mínimos e máximos dos parâmetros medidos na lâmpada fluorescente .	55
Tabela 5 - valores mínimos e máximos dos parâmetros medidos no ventilador (3 velocidades)	62
Tabela 6 - Valores mínimos e máximos dos parâmetros medidos do ferro de passar.....	70
Tabela 7 - Valores mínimos e máximos dos parâmetros medidos no refrigerador	74
Tabela 8 - Valores mínimos e máximos dos parâmetros medidos no forno de micro-ondas .	80
Tabela 9 - Valores mínimos e máximos dos parâmetros medidos na ligação de cargas simultâneas	88

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ANEEL	Agência Nacional de Energia Elétrica
PROCEL	Programa Nacional de Conservação de Energia Elétrica
IBGE	Fundação Instituto Brasileiro de Geografia e Estatística
DHT	Distorção Harmônica Total
UFES	Universidade Federal do Espírito Santo
CEPEL	Centro de Pesquisa de Energia Elétrica

LISTA DE SÍMBOLOS

$v(t)$	Tensão instantânea em um elemento (V)
$i(t)$	Corrente instantânea no elemento (A)
t	Tempo (s)
Σ	Somatório de termos
ω	Velocidade angular de uma função periódica (rad/s)
\int	Operador de integração

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Apresentação	15
1.2	Justificativa.....	18
1.3	Objetivos.....	19
1.3.1	Objetivo geral	19
1.3.2	Objetivos específicos.....	19
1.4	Estrutura do texto	20
1.5	Metodologias para a diferenciação de cargas	20
1.5.1	Escolha do perfil de consumo para análise e testes.....	20
1.5.2	Diferenciação por valores pré-determinados dos parâmetros de regime permanente	22
1.5.3	Limitações do Projeto.....	23
2	O PROTÓTIPO DO DETECTOR DE EVENTOS.....	25
2.1	Embasamento Teórico	25
2.1.1	Valores médios quadráticos.....	25
2.1.2	Potência instantânea	25
2.1.3	Potência ativa	26
2.1.4	Potência Aparente.....	26
2.1.5	Fator de potência	27
2.2	Protótipo do Detector de Eventos.....	28
2.2.1	A plataforma Arduino	28
2.2.2	O sensor de corrente	29
2.2.3	O sensor de tensão.....	30
3	CÁLCULO DOS PARÂMETROS EM TEMPO REAL E COMUNICAÇÃO MODBUS.....	35
3.1	O algoritmo de cálculo dos parâmetros de regime permanente	35
3.2	Definição do valor do <i>Identificador</i>	36
3.3	O protocolo Modbus.....	37
3.4	Composição dos erros dos sensores de tensão e corrente.....	38
4	A INTERFACE DE USUÁRIO E O SCADABR	41

4.1 Dados do tipo Serial Modbus	42
4.2 Dados do tipo Source Meta	43
4.3 A interface com o usuário.....	45
5 DETERMINAÇÃO DOS INTERVALOS DE IDENTIFICAÇÃO DAS CARGAS	
ATRAVÉS DOS PARÂMETROS DE REGIME PERMANENTE.....	48
5.1 Lâmpada incandescente.....	49
5.2 Lâmpada fluorescente.....	53
5.3 Ventilador	57
5.4 Ferro de passar.....	67
5.5 Refrigerador.....	73
5.6 Forno de microondas	76
5.7 Cargas simultâneas: Lâmpadas incandescente e fluorescente.....	87
6 RESULTADOS DOS TESTES DO SISTEMA DE DETECÇÃO DE CARGAS.....	91
6.1 Resultado do teste para a identificação da lâmpada incandescente.....	92
6.2 Resultado do teste para a identificação da lâmpada fluorescente.....	93
6.3 Resultado do teste para a identificação do ventilador	95
6.4 Resultado do teste para a identificação do ferro de passar.....	98
6.5 Resultado do teste para a identificação do refrigerador	100
6.6 Resultado do teste para a identificação do forno de micro-ondas	101
6.7 Resultado do teste para identificação de cargas simultâneas	104
7 CONCLUSÕES E TRABALHOS FUTUROS.....	106
8 REFERÊNCIAS BIBLIOGRÁFICAS	108

1 INTRODUÇÃO

1.1 Apresentação

Nos últimos anos, a preocupação em desenvolver sistemas capazes de auxiliar na preservação do planeta tem aumentado significativamente. Essa preservação, obviamente, passa por um uso racional das diversas formas de energia que dispomos nos dias atuais, como por exemplo, a energia elétrica. Assim, os governos e órgãos responsáveis têm buscado, através dos meios de comunicação, conscientizar as pessoas sobre a contribuição que cada indivíduo pode oferecer em busca de um consumo sustentável de energia. Vários são os estudos realizados no Brasil voltados para o consumo de energia elétrica em busca de sua otimização, tanto no setor Industrial, como no comercial e principalmente no residencial.

O grupo de consumidores residenciais representa hoje cerca de 24,2% do consumo total de energia elétrica no país, sendo o segundo segmento que mais demanda energia elétrica no Brasil, atrás apenas do setor industrial que consome cerca de 40,7 % de toda energia gerada no País [1]. A projeção de crescimento de 4,5% ao ano até 2019 descrita na página 14 do Plano decenal de Expansão de energia [2], torna este subgrupo o alvo principal de estratégias de conscientização e de pesquisas em busca da eficiência energética. Alguns estudos detalhados sobre o perfil dos consumidores residenciais têm sido usados como base para o planejamento energético do Brasil, como feito em [3].

O crescimento econômico evidenciado no Brasil na última década tem mudado drasticamente o mercado energético do País. Essa mudança se dá devido a inserção de novos aparelhos eletrodomésticos nos lares brasileiros, o que tem mudado significativamente o perfil de consumo desta classe [3]. Tarefas que antigamente eram feitas de forma mais rústica ou manual, hoje são auxiliadas por novos aparelhos que facilitam o trabalho dos usuários, como máquinas de lavar e forno micro-ondas. E junto com essa mudança de perfil dos consumidores residenciais, vem a preocupação com o uso consciente da energia elétrica. Essa conscientização vem sendo realizada através de propagandas educacionais, incentivando as pessoas a um melhor uso dos recursos energéticos disponíveis, e ao uso de equipamentos modernos, cujo avanço tecnológico os tornaram mais eficientes.

Dentre os diferentes planos de desenvolvimento, o Ministério de Minas e Energia e a Agência Nacional de Energia Elétrica (ANEEL) têm dado atenção especial para os projetos voltados ao planejamento energético do país, que visa a criação de hábitos racionais do uso da eletricidade nas famílias brasileiras. Com essas campanhas e projetos, no período 2012/2013 foram economizados mais de 400 GWh/ano, o que retirou uma demanda de 200 MW do horário de ponta nesse período [4]. Mas a longo prazo, a dinâmica do mercado energético tornará os atuais métodos de análise de consumo ineficientes e isso poderá representar grandes prejuízos.

Do ponto de vista energético, os projetos desenvolvidos pelas fábricas de eletrodomésticos já são pensados de forma a obter um uso mais eficiente da energia por parte dos dispositivos por elas fabricados. Para regulamentar esses padrões de conservação e eficiência energética, existe o programa PROCEL (Programa Nacional de Conservação de Energia Elétrica) da Eletrobrás, que se faz presente, qualificando os tipos de aparelhos de acordo com sua eficiência energética. O uso de equipamentos com selo Procel gerou, em 2015, uma economia de 11556 GWh, como pode ser visto na página 25 do estudo da Procel [5]. Essa economia evitou ainda que 1,44 milhão de toneladas de CO₂ equivalentes fossem emitidos na atmosfera.

Baseado no que já foi dito anteriormente, é possível observar a necessidade de aprofundar os estudos e análises que permitam melhores avaliações da nova dinâmica do mercado de energia no Brasil em busca de economia, planejamento energético e até mesmo preservação ambiental.

Um método de análise eficiente é a monitoração das cargas de uma instalação elétrica visando criar um perfil energético preciso para essa unidade consumidora. Esse método pode trazer resultados bem detalhados a cerca de um consumidor, porém nesse modelo de análise de dados, para que esse monitoramento ocorra, é necessário ter acesso às cargas e, conseqüentemente, às casas dos consumidores, o que nem sempre é possível. Devido a essa necessidade, esse tipo de monitoramento é chamado de *monitoramento invasivo*.

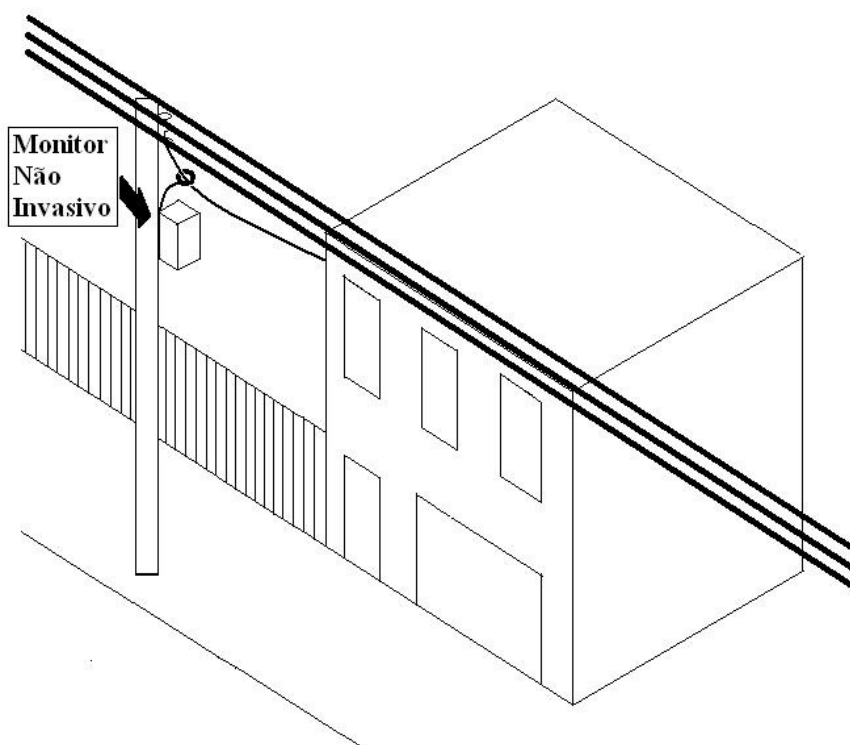
Existe também uma outra forma de monitorar as cargas de uma unidade consumidora: o *monitoramento não invasivo*, que é assim chamado pois, ao contrário do monitoramento invasivo, não necessita de acesso às cargas.

“O Monitoramento Não Invasivo de cargas tem como objetivo básico determinar o consumo individual de cada carga sem ter acesso físico a ela. Isto é feito a partir da análise detalhada da tensão e da corrente na interface com a fonte de energia elétrica

que alimenta a instalação elétrica. As formas de onda da corrente e da tensão no ramal principal são registradas e analisadas. Tal análise das formas de onda da carga total gera estimativas do consumo individual das cargas e outras estatísticas” [6]. Em contrapartida, são sistemas mais complexos e, por consequência, de maior custo em relação ao citado anteriormente.

Como dito acima, o monitoramento não invasivo de cargas analisa apenas os dados na entrada de uma unidade consumidora. A Figura 1 ilustra um sistema não invasivo de cargas e o seu posicionamento em uma unidade consumidora:

Figura 1 - Posicionamento do Monitor na unidade consumidora.

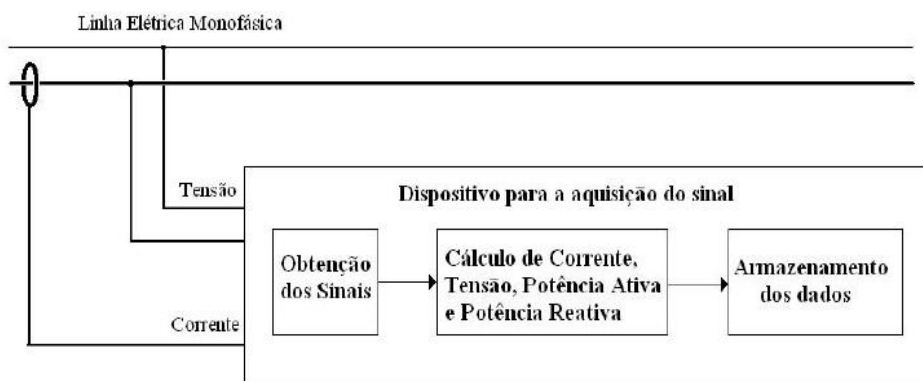


Fonte: Canón, 2009, p. 22.

Através das curvas de tensão e corrente, um *software* processa as informações coletadas, a fim de detectar mudanças de níveis de potência ativa (que indicam mudança de estado de uma carga), ou mesmo detectando padrões pré-estabelecidos para os transitórios de corrente que são comumente chamados de assinaturas e variam de acordo com o tipo de carga que é ligada ou desligada, além da análise das formas de onda de tensão e corrente que dão informações valiosas sobre o tempo de

funcionamento e, conseqüentemente, o consumo das cargas residenciais. A Figura 2 ilustra o bloco de funcionamento de um sistema de monitoramento de cargas.

Figura 2 - Modo de coleta de dados do sistema descrito.



Fonte: Canón, 2009, p. 20.

Esses dados dão informações úteis e precisas sobre o perfil de consumo dos usuários, e podem ser usados para se estabelecer formas de utilização mais eficientes de energia elétrica aos consumidores residenciais.

1.2 Justificativa

O monitoramento do consumo de energia tem várias vantagens, como por exemplo:

1. Permite estudar o consumo de energia elétrica nas residências brasileiras, para saber de que forma esta energia é utilizada pelos usuários a fim de aumentar a eficiência energética do sistema.
2. Permite obter informação de consumo dos horários de ponta e fora ponta e, aliado ao sistema de monitoramento, pode-se desenvolver dispositivos de inteligência capazes de decidir qual eletrodoméstico ligar a cada momento a fim de otimizar o consumo e minimizar o valor da fatura baseando-se na tarifa branca que prevê diferentes valores de tarifa para diferentes períodos do dia.
3. Ajuda e orienta o planejamento de eficiência energética através de propagandas educacionais em populações já conhecidas, a partir das informações do uso da energia elétrica.

4. Permite obter informação sobre dispositivos utilizados em cada uma das residências para estudos de perfil sócio econômico, ou para estudos de mercado para empresas fornecedoras de aparelhos eletrônicos.

Hoje no Brasil, para se realizar estes estudos de perfil de consumo, em geral, é feita uma coleta de dados por meio de enquetes, como realizado em [7]. Porém, por sua natureza estatística, estes dados estão sujeitos a erros como por exemplo, respostas imprecisas dos usuários que são entrevistados. Outro erro significativo vem do fato de que duas pessoas em uma mesma casa podem descrever um perfil de consumo diferente devido aos seus hábitos individuais, o que tornaria os dados coletados pouco representativos em termos do perfil consumidor naquela residência.

Devido aos erros gerados pela falta de precisão na obtenção de dados, se faz necessário desenvolver sistemas que obtenha resultados mais técnicos, precisos e de processamento mais rápido acerca do perfil consumidor da população brasileira, pois dados com o processamento demorado dos resultados podem perder validade, visto que a matriz energética é um mercado tão dinâmico.

1.3 Objetivos

1.3.1 Objetivo geral

Este trabalho objetiva o desenvolvimento de um sistema para detecção de cargas a partir de análises feitas nos sinais de tensão e corrente no ramal de entrada de uma unidade consumidora, utilizando suas características de regime permanente para o reconhecimento do tipo de carga acionada em cada caso.

Este sistema deve ser capaz de analisar eventos de liga/desliga de eletrodomésticos através de sua característica de regime permanente, e relacioná-lo com a carga que o gerou.

1.3.2 Objetivos específicos

1. Desenvolver um *software* no Arduino para fazer as leituras dos sensores de tensão e corrente na entrada da instalação, e com esses dados, calcular os

- valores de tensão e corrente *RMS*, os níveis de potência ativa, potência reativa, potência aparente e o fator de potência da unidade consumidora;
2. A partir dos dados obtidos, criar um perfil de características de regime permanente para cada carga analisada no projeto e então, ler os sinais em tempo real e fazer a identificação do tipo de carga que gerou tais sinais através da comparação com os dados previamente coletados.
 3. Criar uma interface intuitiva para apresentar, de forma clara, tanto os sinais lidos pelos sensores como os resultados obtidos da detecção das cargas conforme elas são ligadas e desligadas.

1.4 Estrutura do texto

Este trabalho está organizado da seguinte forma: O Capítulo 1 aborda, além da justificativa e dos objetivos do projeto, a metodologia utilizada no processo de detecção de cargas. O Capítulo 2 destaca os conceitos teóricos fundamentais e detalha a elaboração do protótipo utilizado na obtenção dos dados. O Capítulo 3 aborda o algoritmo computacional utilizado no processamento dos dados e o protocolo de comunicação serial utilizado. O Capítulo 4 apresenta a plataforma utilizada para a apresentação dos dados para o usuário e sua interface. O Capítulo 5 apresenta os testes realizados com as cargas analisadas e a determinação dos valores dos parâmetros durante a operação das mesmas. Finalmente, o Capítulo 6 apresenta os principais resultados obtidos na execução deste projeto. Além disso, o capítulo 7 apresenta as considerações finais e as possibilidades de trabalhos futuros.

1.5 Metodologias para a diferenciação de cargas

1.5.1 Escolha do perfil de consumo para análise e testes

Neste trabalho, foram selecionados para as análises 5 tipos de cargas de forma a representar da melhor forma possível os hábitos de consumo nos lares brasileiros. Para estas escolhas foram considerados os estudos da Empresa de Pesquisa Energética, EPE, [3] sobre a incidência de cada uma dessas cargas nos lares brasileiros. Estas escolhas foram feitas também considerando que as cargas tivessem

diferentes características de regime permanente (o que diminuiu a complexidade das análises) e de modo que nenhuma das cargas superasse os 30 A nominais suportados pelo sensor de corrente utilizado no projeto. Assim, as cargas escolhidas para basear os estudos desse trabalho foram:

- Lâmpada incandescente (puramente resistiva) e fluorescente (eletrônica)
- Ferro de passar (Eletrodoméstico puramente resistivo)
- Ventilador (motor universal ou de indução monofásico)
- Refrigerador (compressor)
- Forno de microondas (Eletrodoméstico eletrônico)

Para realizar o monitoramento das cargas e sua identificação foi necessário definir algumas classes ou tipos de eletrodomésticos de forma que, de acordo com suas características, as mesmas possam ser agrupadas facilitando assim as análises. Essas análises podem ser feitas a partir de características de regime transitório, permanente ou ambos.

Uma metodologia para a diferenciação de grupos de cargas de acordo com suas características é usar as informações de regime transitório, que geralmente são obtidas da forma de onda do sinal de corrente. Desse sinal, são obtidas informações como distorção harmônica total, valor máximo da corrente durante a partida, tempo de estabilização dos valores de corrente, etc... A partir dessas informações, as cargas são classificadas e um *software* de análise de dados faz a identificação de cada uma delas de acordo com suas características intrínsecas.

Uma outra metodologia que pode ser empregada na identificação de cargas é fazer a análise das suas características de regime permanente, e através dessas características, classificar as cargas. Nessa metodologia de identificação de cargas, as principais informações coletadas são: tensão e corrente eficazes, potência ativa, potência aparente e fator de potência.

Com essas informações é possível caracterizar, identificar e diferenciar grande parte das principais cargas elétricas presentes nos lares dos brasileiros. Essa foi a base da metodologia utilizada nesse projeto, tendo como diferencial a utilização de valores pré-determinados dos parâmetros para a classificação das cargas como explicado a seguir.

1.5.2 Diferenciação por valores pré-determinados dos parâmetros de regime permanente

A ideia nesse projeto foi utilizar valores pré-determinados (empiricamente) dos parâmetros de regime permanente das cargas analisadas gerando um banco de dados e, a partir disso, criar um *software* capaz de calcular esses parâmetros através da leitura e processamento dos sinais recebidos em tempo real por sensores de tensão e corrente e então comparar os valores calculados com os valores pré-determinados a fim de determinar a qual classe de cargas os sinais obtidos em tempo real pertencem.

De forma clara, quando ligamos uma lâmpada incandescente, por exemplo, sabemos que o valor de corrente elétrica calculado através dos sinais lidos pelos sensores deverá ficar próximo ao valor de corrente medido nos testes realizados nesta mesma lâmpada, e sabemos ainda que o valor de fator de potência medido e calculado deverá ser unitário (ou próximo disso, considerando uma margem de erro para os sensores). Já no caso de um ferro de passar roupa, por exemplo, que é um eletrodoméstico muito comum nos lares brasileiros, o fator de potência também será unitário por se tratar de uma carga puramente resistiva, assim como a lâmpada. Porém, em relação à lâmpada incandescente, os níveis de corrente medidos pelos sensores serão muito mais elevados que na lâmpada. Assim, é possível diferenciar as duas cargas simplesmente definindo um nível de corrente no qual é esperado (conforme as medições realizadas previamente) que cada uma dessas cargas opere. Por exemplo, para uma lâmpada incandescente testada previamente com seus níveis de corrente variando ao longo do período de amostragem entre 0,38 e 0,40 A, podemos definir uma faixa razoável para a operação desta lâmpada, considerando uma variação desse intervalo devido ao erro do sensor de corrente escolhido de 1,5% (considerando uma operação em temperatura ambiente), como definido nos dados técnicos do sensor de corrente utilizado, um ACS712 [8]. Assim é razoável afirmar que a operação da lâmpada se dará durante praticamente todo tempo, em termos de corrente elétrica, na seguinte faixa de corrente:

$$0,985 * 0,38 A \leq I_{rms} \leq 1,015 * 0,40 A \quad (1)$$

Ou seja:

$$0,37 \text{ A} \leq I_{rms} \leq 0,41 \quad (2)$$

E sabe-se ainda que, por se tratar de uma carga resistiva, é esperado que:

$$\text{FP} = 1 \quad (3)$$

Assim, é possível criar um código computacional, que avalie se o valor de corrente calculado em tempo real, a partir dos sinais dos sensores, está dentro do intervalo descrito na Equação (2) e se seu fator de potência é unitário. Caso o código valide as condicionais acima e, assegurando que nenhuma outra carga analisada tem essa faixa de valores de corrente em regime permanente, podemos afirmar (considerando apenas as cargas que foram pré-definidas no projeto) que a carga ligada é uma lâmpada incandescente. E podemos diferenciar essa lâmpada de qualquer outra das cargas analisadas utilizando os mesmos princípios. Se as condicionais acima forem satisfeitas sabemos, por exemplo, que não se trata de uma lâmpada fluorescente (carga eletrônica), pois seu fator de potência não seria unitário.

Utilizando-se essa lógica podemos diferenciar todas as cargas analisadas nesse projeto pois as cargas trabalham em faixas de valores de parâmetros de regime permanente que diferem entre si, em pelo menos uma das grandezas analisadas.

1.5.3 Limitações do Projeto

Neste projeto, para a detecção das cargas foram utilizados os valores de parâmetros (tensão, corrente, potência ativa, potência aparente e fator de potência) coletados individualmente em cada carga e esses valores foram salvos em variáveis e comparados com os sinais que eram obtidos pelos sensores. A partir das semelhanças entre as grandezas pré-determinadas e as grandezas obtidas em tempo real, era definido qual tipo de carga estava gerando a excitação dos sensores. Mas, essa metodologia ao passo que gera facilidades na detecção e diferenciação das cargas, gera também algumas limitações.

Quando definimos os parâmetros das cargas individualmente, esbarramos no problema de duas cargas serem ligadas de forma simultânea, tirando os valores da leitura dos sensores dos níveis conhecidos pelo *software*, fazendo com que o programa entenda duas cargas ligadas simultaneamente como um erro na leitura ou

algum tipo de falha nas cargas. Por exemplo, quando definimos que uma lâmpada incandescente irá gerar um sinal de potência ativa em torno de 60 W e fator de potência unitário, o código só reconhecerá essa carga se esses forem os níveis lidos em tempo real. No caso de ligarmos duas lâmpadas incandescentes de forma simultânea, o fator de potência permanecerá unitário, porém os níveis de potência ativa iriam saltar para algo em torno de 120 W, o que estaria fora da faixa conhecida pelo *software* para o funcionamento da lâmpada incandescente, portanto ele não reconheceria aquele sinal e não o identificaria como sendo a lâmpada. Para solucionar este problema, podemos fazer testes com duas cargas ligadas simultaneamente armazenando os valores de regime permanente em uma variável, e então compará-los com os valores lidos em tempo real. Um teste deste tipo foi realizado utilizando uma lâmpada incandescente e uma lâmpada fluorescente funcionando simultaneamente e o resultado está descrito no Capítulo 9 deste relatório.

Outro ponto importante é que, se a unidade consumidora possui duas cargas semelhantes (duas geladeiras de um mesmo fabricante e trabalhando com mesma regulação de termostato), por exemplo, é possível definir que a carga está ligada, mas não qual das duas cargas está ligada. No caso do exemplo citado, seria possível definir que uma geladeira está ligada, mas não qual delas.

Além disso, quando uma nova carga for inserida no sistema, os seus novos parâmetros de funcionamento devem ser adicionados ao *software* para que o mesmo possa fazer a detecção da carga. Ou seja, este projeto exige uma atualização de dados no programa sempre que uma modificação é feita na estrutura ou na configuração da unidade consumidora.

2 O PROTÓTIPO DO DETECTOR DE EVENTOS

Será apresentado neste capítulo o embasamento teórico e a seguir o desenvolvimento do protótipo do detector de eventos.

2.1 Embasamento Teórico

No monitoramento não intrusivo, o reconhecimento do funcionamento das cargas é dado através da análise dos sinais de tensão e corrente instantâneas no ramal principal, conforme já descrito. Com esses dados em mãos, podemos obter algumas informações importantes que compõem a assinatura dessas cargas: seus valores de tensão RMS, corrente RMS, níveis de potência ativa, reativa e aparente, além de seu fator de potência.

2.1.1 Valores médios quadráticos

Para falar de cálculo de potência é preciso definir o conceito de valor eficaz, ou Root Mean Square (RMS). O valor RMS de uma função periódica é definido como a raiz quadrada do valor médio da função ao quadrado, como pode ser visto na equação abaixo, sendo $x(t)$ o valor instantâneo da grandeza (tensão ou corrente) e T o período do sinal [9].

$$X_{rms} = \sqrt{\frac{1}{T} \int_t^{t+T} [x(t)]^2 dt} \quad (4)$$

Esse conceito será importante para a compreensão dos cálculos de potência ativa e aparente realizados à frente.

2.1.2 Potência instantânea

Conhecidos os valores instantâneos de tensão e corrente, podemos definir a potência instantânea como mostrado abaixo:

$$p(t) = v(t) * i(t) \quad (5)$$

Sendo que:

$$v(t) = V * \sin(\omega t) \quad (6)$$

$$i(t) = I * \sin(\omega t - \theta) \quad (7)$$

2.1.3 Potência ativa

Como a potência instantânea varia a cada amostra feita, é conveniente usar o valor médio da potência. Assim, de acordo com o Teorema do Valor Médio de Lagrange e utilizando as identidades trigonométricas e as substituições explicitadas em [9] temos:

$$P = \frac{1}{T} \int_t^{t+T} p(t) dt = V_{RMS} * I_{RMS} * \cos(\theta) = P_{ativa} \quad (8)$$

O resultado da equação acima, fisicamente representa a potência demandada pela carga a cada instante de tempo dividido pelo período em que a grandeza foi integralizada, ou seja, representa a taxa de variação média da potência drenada pela carga. Essa potência é denominada Potência Ativa.

Assim, de forma mais clara, se os valores de potência instantânea são conhecidos, basta integralizar esses valores em um período de tempo T e dividirmos pelo mesmo período T para obtermos a potência ativa.

2.1.4 Potência Aparente

Compreendido o conceito de valor eficaz (RMS) como explicado acima, é mostrado em NILSSON (2007) que a potência aparente (que tem como unidade de medida o Volts-Ampères) pode ser calculada como mostrado na Equação 9. Esse cálculo mostra a relação trigonométrica entre as potências ativa (P), reativa (Q) e aparente (S).

$$S = V_{rms} * I_{rms} \quad (9)$$

Essa potência também pode ser expressa na forma complexa como mostrado a seguir:

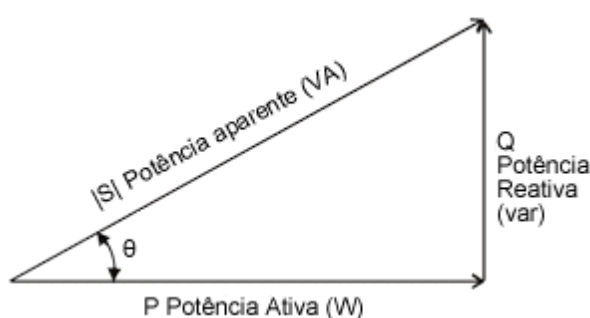
$$S = P + jQ \quad (10)$$

O módulo dessa potência complexa é o que denominamos potência aparente e pode ser escrita como na fórmula abaixo:

$$|S| = \sqrt{P^2 + Q^2} = V_{rms} * I_{rms} \quad (11)$$

A grande vantagem que temos utilizando a forma complexa da potência aparente é a interpretação gráfica que a mesma possibilita. Imaginando as grandezas P, Q e S como os lados de um triângulo retângulo podemos representar esse triângulo de potências como mostrado na Figura 3:

Figura 3 - Triângulo de potências



Fonte: elaborado pelo autor

2.1.5 Fator de potência

Observando o triângulo de potência na Figura 3 e baseando-se nas relações trigonométricas básicas, pode-se afirmar que o fator de potência de um sistema é o cosseno do ângulo θ , ou seja, a razão entre a potência ativa e a potência aparente do sistema. Formulando essa afirmação, tem-se:

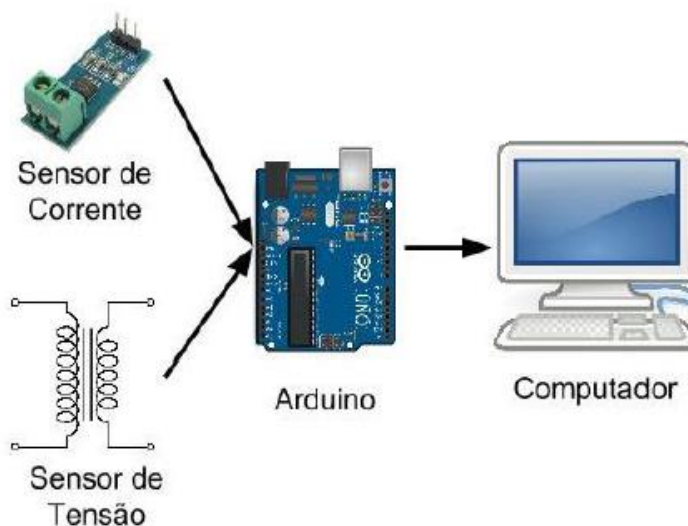
$$FP = \frac{|P|}{|S|} \quad (12)$$

Conceitualmente, se um sistema tem baixo fator de potência em relação a outro, isso implica que ele drenará um maior valor de corrente para realizar o mesmo trabalho. Esse aumento no valor da corrente traz aumento de perdas no sistema, aquecimento, acarreta em um projeto com condutores de maior bitola (o que torna o projeto mais caro), entre outros problemas. Portanto, é desejável que o sistema tenha seu fator de potência o mais próximo possível do valor unitário.

2.2 Protótipo do Detector de Eventos

Como pudemos observar do embasamento teórico anteriormente descrito, para a análise de eventos na curva de carga necessitamos dos sinais de tensão e corrente instantâneos. Utilizando um microcontrolador e sensores de corrente e tensão, podemos processar os sinais lidos e obter as grandezas necessárias (potência ativa, potência aparente, fator de potência) para podermos criar uma assinatura para cada tipo de carga e então diferenciá-las através dessas características conforme as mesmas são ligadas à rede. A Figura 4 abaixo mostra o esquema utilizado na obtenção dos sinais de tensão e corrente através dos sensores e do microcontrolador.

Figura 4 - Esquema de obtenção dos sinais de tensão e corrente



Fonte: Viega, 2016, p. 46.

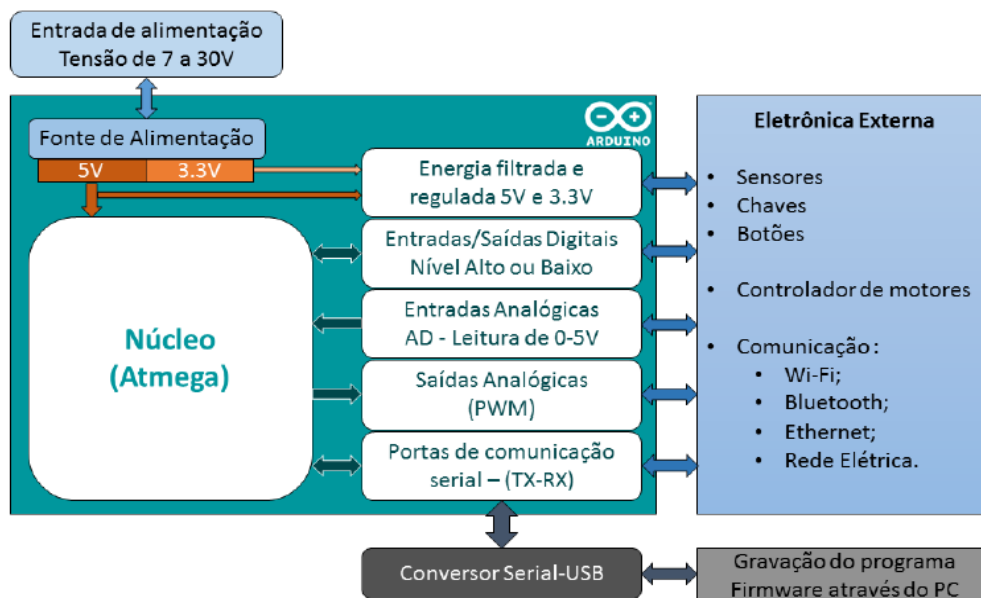
2.2.1 A plataforma Arduino

O Arduino é uma plataforma de prototipagem eletrônica composta por um microcontrolador Atmel, com entradas e saídas analógicas e digitais que podem ser facilmente conectadas a um computador e programado via IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++.

A plataforma Arduino é um projeto totalmente aberto baseado em hardware e *software* flexíveis e de fácil utilização, criado com o objetivo de permitir o desenvolvimento de

controle de sistemas interativos, de baixo custo e acessível a todos. A Figura 5 abaixo apresenta a arquitetura do Arduino.

Figura 5 - Arquitetura da plataforma Arduino

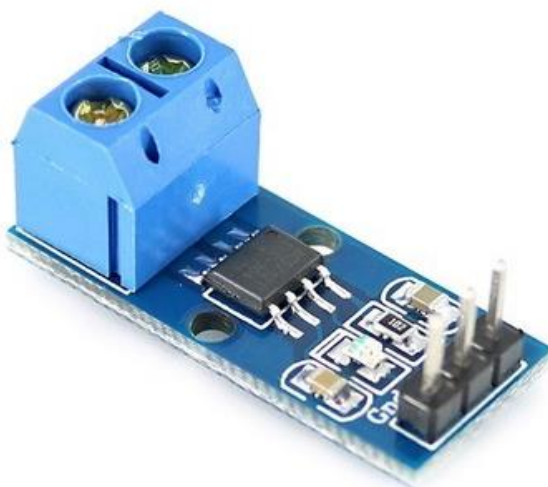


Fonte: www.embarcados.com.br (2014)

2.2.2 O sensor de corrente

Para a obtenção dos sinais de corrente, foi utilizado o sensor ALLEGRO ACS712. Esse sensor utiliza o efeito Hall para detectar o campo magnético gerado pela passagem de corrente, gerando na saída do sensor uma tensão proporcional ao valor da corrente numa proporção de 66mV/A , sendo capaz de medir uma corrente de até 30 Amperes. Outras informações técnicas do sensor podem ser encontradas em ALLEGRO 2017 [8]. A Figura 6 mostra o sensor ACS712.

Figura 6 - Sensor de corrente ACS 712



Fonte: <http://www.filipeflop.com> (2015)

2.2.3 O sensor de tensão

Para a obtenção do sinal de tensão, foi utilizado um transformador comercial com relação 10:1 (como visto na Figura 7) da Hayonik, com erro de transformação de 1,5% em temperatura ambiente como informado no catalogo do fabricante [20], e um circuito auxiliar para a adequação dos níveis de tensão para que ficassem compatíveis com os níveis de leitura na entrada analógica do Arduino.

Figura 7 - Transformador de tensão Hayonik com relação 10:1

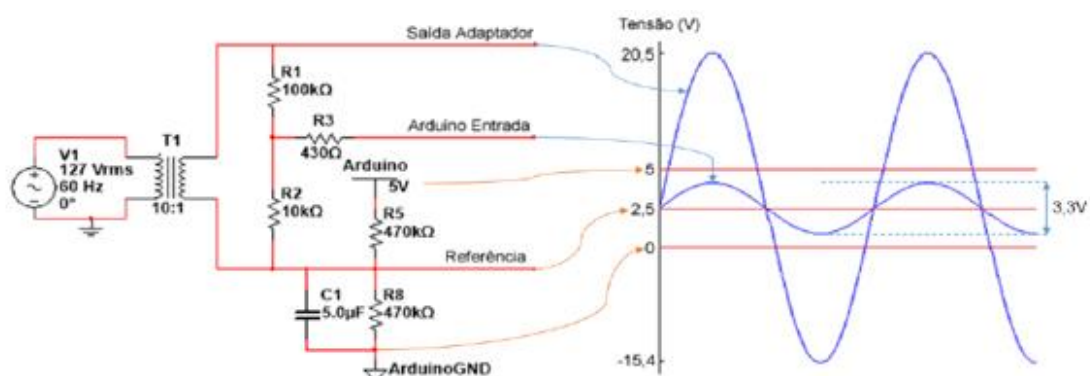


Fonte: <http://www.waltronica.com.br> (2016)

O primário do transformador é conectado ao mesmo ponto (em paralelo) onde as cargas serão ligadas. Assim, quando uma alteração ocorre na tensão das cargas, essa alteração é imediatamente refletida para o secundário do transformador que está conectado ao circuito de adequação de tensão. Desta forma, a partir de variáveis que ajustam as tensões recebidas no Arduino (que tem valores de tensão mais baixos para torná-los compatíveis ao microcontrolador) para os valores reais, é possível obter o valor da tensão na carga em qualquer instante.

O circuito de adequação e a forma de onda de tensão esperada na saída foram montados no *software* de projetos de circuitos elétricos Multisim e são mostrados na Figura 8. Desta forma, a referência para um valor nulo de tensão na carga é uma tensão de 2,5 V na entrada do Arduino, e sua variação máxima em torno da referência ocorrerá numa faixa de 3,3 V, o que garante a integridade física das portas analógicas do microcontrolador.

Figura 8 – Circuito de adequação da tensão e os sinais esperados

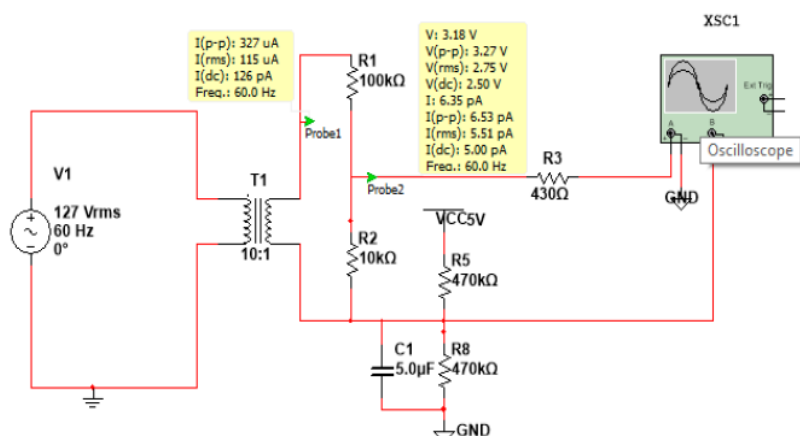


Fonte: elaborada pelo autor

Esta adequação se fez necessária devido às portas analógicas do Arduino trabalharem com valores entre 0V e 5V, e o objetivo deste circuito é fazer com que a referência (quando há sinal nulo de tensão no sensor) fique em 2,5V e que este valor varie senoidalmente entre 0,85V e 4,15V em condições normais (127V).

Com a ajuda do *software* Multisim, foi simulado o funcionamento dos sensores no protótipo para análise dos valores que serão entregues ao Arduino. As Figuras 9 e 10 abaixo mostram o circuito simulado com os sensores e o resultado dessa simulação, respectivamente.

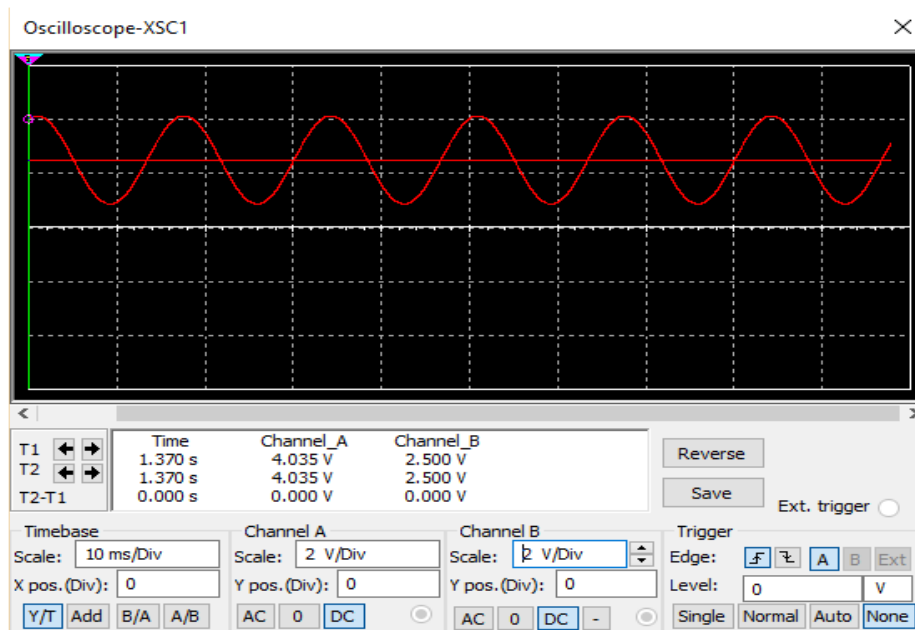
Figura 9 - Simulação do circuito de adequação no Multisim



Fonte: elaborada pelo autor

As ondas geradas são mostradas na Figura 10 e os valores de tensão são mostrados nos canais A e B do display da mesma figura:

Figura 10 - resultado da simulação em Multisim do circuito de adequação



Fonte: elaborada pelo próprio autor

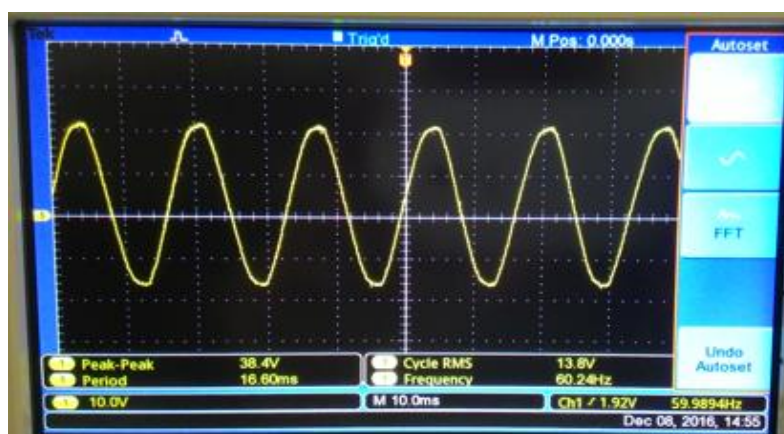
Da Figura 10 podemos observar que, em simulação, os sensores de tensão e corrente funcionavam bem, entregando os valores de tensão e corrente instantâneos quando solicitado e na faixa desejável de operação do Arduino.

Nos laboratórios de Eletrônica, foi montado o circuito de adequação simulado acima para que os testes fossem realizados. É possível notar a eficiência do circuito

observando os resultados obtidos abaixo e os comparando com os resultados esperados de acordo com as simulações.

Na Figura 11 observamos o sinal da saída do secundário do transformador, com uma tensão RMS de 13,0 V, um pouco acima do valor teórico de 12,7 V devido à imprecisão do transformador utilizado. Essa diferença pode ser corrigida no programa principal com a calibração das variáveis associadas ao cálculo da tensão RMS.

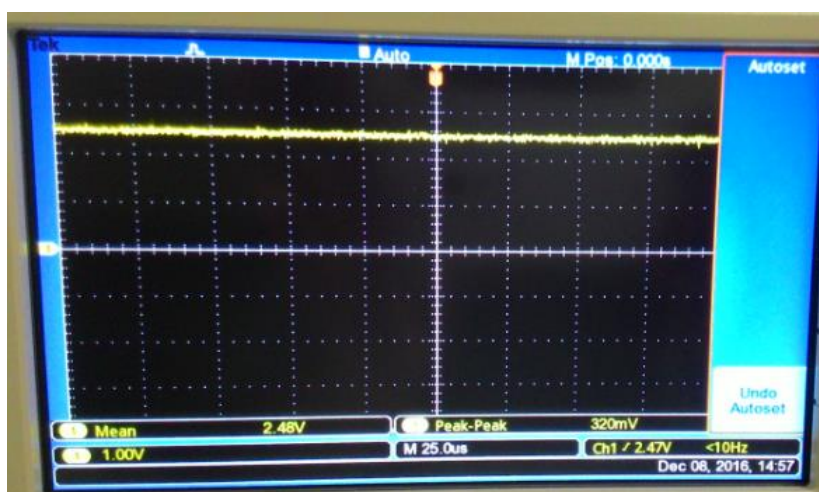
Figura 11 - Resultados obtidos em laboratório para o circuito de adequação da tensão



Fonte: elaborada pelo autor

Na Figura 12 temos o sinal contínuo da referência da tensão em 2,48 V, muito próximo do valor teórico de 2,5 V. Nesse valor de tensão, o microcontrolador entende uma tensão nula na carga

Figura 12 - Sinal de referência da tensão próximo de 2.5 V



Fonte: elaborada pelo autor

Na Figura 13 abaixo, finalmente o sinal de entrada do microcontrolador para um sistema em vazio. A partir disso qualquer distúrbio de tensão na carga seria refletido para o secundário do Trafo e, conseqüentemente, lido pelo microcontrolador.

Figura 13 - Sinal senoidal de entrada da tensão no microcontrolador



Fonte: elaborada pelo autor

Podemos observar que o sinal senoidal tem um *offset* de 2.5 V e uma variação total de 3.5 V, o que o torna adequado para o processamento no microcontrolador Arduino.

3 CÁLCULO DOS PARÂMETROS EM TEMPO REAL E COMUNICAÇÃO MODBUS

O código computacional inserido no processo de identificação das cargas utilizadas neste projeto foi dividido basicamente em duas partes: a primeira foi implementar o código capaz de processar em tempo real as informações dos sensores e executar o cálculo dos parâmetros de regime permanente; e a segunda foi criar uma comunicação via protocolo Modbus entre o computador e o microcontrolador para gerenciar o processo através de uma interface simples e intuitiva.

3.1 O algoritmo de cálculo dos parâmetros de regime permanente

A etapa seguinte do projeto foi o desenvolvimento do algoritmo em linguagem C responsável pelos cálculos de tensão RMS, corrente RMS, das potências ativa e aparente, além do fator de potência baseados nos cálculos mostrados no embasamento teórico visto na seção 2.1. Para auxiliar no desenvolvimento desse código foi utilizada a biblioteca “EmonLib.h” (apêndice B) do projeto de código aberto “OpenEnergyMonitor” que foi criada com o propósito de facilitar projetos na área de sistema de energia [10].

Para o cálculo das tensões e correntes RMS, o código acumula o quadrado dos valores instantâneos lidos pelos sensores em uma variável. Esse processo é repetido até que uma variável temporizada chegue a um valor pré-determinado. Então, a variável com os valores acumulados ao quadrado é dividida pela quantidade de amostras de valores instantâneos coletadas durante o ciclo temporizado, e o resultado dessa divisão é submetido a uma operação de raiz quadrada, resultando no valor RMS da tensão ou da corrente elétrica. E, como descrito no seção 2.1 o produto dos valores RMS da tensão e da corrente resultará na Potência Aparente.

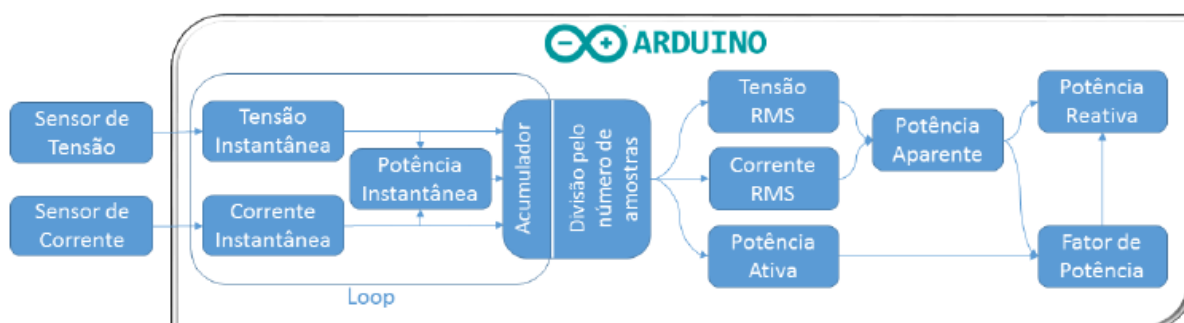
Para o cálculo da potência ativa utilizou-se um método parecido com o utilizado para os cálculos das grandezas RMS. Dentro do mesmo loop temporizado, foi criada uma terceira variável de acumulação, a de potência instantânea, que acumula simplesmente o produto entre tensão e corrente instantâneas. Ao fim do ciclo temporizado, a variável é dividida pela quantidade de amostras lidas no ciclo e então obtém-se a Potência Ativa.

Sendo a potência ativa calculada a partir da integralização do valor médio da potência instantânea, e a potência aparente calculada através do produto entre tensão RMS e corrente RMS, e sabendo que,

$$FP = \frac{P}{S} \quad (13)$$

Podemos facilmente obter então o Fator de Potência. O fluxograma do algoritmo atualizado é mostrado na Figura 14:

Figura 14 – Fluxograma do algoritmo de cálculo dos parâmetros de regime permanente



Fonte: Viegas, 2016, p. 47

3.2 Definição do valor do *Identificador*

Com o algoritmo para o cálculo dos valores dos parâmetros já estabelecido, faltava ainda diferenciar as cargas de acordo com os valores dos parâmetros calculados. Para isso foi criado um novo trecho de código para associar uma variável, chamada *identificador*, com cada tipo de carga (apêndice A). A essa variável é associado um número inteiro de 0 a 7, que está relacionado a uma das cargas de acordo com a Tabela 1:

Tabela 1 - Relação entre o tipo de carga e valor da variável *identificador*

Valor do <i>Identificador</i>	Carga ligada
0	À vazio
1	Lâmpada incandescente
2	Lâmpada fluorescente
3	Ventilador
4	Ferro de passar
5	Refrigerador
6	Forno de micro-ondas
7	Cargas simultâneas

Para definir qual valor a variável *identificador* receberia a cada instante, foi necessário inserir um novo trecho de código em C/C++ para associar o *identificador* aos valores dos parâmetros de regime permanente calculados em tempo real a partir da comparação do que foi calculado pelo programa com os padrões de valores pré-estabelecidos. Esses padrões são obtidos de testes individuais nas cargas e esses dados são salvos no programa.

Quando o *software* define em qual valor está a variável *identificador* (através de comparações com testes feitos previamente), a partir de uma comunicação via protocolo Modbus, essa informação é passada ao *software* ScadaBr, que associa cada valor do *identificador* a uma representação gráfica e o resultado é apresentado ao usuário através de uma interface gráfica.

3.3 O protocolo Modbus

Como já dito, foi utilizado um protocolo Modbus no Arduino para realizar a comunicação do microcontrolador com o supervisor do ScadaBR. Este é o protocolo da camada de aplicação (no modelo OSI) projetado para transportar mensagens e fornecer comunicação cliente/servidor onde os dispositivos podem estar conectados a diferentes tipos de rede e barramentos. É um protocolo padrão na indústria desde 1979, muito bem aceito por ter sido desenvolvido com o foco em aplicações industriais, open-source, fácil de implantar, manter e transmitir dados pela rede sem restrição de fornecedor ou fabricante. (MODBUS, 2016)

O protocolo Modbus é um protocolo de requisição/resposta de mestre/escravo que oferece serviços e funcionalidades especificadas por um *function code*. Neste trabalho foi utilizado o modo de transmissão serial com portas USB e principalmente a função “Ler registrador de *holding*” (Apêndice C). O formato de pacotes utilizado foi o RTU, e o servidor rodando o ScadaBR é o mestre, enquanto o Arduino que recebe os dados da rede elétrica através dos sensores e transmite para o computador é o escravo.

3.4 Composição dos erros dos sensores de tensão e corrente

Um cuidado importante tomado nesse projeto foi considerar o erro associado à imprecisão dos sensores de tensão e corrente utilizados. Para garantir que essa imprecisão não interferisse no processo de detecção das cargas, foi criada uma compensação que considera essa imprecisão dos sensores no cálculo de cada um dos parâmetros de regime permanente.

Usando o valor do erro informado no *datasheet* do sensor de corrente ACS712 [8], podemos considerar seguramente nas medições a mesma margem de erro, que é de 1,5% no valor da amostra para uma temperatura de operação de 25° C. Assim, podemos calcular a variação real de uma amostra coletada nos testes de acordo com a fórmula descrita abaixo:

$$(I - 0,015 * I) \leq I_{real} \leq (I + 0,015 * I) \quad (14)$$

Assim, temos:

$$0,985 * I \leq I_{real} \leq 1,015 * I \quad (15)$$

Para a tensão, podemos ter o mesmo raciocínio, já que, segundo o fabricante [20], também temos um erro aproximado de transformação que chega a 1,5%. Ou seja, os limites superior e inferior da condicional da tensão para a identificação da carga serão:

$$(V - 0,015 * V) \leq V_{real} \leq (V + 0,015 * V) \quad (16)$$

Assim podemos definir a variação das amostras da tensão devido ao erro como sendo:

$$0,985 * V \leq V_{real} \leq 1,015 * V \quad (17)$$

Já para a potência ativa, temos que considerar a forma como a mesma é calculada. Como trata-se de uma integralização do produto de tensão e corrente, podemos unificar o erro desses dois parâmetros como descrito abaixo (como demonstração, será feita apenas a análise para o erro associado ao limite inferior da condicional):

$$p(t) = [0,985 * v(t)] * [0,985 * i(t)] \quad (18)$$

$$P = \frac{1}{T} \int_t^{t+T} 0,97 * p(t) dt = 0,97 * V_{RMS} * I_{RMS} * \cos(\theta) \quad (19)$$

Assim, podemos notar um erro de 3% no cálculo da potência ativa, quando os erros dos sensores de tensão e corrente são compostos. Logo, podemos definir a variação das amostras da potência aparente devido ao erro como sendo:

$$0,97 * P \leq P_{real} \leq 1,03 * P \quad (20)$$

O mesmo raciocínio é usado para calcularmos o erro da potência aparente, já que a mesma é calculada como mostrado abaixo:

$$S = V_{rms} * I_{rms} \quad (21)$$

Compondo os erros temos:

$$S = (0,985 * V_{rms}) * (0,985 * I_{rms}) \quad (22)$$

Assim, temos:

$$S = 0,97 * V_{rms} * I_{rms} \quad (23)$$

Assim, vemos que o erro composto que se reflete na potência aparente devido aos erros dos sensores de tensão e corrente, é de 3%. Assim podemos definir a variação das amostras da potência aparente devido ao erro como sendo:

$$0,97 * S \leq S_{real} \leq 1,03 * S \quad (24)$$

Já para o fator de potência, temos a seguinte expressão:

$$FP = \frac{|P|}{|S|} \quad (25)$$

Como os cálculos de potência ativa e aparente são realizados a cada ciclo a partir de um mesmo conjunto amostral, podemos afirmar com segurança que, se os valores das amostras variam devido ao erro, essa variação afetará na mesma proporção os cálculos de potência ativa e aparente. Assim, compondo os erros de acordo com essa lógica acima e considerando os erros já calculados, temos:

$$FP = \frac{(1+0.03)|P|}{(1+0.03)|S|} \text{ ou } FP = \frac{(1-0.03)|P|}{(1-0.03)|S|} \quad (26)$$

Ou seja, para qualquer um dos casos temos:

$$FP = 1,0 * \frac{|P|}{|S|} \quad (27)$$

Portanto, o erro que se reflete sobre o fator de potência devido a imprecisão dos sensores de tensão e corrente, teoricamente, se anula. Mas para garantir que os valores de fator de potência possam variar dentro de um pequeno intervalo sem que o código perca a capacidade de identificar a carga, vamos definir a variação do fator de potência por erro das amostras como sendo 1%. Assim podemos definir a variação das amostras da potência aparente devido ao erro como sendo:

$$0,99 * FP \leq FP_{real} \leq 1,01 * FP \quad (28)$$

Podemos sintetizar esses valores na Tabela 2 como mostrado abaixo:

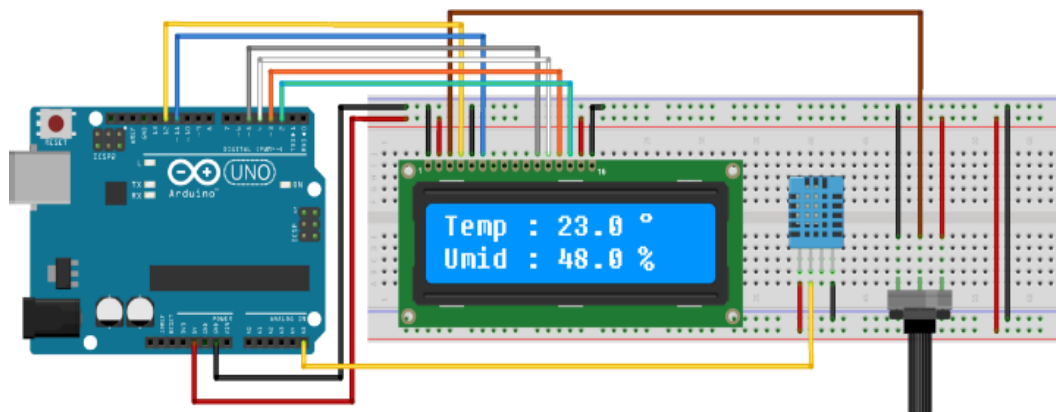
Tabela 2 - Erro associado a cada grandeza devido à imprecisão dos sensores

<i>Parâmetro</i>	<i>Varição devido ao erro dos sensores (%)</i>
Tensão	1.5
Corrente	1.5
Potência ativa	3.0
Potência aparente	3.0
Fator de potência	1.0*

4 A INTERFACE DE USUÁRIO E O SCADABR

Inicialmente o projeto foi pensado totalmente na plataforma Arduino, ou seja, os dados seriam coletados pelos sensores, os cálculos seriam realizados pelo microcontrolador e os resultados seriam apresentados em um display LCD, semelhante ao exemplo mostrado na Figura 15:

Figura 15 - Display de LCD como possibilidade de apresentação de resultados no Arduino



Fonte: www.felipeflop.com

Mas, durante a execução do projeto foi possível observar que, para tornar o produto final algo mais comercial e de compreensão mais simples, seria necessária a criação de uma interface mais intuitiva para a apresentação dos resultados.

Para a criação dessa interface foram analisadas várias possibilidades até que foi pensada uma solução já muito utilizada na indústria de sistemas supervisórios, o SCADABR.

Um sistema supervisório permite o monitoramento de informações e variáveis de um processo produtivo. Esses sistemas são conhecidos como SCADA (*Supervisory Control and Data Acquisition* – Controle Supervisório e Aquisição de Dados). A função principal do SCADA é proporcionar uma apresentação clara das informações do processo para facilitar o monitoramento pelo operador do sistema.

A interface do supervisório desenvolvida para o sistema de detecção de cargas foi criada utilizando-se o programa SCADABR. Trata-se de um *software* específico para a criação de sistemas supervisórios, oferecido em licença Open Source e com diversos usos em várias áreas da indústria como em variados processos produtivos, redes de distribuição de energia, redes de saneamento básico, automação predial e residencial e outras aplicações de sensoriamento diversas.

Sua interface é de fácil utilização e já possui ferramentas de visualização de gráficos, variáveis, configurações, alertas, protocolos, estatísticas, entre outros. [12]

A forma de acesso ao supervisório funciona através de um sistema web server que visa disponibilizar os dados do processo através da rede. Com isto, os clientes em vez de acessarem os dados através de um *software* instalado no computador ou smartphone, podem acessar via browser de internet. O banco de dados utilizado é o Apache Derby, também open source, porém *mySQL* também pode ser utilizado [12]. Para a realização deste projeto, foram utilizados, basicamente, dois tipos de dados compatíveis com o SCADABR: o *Serial Modbus* e o *Data Source Meta*, que são explicados a seguir.

4.1 Dados do tipo Serial Modbus

A porta USB utilizada pelo Arduino foi configurada como sendo o *Data Source* (fonte de dados), ou seja, o Arduino envia os dados por essa porta e esses dados são coletados pelo ScadaBr. Para que o *Data Source* funcione corretamente, é necessário enviar os dados utilizando o protocolo Modbus para a comunicação entre o Arduino e o computador que funcionará como servidor. O *Data Source* envia requisições de leitura dos dados a cada período de tempo a ser determinado na configuração do *Data Source* (no caso deste projeto foi configurado um período de 100 ms entre as requisições) e todas as leituras adquiridas são guardadas nos data points. A configuração dos dados do tipo Serial Modbus é mostrada na Figura 16.

Figura 16 - Configuração dos dados Serial Modbus

The screenshot displays the configuration interface for serial Modbus data in ScadaBR. It is organized into four main panels:

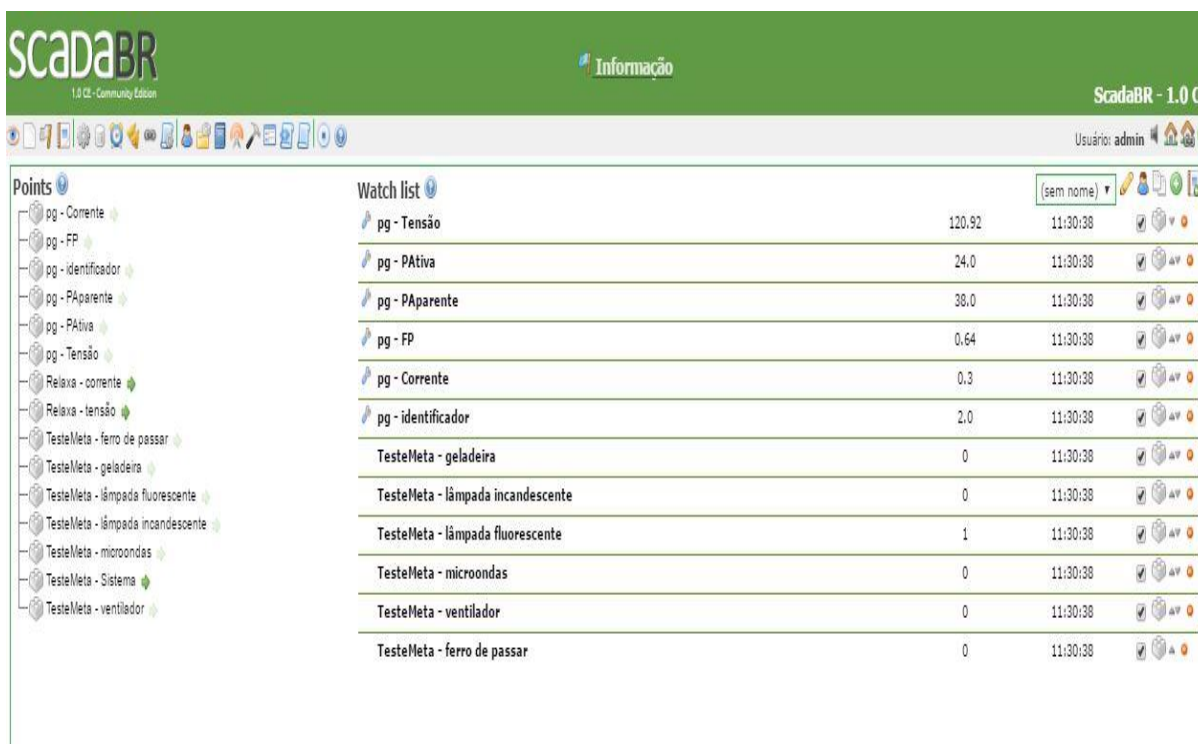
- Propriedades do modbus serial:** Contains fields for 'Nome' (PG), 'Export ID (XID)' (DS_779466), 'Período de atualização' (100 ms), 'Quantificação' (checkbox), 'Timeout (ms)' (2000), 'Tentativas' (2), 'Apenas quantidades contínuas' (checkbox), 'Criar pontos de monitor de escravo' (checkbox), 'Máxima contagem de leitura de bits' (2000), 'Máxima contagem de leitura de registradores' (125), 'Máxima contagem de escrita de registradores' (120), 'Porta' (dropdown), 'Baud rate' (9600), 'Controle de fluxo de entrada' (Nenhum), 'Controle de fluxo de saída' (Nenhum), 'Data bits' (8), 'Stop bits' (1), 'Parity' (Nenhuma), 'Codificação' (RTU), 'Echo' (Desligado), and 'Simultaneidade' (Função).
- Pesquisa de nós modbus:** Includes 'Pesquisar por nós' and 'Cancelar' buttons, and a list area for 'Nós encontrados'.
- Leitura de dados modbus:** Features 'Id do escravo' (1), 'Faixa do registro' (Status do coil), 'Offset (baseado em 0)' (0), and 'Número de registradores' (100), with a 'Ler dados' button.
- Teste de localizador de ponto:** Includes 'Id do escravo' (1), 'Faixa do registro' (Status do coil), 'Tipo de dados modbus' (Binário), 'Offset (baseado em 0)' (0), 'Bit' (0), 'Número de registradores' (0), and 'Codificação de caracteres' (ASCII), with 'Ler' and 'Adicionar ponto' buttons.
- Níveis de alarme de eventos:** Contains 'Exceção de data source' (Nenhum alarme), 'Exceção de leitura de data point' (Nenhum alarme), and 'Exceção de escrita em data point' (Nenhum alarme).

Fonte: *Print screen* do software ScadaBR

4.2 Dados do tipo Source Meta

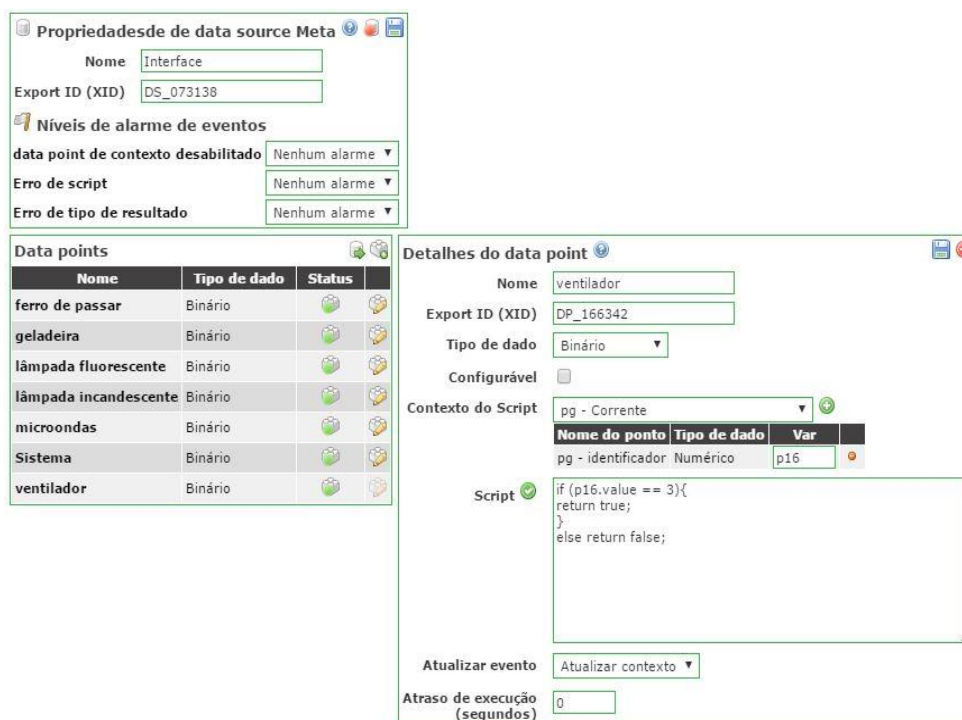
Existem duas formas distintas de se monitorar os dados do ScadaBr: a primeira é simplesmente monitorando o *WatchList*, uma lista dinâmica contendo todas as variáveis do projeto com seus respectivos valores de leitura mostrados em tempo real, como mostrado na Figura 17.

Figura 17 - Imagem da watchlist em funcionamento



Fonte: *Print screen do software ScadaBR*

Outra forma de monitorar os dados no ScadaBR é criando uma representação visual do seu processo, um supervisório, com gráficos, imagens, GIFS, botões, textos e várias outras funcionalidades. Para criar essas representações o ScadaBR oferece uma área chamada “Representação gráfica” que permite ao usuário criar uma interface com grande liberdade utilizando a ferramenta “*Data Source Meta*”, que dá ao usuário a possibilidade de controlar uma série de padrões de imagens, gráficos, textos em HTML e condicionar a apresentação de cada uma dessas imagens a uma ou mais variáveis do processo que está sendo controlado. O exemplo mostrado na Figura 18 mostra a configuração do *Data Source Meta* para a representação do Ventilador e a forma como essa representação gráfica fica condicionada à variável *identificador*.

Figura 18 - Configuração dos dados do tipo *Data Source Meta*

Fonte: *Print screen* do software ScadaBR

Na janela “detalhes do *data point*” vemos que a variável *identificador* (expressa no código pelo nome ‘p16’) é comparada ao valor 3, que é exatamente a identificação do ventilador na Tabela 1. Quando essa variável assume o valor 3, então o *Data Source* muda o estado da Interface, acendendo um led verde abaixo do símbolo do ventilador mostrando que o mesmo está ligado. E assim acontece a configuração para cada uma das cargas analisadas no projeto.

A partir dessas configurações do *Data Source Meta*, foi criada toda a Interface deste projeto como mostrada abaixo.

4.3 A interface com o usuário

A interface criada para monitorar o sistema de detecção de cargas é mostrada na Figura 19 e foi criada de forma intuitiva para facilitar o monitoramento do sistema.

Figura 19 - Interface de detecção de cargas



Fonte: *Print screen do software ScadaBR*

A interface conta com um botão ON/OFF na parte central superior da tela, que indica se o sistema está ou não em funcionamento. Enquanto o botão está vermelho, o sistema está inativo. Se o botão está verde, o sistema está ativo e alguma carga está conectada à rede.

Para o monitoramento das cargas pelo sistema, foi criado um padrão simbólico para cada tipo de carga a fim de gerar uma percepção visual nítida de uma carga que muda de estado. Esse padrão conta com uma imagem de cada carga analisada que muda de estado quando esta carga é ligada ou desligada. Para a lâmpada incandescente, por exemplo, foram utilizadas duas imagens: uma com a lâmpada acesa e uma com a lâmpada apagada. Conforme o Arduino detecta que a lâmpada foi ligada, o ScadaBR lê essa informação e muda o estado das imagens de acordo com a programação feita na criação da interface, mostrando na tela apenas a imagem da lâmpada acesa. O mesmo para todas as outras cargas do sistema.

Para aumentar ainda mais a percepção de ligada ou desligada, foi inserida a imagem de um led abaixo de cada carga, que é vermelho se a carga está inativa e verde quando a carga está ativa. A representação visual das cargas é mostrada mais claramente na Figura 20.

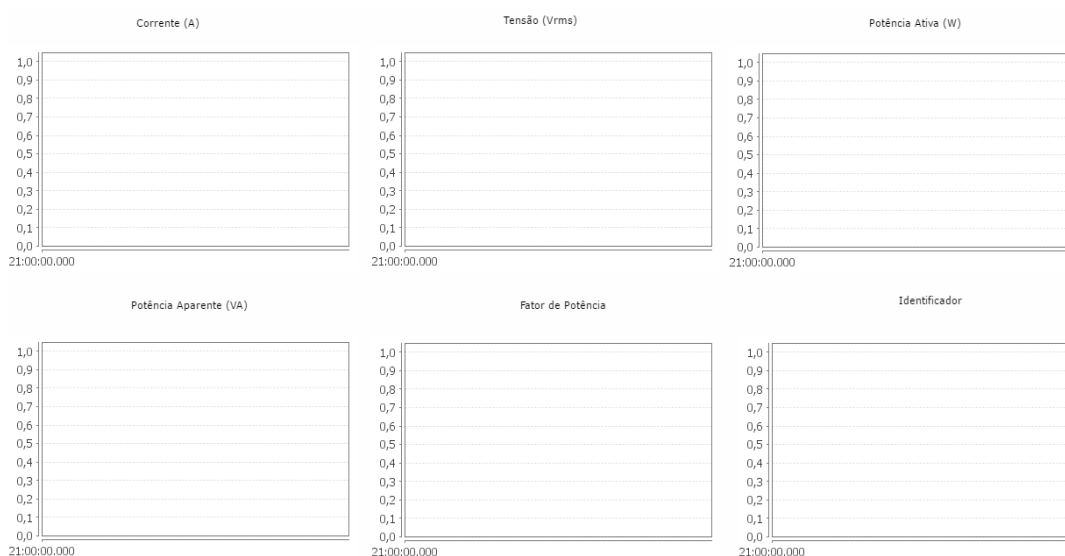
Figura 20 - Representação visual das cargas na interface de detecção



Fonte: *Print screen do software ScadaBR*

Além disso, a interface conta com seis gráficos abaixo da representação das cargas, para dar uma visão mais ampla das principais grandezas envolvidas no processo: tensão e corrente RMS, potência ativa, potência aparente e fator de potência. Há um gráfico ainda para monitorar o valor da variável *identificador* (para se certificar de que o código está funcionando corretamente). Os gráficos foram configurados para mostrar os valores coletados nos últimos 10 minutos de leitura do DataSource, e então eliminar os valores mais antigos para manter a representação sempre atualizada. Os gráficos apresentados na interface são mostrados na Figura 21.

Figura 21 - representação gráfica das grandezas envolvidas no projeto



Fonte: *Print screen do software ScadaBR*

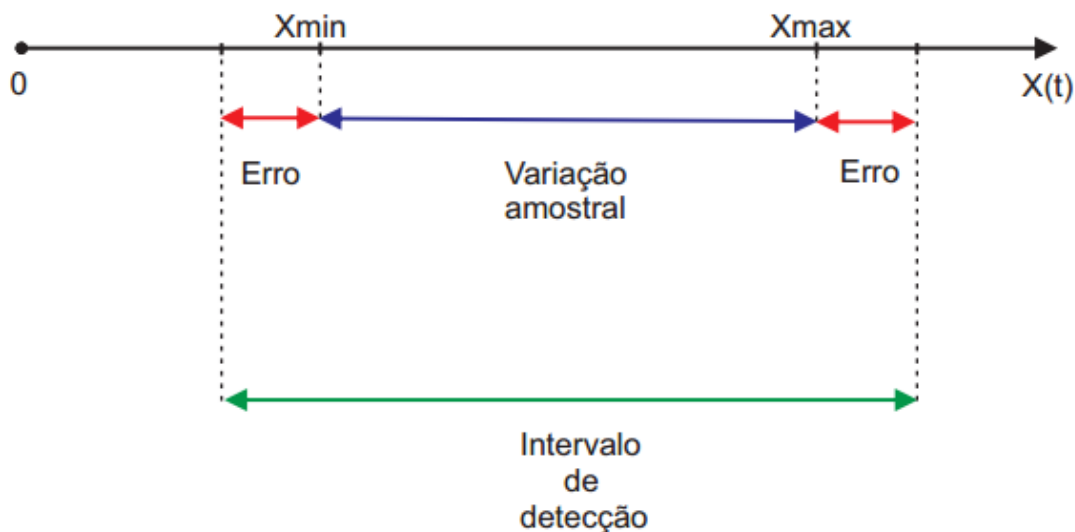
5 DETERMINAÇÃO DOS INTERVALOS DE IDENTIFICAÇÃO DAS CARGAS ATRAVÉS DOS PARÂMETROS DE REGIME PERMANENTE

Com a comunicação entre o ScadaBr e o Arduino já estabelecida, e a interface construída de forma intuitiva, faltava caracterizar individualmente as cargas através de seus parâmetros de regime permanente para poder criar um banco de dados que permitisse a detecção individual dessas cargas quando todas elas fossem integradas ao sistema.

Para a determinação das faixas de valores dos parâmetros de regime permanente que seriam usados para a caracterização de cada tipo de carga analisada no projeto, foram realizados testes de forma individual de modo a determinar os níveis de operação dessas cargas, considerando como variação desses níveis apenas o erro composto devido à imprecisão dos sensores, como descrito na Tabela 2.

Estes testes foram realizados ligando-se uma carga de cada vez e utilizando o *Serial Monitor* do Arduino para monitorar os parâmetros de regime permanente a fim de definir os intervalos de operação da carga para cada um dos parâmetros. Para cada conjunto amostral coletado no *Serial monitor*, utilizou-se o menor e o maior valor de cada parâmetro, associando-os ao seu erro (conforme a Tabela 2), e assim foi obtido o intervalo em que se esperava que aqueles parâmetros operassem possibilitando assim a identificação da carga quando a mesma fosse integrada ao sistema. O diagrama mostrado na Figura 22 ajuda a ilustrar melhor o modo como o código define o intervalo de detecção para o reconhecimento das cargas.

Figura 22 – Diagrama de definição do intervalo de detecção de um parâmetro X



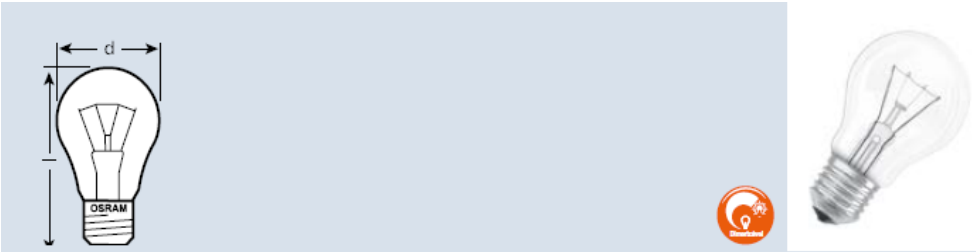
Fonte: elaborado pelo autor

Como já foi dito e podemos ver na Figura 22, o intervalo de detecção que é usado para definir se o parâmetro calculado está dentro do que foi pré-determinado, é exatamente o intervalo entre a menor e a maior amostra obtida nos testes considerando ainda o erro associado a cada parâmetro conforme a Tabela 2.

5.1 Lâmpada incandescente

Para o teste com a lâmpada incandescente foi utilizada uma lâmpada “Classic A CL” de 40W da Osram, como visto na Figura 23 abaixo. Seus dados técnicos podem ser encontrados na página 7 do catálogo de produtos da fabricante [14].

Figura 23 - Lâmpada incandescente utilizada nos testes de detecção



	V	W	lm	∅ d (mm)	l max. (mm)	Base	t(h)
	Tensão	Potência	Fluxo Luminoso	Diâmetro	Compr. máx.		Vida Média
CLASSIC A - BULBO CLARO							
CLAS A CL 25 W	127	25	230	55	94	E27	750
CLAS A CL 40 W	127	40	516	55	94	E27	750
CLAS A CL 25 W	220	25	210	55	94	E27	1000
CLAS A CL 40 W	220	40	415	55	94	E27	1000

Fonte: www.osram.com.br

A Figura 24 mostra um *frame* do *Serial Monitor* com os valores dos parâmetros de regime permanente obtidos pelo microcontrolador durante alguns instantes de tempo. Este quadro apresenta **34 amostras**, sendo coletadas a cada **500 ms** (17 segundos de observação são mostrados no quadro), e vale ressaltar que, para aumentar a confiabilidade dos dados, o *frame* só é coletado quando se observa uma boa estabilização dos valores dos parâmetros no *Serial Monitor*.

Figura 24 – Dados do *Serial Monitor* durante a ligação da lâmpada incandescente

```

Preal: 36.83 Papar: 37.13 Vrms: 121.70 Irms: 0.31 FP: 1.00
Preal: 36.82 Papar: 37.16 Vrms: 121.82 Irms: 0.31 FP: 1.00
Preal: 36.69 Papar: 37.02 Vrms: 121.83 Irms: 0.30 FP: 1.00
Preal: 36.63 Papar: 36.95 Vrms: 121.93 Irms: 0.30 FP: 1.00
Preal: 36.76 Papar: 37.10 Vrms: 121.86 Irms: 0.30 FP: 1.00
Preal: 36.39 Papar: 36.74 Vrms: 121.88 Irms: 0.30 FP: 1.00
Preal: 36.63 Papar: 37.03 Vrms: 122.13 Irms: 0.30 FP: 1.00
Preal: 36.81 Papar: 37.14 Vrms: 121.97 Irms: 0.30 FP: 1.00
Preal: 36.85 Papar: 37.18 Vrms: 121.85 Irms: 0.31 FP: 1.00
Preal: 36.93 Papar: 37.25 Vrms: 122.01 Irms: 0.31 FP: 1.00
Preal: 36.77 Papar: 37.09 Vrms: 121.79 Irms: 0.30 FP: 1.00
Preal: 37.03 Papar: 37.35 Vrms: 122.11 Irms: 0.31 FP: 1.00
Preal: 36.53 Papar: 36.85 Vrms: 121.95 Irms: 0.30 FP: 1.00
Preal: 37.01 Papar: 37.36 Vrms: 121.92 Irms: 0.31 FP: 1.00
Preal: 36.89 Papar: 37.21 Vrms: 122.04 Irms: 0.30 FP: 1.00
Preal: 37.02 Papar: 37.34 Vrms: 122.13 Irms: 0.31 FP: 1.00
Preal: 36.94 Papar: 37.24 Vrms: 122.05 Irms: 0.31 FP: 1.00
Preal: 36.60 Papar: 36.90 Vrms: 122.05 Irms: 0.30 FP: 1.00
Preal: 36.97 Papar: 37.24 Vrms: 122.02 Irms: 0.31 FP: 1.00
Preal: 36.77 Papar: 37.13 Vrms: 122.02 Irms: 0.30 FP: 1.00
Preal: 36.46 Papar: 36.79 Vrms: 122.03 Irms: 0.30 FP: 1.00
Preal: 36.72 Papar: 37.05 Vrms: 121.99 Irms: 0.30 FP: 1.00
Preal: 36.76 Papar: 37.05 Vrms: 122.02 Irms: 0.30 FP: 1.00
Preal: 36.70 Papar: 37.02 Vrms: 121.90 Irms: 0.30 FP: 1.00
Preal: 36.67 Papar: 37.00 Vrms: 121.93 Irms: 0.30 FP: 1.00
Preal: 36.69 Papar: 37.02 Vrms: 121.97 Irms: 0.30 FP: 1.00
Preal: 36.68 Papar: 36.97 Vrms: 121.94 Irms: 0.30 FP: 1.00
Preal: 36.53 Papar: 36.85 Vrms: 122.04 Irms: 0.30 FP: 1.00
Preal: 36.70 Papar: 37.06 Vrms: 122.02 Irms: 0.30 FP: 1.00
Preal: 36.79 Papar: 37.15 Vrms: 121.93 Irms: 0.30 FP: 1.00
Preal: 36.58 Papar: 36.97 Vrms: 121.94 Irms: 0.30 FP: 1.00
Preal: 36.76 Papar: 37.13 Vrms: 122.07 Irms: 0.30 FP: 1.00
Preal: 36.87 Papar: 37.18 Vrms: 121.94 Irms: 0.30 FP: 1.00
Preal: 36.93 Papar: 37.23 Vrms: 121.98 Irms: 0.31 FP: 1.00
Preal: 37.10 Papar: 37.40 Vrms: 121.92 Irms: 0.31 FP: 1.00

```

Fonte: *Print screen* do software Arduino

A partir desses valores, podemos criar a condição matemática na qual o *software* possa identificar a ligação desta lâmpada incandescente quando todas as cargas forem integradas ao sistema. Observando o quadro acima, foi montada a Tabela 3 que apresenta o valor máximo e o valor mínimo que cada um dos parâmetros poderá apresentar.

Tabela 3 – Valores mínimos e máximos dos parâmetros medidos na lâmpada incandescente

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Lâmpada incandescente Osram Centra A 60W	122.13	0.31	37.10	37.40	1
Valor Mínimo	Lâmpada incandescente Osram Centra A 60W	121.70	0.30	36.39	36.85	1

Observando os valores da tabela e aplicando os erros associados a cada parâmetro como mostrado na Tabela 2, temos que a identificação da lâmpada incandescente obedecerá a seguinte estrutura condicional:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * V_{max} \quad (29)$$

Ou seja:

$$0,985 * 121,7 \leq V_{rms} \leq 1,015 * 122,13 \quad (30)$$

Assim, temos:

$$119,9 \text{ V} \leq V_{rms} \leq 123,9 \text{ V} \quad (31)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (32)$$

Ou seja:

$$0,985 * 0,30 \leq I_{rms} \leq 1,015 * 0,31 \quad (33)$$

Assim, temos:

$$0,295 \text{ A} \leq I_{rms} \leq 0,315 \text{ A} \quad (34)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (35)$$

Ou seja:

$$0,97 * 36,39 \leq P \leq 1,03 * 37,1 \quad (36)$$

Assim, temos:

$$35,30 W \leq P \leq 38,21 W \quad (37)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (38)$$

Ou seja:

$$0,97 * 35,34 \leq S \leq 1,03 * 35,89 \quad (39)$$

Assim, temos:

$$34,27 VA \leq S \leq 36,97 VA \quad (40)$$

- **Fator de Potência (erro de 1.0%)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (41)$$

Ou seja

$$0,99 * 1 \leq FP \leq 1,01 * 1 \quad (42)$$

Assim, temos:

$$0,99 \leq FP \leq 1,01 \quad (43)$$

Se todas as condicionais forem satisfeitas, o *software* entende a carga como sendo uma lâmpada incandescente e, de acordo com a Tabela 2 o valor da variável *Identificador* será igual a 1.

5.2 Lâmpada fluorescente

Para os testes com a lâmpada fluorescente foi utilizada a lâmpada compacta CFLI-3U-AM da Avant, como visto na Figura 25. Seus dados técnicos podem ser encontrados na página 2 do catálogo de produtos da fabricante [15].

Figura 25 - Lâmpada fluorescente compacta usada nos testes de detecção



Fonte: www.avant.com.br

A Figura 26 mostra o *Serial Monitor* do Arduino enquanto a lâmpada fluorescente está ligada. Como se trata de uma lâmpada eletrônica era esperado um fator de potência abaixo do unitário.

Figura 26 – Dados do *Serial Monitor* durante a ligação da lâmpada fluorescente

```

Preal: 21.87 Papar: 33.12 Vrms: 121.23 Irms: 0.27 FP: 0.66
Preal: 21.66 Papar: 33.58 Vrms: 120.76 Irms: 0.28 FP: 0.64
Preal: 20.59 Papar: 31.83 Vrms: 121.40 Irms: 0.26 FP: 0.65
Preal: 21.43 Papar: 33.85 Vrms: 121.36 Irms: 0.28 FP: 0.63
Preal: 22.17 Papar: 34.38 Vrms: 121.22 Irms: 0.28 FP: 0.64
Preal: 21.98 Papar: 34.19 Vrms: 121.28 Irms: 0.28 FP: 0.64
Preal: 22.25 Papar: 34.29 Vrms: 121.54 Irms: 0.28 FP: 0.65
Preal: 20.38 Papar: 33.12 Vrms: 121.42 Irms: 0.27 FP: 0.62
Preal: 20.89 Papar: 31.60 Vrms: 121.43 Irms: 0.26 FP: 0.66
Preal: 19.69 Papar: 30.33 Vrms: 115.09 Irms: 0.26 FP: 0.65
Preal: 21.72 Papar: 33.31 Vrms: 120.95 Irms: 0.28 FP: 0.65
Preal: 20.67 Papar: 33.41 Vrms: 120.73 Irms: 0.28 FP: 0.62
Preal: 21.03 Papar: 32.81 Vrms: 120.69 Irms: 0.27 FP: 0.64
Preal: 19.63 Papar: 31.24 Vrms: 120.84 Irms: 0.26 FP: 0.63
Preal: 19.66 Papar: 31.34 Vrms: 120.63 Irms: 0.26 FP: 0.63
Preal: 21.41 Papar: 33.72 Vrms: 120.48 Irms: 0.28 FP: 0.64
Preal: 21.89 Papar: 33.14 Vrms: 120.74 Irms: 0.27 FP: 0.66
Preal: 21.67 Papar: 34.23 Vrms: 120.86 Irms: 0.28 FP: 0.63
Preal: 19.77 Papar: 31.75 Vrms: 120.72 Irms: 0.26 FP: 0.62
Preal: 20.61 Papar: 31.98 Vrms: 120.64 Irms: 0.27 FP: 0.64
Preal: 19.46 Papar: 31.44 Vrms: 120.74 Irms: 0.26 FP: 0.62
Preal: 21.81 Papar: 34.25 Vrms: 120.67 Irms: 0.28 FP: 0.64
Preal: 20.98 Papar: 32.08 Vrms: 120.66 Irms: 0.27 FP: 0.65
Preal: 19.77 Papar: 31.85 Vrms: 120.74 Irms: 0.26 FP: 0.62
Preal: 20.25 Papar: 32.45 Vrms: 120.61 Irms: 0.27 FP: 0.62
Preal: 21.57 Papar: 33.93 Vrms: 120.67 Irms: 0.28 FP: 0.64
Preal: 20.38 Papar: 33.16 Vrms: 120.74 Irms: 0.27 FP: 0.61
Preal: 19.92 Papar: 32.05 Vrms: 120.65 Irms: 0.27 FP: 0.62
Preal: 21.02 Papar: 32.28 Vrms: 120.63 Irms: 0.27 FP: 0.65
Preal: 19.33 Papar: 31.42 Vrms: 120.85 Irms: 0.26 FP: 0.62
Preal: 21.71 Papar: 32.73 Vrms: 120.68 Irms: 0.27 FP: 0.66
Preal: 21.50 Papar: 34.15 Vrms: 120.51 Irms: 0.28 FP: 0.63
Preal: 20.25 Papar: 32.69 Vrms: 120.62 Irms: 0.27 FP: 0.62
Preal: 19.97 Papar: 31.30 Vrms: 120.68 Irms: 0.26 FP: 0.64
Preal: 20.46 Papar: 31.74 Vrms: 120.72 Irms: 0.26 FP: 0.64

```

Fonte: *Print screen* do *software* Arduino

A partir desses valores, podemos criar a condição matemática na qual o *software* possa identificar a ligação desta lâmpada fluorescente quando todas as cargas forem integradas ao sistema. Observando o quadro acima, foi montada a Tabela 4 abaixo que apresenta o valor máximo e o valor mínimo que cada um dos parâmetros apresentou no quadro coletado:

Tabela 4 - Valores mínimos e máximos dos parâmetros medidos na lâmpada fluorescente

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Lâmpada Fluorescente	121.43	0.28	22.25	34.38	0.66
Valor Mínimo		120.51	0.26	19.33	30.33	0.61

Observando os valores da tabela e aplicando os erros associados a cada parâmetro como mostrado na Tabela 2, temos que a identificação da lâmpada incandescente obedecerá a seguinte estrutura condicional:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * V_{max} \quad (44)$$

Ou seja:

$$0,985 * 120,51 \leq V_{rms} \leq 1,015 * 121,43 \quad (45)$$

Assim, temos:

$$118,7 V \leq V_{rms} \leq 123,25 V \quad (46)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (47)$$

Ou seja

$$0,985 * 0,26 \leq X \leq 1,015 * 0,28 \quad (48)$$

Assim, temos:

$$0,26 A \leq I_{rms} \leq 0,28 A \quad (49)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (50)$$

Ou seja

$$0,97 * 19,33 \leq P \leq 1,03 * 22,25 \quad (51)$$

Assim, temos:

$$18,75 W \leq P \leq 22,91 W \quad (52)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (53)$$

Ou seja

$$0,97 * 30,33 \leq S \leq 1,03 * 34,38 \quad (54)$$

Assim, temos:

$$\mathbf{29,42 VA \leq S \leq 35,41 VA} \quad (55)$$

- **Fator de Potência (erro de 1.0%)**

$$0,938 * FP_{min} \leq FP \leq 1,062 * FP_{max} \quad (56)$$

Ou seja

$$0,99 * 0,61 \leq FP \leq 1,01 * 0,66 \quad (57)$$

Assim, temos:

$$\mathbf{0,60 \leq FP \leq 0,67} \quad (58)$$

Se todas as condicionais acima forem satisfeitas, o *software* entende a carga como sendo uma lâmpada fluorescente e, de acordo com a Tabela 1, o valor da variável *Identificador* será igual a 2.

5.3 Ventilador

Para os testes com o ventilador de mesa foi utilizado o ventilador NV-15 da Mondial, como visto na Figura 27. Seus dados técnicos podem ser encontrados no cadastro de produtos da fabricante [16].

Figura 27 - Ventilador utilizado nos testes de detecção



Fonte: www.mondialine.com.br

O funcionamento do ventilador utilizado nos testes se dá a partir de um motor monofásico universal. A fabricante não fornece detalhes técnicos especificamente sobre o motor utilizado, mas trata-se de um motor de 50 W, com 3 velocidades e que opera tanto em 127 V quanto em 220 V. Nos testes realizados, o motor foi operado apenas em 127 V.

Como o ventilador opera em 3 velocidades, e cada um desses pontos de operação tem diferentes valores de parâmetros de regime permanente, foi necessário fazer uma condicional para cada ponto de operação da carga.

A Figura 28 mostra o *Serial Monitor* do Arduino enquanto o ventilador funciona na potência mínima (velocidade 1), que seria a velocidade mais baixa de operação do motor.

Figura 28 – Dados do *Serial Monitor* durante a ligação do ventilador em velocidade 1

```
Preal: 35.18 Papar: 36.64 Vrms: 117.52 Irms: 0.31 FP: 0.96
Preal: 34.95 Papar: 36.35 Vrms: 117.63 Irms: 0.31 FP: 0.96
Preal: 35.13 Papar: 36.53 Vrms: 117.62 Irms: 0.31 FP: 0.96
Preal: 34.92 Papar: 36.30 Vrms: 117.55 Irms: 0.31 FP: 0.96
Preal: 35.15 Papar: 36.52 Vrms: 117.36 Irms: 0.31 FP: 0.96
Preal: 35.08 Papar: 36.44 Vrms: 117.44 Irms: 0.31 FP: 0.96
Preal: 35.15 Papar: 36.51 Vrms: 117.53 Irms: 0.31 FP: 0.96
Preal: 35.03 Papar: 36.48 Vrms: 117.59 Irms: 0.31 FP: 0.96
Preal: 35.24 Papar: 36.65 Vrms: 117.63 Irms: 0.31 FP: 0.96
Preal: 35.06 Papar: 36.44 Vrms: 117.80 Irms: 0.31 FP: 0.96
Preal: 35.08 Papar: 36.51 Vrms: 117.50 Irms: 0.31 FP: 0.96
Preal: 35.49 Papar: 36.86 Vrms: 117.67 Irms: 0.31 FP: 0.96
Preal: 34.89 Papar: 36.33 Vrms: 117.53 Irms: 0.31 FP: 0.96
Preal: 34.80 Papar: 36.15 Vrms: 117.23 Irms: 0.31 FP: 0.96
Preal: 35.45 Papar: 36.86 Vrms: 117.66 Irms: 0.31 FP: 0.96
Preal: 34.89 Papar: 36.26 Vrms: 117.25 Irms: 0.31 FP: 0.96
Preal: 34.92 Papar: 36.35 Vrms: 117.76 Irms: 0.31 FP: 0.96
Preal: 34.84 Papar: 36.20 Vrms: 117.18 Irms: 0.31 FP: 0.96
Preal: 34.98 Papar: 36.39 Vrms: 117.57 Irms: 0.31 FP: 0.96
Preal: 35.35 Papar: 36.80 Vrms: 117.58 Irms: 0.31 FP: 0.96
Preal: 35.30 Papar: 36.68 Vrms: 117.45 Irms: 0.31 FP: 0.96
Preal: 35.01 Papar: 36.44 Vrms: 117.45 Irms: 0.31 FP: 0.96
Preal: 35.31 Papar: 36.72 Vrms: 117.60 Irms: 0.31 FP: 0.96
Preal: 35.05 Papar: 36.53 Vrms: 117.39 Irms: 0.31 FP: 0.96
Preal: 34.91 Papar: 36.26 Vrms: 117.35 Irms: 0.31 FP: 0.96
Preal: 35.41 Papar: 36.79 Vrms: 117.67 Irms: 0.31 FP: 0.96
Preal: 35.19 Papar: 36.67 Vrms: 117.41 Irms: 0.31 FP: 0.96
Preal: 35.01 Papar: 36.41 Vrms: 117.44 Irms: 0.31 FP: 0.96
Preal: 35.22 Papar: 36.62 Vrms: 117.64 Irms: 0.31 FP: 0.96
Preal: 35.30 Papar: 36.73 Vrms: 117.57 Irms: 0.31 FP: 0.96
Preal: 35.21 Papar: 36.59 Vrms: 117.50 Irms: 0.31 FP: 0.96
Preal: 35.04 Papar: 36.39 Vrms: 117.32 Irms: 0.31 FP: 0.96
Preal: 34.98 Papar: 36.45 Vrms: 117.67 Irms: 0.31 FP: 0.96
Preal: 35.02 Papar: 36.50 Vrms: 117.76 Irms: 0.31 FP: 0.96
Preal: 35.26 Papar: 36.65 Vrms: 117.50 Irms: 0.31 FP: 0.96
```

Fonte: *Print screen* do software Arduino

Já a Figura 29 mostra o *Serial Monitor* enquanto o ventilador funciona na velocidade 2, que seria a velocidade intermediária de operação do motor.

Figura 29 - dados do *Serial Monitor* durante a ligação do ventilador em velocidade 2

```

Preal: 44.59 Papar: 46.46 Vrms: 121.80 Irms: 0.37 FP: 0.96
Preal: 44.69 Papar: 46.59 Vrms: 121.81 Irms: 0.37 FP: 0.96
Preal: 44.93 Papar: 46.81 Vrms: 121.83 Irms: 0.37 FP: 0.96
Preal: 44.92 Papar: 46.79 Vrms: 121.95 Irms: 0.37 FP: 0.96
Preal: 44.94 Papar: 46.81 Vrms: 122.02 Irms: 0.37 FP: 0.96
Preal: 45.18 Papar: 47.06 Vrms: 122.48 Irms: 0.37 FP: 0.96
Preal: 45.20 Papar: 47.08 Vrms: 121.95 Irms: 0.37 FP: 0.96
Preal: 44.77 Papar: 46.69 Vrms: 122.03 Irms: 0.37 FP: 0.96
Preal: 45.18 Papar: 47.05 Vrms: 121.99 Irms: 0.37 FP: 0.96
Preal: 44.76 Papar: 46.64 Vrms: 122.03 Irms: 0.37 FP: 0.96
Preal: 44.92 Papar: 46.82 Vrms: 122.04 Irms: 0.37 FP: 0.96
Preal: 44.87 Papar: 46.73 Vrms: 122.08 Irms: 0.37 FP: 0.96
Preal: 44.97 Papar: 46.84 Vrms: 122.08 Irms: 0.37 FP: 0.96
Preal: 44.79 Papar: 46.64 Vrms: 121.98 Irms: 0.37 FP: 0.96
Preal: 45.00 Papar: 46.88 Vrms: 122.16 Irms: 0.37 FP: 0.96
Preal: 44.99 Papar: 46.86 Vrms: 122.02 Irms: 0.37 FP: 0.96
Preal: 44.73 Papar: 46.61 Vrms: 121.87 Irms: 0.37 FP: 0.96
Preal: 45.07 Papar: 46.94 Vrms: 121.94 Irms: 0.37 FP: 0.96
Preal: 44.95 Papar: 46.80 Vrms: 122.00 Irms: 0.37 FP: 0.96
Preal: 44.91 Papar: 46.81 Vrms: 121.86 Irms: 0.37 FP: 0.96
Preal: 44.94 Papar: 46.81 Vrms: 122.06 Irms: 0.37 FP: 0.96
Preal: 44.84 Papar: 46.69 Vrms: 121.97 Irms: 0.37 FP: 0.96
Preal: 44.77 Papar: 46.63 Vrms: 121.89 Irms: 0.37 FP: 0.96
Preal: 44.89 Papar: 46.78 Vrms: 121.82 Irms: 0.37 FP: 0.96
Preal: 45.09 Papar: 47.00 Vrms: 121.89 Irms: 0.37 FP: 0.96
Preal: 45.45 Papar: 47.33 Vrms: 122.61 Irms: 0.37 FP: 0.96
Preal: 44.91 Papar: 46.80 Vrms: 122.17 Irms: 0.37 FP: 0.96
Preal: 44.87 Papar: 46.77 Vrms: 122.01 Irms: 0.37 FP: 0.96
Preal: 44.89 Papar: 46.77 Vrms: 121.88 Irms: 0.37 FP: 0.96
Preal: 45.34 Papar: 47.18 Vrms: 122.58 Irms: 0.37 FP: 0.96
Preal: 44.97 Papar: 46.84 Vrms: 122.04 Irms: 0.37 FP: 0.96
Preal: 44.88 Papar: 46.75 Vrms: 122.08 Irms: 0.37 FP: 0.96
Preal: 44.95 Papar: 46.79 Vrms: 121.99 Irms: 0.37 FP: 0.96
Preal: 45.25 Papar: 47.11 Vrms: 122.37 Irms: 0.37 FP: 0.96
Preal: 44.76 Papar: 46.63 Vrms: 121.97 Irms: 0.37 FP: 0.96

```

Fonte: *Print screen* do software Arduino

Por fim, na Figura 30, o *Serial monitor* mostra os valores dos parâmetros de regime permanente para o ventilador funcionando em sua velocidade máxima, que é a velocidade 3 de operação do motor.

Figura 30 - Dados do *Serial Monitor* durante a ligação do ventilador em velocidade 3

```

Preal: 47.58 Papar: 50.09 Vrms: 121.76 Irms: 0.41 FP: 0.95
Preal: 47.78 Papar: 50.32 Vrms: 121.68 Irms: 0.41 FP: 0.95
Preal: 47.74 Papar: 50.33 Vrms: 122.01 Irms: 0.41 FP: 0.95
Preal: 47.66 Papar: 50.13 Vrms: 121.92 Irms: 0.41 FP: 0.95
Preal: 47.80 Papar: 50.47 Vrms: 122.02 Irms: 0.41 FP: 0.95
Preal: 47.76 Papar: 50.39 Vrms: 122.00 Irms: 0.41 FP: 0.95
Preal: 48.29 Papar: 50.85 Vrms: 122.05 Irms: 0.42 FP: 0.95
Preal: 47.93 Papar: 50.43 Vrms: 121.99 Irms: 0.41 FP: 0.95
Preal: 47.75 Papar: 50.30 Vrms: 122.01 Irms: 0.41 FP: 0.95
Preal: 47.77 Papar: 50.26 Vrms: 121.95 Irms: 0.41 FP: 0.95
Preal: 48.30 Papar: 50.92 Vrms: 122.57 Irms: 0.42 FP: 0.95
Preal: 47.38 Papar: 49.92 Vrms: 121.95 Irms: 0.41 FP: 0.95
Preal: 48.25 Papar: 50.82 Vrms: 122.52 Irms: 0.41 FP: 0.95
Preal: 47.98 Papar: 50.61 Vrms: 122.03 Irms: 0.41 FP: 0.95
Preal: 47.74 Papar: 50.32 Vrms: 121.87 Irms: 0.41 FP: 0.95
Preal: 47.66 Papar: 50.09 Vrms: 121.93 Irms: 0.41 FP: 0.95
Preal: 47.84 Papar: 50.36 Vrms: 121.89 Irms: 0.41 FP: 0.95
Preal: 48.12 Papar: 50.71 Vrms: 122.04 Irms: 0.42 FP: 0.95
Preal: 48.36 Papar: 50.84 Vrms: 122.54 Irms: 0.41 FP: 0.95
Preal: 48.05 Papar: 50.57 Vrms: 122.08 Irms: 0.41 FP: 0.95
Preal: 47.74 Papar: 50.30 Vrms: 121.98 Irms: 0.41 FP: 0.95
Preal: 47.74 Papar: 50.24 Vrms: 121.92 Irms: 0.41 FP: 0.95
Preal: 47.65 Papar: 50.25 Vrms: 122.07 Irms: 0.41 FP: 0.95
Preal: 47.70 Papar: 50.21 Vrms: 121.82 Irms: 0.41 FP: 0.95
Preal: 48.08 Papar: 50.56 Vrms: 121.83 Irms: 0.42 FP: 0.95
Preal: 47.86 Papar: 50.33 Vrms: 121.86 Irms: 0.41 FP: 0.95
Preal: 47.87 Papar: 50.36 Vrms: 121.83 Irms: 0.41 FP: 0.95
Preal: 47.79 Papar: 50.32 Vrms: 121.84 Irms: 0.41 FP: 0.95
Preal: 47.82 Papar: 50.35 Vrms: 121.94 Irms: 0.41 FP: 0.95
Preal: 47.59 Papar: 50.21 Vrms: 121.89 Irms: 0.41 FP: 0.95
Preal: 48.32 Papar: 50.80 Vrms: 122.38 Irms: 0.42 FP: 0.95
Preal: 47.83 Papar: 50.38 Vrms: 121.90 Irms: 0.41 FP: 0.95
Preal: 48.03 Papar: 50.64 Vrms: 122.23 Irms: 0.41 FP: 0.95
Preal: 47.73 Papar: 50.30 Vrms: 121.94 Irms: 0.41 FP: 0.95
Preal: 47.85 Papar: 50.44 Vrms: 121.95 Irms: 0.41 FP: 0.95

```

Fonte: *Print screen* do *software* Arduino

A partir desses valores, podemos criar as condições matemáticas nas quais o *software* possa identificar a ligação desse ventilador, em qualquer um dos três pontos de operação, quando o ventilador for integrado a todo o sistema. Observando o quadro acima, foi montada a Tabela 5 que apresenta o valor máximo e o valor mínimo que cada um dos parâmetros apresentou no quadro coletado em cada um dos 3 pontos de operação:

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Ventilador	117.18	0.31	35.35	36.86	0,96
Valor Mínimo	Velocidade 1	117.80	0.31	34.80	36.15	0,96
Valor Máximo	Ventilador	122.61	0.37	45.45	47.33	0,96
Valor Mínimo	Velocidade 2	121.80	0.37	44.59	46.46	0,96
Valor Máximo	Ventilador	122.54	0.42	48.32	50.92	0,95
Valor Mínimo	Velocidade 3	121.68	0.41	47.58	50.09	0,95

Tabela 5 - valores mínimos e máximos dos parâmetros medidos no ventilador (3 velocidades)

Observando os valores da tabela e aplicando os erros associados a cada parâmetro como mostrado na Tabela 1, a identificação do ventilador deverá obedecer à seguinte estrutura condicional:

➤ **Para a Velocidade 1 do motor:**

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (59)$$

Ou seja:

$$0,985 * 117,18 \leq V_{rms} \leq 1,015 * 117,80 \quad (60)$$

Assim, temos:

$$115,4 V \leq V_{rms} \leq 119,6 V \quad (61)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (62)$$

Ou seja

$$0,985 * 0,31 \leq I_{rms} \leq 1,015 * 0,31 \quad (63)$$

Assim, temos:

$$\mathbf{0,30 \leq I_{rms} \leq 0,32} \quad (64)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (65)$$

Ou seja

$$0,97 * 34,80 \leq P \leq 1,03 * 35,35 \quad (66)$$

Assim, temos:

$$\mathbf{33,76 W \leq P \leq 36,41 W} \quad (67)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (68)$$

Ou seja

$$0,97 * 36,15 \leq S \leq 1,03 * 36,86 \quad (69)$$

Então, temos:

$$\mathbf{35,07 VA \leq S \leq 37,97 VA} \quad (70)$$

- **Fator de Potência (erro de 1.0%)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (71)$$

Ou seja

$$0,99 * 0,96 \leq FP \leq 1,01 * 0,96 \quad (72)$$

Então, temos:

$$0,95 \leq FP \leq 0,97 \quad (73)$$

➤ **Para a Velocidade 2 do motor:**

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (74)$$

Ou seja:

$$0,985 * 121,80 \leq V_{rms} \leq 1,015 * 122,61 \quad (75)$$

Assim, temos:

$$119,97 V \leq V_{rms} \leq 124,45 V \quad (76)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (77)$$

Ou seja:

$$0,985 * 0,37 \leq I_{rms} \leq 1,015 * 0,37 \quad (78)$$

Assim, temos:

$$0,36 A \leq I_{rms} \leq 0,38 A \quad (79)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (80)$$

Ou seja:

$$0,97 * 44,59 \leq P \leq 1,03 * 45,45 \quad (81)$$

Assim, temos:

$$43,25 W \leq P \leq 46,81 W \quad (82)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (83)$$

Ou seja:

$$0,97 * 46,46 \leq S \leq 1,03 * 47,33 \quad (84)$$

Assim, temos:

$$45,06 VA \leq S \leq 49,48 VA \quad (85)$$

- **Fator de Potência (erro de 1,0 %)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (86)$$

Ou seja:

$$0,99 * 0,95 \leq FP \leq 1,01 * 0,95 \quad (87)$$

Assim, temos:

$$0,94 \leq FP \leq 0,96 \quad (88)$$

- **Para a Velocidade 3 do motor:**

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (89)$$

Ou seja:

$$0,985 * 121,68 \leq V_{rms} \leq 1,015 * 122,54 \quad (90)$$

Assim, temos:

$$\mathbf{119,85 V \leq V_{rms} \leq 124,38 V} \quad (91)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (92)$$

Ou seja

$$0,985 * 0,41 \leq I_{rms} \leq 1,015 * 0,42 \quad (93)$$

Assim, temos:

$$\mathbf{0,40 A \leq I_{rms} \leq 0,43 A} \quad (94)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (95)$$

Ou seja:

$$0,97 * 47,58 \leq P \leq 1,03 * 48,32 \quad (96)$$

Assim, temos:

$$\mathbf{46,15 W \leq P \leq 49,76 W} \quad (97)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (98)$$

Ou seja:

$$0,97 * 50,09 \leq S \leq 1,03 * 50,92 \quad (99)$$

Assim, temos:

$$48,59 VA \leq S \leq 52,45 VA \quad (100)$$

- **Fator de Potência (erro de 6,2%)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (101)$$

Ou seja:

$$0,99 * 0,95 \leq FP \leq 1,01 * 0,95 \quad (102)$$

Assim, temos:

$$0,94 \leq FP \leq 0,96 \quad (103)$$

Assim, se qualquer um dos 3 grupos de condicionais acima forem validados pelo *software*, o ventilador é detectado e, conforme a Tabela 1, a variável *identificador* passa a ter valor 3.

5.4 Ferro de passar

Para os testes com o ferro de passar foi utilizado o ferro Maximus 31 Black da Mondial, como visto na Figura 31. Seus dados técnicos podem ser encontrados na página 1 do catálogo de produtos da fabricante [17].

Figura 31 - Ferro de passar utilizado nos testes de detecção de cargas



Fonte: www.mondialine.com.br

O ferro elétrico é uma carga resistiva, com um potenciômetro que regula o quanto de potência será dissipada sobre seu resistor a fim de regular a temperatura da prancha do ferro (parte metálica que toca a roupa), para se adequar aos diferentes tipos de tecidos que serão submetidos ao processo. Para cada uma dessas posições em que o potenciômetro pode se encontrar, está associado um valor de corrente elétrica e, conseqüentemente, um valor de potência ativa. Sendo assim, existem infinitos valores de corrente e potência ativa nos quais um ferro de passar pode estar operando, o que dificulta determinar em quais valores os parâmetros de regime permanente poderão estar durante a operação.

Para solucionar esse problema, foram realizados dois testes distintos: o primeiro com o ferro de passar na posição de potência mínima, ou seja, na menor temperatura de operação possível; e um segundo teste com o potenciômetro do ferro de passar na posição de potência máxima, ou seja, a maior temperatura de operação do ferro. Assim, foi possível determinar o menor valor de cada parâmetro para a operação em potência mínima e o maior valor de cada parâmetro para a operação em potência máxima, e assim criar um intervalo no qual a condicional valide a detecção do ferro de passar.

A Figura 32 mostra o *Serial Monitor* do Arduino enquanto o ferro de passar funciona na potência mínima, que seria a temperatura mais baixa de operação da carga.

Figura 32 - Dados do *Serial Monitor* durante a ligação do ferro de passar em potência mínima

```

Preal: 663.94 Papar: 664.85 Vrms: 114.58 Irms: 5.76 FP: 1.00
Preal: 664.71 Papar: 665.65 Vrms: 114.68 Irms: 5.76 FP: 1.00
Preal: 665.35 Papar: 666.23 Vrms: 114.69 Irms: 5.77 FP: 1.00
Preal: 665.57 Papar: 666.50 Vrms: 114.71 Irms: 5.77 FP: 1.00
Preal: 666.59 Papar: 667.50 Vrms: 114.86 Irms: 5.77 FP: 1.00
Preal: 667.24 Papar: 668.17 Vrms: 114.88 Irms: 5.77 FP: 1.00
Preal: 665.64 Papar: 666.56 Vrms: 114.74 Irms: 5.77 FP: 1.00
Preal: 668.06 Papar: 668.97 Vrms: 114.90 Irms: 5.78 FP: 1.00
Preal: 666.17 Papar: 667.04 Vrms: 114.75 Irms: 5.77 FP: 1.00
Preal: 668.02 Papar: 668.95 Vrms: 114.92 Irms: 5.78 FP: 1.00
Preal: 667.80 Papar: 668.72 Vrms: 114.90 Irms: 5.78 FP: 1.00
Preal: 668.30 Papar: 669.19 Vrms: 114.93 Irms: 5.78 FP: 1.00
Preal: 666.16 Papar: 667.04 Vrms: 114.76 Irms: 5.77 FP: 1.00
Preal: 667.77 Papar: 668.71 Vrms: 114.84 Irms: 5.78 FP: 1.00
Preal: 667.56 Papar: 668.46 Vrms: 114.88 Irms: 5.78 FP: 1.00
Preal: 667.53 Papar: 668.45 Vrms: 114.88 Irms: 5.78 FP: 1.00
Preal: 667.19 Papar: 668.06 Vrms: 114.83 Irms: 5.77 FP: 1.00
Preal: 666.34 Papar: 667.25 Vrms: 114.76 Irms: 5.77 FP: 1.00
Preal: 668.14 Papar: 669.04 Vrms: 114.93 Irms: 5.78 FP: 1.00
Preal: 671.12 Papar: 672.00 Vrms: 115.17 Irms: 5.79 FP: 1.00
Preal: 662.93 Papar: 663.85 Vrms: 114.47 Irms: 5.76 FP: 1.00
Preal: 666.14 Papar: 667.05 Vrms: 114.76 Irms: 5.77 FP: 1.00
Preal: 667.91 Papar: 668.81 Vrms: 114.91 Irms: 5.78 FP: 1.00
Preal: 666.75 Papar: 667.67 Vrms: 114.79 Irms: 5.77 FP: 1.00
Preal: 668.47 Papar: 669.39 Vrms: 114.93 Irms: 5.78 FP: 1.00
Preal: 666.72 Papar: 667.63 Vrms: 114.83 Irms: 5.77 FP: 1.00
Preal: 667.45 Papar: 668.35 Vrms: 114.85 Irms: 5.78 FP: 1.00
Preal: 665.54 Papar: 666.48 Vrms: 114.70 Irms: 5.77 FP: 1.00
Preal: 664.82 Papar: 665.75 Vrms: 114.69 Irms: 5.76 FP: 1.00
Preal: 665.25 Papar: 666.20 Vrms: 114.67 Irms: 5.77 FP: 1.00
Preal: 664.42 Papar: 665.33 Vrms: 114.57 Irms: 5.76 FP: 1.00
Preal: 665.49 Papar: 666.44 Vrms: 114.65 Irms: 5.77 FP: 1.00
Preal: 666.77 Papar: 667.66 Vrms: 114.79 Irms: 5.77 FP: 1.00
Preal: 664.44 Papar: 665.36 Vrms: 114.60 Irms: 5.76 FP: 1.00
Preal: 665.11 Papar: 666.03 Vrms: 114.65 Irms: 5.77 FP: 1.00

```

Fonte: *Print screen* do software Arduino

Já a Figura 33 mostra o *Serial Monitor* do Arduino enquanto o ferro de passar funciona na potência máxima, que seria a temperatura mais alta de operação da carga.

Figura 33 - Dados do *Serial Monitor* durante a ligação do ferro de passar em potência máxima

```

Preal: 697.19 Papar: 697.99 Vrms: 117.05 Irms: 5.92 FP: 1.00
Preal: 698.47 Papar: 699.24 Vrms: 117.18 Irms: 5.92 FP: 1.00
Preal: 698.12 Papar: 698.85 Vrms: 117.20 Irms: 5.92 FP: 1.00
Preal: 697.61 Papar: 698.36 Vrms: 117.07 Irms: 5.92 FP: 1.00
Preal: 698.07 Papar: 698.85 Vrms: 117.23 Irms: 5.92 FP: 1.00
Preal: 698.96 Papar: 699.70 Vrms: 117.29 Irms: 5.92 FP: 1.00
Preal: 698.12 Papar: 698.87 Vrms: 117.14 Irms: 5.92 FP: 1.00
Preal: 697.84 Papar: 698.60 Vrms: 117.09 Irms: 5.92 FP: 1.00
Preal: 697.12 Papar: 697.86 Vrms: 117.01 Irms: 5.92 FP: 1.00
Preal: 697.70 Papar: 698.48 Vrms: 117.14 Irms: 5.92 FP: 1.00
Preal: 696.84 Papar: 697.61 Vrms: 116.99 Irms: 5.92 FP: 1.00
Preal: 697.12 Papar: 697.88 Vrms: 117.04 Irms: 5.92 FP: 1.00
Preal: 697.57 Papar: 698.33 Vrms: 117.16 Irms: 5.92 FP: 1.00
Preal: 696.29 Papar: 697.04 Vrms: 116.97 Irms: 5.92 FP: 1.00
Preal: 697.98 Papar: 698.78 Vrms: 117.13 Irms: 5.92 FP: 1.00
Preal: 696.61 Papar: 697.34 Vrms: 116.95 Irms: 5.92 FP: 1.00
Preal: 697.72 Papar: 698.47 Vrms: 117.05 Irms: 5.92 FP: 1.00
Preal: 697.97 Papar: 698.75 Vrms: 117.20 Irms: 5.92 FP: 1.00
Preal: 697.07 Papar: 697.86 Vrms: 117.01 Irms: 5.92 FP: 1.00
Preal: 696.84 Papar: 697.59 Vrms: 116.97 Irms: 5.92 FP: 1.00
Preal: 697.60 Papar: 698.38 Vrms: 117.07 Irms: 5.92 FP: 1.00
Preal: 697.38 Papar: 698.13 Vrms: 117.09 Irms: 5.92 FP: 1.00
Preal: 696.38 Papar: 697.16 Vrms: 116.93 Irms: 5.92 FP: 1.00
Preal: 696.85 Papar: 697.58 Vrms: 117.00 Irms: 5.92 FP: 1.00
Preal: 698.39 Papar: 699.21 Vrms: 117.20 Irms: 5.92 FP: 1.00
Preal: 698.26 Papar: 699.03 Vrms: 117.16 Irms: 5.92 FP: 1.00
Preal: 696.40 Papar: 697.15 Vrms: 116.94 Irms: 5.92 FP: 1.00
Preal: 696.70 Papar: 697.42 Vrms: 117.01 Irms: 5.92 FP: 1.00
Preal: 695.53 Papar: 696.26 Vrms: 116.90 Irms: 5.91 FP: 1.00
Preal: 696.15 Papar: 696.89 Vrms: 117.03 Irms: 5.91 FP: 1.00
Preal: 695.21 Papar: 696.00 Vrms: 116.91 Irms: 5.91 FP: 1.00
Preal: 696.58 Papar: 697.36 Vrms: 117.07 Irms: 5.91 FP: 1.00
Preal: 696.55 Papar: 697.34 Vrms: 117.00 Irms: 5.92 FP: 1.00
Preal: 696.48 Papar: 697.22 Vrms: 117.04 Irms: 5.91 FP: 1.00
Preal: 697.33 Papar: 698.12 Vrms: 117.11 Irms: 5.92 FP: 1.00

```

Fonte: *Print screen* do *software* Arduino

Com os valores lidos nos dois quadros acima foi montada a Tabela 6, apresentando os valores dos parâmetros de regime permanente para dois pontos de operação do ferro de passar: em potência mínima e em potência máxima.

Tabela 6 - Valores mínimos e máximos dos parâmetros medidos do ferro de passar

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Ferro de passar (potência mínima)	115.17	5.79	668.47	669.39	1
Valor Mínimo		114.57	5.76	664.42	665.33	1
Valor Máximo	Ferro de passar (potência máxima)	117.29	5.92	698.96	699.70	1
Valor Mínimo		116.90	5.91	695.21	696.00	1

A partir dos dados da Tabela 6, foram definidos os intervalos de valores de parâmetros que validam a condicional para a identificação do ferro de passar. Para determinar o limite inferior de cada parâmetro na condicional, foi utilizada a primeira parte da tabela

6, que são as linhas da tabela que apresentam os dados do ferro de passar quando o mesmo está ligado em potência mínima. Pela mesma lógica, para definir o limite superior do intervalo válido para a condicional, os dados utilizados foram os da segunda parte da Tabela 6 (olhando para a linha de valor máximo de cada parâmetro). Assim, seguindo o método descrito acima, a identificação do ferro de passar obedecerá a estrutura condicional abaixo:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min}(Pmin) \leq V_{rms} \leq 1,015 * V_{max}(Pmax) \quad (104)$$

Ou seja:

$$0,985 * 114,57 \leq V_{rms} \leq 1,015 * 117,29 \quad (105)$$

Assim, temos:

$$\mathbf{112,85 V} \leq V_{rms} \leq \mathbf{119,05 V} \quad (106)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min}(Pmin) \leq I_{rms} \leq 1,015 * I_{max}(Pmax) \quad (107)$$

Ou seja:

$$0,985 * 5,76 \leq I_{rms} \leq 1,015 * 5,92 \quad (108)$$

Assim, temos:

$$\mathbf{5,67 A} \leq I_{rms} \leq \mathbf{6,01 A} \quad (109)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min}(Pmin) \leq P \leq 1,03 * P_{max}(Pmax) \quad (110)$$

Ou seja

$$0,97 * 664,42 \leq P \leq 1,03 * 698,96 \quad (111)$$

Assim, temos:

$$644,49 W \leq P \leq 719,93 W \quad (112)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{\min}(P_{\min}) \leq S \leq 1,03 * S_{\max}(P_{\max}) \quad (113)$$

Ou seja:

$$0,97 * 665,33 \leq S \leq 1,03 * 699,70 \quad (114)$$

Assim, temos:

$$645,05 VA \leq S \leq 720,69 VA \quad (115)$$

- **Fator de Potência (erro de 1,0 %)**

$$0,99 * FP_{\min} \leq FP \leq 1,01 * FP_{\max} \quad (116)$$

Ou seja:

$$0,99 * 1 \leq FP \leq 1,01 * 1 \quad (117)$$

Assim, temos:

$$0,99 \leq FP \leq 1,01 \quad (118)$$

Se todas essas condições forem satisfeitas, o *software* entende a carga como sendo o ferro de passar e o valor da variável *identificador* será igual a 4, de acordo com a Tabela 1.

5.5 Refrigerador

Para os testes com o refrigerador, foi utilizado o modelo CRA30 da Consul como o visto na Figura 34. Seus dados técnicos podem ser encontrados na página 6 do guia rápido fornecido pelo fabricante [18].

Figura 34 - Refrigerador utilizado nos testes de detecção



Fonte: loja.consul.com.br

O refrigerador, é uma máquina térmica que trabalha em ciclos bem definidos baseados em transformações termodinâmicas dos fluidos envolvidos no processo. São basicamente 5 estágios: compressão adiabática, resfriamento isobárico, condensação, expansão adiabática e vaporização isobárica.

A Figura 35 abaixo mostra o *Serial Monitor* do Arduino enquanto o refrigerador está ligada. É possível observar uma boa estabilidade nos valores dos parâmetros, o que facilita a detecção da carga.

Figura 35 - Dados do *Serial Monitor* durante a ligação do refrigerador

```

Preal: 64.36 Papar: 113.11 Vrms: 120.62 Irms: 0.93 FP: 0.57
Preal: 64.61 Papar: 113.38 Vrms: 120.64 Irms: 0.93 FP: 0.57
Preal: 64.92 Papar: 113.43 Vrms: 120.63 Irms: 0.93 FP: 0.57
Preal: 64.87 Papar: 113.22 Vrms: 120.53 Irms: 0.93 FP: 0.57
Preal: 64.62 Papar: 113.40 Vrms: 120.81 Irms: 0.93 FP: 0.57
Preal: 64.67 Papar: 113.26 Vrms: 120.65 Irms: 0.93 FP: 0.57
Preal: 64.64 Papar: 113.09 Vrms: 120.80 Irms: 0.93 FP: 0.57
Preal: 64.78 Papar: 113.48 Vrms: 120.67 Irms: 0.93 FP: 0.57
Preal: 64.60 Papar: 113.14 Vrms: 120.61 Irms: 0.93 FP: 0.57
Preal: 64.54 Papar: 113.29 Vrms: 120.63 Irms: 0.93 FP: 0.57
Preal: 64.56 Papar: 113.05 Vrms: 120.75 Irms: 0.93 FP: 0.57
Preal: 64.67 Papar: 113.39 Vrms: 120.81 Irms: 0.93 FP: 0.57
Preal: 64.44 Papar: 113.35 Vrms: 120.87 Irms: 0.93 FP: 0.57
Preal: 64.58 Papar: 113.28 Vrms: 120.71 Irms: 0.93 FP: 0.57
Preal: 64.54 Papar: 112.99 Vrms: 120.57 Irms: 0.93 FP: 0.57
Preal: 64.35 Papar: 113.28 Vrms: 120.75 Irms: 0.93 FP: 0.57
Preal: 64.49 Papar: 113.17 Vrms: 120.64 Irms: 0.93 FP: 0.57
Preal: 64.60 Papar: 113.14 Vrms: 120.77 Irms: 0.93 FP: 0.57
Preal: 64.39 Papar: 113.51 Vrms: 120.72 Irms: 0.93 FP: 0.57
Preal: 64.35 Papar: 113.20 Vrms: 120.69 Irms: 0.93 FP: 0.57
Preal: 64.34 Papar: 113.03 Vrms: 120.67 Irms: 0.93 FP: 0.57
Preal: 64.75 Papar: 113.23 Vrms: 120.62 Irms: 0.93 FP: 0.57
Preal: 64.39 Papar: 112.96 Vrms: 120.59 Irms: 0.93 FP: 0.57
Preal: 64.36 Papar: 113.13 Vrms: 120.64 Irms: 0.93 FP: 0.57
Preal: 64.41 Papar: 112.93 Vrms: 120.56 Irms: 0.93 FP: 0.57
Preal: 63.88 Papar: 112.07 Vrms: 119.98 Irms: 0.93 FP: 0.57
Preal: 64.51 Papar: 113.10 Vrms: 120.62 Irms: 0.93 FP: 0.57
Preal: 64.42 Papar: 113.41 Vrms: 120.75 Irms: 0.93 FP: 0.57
Preal: 64.02 Papar: 112.92 Vrms: 120.58 Irms: 0.93 FP: 0.57
Preal: 64.49 Papar: 113.31 Vrms: 120.82 Irms: 0.93 FP: 0.57
Preal: 64.64 Papar: 113.27 Vrms: 120.73 Irms: 0.93 FP: 0.57
Preal: 64.67 Papar: 113.12 Vrms: 120.71 Irms: 0.93 FP: 0.57
Preal: 64.30 Papar: 112.96 Vrms: 120.60 Irms: 0.93 FP: 0.57
Preal: 64.11 Papar: 112.52 Vrms: 120.19 Irms: 0.93 FP: 0.57
Preal: 64.38 Papar: 113.11 Vrms: 120.65 Irms: 0.93 FP: 0.57

```

A partir desses valores, podemos criar a condição matemática na qual o *software* possa identificar a ligação deste refrigerador quando todas as cargas forem integradas ao sistema. Observando o quadro acima, foi montada a Tabela 7 que apresenta o valor máximo e o valor mínimo que cada um dos parâmetros apresentou no quadro coletado:

Tabela 7 - Valores mínimos e máximos dos parâmetros medidos no refrigerador

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Refrigerador	120.87	0.93	64.92	113.43	0.57
Valor Mínimo		119.98	0.93	64.11	112.07	0.57

Observando os valores da tabela e aplicando os erros associados a cada parâmetro como mostrado na Tabela 2, temos que a identificação do refrigerador obedecerá a seguinte estrutura condicional:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * V_{max} \quad (119)$$

Ou seja:

$$0,985 * 119,98 \leq V_{rms} \leq 1,015 * 120,87 \quad (120)$$

Assim, temos:

$$\mathbf{118,18 V \leq V_{rms} \leq 122,68 V} \quad (121)$$

- **Corrente RMS (erro de 1,5%)**

(122)

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max}$$

Ou seja

$$0,985 * 0,93 \leq I_{rms} \leq 1,015 * 0,93 \quad (123)$$

Assim, temos:

$$\mathbf{0,92 A \leq I_{rms} \leq 0,94 A} \quad (124)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (125)$$

Ou seja:

$$0,97 * 64,11 W \leq P \leq 1,03 * 64,92 W \quad (126)$$

Assim, temos:

$$\mathbf{62,18 W \leq P \leq 66,87 W} \quad (127)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (128)$$

Ou seja:

$$0,97 * 112,07 \leq S \leq 1,03 * 113,43 \quad (129)$$

Assim, temos:

$$108,71 VA \leq S \leq 116,83 VA \quad (130)$$

- **Fator de Potência (erro de 6.2%)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (131)$$

Ou seja:

$$0,99 * 0,96 \leq FP \leq 1,01 * 0,96 \quad (132)$$

Assim, temos:

$$0,95 \leq FP \leq 0,97 \quad (133)$$

Se todas essas condições forem satisfeitas, o *software* entende a carga como sendo o refrigerador e o valor da variável *identificador* será igual a 5, de acordo com a Tabela 1.

5.6 Forno de microondas

Para os testes com o forno de microondas, foi utilizado o modelo MEF41 da Electrolux como o visto na Figura 36 abaixo. Seus dados técnicos podem ser encontrados na página 34 do catálogo de produtos da fabricante [19].

Figura 36 - Forno de microondas utilizado nos testes de detecção de cargas



Fonte: *Print screen do software Arduino*

O funcionamento de um forno de micro-ondas é dado basicamente através de um transformador e um magnetron, um equipamento que utiliza a vibração dos elétrons para gerar um campo magnético (um campo elétrico variável gera um campo magnético).

O microondas é projetado para emitir ondas eletromagnéticas de 2450 MHz (frequência de ressonância da água), pois nessa frequência, essas ondas são absorvidas em forma de vibração pelas moléculas de água contidas nos alimentos e essa vibração das moléculas causa o aquecimento. [13]

Devido ao seu sistema de controle e aos elementos de armazenamento de energia que o compõe (componentes indutivos e capacitivos), um microondas é uma carga de fator de potência distante do unitário. Esse é um parâmetro importante na identificação desse tipo de carga.

Uma outra característica importante a ser citada, é a quantidade de níveis de potência no qual um forno de microondas pode funcionar. A função de aquecimento, por exemplo, exige uma distribuição de potência mais contínua ao longo do ciclo, diferente da função de descongelamento de alimentos. E essas foram as duas funções do microondas escolhidas para esta análise. Logo, foi necessário tratar essa carga trabalhando em dois pontos de operação distintos: aquecimento e descongelamento. A Figura 37 mostra o *Serial Monitor* do Arduino enquanto o microondas funciona na função de aquecimento.

Figura 37 - Dados do *Serial Monitor* para a operação do micro-ondas em modo 'aquecimento'

```
Preal: 1134.51 Papar: 1180.83 Vrms: 111.42 Irms: 10.59 FP: 0.96
Preal: 1135.18 Papar: 1181.43 Vrms: 111.42 Irms: 10.59 FP: 0.96
Preal: 1138.22 Papar: 1183.81 Vrms: 111.28 Irms: 10.63 FP: 0.96
Preal: 1138.60 Papar: 1184.51 Vrms: 111.37 Irms: 10.63 FP: 0.96
Preal: 1138.99 Papar: 1184.77 Vrms: 111.34 Irms: 10.63 FP: 0.96
Preal: 1140.10 Papar: 1186.36 Vrms: 111.28 Irms: 10.65 FP: 0.96
Preal: 1138.84 Papar: 1185.60 Vrms: 111.21 Irms: 10.65 FP: 0.96
Preal: 1138.89 Papar: 1185.34 Vrms: 111.29 Irms: 10.64 FP: 0.96
Preal: 1138.96 Papar: 1186.33 Vrms: 111.57 Irms: 10.62 FP: 0.96
Preal: 1138.23 Papar: 1185.58 Vrms: 111.56 Irms: 10.62 FP: 0.96
Preal: 1137.29 Papar: 1183.77 Vrms: 111.49 Irms: 10.61 FP: 0.96
Preal: 1135.83 Papar: 1181.46 Vrms: 111.29 Irms: 10.61 FP: 0.96
Preal: 1136.25 Papar: 1181.34 Vrms: 111.32 Irms: 10.60 FP: 0.96
Preal: 1136.11 Papar: 1180.95 Vrms: 111.32 Irms: 10.60 FP: 0.96
Preal: 1136.35 Papar: 1181.04 Vrms: 111.34 Irms: 10.60 FP: 0.96
Preal: 1136.56 Papar: 1182.29 Vrms: 111.34 Irms: 10.61 FP: 0.96
Preal: 1139.15 Papar: 1185.99 Vrms: 111.52 Irms: 10.63 FP: 0.96
Preal: 1138.21 Papar: 1184.46 Vrms: 111.40 Irms: 10.62 FP: 0.96
Preal: 1139.30 Papar: 1185.49 Vrms: 111.47 Irms: 10.63 FP: 0.96
Preal: 1140.64 Papar: 1187.68 Vrms: 111.53 Irms: 10.64 FP: 0.96
Preal: 1138.60 Papar: 1187.54 Vrms: 111.39 Irms: 10.65 FP: 0.96
Preal: 1138.83 Papar: 1188.15 Vrms: 111.42 Irms: 10.65 FP: 0.96
Preal: 1137.15 Papar: 1185.47 Vrms: 111.30 Irms: 10.64 FP: 0.96
Preal: 1137.37 Papar: 1184.91 Vrms: 111.37 Irms: 10.63 FP: 0.96
Preal: 1134.76 Papar: 1181.97 Vrms: 111.32 Irms: 10.61 FP: 0.96
Preal: 1135.67 Papar: 1181.28 Vrms: 111.25 Irms: 10.61 FP: 0.96
Preal: 1136.43 Papar: 1181.27 Vrms: 111.35 Irms: 10.60 FP: 0.96
Preal: 1137.25 Papar: 1181.60 Vrms: 111.40 Irms: 10.60 FP: 0.96
Preal: 1137.82 Papar: 1181.43 Vrms: 111.40 Irms: 10.60 FP: 0.96
Preal: 1137.98 Papar: 1181.75 Vrms: 111.34 Irms: 10.60 FP: 0.96
Preal: 1137.21 Papar: 1181.76 Vrms: 111.19 Irms: 10.62 FP: 0.96
Preal: 1136.61 Papar: 1182.66 Vrms: 111.28 Irms: 10.62 FP: 0.96
Preal: 1136.03 Papar: 1181.88 Vrms: 111.25 Irms: 10.62 FP: 0.96
Preal: 1135.50 Papar: 1180.71 Vrms: 111.19 Irms: 10.61 FP: 0.96
Preal: 1139.90 Papar: 1185.00 Vrms: 111.45 Irms: 10.62 FP: 0.96
```

Fonte: *Print screen* do software Arduino

Podemos observar certa estabilidade nos valores dos parâmetros durante o ciclo de aquecimento. Já a Figura 38 mostra o *Serial Monitor* enquanto o microondas funciona na função descongelamento.

Figura 38 - Dados do *Serial Monitor* para a operação do micro-ondas em modo 'descongelamento'

```

Preal: 35.82 Papar: 47.06 Vrms: 120.97 Irms: 0.39 FP: 0.76
Preal: 35.86 Papar: 47.19 Vrms: 121.01 Irms: 0.39 FP: 0.76
Preal: 35.80 Papar: 47.18 Vrms: 121.04 Irms: 0.39 FP: 0.76
Preal: 35.44 Papar: 46.87 Vrms: 121.09 Irms: 0.39 FP: 0.76
Preal: 35.75 Papar: 47.07 Vrms: 121.07 Irms: 0.39 FP: 0.76
Preal: 35.87 Papar: 47.07 Vrms: 121.03 Irms: 0.39 FP: 0.76
Preal: 150.71 Papar: 643.49 Vrms: 119.19 Irms: 5.40 FP: 0.23
Preal: 112.02 Papar: 720.38 Vrms: 119.02 Irms: 6.05 FP: 0.16
Preal: 107.93 Papar: 721.38 Vrms: 119.09 Irms: 6.06 FP: 0.15
Preal: 105.42 Papar: 724.18 Vrms: 119.09 Irms: 6.08 FP: 0.15
Preal: 104.00 Papar: 727.10 Vrms: 119.10 Irms: 6.10 FP: 0.14
Preal: 109.14 Papar: 723.34 Vrms: 119.09 Irms: 6.07 FP: 0.15
Preal: 122.21 Papar: 721.58 Vrms: 119.02 Irms: 6.06 FP: 0.17
Preal: 169.00 Papar: 714.98 Vrms: 118.57 Irms: 6.03 FP: 0.24
Preal: 678.83 Papar: 907.85 Vrms: 114.68 Irms: 7.92 FP: 0.75
Preal: 1160.48 Papar: 1222.09 Vrms: 110.71 Irms: 11.04 FP: 0.95
Preal: 1161.12 Papar: 1221.58 Vrms: 110.61 Irms: 11.04 FP: 0.95
Preal: 1165.62 Papar: 1225.39 Vrms: 110.88 Irms: 11.05 FP: 0.95
Preal: 1167.22 Papar: 1227.33 Vrms: 110.89 Irms: 11.07 FP: 0.95
Preal: 1166.89 Papar: 1225.60 Vrms: 110.82 Irms: 11.06 FP: 0.95
Preal: 1164.31 Papar: 1223.07 Vrms: 110.63 Irms: 11.06 FP: 0.95
Preal: 1164.61 Papar: 1222.44 Vrms: 110.57 Irms: 11.06 FP: 0.95
Preal: 1164.59 Papar: 1223.13 Vrms: 110.63 Irms: 11.06 FP: 0.95
Preal: 1163.66 Papar: 1221.63 Vrms: 110.54 Irms: 11.05 FP: 0.95
Preal: 1163.59 Papar: 1223.17 Vrms: 110.60 Irms: 11.06 FP: 0.95
Preal: 1163.30 Papar: 1223.19 Vrms: 110.65 Irms: 11.05 FP: 0.95
Preal: 1164.67 Papar: 1225.49 Vrms: 110.69 Irms: 11.07 FP: 0.95
Preal: 1166.37 Papar: 1225.83 Vrms: 110.70 Irms: 11.07 FP: 0.95
Preal: 36.21 Papar: 48.52 Vrms: 121.26 Irms: 0.40 FP: 0.75
Preal: 35.52 Papar: 47.12 Vrms: 120.88 Irms: 0.39 FP: 0.75
Preal: 35.62 Papar: 47.10 Vrms: 120.90 Irms: 0.39 FP: 0.76
Preal: 35.99 Papar: 47.34 Vrms: 121.16 Irms: 0.39 FP: 0.76
Preal: 35.76 Papar: 47.27 Vrms: 120.98 Irms: 0.39 FP: 0.76
Preal: 35.60 Papar: 47.00 Vrms: 120.80 Irms: 0.39 FP: 0.76
Preal: 36.04 Papar: 47.34 Vrms: 120.99 Irms: 0.39 FP: 0.76

```

Fonte: *Print screen* do software Arduino

Na função de descongelamento, podemos observar basicamente 3 estágios dentro deste ciclo. Cada estágio em um nível de potência, de corrente e, conseqüentemente, de fator de potência. Interessante ainda observar que, o estágio de maior potência na função de descongelamento é uma operação em níveis de potência muito parecidos com a função de aquecimento.

A partir desses valores, podemos criar as condições matemáticas nas quais o *software* possa identificar o acionamento deste microondas, em qualquer uma das duas funções (aquecimento e descongelamento) quando o microondas for integrado a todo o sistema. Para a função de aquecimento, teremos apenas um grupo de condicionais controlando a detecção da carga, pois, como é possível observar na Figura 36, o aquecimento se dá com os parâmetros mantendo certa uniformidade. Já a função descongelamento, trabalha com um ciclo que inclui 3 estágios com valores de

parâmetros distintos entre si. Para que a detecção da carga ocorra em qualquer um destes 3 estágios, precisamos de 3 grupos de condicionais, um para cada estágio. Observando o quadro acima, foi montada a Tabela 8 que apresenta o valor máximo e o valor mínimo que cada um dos parâmetros apresentou no quadro coletado em cada uma das funções do forno de microondas:

Tabela 8 - Valores mínimos e máximos dos parâmetros medidos no forno de micro-ondas

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Micro-ondas	111.57	10.65	1140.64	1188.15	0,96
Valor Mínimo	Aquecimento	111.19	10.59	1134.51	1180.83	0,96
Valor Máximo	Micro-ondas	121.16	0.40	36.21	48.52	0.76
Valor Mínimo	Descongelamento (ciclo 1)	120.80	0.39	35.44	46.87	0.76
Valor Máximo	Micro-ondas	119.19	6.03	169.00	727.10	0.23
Valor Mínimo	Descongelamento (ciclo 2)	118.57	5.40	104.00	643.49	0.15
Valor Máximo	Micro-ondas	110.89	11.07	1167.22	1227.33	0,95
Valor Mínimo	Descongelamento (ciclo 3)	110.54	11.04	1160.48	1222.09	0,95

Observando os valores da tabela e aplicando os erros associados a imprecisão dos sensores (como descrito na Tabela 2), temos que a identificação do forno de microondas obedecerá a seguinte estrutura condicional:

➤ **Para o Microondas em modo aquecimento:**

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * V_{max} \quad (134)$$

Ou seja:

$$0,985 * 111,09 \leq V_{rms} \leq 1,015 * 111,57 \quad (135)$$

Assim, temos:

$$\mathbf{109,42 V \leq V_{rms} \leq 113,24 V} \quad (136)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (137)$$

Ou seja:

$$0,985 * 10,59 \leq I_{rms} \leq 1,015 * 10,65 \quad (138)$$

Assim, temos:

$$\mathbf{10,43 A \leq I_{rms} \leq 10,81 A} \quad (139)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (140)$$

Ou seja

$$0,97 * 1134,51 \leq P \leq 1,03 * 1140,64 \quad (141)$$

Assim, temos:

$$\mathbf{1100,47 W \leq P \leq 1174,86 W} \quad (142)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (143)$$

Ou seja:

$$0,97 * 1180,83 \leq S \leq 1,03 * 1188,15 \quad (144)$$

Assim, temos:

$$1145,40 \text{ VA} \leq S \leq 1223,80 \text{ VA} \quad (145)$$

- **Fator de Potência (erro de 1%)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (146)$$

Ou seja:

$$0,99 * 0,96 \leq FP \leq 1,01 * 0,96 \quad (147)$$

Assim, temos:

$$0,95 \leq FP \leq 0,97 \quad (148)$$

- **Para o forno de microondas em modo de descongelamento:**

i. Para o ciclo 1:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (149)$$

Ou seja:

$$0,985 * 120,80 \leq V_{rms} \leq 1,015 * 121,16 \quad (150)$$

Assim, temos:

$$118,99 \text{ V} \leq V_{rms} \leq 122,98 \text{ V} \quad (151)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (152)$$

Ou seja:

$$0,985 * 0,39 \leq I_{rms} \leq 1,015 * 0,40 \quad (153)$$

Assim, temos:

$$\mathbf{0,38 A \leq I_{rms} \leq 0,41 A} \quad (154)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (155)$$

Ou seja:

$$0,97 * 35,44 \leq P \leq 1,03 * 36,21 \quad (156)$$

Assim, temos:

$$\mathbf{34,38 W \leq P \leq 37,29 W} \quad (157)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (158)$$

Ou seja:

$$0,97 * 46,87 \leq S \leq 1,03 * 48,52 \quad (159)$$

Assim, temos:

$$\mathbf{45,46 VA \leq S \leq 49,97 VA} \quad (160)$$

- **Fator de Potência (erro de 1,0 %)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (161)$$

Ou seja:

$$0,99 * 0,76 \leq FP \leq 1,01 * 0,76 \quad (162)$$

Assim, temos:

$$0,75 \leq FP \leq 0,77 \quad (163)$$

ii. Para o ciclo 2:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (164)$$

Ou seja:

$$0,985 * 118,57 \leq V_{rms} \leq 1,015 * 119,19 \quad (165)$$

Assim, temos:

$$116,79 V \leq V_{rms} \leq 120,98 V \quad (166)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (167)$$

Ou seja:

$$0,985 * 5,40 \leq I_{rms} \leq 1,015 * 6,03 \quad (168)$$

Assim, temos:

$$5,32 A \leq I_{rms} \leq 6,12 A \quad (169)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (170)$$

Ou seja:

$$0,97 * 104,00 \leq P \leq 1,03 * 169,00 \quad (171)$$

Assim, temos:

$$100,88 W \leq P \leq 174,07 W \quad (172)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (173)$$

Ou seja:

$$0,97 * 643,49 \leq S \leq 1,03 * 727,10 \quad (174)$$

Assim, temos:

$$\mathbf{624,18 VA \leq S \leq 748,91 VA} \quad (175)$$

- **Fator de Potência (erro de 1,0 %)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (176)$$

Ou seja:

$$0,99 * 0,15 \leq FP \leq 1,01 * 0,23 \quad (177)$$

Assim, temos:

$$\mathbf{0,15 \leq FP \leq 0,23} \quad (178)$$

iii. Para o ciclo 3:

- **Tensão RMS (erro de 1,5 %)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (179)$$

Ou seja:

$$0,985 * 110,54 \leq V_{rms} \leq 1,015 * 110,89 \quad (180)$$

Assim, temos:

$$\mathbf{108,89 V \leq V_{rms} \leq 112,55 V} \quad (181)$$

- **Corrente RMS (erro de 1,5 %)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (182)$$

Ou seja:

$$0,985 * 11,04 \leq I_{rms} \leq 1,015 * 11,07 \quad (183)$$

Assim, temos:

$$\mathbf{10,87 A \leq I_{rms} \leq 11,23 A} \quad (184)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (185)$$

Ou seja:

$$0,97 * 1160,48 \leq P \leq 1,03 * 1167,22 \quad (186)$$

Assim, temos:

$$\mathbf{1125,66 W \leq P \leq 1202,24 W} \quad (187)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (188)$$

Ou seja:

$$0,97 * 1222,09 \leq S \leq 1,03 * 1227,33 \quad (189)$$

Assim, temos:

$$\mathbf{1185,42 VA \leq S \leq 1264,15 VA} \quad (190)$$

- **Fator de Potência (erro de 6,2%)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (191)$$

Ou seja:

$$0,99 * 0,95 \leq FP \leq 1,01 * 0,95 \quad (192)$$

Assim, temos:

$$0,94 \leq FP \leq 0,96 \quad (193)$$

Assim, se qualquer um dos grupos de condicionais acima forem validados pelo *software*, o forno de microondas é detectado e a variável *identificador* passa a ter valor 6, conforme a Tabela 2.

5.7 Cargas simultâneas: Lâmpadas incandescente e fluorescente

Até agora neste projeto, foram citados apenas os testes individuais das cargas, mas, com a mesma metodologia adotada até aqui, podemos identificar várias cargas sendo ligadas simultaneamente. Para provar essa afirmação, foram escolhidas duas cargas entre as analisadas até aqui: a lâmpada incandescente e a lâmpada fluorescente. Os modelos utilizados são os mesmos dos testes já descritos.

A Figura 39 mostra um *frame* do *Serial Monitor* com os valores dos parâmetros de regime permanente obtidos pelo microcontrolador quando as duas cargas são ligadas de forma simultânea. Como a lâmpada fluorescente não apresenta fator de potência unitário, é natural que a integração entre uma lâmpada incandescente e uma lâmpada fluorescente não apresente fator de potência unitário. Mas pelo triângulo de potências anteriormente apresentado na Figura 3, podemos imaginar que o fator de potência deve aumentar no conjunto em relação ao teste individual da lâmpada fluorescente, já que a potência ativa nesse caso aumenta devido à integração da lâmpada incandescente ao conjunto.

Figura 39 - Dados do *Serial Monitor* para a ligação simultânea das lâmpadas incandescente e fluorescente

```

Preal: 58.18 Papar: 64.64 Vrms: 119.82 Irms: 0.53 FP: 0.90
Preal: 58.29 Papar: 64.92 Vrms: 120.03 Irms: 0.53 FP: 0.90
Preal: 57.82 Papar: 64.09 Vrms: 119.46 Irms: 0.53 FP: 0.90
Preal: 58.64 Papar: 65.06 Vrms: 120.32 Irms: 0.53 FP: 0.90
Preal: 57.94 Papar: 64.18 Vrms: 119.60 Irms: 0.53 FP: 0.90
Preal: 58.30 Papar: 64.56 Vrms: 120.34 Irms: 0.53 FP: 0.90
Preal: 58.09 Papar: 64.55 Vrms: 119.88 Irms: 0.53 FP: 0.90
Preal: 58.38 Papar: 64.84 Vrms: 120.24 Irms: 0.53 FP: 0.90
Preal: 58.50 Papar: 64.97 Vrms: 120.13 Irms: 0.53 FP: 0.90
Preal: 58.32 Papar: 64.88 Vrms: 120.25 Irms: 0.53 FP: 0.90
Preal: 58.44 Papar: 64.93 Vrms: 120.15 Irms: 0.53 FP: 0.90
Preal: 58.49 Papar: 65.01 Vrms: 120.23 Irms: 0.53 FP: 0.90
Preal: 58.39 Papar: 64.94 Vrms: 120.28 Irms: 0.53 FP: 0.90
Preal: 58.46 Papar: 64.91 Vrms: 120.21 Irms: 0.53 FP: 0.90
Preal: 57.88 Papar: 64.30 Vrms: 119.68 Irms: 0.53 FP: 0.90
Preal: 58.71 Papar: 65.03 Vrms: 120.29 Irms: 0.53 FP: 0.90
Preal: 57.88 Papar: 64.29 Vrms: 119.78 Irms: 0.53 FP: 0.90
Preal: 58.43 Papar: 64.92 Vrms: 120.17 Irms: 0.53 FP: 0.90
Preal: 58.20 Papar: 64.65 Vrms: 119.57 Irms: 0.53 FP: 0.90
Preal: 58.33 Papar: 64.84 Vrms: 120.05 Irms: 0.53 FP: 0.90
Preal: 57.71 Papar: 64.16 Vrms: 119.55 Irms: 0.53 FP: 0.90
Preal: 58.43 Papar: 64.79 Vrms: 120.08 Irms: 0.53 FP: 0.90
Preal: 58.37 Papar: 64.83 Vrms: 120.12 Irms: 0.53 FP: 0.90
Preal: 58.60 Papar: 65.03 Vrms: 120.14 Irms: 0.53 FP: 0.90
Preal: 57.96 Papar: 64.38 Vrms: 119.64 Irms: 0.53 FP: 0.90
Preal: 58.26 Papar: 64.63 Vrms: 120.07 Irms: 0.53 FP: 0.90
Preal: 58.60 Papar: 65.09 Vrms: 120.15 Irms: 0.53 FP: 0.90
Preal: 58.60 Papar: 65.08 Vrms: 120.17 Irms: 0.53 FP: 0.90
Preal: 57.99 Papar: 64.31 Vrms: 119.68 Irms: 0.53 FP: 0.90
Preal: 58.34 Papar: 64.77 Vrms: 120.22 Irms: 0.53 FP: 0.90
Preal: 57.76 Papar: 64.10 Vrms: 119.66 Irms: 0.53 FP: 0.90
Preal: 58.59 Papar: 65.00 Vrms: 120.32 Irms: 0.53 FP: 0.90
Preal: 58.20 Papar: 64.60 Vrms: 119.87 Irms: 0.53 FP: 0.90
Preal: 58.43 Papar: 65.02 Vrms: 120.38 Irms: 0.53 FP: 0.90
Preal: 57.88 Papar: 64.31 Vrms: 119.75 Irms: 0.53 FP: 0.90

```

Fonte: *Print screen* do software Arduino

A partir desses valores, podemos criar a condição matemática na qual o *software* possa identificar a ligação simultânea das cargas quando todas as cargas forem integradas ao sistema. Observando o quadro acima, foi montada a Tabela 9 que apresenta o valor máximo e o valor mínimo que cada um dos parâmetros poderá apresentar.

Tabela 9 - Valores mínimos e máximos dos parâmetros medidos na ligação de cargas simultâneas

	Tipo de carga	Tensão RMS (V)	Corrente RMS (A)	Potência ativa (W)	Potência aparente (VA)	Fator de potência
Valor Máximo	Lâmpada incandescente e lâmpada	120.38	0.53	58.59	65.08	0.90
Valor Mínimo	fluorescente (simultaneamente)	119.46	0.53	57.76	64.09	0.90

Observando os valores da Tabela 9 e aplicando os erros associados à imprecisão dos sensores como descrito pela Tabela 2, temos que a identificação da ligação simultânea das duas cargas obedecerá a seguinte estrutura condicional:

- **Tensão RMS (erro de 1,5%)**

$$0,985 * V_{min} \leq V_{rms} \leq 1,015 * X_{max} \quad (194)$$

Ou seja:

$$0,985 * 119,46 \leq V_{rms} \leq 1,015 * 120,38 \quad (195)$$

Assim, temos,

$$117,67 V \leq V_{rms} \leq 122,19 V \quad (196)$$

- **Corrente RMS (erro de 1,5%)**

$$0,985 * I_{min} \leq I_{rms} \leq 1,015 * I_{max} \quad (197)$$

Ou seja:

$$0,985 * 0,53 \leq I_{rms} \leq 1,015 * 0,53 \quad (198)$$

Assim, temos:

$$0,52 A \leq I_{rms} \leq 0,54 A \quad (199)$$

- **Potência Ativa (erro de 3%)**

$$0,97 * P_{min} \leq P \leq 1,03 * P_{max} \quad (200)$$

Ou seja:

$$0,97 * 57,76 \leq P \leq 1,03 * 58,59 \quad (201)$$

Assim, temos:

$$56,03 W \leq P \leq 60,34 W \quad (202)$$

- **Potência Aparente (erro de 3%)**

$$0,97 * S_{min} \leq S \leq 1,03 * S_{max} \quad (203)$$

Ou seja:

$$0,97 * 64,09 \leq S \leq 1,03 * 65,08 \quad (204)$$

Assim, temos:

$$\mathbf{62,17 VA \leq S \leq 67,03 VA} \quad (205)$$

- **Fator de Potência (erro de 1,0 %)**

$$0,99 * FP_{min} \leq FP \leq 1,01 * FP_{max} \quad (206)$$

Ou seja:

$$0,99 * 0,90 \leq FP \leq 1,01 * 0,90 \quad (207)$$

Assim, temos:

$$\mathbf{0,89 \leq FP \leq 0,91} \quad (208)$$

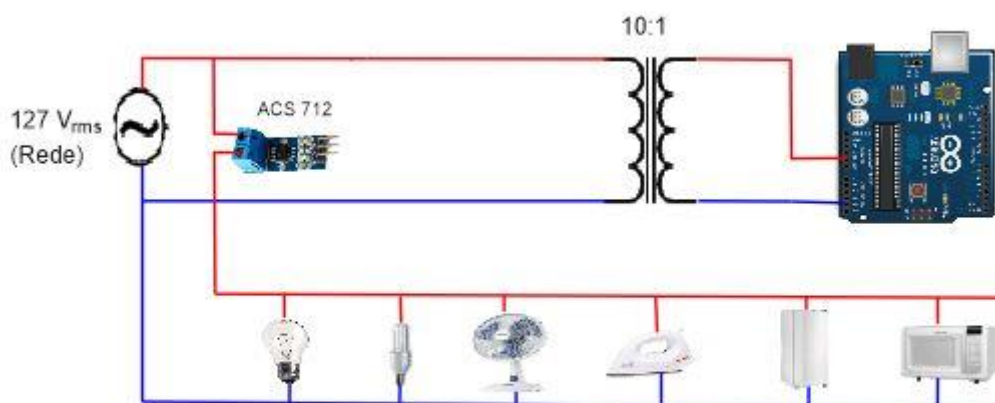
Assim, se as condicionais acima forem validadas pelo *software*, a ligação simultânea das lâmpadas incandescente e fluorescente é detectado e, de acordo com a Tabela 2, a variável *identificador* passa a ter valor 7.

6 RESULTADOS DOS TESTES DO SISTEMA DE DETECÇÃO DE CARGAS

Como foi dito inicialmente, a proposta desse projeto é o desenvolvimento de um sistema para detecção de cargas a partir de análises feitas nos sinais de tensão e corrente, utilizando suas características de regime permanente para a reconhecimento do tipo de carga acionada em cada caso. Para isso foi criado um *software* capaz de analisar os dados recebidos e processá-los de forma a identificar o tipo de carga acionada a cada instante de tempo. A partir disso, foi criada uma interface simples e intuitiva para a apresentação dos resultados obtidos.

Como mostrado até aqui, toda a ideia inicial foi executada faltando apenas apresentar os resultados obtidos na detecção das cargas a partir da metodologia aqui descrita. Para realizar os testes de detecção, todas as cargas foram integradas ao sistema, Figura 39, e foram ligadas conforme apresentado a seguir.

Figura 39 – Esquema de ligação do sistema com as cargas já integradas



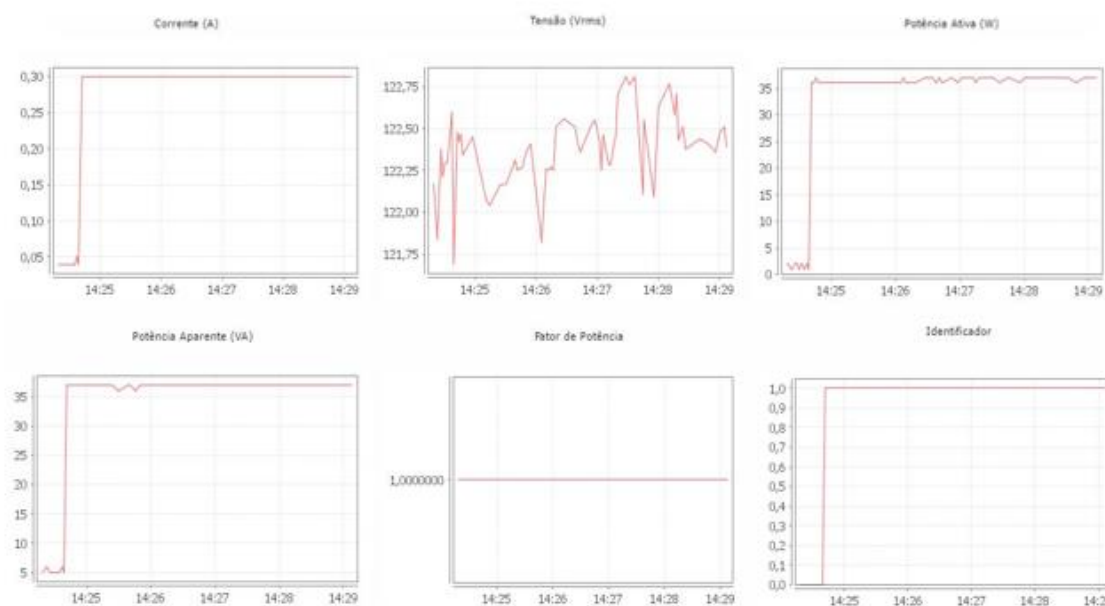
Fonte: *Print screen do software ScadaBR*

Com todas as cargas integradas, os testes de detecção foram realizados conforme a ordem descrita nas seções abaixo.

6.1 Resultado do teste para a identificação da lâmpada incandescente

Com todo o sistema integrado, apenas a lâmpada incandescente é ligada e o resultado da leitura dos sensores é mostrado nos gráficos da Figura 40:

Figura 40 - Gráficos obtidos durante a ligação da lâmpada incandescente



Fonte: Print screen do software ScadaBR

A parte inicial do gráfico de corrente mostra o degrau que ocorre no momento em que a lâmpada incandescente é ligada. Nesse momento ocorrem degraus também nas potências e o fator de potência se mantém unitário como era esperado, visto termos ligada uma carga resistiva.

Comparando os valores obtidos nos gráficos com as equações da seção 5.1, vemos que todas as condicionais são satisfeitas, tornando o valor da variável *identificador* igual a 1 como era esperado. Podemos notar também pelo último gráfico da Figura 40 que durante a amostragem (no gráfico estão mostrados aproximadamente 4 minutos de amostras coletadas), o valor do *identificador* vai de 0 a 1 quando a lâmpada é ligada e se mantém em 1, mostrando a confiabilidade do processo de detecção.

A detecção da carga é representada na interface gráfica de modo intuitivo, com a imagem da lâmpada incandescente se acendendo na tela, como mostrado na Figura 41. Também o led abaixo da lâmpada acende e o botão On/Off fica verde para indicar que existe uma carga ligada no sistema.

Figura 41 - Interface de detecção durante a ligação da lâmpada incandescente



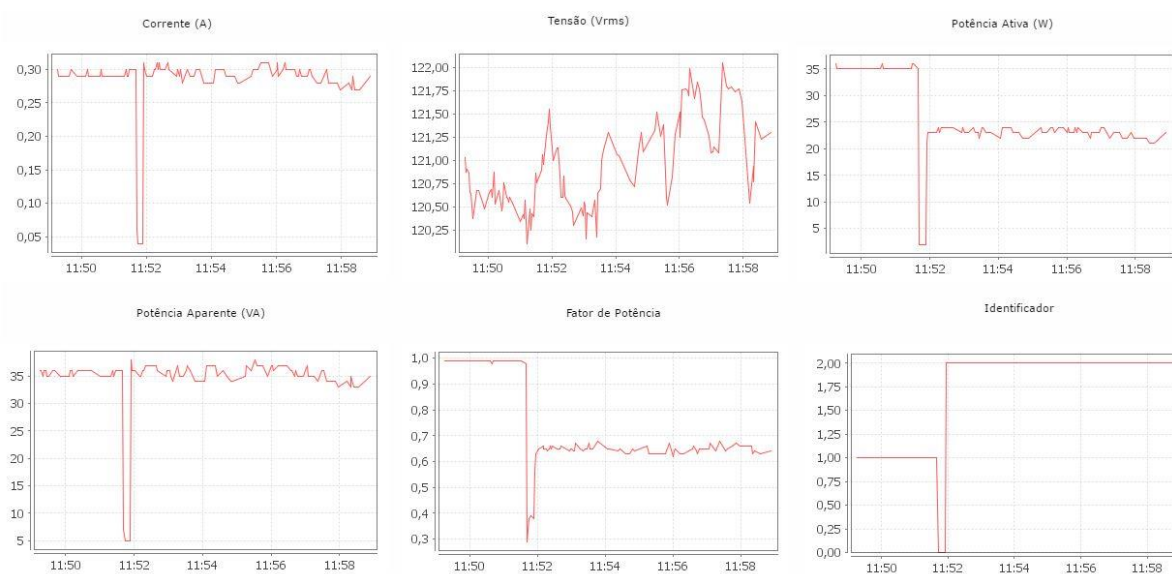
Fonte: *Print screen do software ScadaBR*

Podemos observar na interface que o reconhecimento da lâmpada incandescente funcionou de forma satisfatória.

6.2 Resultado do teste para a identificação da lâmpada fluorescente

Agora, desligando a lâmpada incandescente e ligando a lâmpada fluorescente, obtemos os gráficos mostrados na figura abaixo. Podemos observar que, no momento imediatamente após desligarmos a lâmpada incandescente, o gráfico de corrente e das potências vai a um valor próximo de zero e, imediatamente após a lâmpada fluorescente ser ligada esses valores tem um novo degrau, sendo que a potência ativa tem uma redução em relação aos níveis da lâmpada incandescente, enquanto a potência aparente permanece em níveis próximos ao da carga anterior. Isso leva a uma queda no fator de potência do sistema como é possível observar no quinto gráfico da Figura 42.

Figura 42 - Gráficos obtidos durante a ligação da lâmpada fluorescente



Fonte: *Print screen do software ScadaBR*

É interessante notar também que o valor da variável *identificador* (sexto gráfico) era 1 enquanto a lâmpada incandescente estava ligada, quando a carga é desligada o valor vai para zero e imediatamente após a lâmpada fluorescente ser ligada, o *identificador* assume o valor 2, estando em conformidade com a Tabela 1 e mostrando o bom funcionamento do *identificador*.

O funcionamento do sistema de detecção expresso nos gráficos gera a identificação confiável da lâmpada fluorescente, como visto na Figura 43.

Figura 43 - Interface de detecção durante a ligação da lâmpada incandescente



Fonte: *Print screen do software ScadaBR*

Podemos observar na interface apresentada de modo intuitivo, que a detecção da lâmpada fluorescente funcionou de forma adequada.

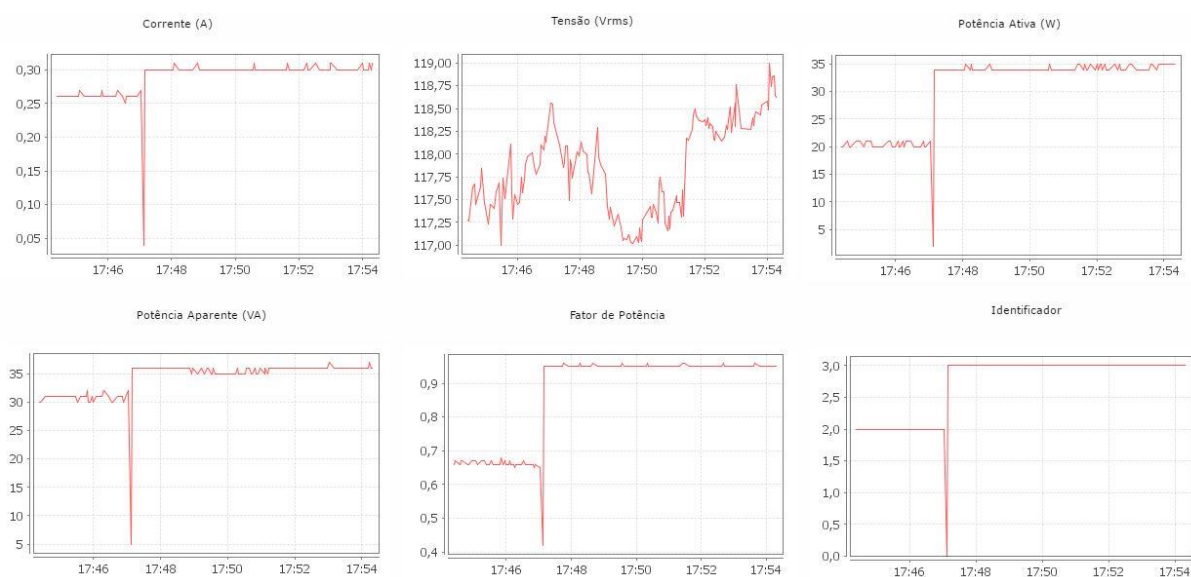
6.3 Resultado do teste para a identificação do ventilador

Como dito anteriormente, os testes no ventilador de mesa escolhido foram realizados em 3 pontos de operação distintos (3 velocidades de operação). De forma equivalente, os resultados da detecção de cargas serão apresentados nos três pontos de operação analisados.

➤ Potência mínima (velocidade 1)

Neste teste, foi desligada a lâmpada fluorescente e imediatamente após, o ventilador foi ligado em potência mínima. Podemos observar o bom funcionamento do sistema de detecção observando na Figura 44 que no exato instante em que a carga muda (instante em que os parâmetros têm uma variação abrupta em seus valores), a variável *Identificador* muda seu valor e assume o valor 3, que de acordo com a Tabela 1, corresponde à detecção do ventilador, como era esperado.

Figura 44 - Gráficos obtidos durante a ligação do ventilador em potência mínima

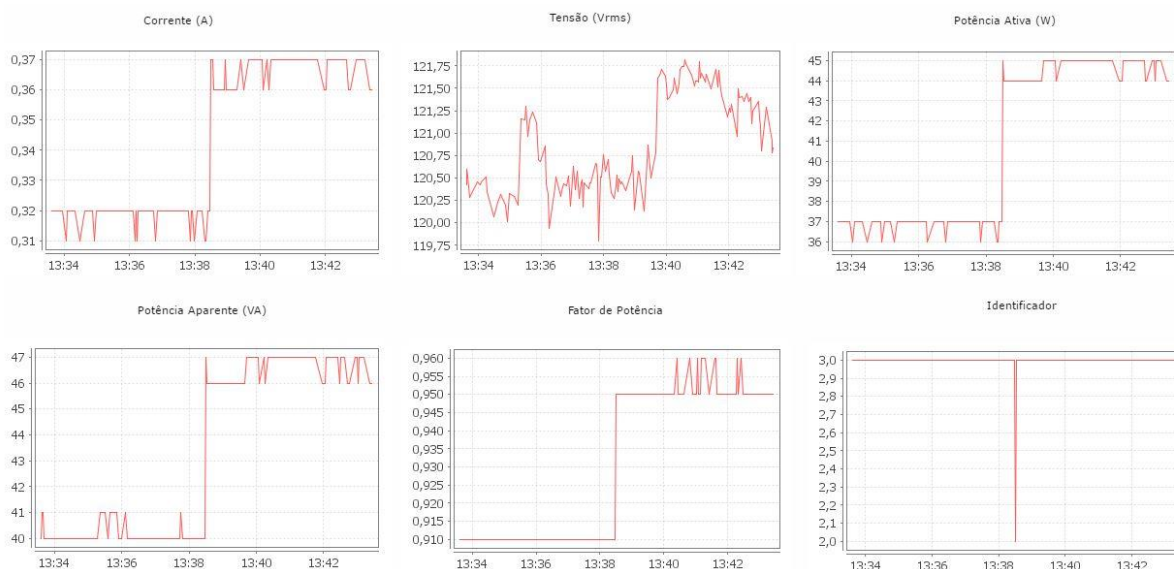


Fonte: *Print screen do software ScadaBR*

➤ Potência média (velocidade 2)

Agora, mudando o ponto de operação, aumentando a potência e a velocidade do ventilador, temos os gráficos mostrados na Figura 45.

Figura 45 - Gráficos obtidos durante a ligação do ventilador em potência média



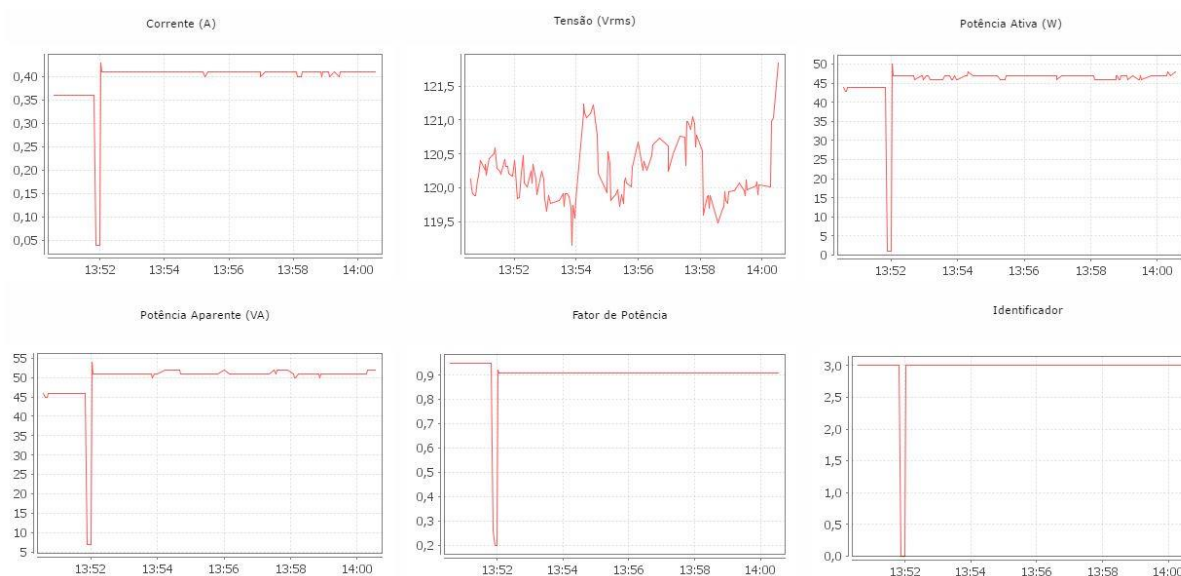
Fonte: *Print screen do software ScadaBR*

Podemos observar que no momento em que se muda o ponto de operação da carga ocorre um degrau nos gráficos de corrente e das potências, além de um aumento no fator de potência. Apesar disso, devido a definição das condicionais para cada ponto de operação, a variável *identificador* permanece no valor 3 mesmo com a variação dos parâmetros de regime permanente.

➤ Potência máxima (velocidade 3)

Finalmente, o ventilador é operado em sua potência máxima, os gráficos são mostrados na Figura 46 abaixo.

Figura 46 - Gráficos obtidos durante a ligação do ventilador em potência máxima



Fonte: *Print screen do software ScadaBR*

Novamente, é possível observar um degrau nos gráficos de corrente e das potências, no exato momento em que se muda o ponto de operação da carga. Há ainda uma pequena variação negativa no fator de potência. Porém, apesar da variação dos parâmetros, as condicionais garantem o valor da variável *identificador* se mantendo em 3, indicando que a carga analisada é o ventilador (conforme Tabela 2). A Figura 47 abaixo mostra o resultado dessa identificação na interface de detecção de cargas.

Figura 47 - Interface de detecção durante a ligação da lâmpada incandescente



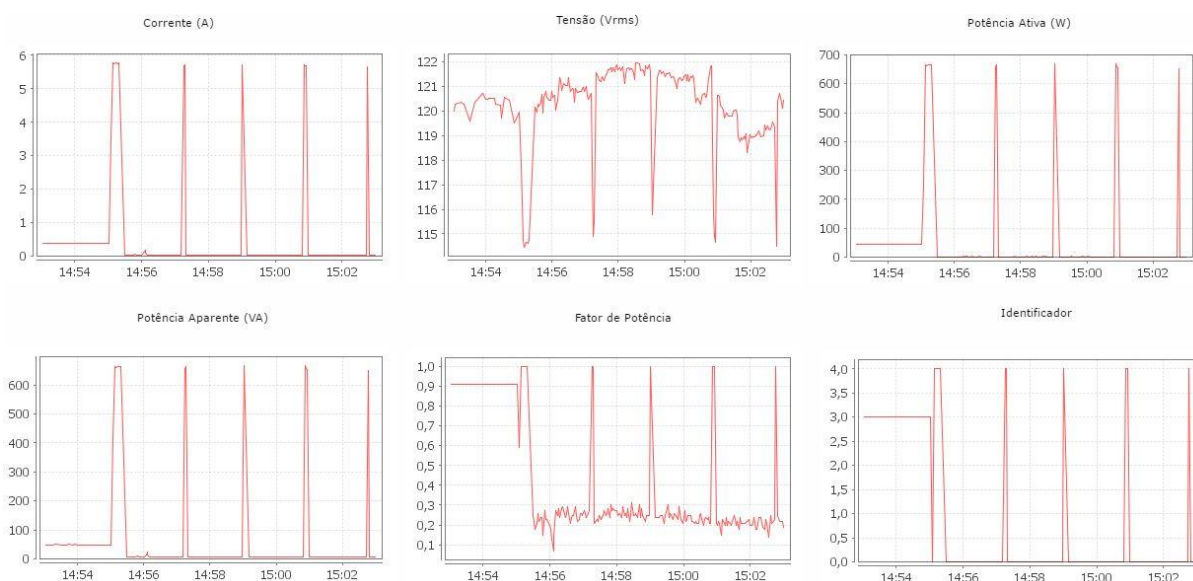
Fonte: *Print screen do software ScadaBR*

Podemos observar na interface apresentada de modo intuitivo, que a detecção do ventilador funcionou de forma adequada.

6.4 Resultado do teste para a identificação do ferro de passar

Para os testes com o ferro de passar, o ventilador foi desligado e imediatamente após, o ferro de passar foi ligado em sua potência mínima (menor temperatura de operação) e os gráficos mostrados na Figura 48 abaixo apresentam os resultados.

Figura 48 - Gráficos obtidos durante a ligação do ferro de passar em potência mínima

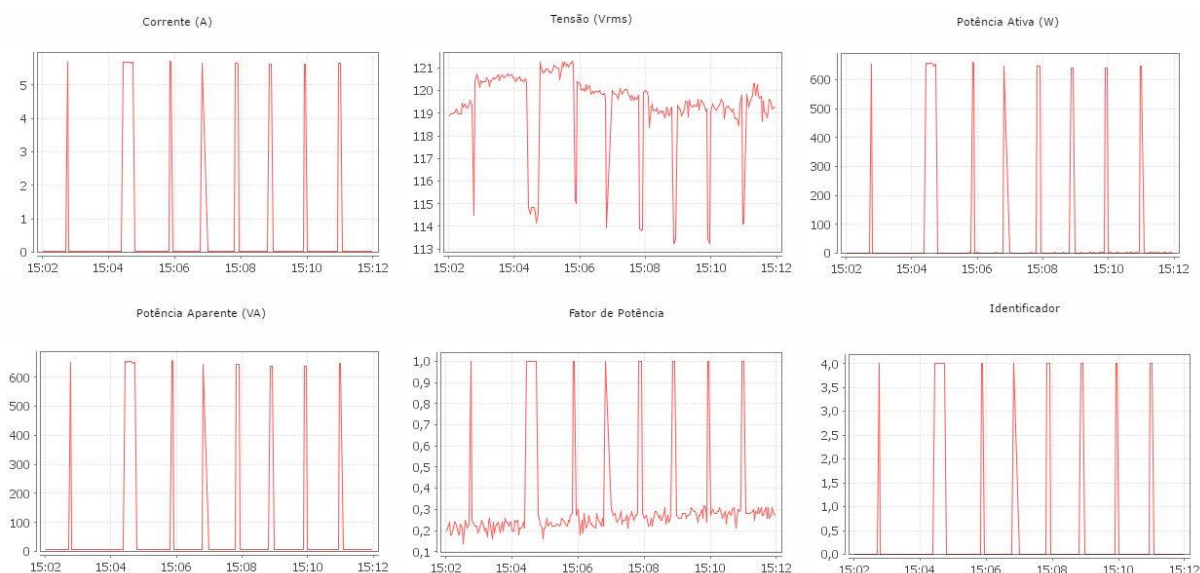


Fonte: *Print screen do software ScadaBR*

Primeiramente, podemos observar o gráfico da variável *identificador* e notar que a mesma inicialmente tem valor 3, pois a carga anterior era o ventilador. No momento em que o ventilador é desligado e o ferro de passar é ligado, há um grande degrau nos gráficos de corrente e potência, além de um afundamento no gráfico de tensão. Porém, o ferro de passar é uma carga que trabalha em modo on-off controlado por um termostato. Ou seja, quando a prancha metálica atinge determinada temperatura, o termostato aciona um relê que desconecta a carga do sistema. Assim, o *identificador* não considera o ferro de passar em modo ON na interface durante todo seu tempo de uso, mas apenas quando o mesmo está consumindo a energia da rede, ou seja, durante os picos de corrente e potências observados nos gráficos das figuras acima. Isso pode ser mostrado também, observando que a variável *identificador* só assume o valor 4 (que está relacionado ao ferro de passar, conforme a Tabela 1) exatamente nos mesmos instantes em que acontecem os picos dos gráficos de corrente e das potências.

Regulando a temperatura do ferro de passar para a potência máxima, obtemos os seguintes gráficos mostrados na Figura 49:

Figura 49 - Gráficos obtidos durante a ligação do ferro de passar em potência máxima

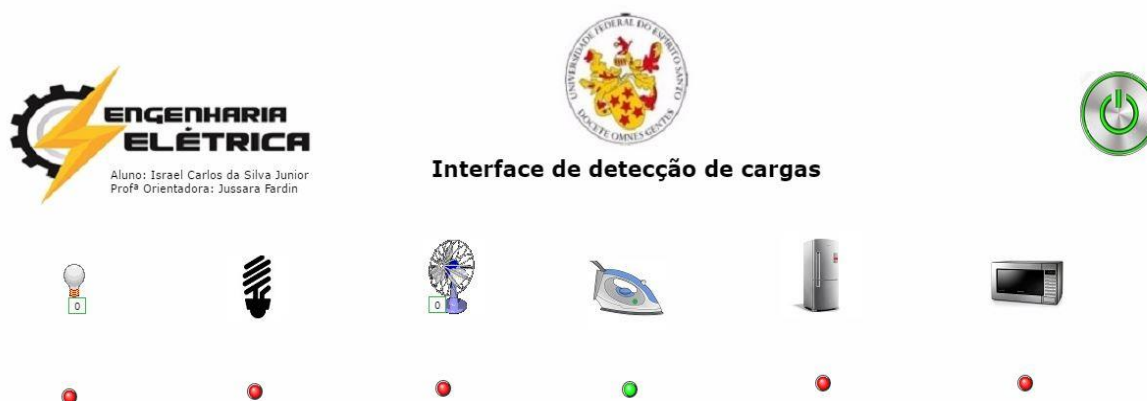


Fonte: *Print screen do software ScadaBR*

Todas as observações feitas para o ferro em potência mínima podem ser feitas para o seu funcionamento em potência máxima. A diferença está apenas nos níveis de potência e corrente, que se elevam conforme a regulagem de temperatura vai aumentando. Mas é possível perceber pela variável *identificador* que a detecção da carga ocorreu satisfatoriamente também para o funcionamento do ferro de passar em potência máxima, com a variável assumindo o valor 4 (quatro) sempre que a carga é conectada ao sistema pelo relé de on-off.

O bom funcionamento da detecção de cargas para a ligação do ferro de passar é expresso na interface conforme mostrado na Figura 50.

Figura 50 -Interface de detecção durante a ligação do ferro de passar



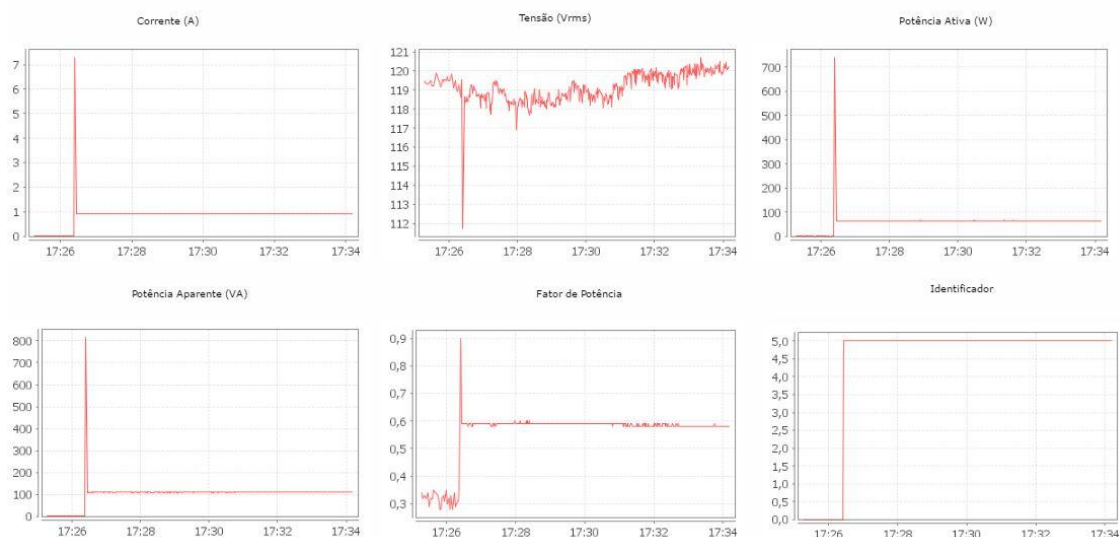
Fonte: *Print screen do software ScadaBR*

Podemos observar na interface apresentada de modo intuitivo, que a detecção do ferro de passar funcionou de forma adequada.

6.5 Resultado do teste para a identificação do refrigerador

Com todo o sistema integrado, apenas o refrigerador é ligado e o resultado da leitura dos sensores é mostrado nos gráficos da Figura 51:

Figura 51 - Gráficos obtidos durante a ligação do refrigerador



Fonte: *Print screen do software ScadaBR*

Podemos observar que existe um pico de corrente na partida do compressor e, simultaneamente, um afundamento na tensão. Mas rapidamente os valores se

estabilizam e é possível notar que a variável *identificador* recebe o valor 5 (cinco), o que está em conformidade com os valores de *identificador* associados a cada carga na Tabela 1.

O bom funcionamento da detecção de cargas para a ligação do ferro de passar é expresso na interface conforme mostrado na Figura 52.

Figura 52 - Interface de detecção durante a ligação do refrigerador



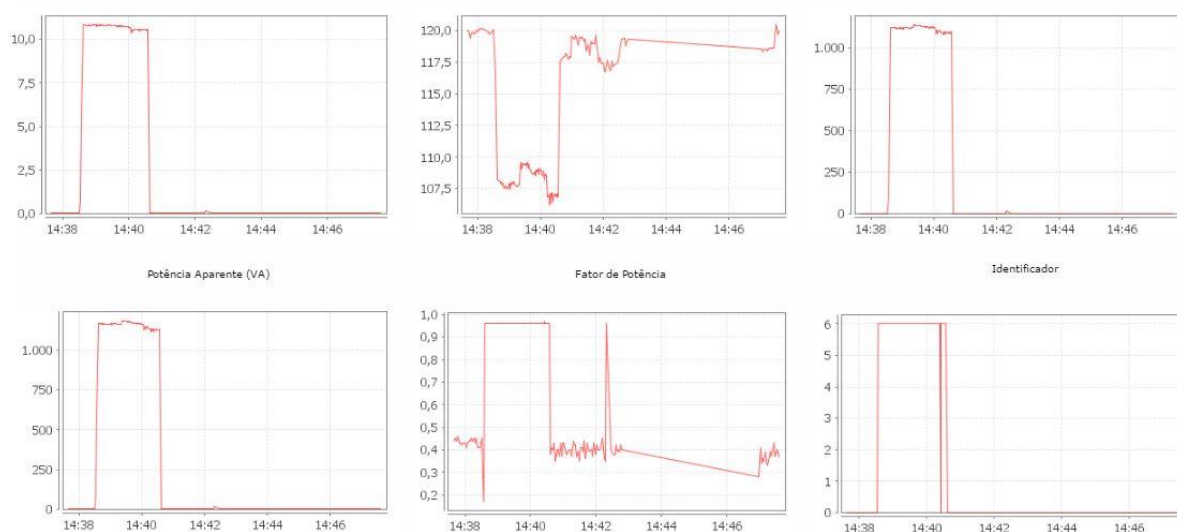
Fonte: *Print screen* do software ScadaBR

Podemos observar na interface apresentada, que a detecção do refrigerador funcionou de forma adequada.

6.6 Resultado do teste para a identificação do forno de micro-ondas

Como explicado na seção 5.6, o micro-ondas foi testado em duas funções diferente: aquecimento e descongelamento de alimentos. Para a ligação da função 'aquecimento', os gráficos dos parâmetros são mostrados na Figura 53:

Figura 53 - Gráficos obtidos durante a ligação do forno de micro-ondas em modo 'aquecimento'

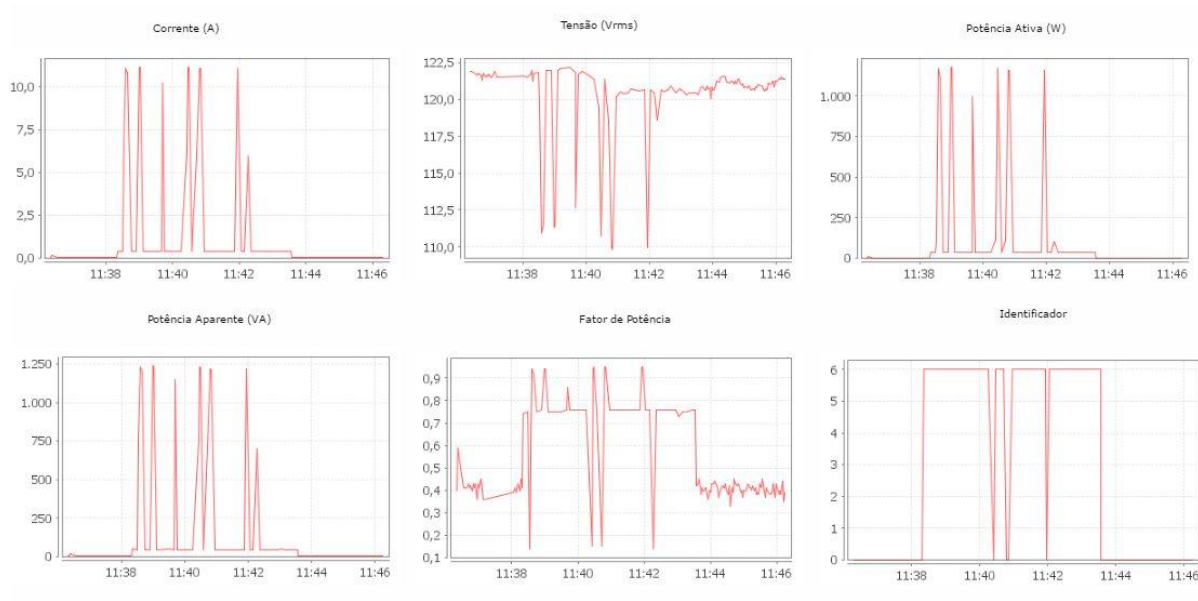


Fonte: *Print screen do software ScadaBR*

Vemos que no momento da ligação existe um degrau nos gráficos de corrente e potência, e em contraponto, observamos um afundamento considerável na tensão do sistema. Após o ciclo de aquecimento temos, como esperado, uma queda abrupta nos valores de corrente e potência e o restabelecimento do sinal de tensão. Nota-se ainda, a partir do último gráfico da Figura 53, que a variável *identificador* se manteve no valor 6 (o que era o esperado de acordo com a Tabela 2) por praticamente todo o tempo amostral, falhando uma única vez e durante um curto intervalo de tempo.

Já na função de descongelamento, como mostrado na seção 8.6, o forno de microondas trabalha em 3 níveis de potência diferentes, distribuindo de forma não uniforme sua potência ao longo do tempo. Os gráficos do microondas na função 'descongelamento' são mostrados na Figura 54:

Figura 54 - Gráficos obtidos durante a ligação do forno de micro-ondas em modo 'descongelamento'



Fonte: *Print screen do software ScadaBR*

Como já havia sido possível se observar no *Serial Monitor*, o descongelamento ocorre com a variação abrupta dos níveis de potência. Mas como já explicado na seção 5.6, a identificação do microondas foi programada para cada um dos três níveis de potência do ciclo de descongelamento observados no gráfico: tanto em baixa quanto em alta potência, cobrindo também parte dessa transição. Mas, durante essa variação de níveis de potência, algumas amostras acabam ficando fora dos intervalos de valores para a identificação, causando algumas falhas da detecção da carga. Podemos observar pelo gráfico da variável *identificador* que, em alguns momentos do período amostral, a mesma não manteve seu valor 6 (como era esperado), porém na maior parte do tempo, a identificação aconteceu de forma correta. O bom funcionamento do sistema de detecção expresso nos gráficos gera a identificação do forno de micro-ondas, como visto na Figura 55.

Figura 55 - Interface de detecção durante a ligação do forno de micro-ondas



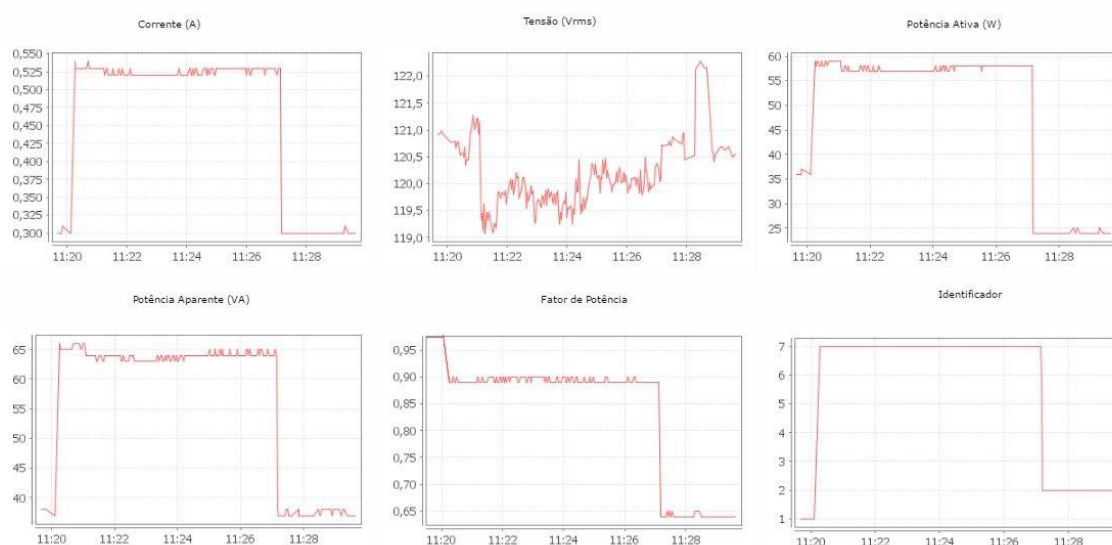
Fonte: *Print screen do software ScadaBR*

Podemos observar na interface que o reconhecimento do forno de micro-ondas funcionou de forma satisfatória.

6.7 Resultado do teste para identificação de cargas simultâneas

Para a demonstração de que seria possível executar a detecção de múltiplas cargas ligadas simultaneamente, foram ligadas as lâmpadas incandescente e fluorescente simultaneamente como mostrado em 5.7. Para a demonstração foi executada a seguinte ordem: Com a lâmpada incandescente já ligada, é ligada também a lâmpada fluorescente e, em seguida, a lâmpada incandescente é desligada. Os resultados deste teste são mostrados nos gráficos da Figura 56.

Figura 56 - Gráficos obtidos durante a ligação simultânea das lâmpadas incandescente e fluorescente



Fonte: *Print screen do software ScadaBR*

Olhando para o último gráfico da Figura 56, vemos que a variável *identificador* começa com o valor 1, o que era esperado já que apenas a lâmpada incandescente está ligada naquele momento. Quando a lâmpada fluorescente é ligada é possível perceber um salto no valor da potência ativa, porém o degrau na potência aparente é maior, o que causa uma queda no fator de potência. Simultaneamente, a variável *identificador* recebe o valor 7 (sete), indicando que as lâmpadas estão ligadas simultaneamente. No fim do teste, a lâmpada incandescente é desligada, os níveis de potência ativa diminuem mais que os níveis de potência aparente, o que causa uma nova queda do fator de potência. Já a variável *identificador* recebe o valor 2 (dois), já que nesse momento só a lâmpada incandescente está ligada.

O bom funcionamento do sistema de detecção para cargas simultâneas expresso nos gráficos gera a identificação confiável das lâmpadas incandescente e fluorescente, como visto na Figura 57.

Figura 57 - Interface de detecção durante a ligação de duas cargas simultaneamente (lâmpadas)



Fonte: *Print screen do software ScadaBR*

Podemos observar na interface apresentada, que a detecção das lâmpadas incandescente e fluorescente ligadas simultaneamente funcionou de forma adequada.

7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado um sistema de monitoramento para realizar a detecção de vários tipos de cargas partindo apenas dos sinais de tensão e corrente geradas por essas cargas no ramal de entrada de uma unidade consumidora. Esse sistema além de detectar o tipo de carga que gerou o sinal lido, calcula ainda as tensões e correntes RMS, potências ativa e aparente, além do fator de potência. O resultado desse processamento e da detecção de cada tipo de carga, foi mostrado em gráficos e imagens em uma interface intuitiva montada na plataforma de criação de supervisórios ScadaBR.

O processador Atmel do microcontrolador Arduino, atendeu perfeitamente às expectativas sendo capaz de processar os sinais recebidos pelos sensores de tensão e corrente e calcular os valores de todos os parâmetros desejados no projeto. O Arduino atendeu de forma satisfatória também à comunicação com o ScadaBr, a partir de um protocolo *Modbus* (com a arquitetura mestre-escravo, onde o dispositivo mestre é o computador onde está instalado o ScadaBr e o escravo é o microcontrolador que processa os sinais dos sensores), entregando os dados calculados pelo microcontrolador ao supervisor, e permitindo uma atualização desses dados a cada 100 ms (tempo de atualização escolhido na configuração do *Data Source*), o que possibilitou apresentar os gráficos dos parâmetros de forma precisa e fiel aos sinais recebidos nos sensores.

Na apresentação dos dados, o ScadaBr atendeu de forma satisfatória ao que lhe foi proposto. Os recursos de múltiplos gráficos, de texto *HTML* e de *Metadados* (mudança de estado de uma representação gráfica) permitiram a criação de uma interface simples, dinâmica e intuitiva para a apresentação dos resultados da detecção das diversas cargas analisadas.

Os resultados obtidos neste projeto mostraram a viabilidade do sistema de detecção de cargas a partir da comparação dos cálculos em tempo real com os valores dos parâmetros pré-determinados. Foi mostrado ainda que é possível a detecção de cargas ligadas simultaneamente partindo da mesma metodologia de identificação para uma carga ligada de forma isolada.

Um possível trabalho futuro constituirá em automatizar o processo de pré-determinação dos parâmetros de regime permanente de cada carga, ou seja, o próprio sistema de detecção seria capaz de criar o perfil de uma carga a partir do momento

que a mesma é conectada à rede, e a partir disso, o processo de identificação seria feito da mesma forma que foi realizado neste projeto, através da comparação desse perfil criado com os sinais lidos em tempo real.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BALANÇO ENERGÉTICO NACIONAL, **Relatório Final – Consumo de Energia por Setor** – Capítulo 3, Ano Base 2015
- [2] MINISTÉRIO DE MINAS E ENERGIA & EMPRESA DE PESQUISA ENERGÉTICA, **Relatório Plano Decenal de Expansão 2008-2017, Capítulo II.**
- [3] EPE: **Eficiência energética, Estudos de demanda** (2012-2021), Capítulo V.
- [4] CAÑÓN, A. D. O., **Sistema Neural De Monitoração Não Invasiva De Carga Elétrica Residencial**, Dissertação de Mestrado, UFRJ, 2009.
- [5] PROCEL: **Tabela de Produtos contemplados com o selo PROCEL.** Disponível em: <http://www.procelinfo.com.br/resultadosprocel2016/docs/rel_procel2016_web.pdf>. Acesso em: 20 fev. 2017
- [6] AZZINI, H. A. D., **Avaliação de Técnicas para Monitoramento Não Intrusivo de Cargas Residenciais com Fins de Auditoria Energética**, Dissertação de Mestrado, UFES, 2012.
- [7] SILVA, T., **Estimativa De Padrão De Consumo Energético De Comunidades Carentes, Estudo De Caso – Complexo Da Maré.** Dissertação de Mestrado, UFRJ, 2003.
- [8] ALLEGRO Microsystems, LCC. **Datasheet Hall Effect-Based Linear Current Sensor IC:**
Disponível em: <<http://www.allegromicro.com/~media/files/datasheets/acs712-datasheet.ashx>>. Acesso em: 21 de fevereiro. 2017.
- [9] NILSSON, J. **Electric circuits (8th Edition).** Upper Sadle River: Prentice Hall, 2007.
- [10] OpenEnergyMonitor, **Open-source energy monitor.** Disponível em: <<https://openenergymonitor.org/emon/>>. Acesso em: 21 de fevereiro. 2017.
- [11] MODBUS, **The Modbus Organization.** Disponível em: <<http://www.modbus.org/>>. Acesso em: 21 de fevereiro. 2017.
- [12] SCADABR, **Supervisório Open-Source.** Disponível em: <<http://www.scadabr.com.br/>>. Acesso em: 22 fevereiro. 2017.
- [13] EFEITO JOULE, **Como funciona o forno de microondas.** Disponível em: <<http://www.efeitojoule.com/2008/09/como-funciona-forno-microondas.html>>. Acesso em 22 de fevereiro. 2017

- [14] OSRAM, **Lâmpadas incandescentes**. Disponível em: <<http://www.osram.com.br/media/resource/HIRES/617126/catlogo-de-produtos-osram---lmpadas-incandescentes-pt.pdf>>. Acesso em: 22 fevereiro. 2017.
- [15] AVANT, **Lâmpadas fluorescentes compactas integradas**. Disponível em: <avantlux.com.br/wp-content/uploads/2016/01/TRADICIONAL.pdf>. Acesso em: 22 fevereiro. 2017.
- [16] MONDIAL, **Ventiladores de mesa**. Disponível em: <<http://www.mondialine.com.br/ventilacao/vent-domestico/ventilador-de-mesa/p/nv-15-ventilador-30cm>>. Acesso em: 22 fevereiro. 2017.
- [17] MONDIAL, **Ferros de passar**. Disponível em: <<http://www.mondialine.com.br/portateis/lar/ferro-seco>>. Acesso em: 22 fevereiro. 2017.
- [18] CONSUL, **Refrigeradores**. Disponível em: <<http://www.consul.com.br/produto/geladeira-consul-degelo-seco-261l/>>. Acesso em: 25 fevereiro. 2017.
- [19] ELECTROLUX, **Forno de micro-ondas**. Disponível em: <<http://www.electrolux.com.br/produtos/cozinha/micro-ondas/mef41/>>. Acesso em: 25 fevereiro. 2017.
- [20] HAYONIK, **Transformadores**. Disponível em: <<http://www.hayonik.com/categoria/energia-transformadores>>. Acesso em: 23 Março. 2017
- [21] DOS SANTOS, W. G. V., **Transmissão de dados via rede elétrica: Tomada inteligente**. Projeto de graduação, UFES, 2017.

APÊNDICE A – PROGRAMA PRINCIPAL

```

// Início do Código principal

#include "SimpleModbusSlave.h"
#include "EmonLib.h"
EnergyMonitor emon1;      // Create an instance

#define LED 13

////////// registers of your slave //////////
enum
{
  // just add or remove registers and your good to go...
  // The first register starts at address 0
  // ADC_VAL,
  // PWM_VAL,
  VALOR_CORRENTE,
  VALOR_TENSAO,
  VALOR_PATIVA,
  VALOR_PAPARENTE,
  VALOR_FP,
  VALOR_IDENTIFICADOR,
  HOLDING_REGS_SIZE // leave this one
  // total number of registers for function 3 and 16 share the same register array
  // i.e. the same address space
};

unsigned int holdingRegs[HOLDING_REGS_SIZE]; // function 3 and 16 register array
//////////

unsigned long int previousMillis = 0;
unsigned long int currentMillis;
int identificador;

void setup()

```

```

{
  Serial.begin(9600);

  modbus_configure(&Serial, 9600, SERIAL_8N1, 1, 5, HOLDING_REGS_SIZE, holdingRegs);

  // modbus_update_comms(baud, byteFormat, id) is not needed but allows for easy update of the
  // port variables and slave id dynamically in any function.
  modbus_update_comms(9600, SERIAL_8N1, 1); // modbus_update_comms(9600, SERIAL_8N1, 1)
  pinMode(LED, OUTPUT);

  //-----
  emon1.voltage(2, 127 * 1.0570 * 3 / 4, 1.5); // Voltage: input pin, calibration, phase_shift 234.26 1.7
  emon1.current(1, 13.000); // Current: input pin, calibration. / 111.1
  //-----
}

void loop()
{
  modbus_update();

  //-----
  //emon1.calcVI(20,2000); // Calculate all. No.of half wavelengths (crossings), time-out
  emon1.calcVI(20, 500); // Calculate all. No.of half wavelengths (crossings), time-out .....
  mudei pra 500, valor antigo era 100
  emon1.serialprint(); // Print out all variables (realpower, apparent power, Vrms, Irms, power
  factor)

  float realPower = emon1.realPower; //extract Real Power into variable
  float apparentPower = emon1.apparentPower; //extract Apparent Power into variable
  float powerFACTOR = emon1.powerFactor; //extract Power Factor into Variable
  float supplyVoltage = emon1.Vrms; //extract Vrms into Variable
  float Irms = emon1.Irms; //extract Irms into Variable

  holdingRegs [VALOR_CORRENTE] = (int) (Irms * 100);
  holdingRegs [VALOR_TENSAO] = (int) (supplyVoltage * 100);
  holdingRegs [VALOR_PATIVA] = (int) (realPower * 1);
  holdingRegs [VALOR_PAPARENTE] = (int)(apparentPower * 1);
  holdingRegs [VALOR_FP] = (int)(powerFactor * 100);
  holdingRegs [VALOR_IDENTIFICADOR] = identificador;

```


// Condicionais para a diferenciação entre as cargas a partir dos valores dos parâmetros de regime permanente

```

if ((supplyVoltage >= 119.9 && supplyVoltage <= 123.2 ) && (Irms >= 0.295 && Irms <= 0.315 ) &&
(realPower >= 35.3 && realPower <= 38.21 ) && (apparentPower >= 34.27 && apparentPower <= 36.97
) && ( powerFActor >= 0,99 && powerFActor <= 1 ))
{
    identificador = 1;
}
else if ((supplyVoltage >= 118.7 && supplyVoltage <= 123.25 ) && (Irms >= 0.26 && Irms <= 0.28)
&& (realPower >= 18.75 && realPower <= 22.91 ) && (apparentPower >= 29.42 && apparentPower <=
35.41 ) && (powerFActor >= 0,95 && powerFActor <= 0,97 ))
{
    identificador = 2;
}
else if ((supplyVoltage >= 115.4 && supplyVoltage <= 119.6 )&&(Irms >= 0.30 && Irms <= 0.32
)&&(realPower >= 33.76 && realPower <= 36.41 ) && (apparentPower >= 35.07 && apparentPower <=
37.97 ) && (powerFActor >= 0,94 && powerFActor <= 0,96 ))
{
    identificador = 3;
}

else if ((supplyVoltage >= 112.85 && supplyVoltage <= 119.05 )&&(Irms >= 5.67 && Irms <= 6.01
)&&(realPower >= 644.49 && realPower <= 719.93 ) && (apparentPower >= 645.05 && apparentPower
<= 720.69 ) && (powerFActor >= 0,99 && powerFActor <= 1.0 ))
{
    identificador = 4;
}
else if ((supplyVoltage >= 118.18 && supplyVoltage <= 122.68 )&&(Irms >= 0.92 && Irms <= 0,94
)&&(realPower >=62.18 && realPower <= 66.87 ) && (apparentPower >= 108.71 && apparentPower <=
116.83 ) && (powerFActor >= 0,95 && powerFActor <= 0,97 ))
{
    identificador = 5;
}

else if ((supplyVoltage >= 100 && supplyVoltage <= 127 )&&((Irms >= 10.43 && Irms <= 10.81))|(Irms
>= 0.38 && Irms <= 0.41))|(Irms >= 5.32 && Irms <= 6.12))&&((realPower >= 1100.47 && realPower <=
1174.86))|(realPower >= 34.38 && realPower <= 37.29))|(realPower >= 100.88 && realPower <=
174.07)) && ((apparentPower >= 1145.40 && apparentPower <= 1223.80))| (apparentPower >= 45.46

```

```

&& apparentPower <= 49.97)||((apparentPower >= 624.18 && apparentPower <= 748.91 )) &&
((powerFactor >= 0.75 && powerFactor <= 0.77 )||(powerFactor >= 0.15 && powerFactor <= 0.23
)||((powerFactor >= 0,94 && powerFactor <= 0,96 )))
{
  identificador = 6;
}
else if ((supplyVoltage >= 117.69 && supplyVoltage <= 122.19 )&&(Irms >= 0.52 && Irms <= 0.54
)&&(realPower >= 56.03 && realPower <= 60.34 ) && (apparentPower >= 62.17 && apparentPower <=
67.03 ) && (powerFactor >= 0.8 && powerFactor <= 0.93 ))
{
  identificador = 7;
}
else identificador = 0;

}
//-----

```

APÊNDICE B – CÓDIGO USADO PARA O CÁLCULO DOS PARÂMETROS

Código OpenSource da biblioteca EmonLib para o cálculo em tempo real das grandezas elétricas envolvidas no projeto.

```

//#include "WProgram.h" un-comment for use on older versions of Arduino IDE
#include "EmonLib.h"

#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

//-----
// Sets the pins to be used for voltage and current sensors
//-----
void EnergyMonitor::voltage(unsigned int _inPinV, double _VCAL, double _PHASECAL)
{

```

```

inPinV = _inPinV;
VCAL = _VCAL;
PHASECAL = _PHASECAL;
offsetV = ADC_COUNTS>>1; //512
}

void EnergyMonitor::current(unsigned int _inPinI, double _ICAL)
{
inPinI = _inPinI;
ICAL = _ICAL;
offsetI = ADC_COUNTS>>1; //512
}

//-----
// Sets the pins to be used for voltage and current sensors based on emontx pin map
//-----
void EnergyMonitor::voltageTX(double _VCAL, double _PHASECAL)
{
inPinV = 2;
VCAL = _VCAL;
PHASECAL = _PHASECAL;
offsetV = ADC_COUNTS>>1;
}

void EnergyMonitor::currentTX(unsigned int _channel, double _ICAL)
{
if (_channel == 1) inPinI = 3;
if (_channel == 2) inPinI = 0;
if (_channel == 3) inPinI = 1;
ICAL = _ICAL;
offsetI = ADC_COUNTS>>1;
}

//-----
// emon_calc procedure
// Calculates realPower,apparentPower,powerFactor,Vrms,Irms,kWh increment
// From a sample window of the mains AC voltage and current.
// The Sample window length is defined by the number of half wavelengths or crossings we choose to
measure.
//-----

```

```

void EnergyMonitor::calcVI(unsigned int crossings, unsigned int timeout)
{
    #if defined emonTxV3
        int SupplyVoltage=3300;
    #else
        int SupplyVoltage = readVcc();
    #endif

    unsigned int crossCount = 0;           //Used to measure number of times threshold is crossed.
    unsigned int numberOfSamples = 0;      //This is now incremented

    //-----
    // 1) Waits for the waveform to be close to 'zero' (mid-scale adc) part in sin curve.
    //-----

    boolean st=false;                    //an indicator to exit the while loop

    unsigned long start = millis(); //millis()-start makes sure it doesnt get stuck in the loop if there is an
    error.

    while(st==false)                    //the while loop...
    {
        startV = analogRead(inPinV);     //using the voltage waveform
        if ((startV < (ADC_COUNTS*0.55)) && (startV > (ADC_COUNTS*0.45))) st=true; //check its within
    range
        if ((millis()-start)>timeout) st = true;
    }

    //-----
    // 2) Main measurement loop
    //-----

    start = millis();

    while ((crossCount < crossings) && ((millis()-start)<timeout))
    {
        numberOfSamples++;                //Count number of times looped.
        lastFilteredV = filteredV;        //Used for delay/phase compensation

        //-----
        // A) Read in raw voltage and current samples
        //-----

```

```

sampleV = analogRead(inPinV);          //Read in raw voltage signal
sampleI = analogRead(inPinI);          //Read in raw current signal

//-----
// B) Apply digital low pass filters to extract the 2.5 V or 1.65 V dc offset,
// then subtract this - signal is now centred on 0 counts.
//-----
offsetV = offsetV + ((sampleV-offsetV)/1024);
    filteredV = sampleV - offsetV;
offsetI = offsetI + ((sampleI-offsetI)/1024);
    filteredI = sampleI - offsetI;

//-----
// C) Root-mean-square method voltage
//-----
sqV= filteredV * filteredV;           //1) square voltage values
sumV += sqV;                          //2) sum

//-----
// D) Root-mean-square method current
//-----
sql = filteredI * filteredI;          //1) square current values
sumI += sql;                          //2) sum

//-----
// E) Phase calibration
//-----
phaseShiftedV = lastFilteredV + PHASECAL * (filteredV - lastFilteredV);

//-----
// F) Instantaneous power calc
//-----
instP = phaseShiftedV * filteredI;    //Instantaneous Power
sumP +=instP;                          //Sum

//-----
// G) Find the number of times the voltage has crossed the initial voltage
// - every 2 crosses we will have sampled 1 wavelength
// - so this method allows us to sample an integer number of half wavelengths which increases
accuracy

```

```

//-----
lastVCross = checkVCross;
if (sampleV > startV) checkVCross = true;
    else checkVCross = false;
if (numberOfSamples==1) lastVCross = checkVCross;

if (lastVCross != checkVCross) crossCount++;
}

//-----
// 3) Post loop calculations
//-----
//Calculation of the root of the mean of the voltage and current squared (rms)
//Calibration coefficients applied.

double V_RATIO = VCAL *((SupplyVoltage/1000.0) / (ADC_COUNTS));
Vrms = V_RATIO * sqrt(sumV / numberOfSamples);

double I_RATIO = ICAL *((SupplyVoltage/1000.0) / (ADC_COUNTS));
Irms = I_RATIO * sqrt(sumI / numberOfSamples);

//Calculation power values
realPower = V_RATIO * I_RATIO * sumP / numberOfSamples;
apparentPower = Vrms * Irms + 1;
powerFactor=realPower / apparentPower;
// if (powerFactor > 1 );
// { powerFactor = 1; // garantir fator de potência menor que o unitário
// }

//Reset accumulators
sumV = 0;
sumI = 0;
sumP = 0;
//-----
}

//-----
double EnergyMonitor::calcIrms(unsigned int Number_of_Samples)
{

```

```

#if defined emonTxV3
    int SupplyVoltage=3300;
#else
    int SupplyVoltage = readVcc();
#endif

for (unsigned int n = 0; n < Number_of_Samples; n++)
{
    sampleI = analogRead(inPinI);

    // Digital low pass filter extracts the 2.5 V or 1.65 V dc offset,
    // then subtract this - signal is now centered on 0 counts.
    offsetI = (offsetI + (sampleI-offsetI)/1024);
    filteredI = sampleI - offsetI;

    // Root-mean-square method current
    // 1) square current values
    sqI = filteredI * filteredI;
    // 2) sum
    sumI += sqI;
}

double I_RATIO = ICAL * ((SupplyVoltage/1000.0) / (ADC_COUNTS));
Irms = I_RATIO * sqrt(sumI / Number_of_Samples);

//Reset accumulators
sumI = 0;
//-----

return Irms;
}

void EnergyMonitor::serialprint()
{
    Serial.print("Preal: ");
    Serial.print(realPower);
    Serial.print(' ');
    Serial.print("Papar: ");
    Serial.print(apparentPower);
}

```

```

Serial.print(' ');
Serial.print("Vrms: ");
Serial.print(Vrms);
Serial.print(' ');
Serial.print("Irms: ");
Serial.print(Irms);
Serial.print(' ');
Serial.print("FP: ");
Serial.print(powerFactor);
Serial.println(' ');
delay(100);
}

//thanks to http://hacking.majenko.co.uk/making-accurate-adc-readings-on-arduino
//and Jérôme who alerted us to http://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/

long EnergyMonitor::readVcc() {
    long result;

    //not used on emonTx V3 - as Vcc is always 3.3V - eliminates bandgap error and need for calibration
    http://harizanov.com/2013/09/thoughts-on-avr-adc-accuracy/

    #if defined(__AVR_ATmega168__) || defined(__AVR_ATmega328__) || defined
    (__AVR_ATmega328P__)
        ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
    #elif defined(__AVR_ATmega32U4__) || defined(__AVR_ATmega1280__) ||
    defined(__AVR_ATmega2560__) || defined(__AVR_AT90USB1286__)
        ADMUX = _BV(REFS0) | _BV(MUX4) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
        ADCSRB &= ~_BV(MUX5); // Without this the function always returns -1 on the ATmega2560
        http://openenergymonitor.org/emon/node/2253#comment-11432
    #elif defined (__AVR_ATtiny24__) || defined(__AVR_ATtiny44__) || defined(__AVR_ATtiny84__)
        ADMUX = _BV(MUX5) | _BV(MUX0);
    #elif defined (__AVR_ATtiny25__) || defined(__AVR_ATtiny45__) || defined(__AVR_ATtiny85__)
        ADMUX = _BV(MUX3) | _BV(MUX2);

    #endif

    #if defined(__AVR__)
        delay(2); // Wait for Vref to settle
    #endif
}

```



```

ADCSRA |= _BV(ADSC);           // Convert
while (bit_is_set(ADCSRA,ADSC));
result = ADCL;
result |= ADCH<<8;
result = READVCC_CALIBRATION_CONST / result; //1100mV*1024 ADC steps
http://openenergymonitor.org/emon/node/1186
return result;
#elif defined(__arm__)
return (3300);                 //Arduino Due
#else
return (3300);                 //Guess that other un-supported architectures will be running a
3.3V!
#endif
}

```

APÊNDICE C – CÓDIGO USADO PARA A COMUNICAÇÃO MODBUS ENTRE ARDUINO E SCADABR

```

#include "SimpleModbusSlave.h"
#include "HardwareSerial.h"

// SimpleModbusSlaveV10

#define BUFFER_SIZE 64

// frame[] is used to receive and transmit packages.

unsigned char frame[BUFFER_SIZE];
unsigned int holdingRegsSize; // size of the register array
unsigned int* regs; // user array address
unsigned char broadcastFlag;
unsigned char slaveID;
unsigned char function;
unsigned char TxEnablePin;
unsigned int errorCount;
unsigned int T1_5; // inter character time out
unsigned int T3_5; // frame delay
HardwareSerial* ModbusPort;

// function definitions
void exceptionResponse(unsigned char exception);

```

```

unsigned int calculateCRC(unsigned char bufferSize);
void sendPacket(unsigned char bufferSize);

void modbus_configure(HardwareSerial *SerialPort,
                    long baud,
                    unsigned char
byteFormat,
                    unsigned char
_slaveID,
                    unsigned char _TxEnablePin,
                    unsigned int
_holdingRegsSize,
                    unsigned int* _regs)
{
    ModbusPort = SerialPort;
    modbus_update_comms(baud, byteFormat, _slaveID);
    holdingRegsSize = _holdingRegsSize;
    regs = _regs;
    TxEnablePin = _TxEnablePin;
    pinMode(TxEnablePin, OUTPUT);
    digitalWrite(TxEnablePin, LOW);
    errorCount = 0; // initialize errorCount
}

void modbus_update_comms(long baud, unsigned char byteFormat, unsigned char _slaveID)
{
    (*ModbusPort).begin(baud, byteFormat);
    slaveID = _slaveID;

    // Modbus states that a baud rate higher than 19200 must use a fixed 750 us
    // for inter character time out and 1.75 ms for a frame delay for baud rates
    // below 19200 the timing is more critical and has to be calculated.
    // E.g. 9600 baud in a 10 bit packet is 960 characters per second
    // In milliseconds this will be 960characters per 1000ms. So for 1 character
    // 1000ms/960characters is 1.04167ms per character and finally modbus states
    // an inter-character must be 1.5T or 1.5 times longer than a character. Thus
    // 1.5T = 1.04167ms * 1.5 = 1.5625ms. A frame delay is 3.5T.

    if (baud > 19200)
    {

```

```

        T1_5 = 750;
        T3_5 = 1750;
    }
    else
    {
        T1_5 = 15000000/baud; // 1T * 1.5 = T1.5
        T3_5 = 35000000/baud; // 1T * 3.5 = T3.5
    }
}

unsigned int modbus_update()
{
    if ((*ModbusPort).available())
    {
        unsigned char buffer = 0;
        unsigned char overflow = 0;

        while ((*ModbusPort).available())
        {
            // If more bytes is received than the BUFFER_SIZE the overflow flag will be set and
            the
            // serial buffer will be red untill all the data is cleared from the receive buffer.
            if (overflow)
                (*ModbusPort).read();
            else
            {
                if (buffer == BUFFER_SIZE)
                    overflow = 1;
                frame[buffer] = (*ModbusPort).read();
                buffer++;
            }
            delayMicroseconds(T1_5); // inter character time out
        }

        // If an overflow occurred increment the errorCount
        // variable and return to the main sketch without
        // responding to the request i.e. force a timeout
        if (overflow)
            return errorCount++;
    }
}

```

```

// The minimum request packet is 8 bytes for function 3 & 16
if (buffer > 7)
{
    unsigned char id = frame[0];

    broadcastFlag = 0;

    if (id == 0)
        broadcastFlag = 1;

    if (id == slaveID || broadcastFlag) // if the recieved ID matches the slaveID or broadcasting id (0),
continue
    {
        unsigned int crc = ((frame[buffer - 2] << 8) | frame[buffer - 1]); // combine the crc Low & High bytes
        if (calculateCRC(buffer - 2) == crc) // if the calculated crc matches the recieved crc continue
        {
            function = frame[1];
            unsigned int startingAddress = ((frame[2] << 8) | frame[3]); // combine
the starting address bytes
            unsigned int no_of_registers = ((frame[4] << 8) | frame[5]); // combine
the number of register bytes
            unsigned int maxData = startingAddress + no_of_registers;
            unsigned char index;
            unsigned char address;
            unsigned int crc16;

            // broadcasting is not supported for function 3
            if (!broadcastFlag && (function == 3))
            {
                if (startingAddress < holdingRegsSize) // check exception 2
                ILLEGAL DATA ADDRESS
                {
                    if (maxData <= holdingRegsSize) // check exception 3
                    ILLEGAL DATA VALUE
                    {
                        unsigned char noOfBytes = no_of_registers *
2;
                        // ID, function, noOfBytes, (dataLo + dataHi)*number of registers,
                        // crcLo, crcHi

```

```

noOfBytes;
maxData; index++)
the register into 2 bytes
2);
// split crc into 2 bytes
0xFF;
ILLEGAL DATA VALUE
ADDRESS
exception 2 ILLEGAL DATA ADDRESS

unsigned char responseFrameSize = 5 +
frame[0] = slaveID;
frame[1] = function;
frame[2] = noOfBytes;
address = 3; // PDU starts at the 4th byte
unsigned int temp;

for (index = startingAddress; index <
{
temp = regs[index];
frame[address] = temp >> 8; // split

address++;
frame[address] = temp & 0xFF;
address++;
}

crc16 = calculateCRC(responseFrameSize -
frame[responseFrameSize - 2] = crc16 >> 8;
frame[responseFrameSize - 1] = crc16 &

sendPacket(responseFrameSize);
}
else
exceptionResponse(3); // exception 3
}
else
exceptionResponse(2); // exception 2 ILLEGAL DATA
}
else if (function == 6)
{
if (startingAddress < holdingRegsSize) // check
{

```

```

regs[startingAddress] = ((frame[4] << 8) |
frame[5]); // the 4th and 5th elements in frame is the 16 bit data value

// only the first 6 bytes are used for CRC
calculation
crc16 = calculateCRC(6);
frame[6] = crc16 >> 8; // split crc into 2 bytes
frame[7] = crc16 & 0xFF;

// a function 16 response is an echo of the first
6 bytes from
// the request + 2 crc bytes

broadcast message
if (!broadcastFlag) // don't respond if it's a

sendPacket(8);
}
else
exceptionResponse(2); // exception 2 ILLEGAL

DATA ADDRESS
}
else if (function == 16)
{
// Check if the recieved number of bytes matches the
calculated bytes
// minus the request bytes.
// id + function + (2 * address bytes) + (2 * no of register bytes)
+
// byte count + (2 * CRC bytes) = 9 bytes
if (frame[6] == (buffer - 9))
{
if (startingAddress < holdingRegsSize) // check
exception 2 ILLEGAL DATA ADDRESS
{
if (maxData <= holdingRegsSize) // check
exception 3 ILLEGAL DATA VALUE
{
address = 7; // start at the 8th byte in
the frame

```

```

maxData; index++)
    ((frame[address] << 8) | frame[address + 1]);

CRC calculation
    2 bytes

the first 6 bytes from
    // the request + 2 crc bytes

it's a broadcast message

ILLEGAL DATA VALUE
    ILLEGAL DATA ADDRESS

}
else
    errorCount++; // corrupted packet

}
else
    exceptionResponse(1); // exception 1 ILLEGAL FUNCTION

} // incorrect id
}

```

```

for (index = startingAddress; index <
{
    regs[index] =
    address += 2;
}

// only the first 6 bytes are used for

crc16 = calculateCRC(6);
frame[6] = crc16 >> 8; // split crc into

frame[7] = crc16 & 0xFF;

// a function 16 response is an echo of

if (!broadcastFlag) // don't respond if

    sendPacket(8);
}
else
    exceptionResponse(3); // exception 3

}
else
    exceptionResponse(2); // exception 2

}
else // checksum failed
    errorCount++;
} // incorrect id
}

```

```

        else if (buffer > 0 && buffer < 8)
            errorCount++; // corrupted packet
    }
    return errorCount;
}

void exceptionResponse(unsigned char exception)
{
    // each call to exceptionResponse() will increment the errorCount
    errorCount++;
    if (!broadcastFlag) // don't respond if its a broadcast message
    {
        frame[0] = slaveID;
        frame[1] = (function | 0x80); // set MSB bit high, informs the master of an exception
        frame[2] = exception;
        unsigned int crc16 = calculateCRC(3); // ID, function|0x80, exception code
        frame[3] = crc16 >> 8;
        frame[4] = crc16 & 0xFF;
        // exception response is always 5 bytes
        // ID, function + 0x80, exception code, 2 bytes crc
        sendPacket(5);
    }
}

unsigned int calculateCRC(unsigned char bufferSize)
{
    unsigned int temp, temp2, flag;
    temp = 0xFFFF;
    for (unsigned char i = 0; i < bufferSize; i++)
    {
        temp = temp ^ frame[i];
        for (unsigned char j = 1; j <= 8; j++)
        {
            flag = temp & 0x0001;
            temp >>= 1;
            if (flag)
                temp ^= 0xA001;
        }
    }
    // Reverse byte order.

```



```
temp2 = temp >> 8;
temp = (temp << 8) | temp2;
temp &= 0xFFFF;
// the returned value is already swapped
// crcLo byte is first & crcHi byte is last
return temp;
}

void sendPacket(unsigned char bufferSize)
{
digitalWrite(TxEnablePin, HIGH);

for (unsigned char i = 0; i < bufferSize; i++)
    (*ModbusPort).write(frame[i]);

    (*ModbusPort).flush();

    // allow a frame delay to indicate end of transmission
    delayMicroseconds(T3_5);

digitalWrite(TxEnablePin, LOW);
}
```