

**CENTRO UNIVERSITÁRIO ASSIS GURGACZ
THIAGO FERNANDES DA COSTA**

**PROJETO E DESENVOLVIMENTO DE UM MEDIDOR DE GRANDEZAS
ELÉTRICAS PARA ANÁLISE DA QUALIDADE DA ENERGIA ELÉTRICA EM
RESIDÊNCIAS**

**CASCADEL
2018**

**CENTRO UNIVERSITÁRIO ASSIS GURGACZ
THIAGO FERNANDES DA COSTA**

**PROJETO E DESENVOLVIMENTO DE UM MEDIDOR DE GRANDEZAS
ELÉTRICAS PARA ANÁLISE DA QUALIDADE DA ENERGIA ELÉTRICA EM
RESIDÊNCIAS**

Projeto de pesquisa apresentado ao Curso de Graduação em Engenharia Elétrica do Centro Universitário Assis Gurgacz para elaboração do Trabalho de Conclusão de Curso II.

Professor Orientador: Ederson Zanchet

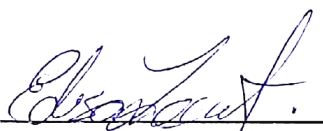
**CASCABEL
2018**

**CENTRO UNIVERSITÁRIO ASSIS GURGACZ
THIAGO FERNANDES DA COSTA**

**PROJETO E DESENVOLVIMENTO DE UM MEDIDOR DE GRANDEZAS
ELÉTRICAS, PARA ANÁLISE DA QUALIDADE DA ENERGIA ELÉTRICA EM
RESIDÊNCIAS.**

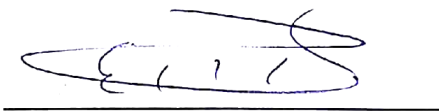
Trabalho apresentado no Curso de Engenharia Elétrica, Centro Universitário Assis Gurgacz, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica, sob a orientação do Professor Ederson Zanchet.

BANCA EXAMINADORA



Ederson Zanchet

Centro Universitário Assis Gurgacz



Ewerson Luiz Poisk

Centro Universitário Assis Gurgacz



Mario Elias Carvalho do Nascimento

Centro Universitário Assis Gurgacz

Cascavel/PR., 08 de dezembro de 2018.

AGRADECIMENTOS

A Deus, pelo dom da vida e por me possibilitar realizar o curso de Engenharia Elétrica, estando sempre em minha companhia nas horas de alegria e por me guiar nos momentos mais difíceis.

Ao meu filho Matheus e minha esposa Morgana, por serem a minha motivação diária e por me acompanharem nesta longa caminhada e por iluminarem a minha vida, amo vocês.

A minha mãe Marli, por me cuidar, educar e dar sempre o seu melhor para que seus filhos possam ter o melhor. Obrigado por tudo.

Ao professor Ederson, pela paciência e pelos ensinamentos, obrigado pela parceria neste período.

Nenhum homem é tão sábio que não tenha algo para aprender,
e nem tão tolo que não tenha algo para ensinar.
Blaise Pascal

RESUMO

Com a crescente necessidade de energia elétrica para utilização de aparelhos eletrônicos cada vez mais tecnológicos e sensíveis, é de suma importância o fornecimento de energia elétrica de qualidade por parte da concessionária de energia. Existem indicadores monitorados pela ANEEL, que garantem que a energia esteja dentro de padrões pré-estabelecidos, como o DIC, FIC, DMIC e DICR. Estes indicadores citam poucas informações para os consumidores, o que vai contra a tendência de aumento pela aquisição de informações cada vez mais detalhadas por parte da população em geral, diante disto, surge a oportunidade de implementação de um sistema que busca monitorar a energia elétrica recebida e apresentar indicadores mais eficientes, sendo este, o assunto desta análise. O projeto desenvolvido possibilitará conferência dos dados fornecidos pela concessionária, levando ao consumidor maior confiabilidade em relação à veracidade dos indicadores apresentados e flexibilidade quanto as demais informações que poderão ser visualizadas no sistema supervisorio. Para isso, será utilizado um sistema de processamento de dados baseado em um microcontrolador na plataforma *Arduino*® e visualização dos dados via sistema supervisorio na linguagem *SCADA*®. Através da medição de tensão e corrente, determinar por meio de cálculos a potência ativa, reativa, aparente, fator de potência e consumo em kWh. Após a confecção do projeto, será realizada a coleta dos dados medidos pelo protótipo e, também, pelo equipamento de referência, sendo que, ambos realizarão a medição em um mesmo ponto. Por meio deste, determinar a precisão do protótipo, analisando os dados coletados, comparando-os para determinar o erro de medição. Ao fim, verificar também a viabilidade do projeto.

Palavras Chave: Microcontrolador. Medidor de Energia Elétrica. Supervisorio *SCADA*®. Qualidade de Energia Elétrica.

ABSTRACT

With the increasing need for electric energy to use increasingly technological and sensitive electronic devices, it is of paramount importance to provide quality electric power from the utility. There are indicators monitored by ANEEL, which ensure that energy is within pre-established standards, such as DIC, FIC, DMIC and DICR. These indicators cite little information for consumers, which goes against the trend of increasing the acquisition of increasingly detailed information by the population in general, given the opportunity to implement a system that seeks to monitor the electricity received and to present more efficient indicators, this being the subject of this analysis. The developed project will allow the verification of the data provided by the concessionaire, leading to the consumer greater reliability in relation to the veracity of the presented indicators and flexibility as the other information that can be visualized in the supervisory system. For this, a data-processing system based on a microcontroller in the Arduino® platform will be used and data visualization via supervisory system in the SCADA® language. Through the measurement of voltage and current, determine by calculation the active, reactive, apparent power, power factor and consumption in kWh. After the design of the project, the data measured by the prototype and the reference equipment will be collected, both of which will carry out the measurement at the same measurement point. Through this, determine the accuracy of the prototype by analyzing the data collected, comparing them to determine the measurement error. Finally, check also the feasibility of the project.

Keywords: Microcontroller. Electric Power Meter. SCADA® Supervisory. Quality of Electric Power.

LISTA DE FIGURAS

Figura 2.1 - Triângulo das Potências.....	19
Figura 2.2 - <i>Arduino</i> ® UNO Rev3.....	21
Figura 2.3 - <i>Ethernet Shield</i> W5100	22
Figura 2.4 - Transdutor de Corrente SCT-013-050	23
Figura 2.5 - Transdutor de Tensão ZMPT101B.....	24
Figura 3.1 - Fluxograma de Atividades Confecção Protótipo	31
Figura 3.2 - Integração do <i>Hardware</i> do Protótipo	34
Figura 3.3 - Circuito de ligação do transdutor SCT-013-050	35
Figura 3.4 - Circuito de ligação do transdutor ZMPT101B	36
Figura 3.5 - Alimentação do Microcontrolador <i>Arduino</i> ® Uno Rev3.....	37
Figura 3.6 - Conexão <i>Arduino</i> ® <i>Ethernet Shield</i> W5100.....	39
Figura 3.7 - Protótipo Medidor de Energia Elétrica.....	39
Figura 3.8 - Processamento e Comunicação do Protótipo	40
Figura 3.9 - Alimentação e Coleta de Tensão do Protótipo.....	41
Figura 3.10 - Conexões de Entrada do Protótipo	42
Figura 3. 11 - Ligação dos Componentes de Entrada de Tensão e Corrente	42
Figura 3.12 - Conexão de Programação e Comunicação Protótipo	43
Figura 3.13 - Fonte de Dados.....	53
Figura 3.14 - Programação da Fonte de Dados	54
Figura 3.15 - Programação do Ponto de Dados	55
Figura 3.16 - Acesso ao Supervisório	56
Figura 4.1 - Medição Anterior a Calibração.....	57
Figura 4.2 - Medição Posterior a Calibração	58
Figura 4.3 - Tela Inicial de Medição Supervisório.....	64
Figura 4.4 - Supervisório Alarme de Falta de Energia.....	65
Figura 4.5 - Supervisório Alarme de Nível Precário de Tensão Inferior	66
Figura 4.6 - Supervisório Alarme de Nível Precário de Tensão Superior	66
Figura 4.7 - Supervisório Alarme de Nível Crítico de Tensão Inferior	67
Figura 4.8 - Supervisório Alarme de Nível Crítico de Tensão Superior	67

LISTA DE TABELAS

Tabela 3.1 - Custos de Implementação do Protótipo	33
Tabela 4.1 - Resultados de Medição Tensão e Corrente Carga Resistiva.....	59
Tabela 4.2 - Resultados de Medição Tensão e Corrente Carga <i>RLC</i>	59
Tabela 4.3 - Resultados de Potência Ativa (W), Aparente (VA) e Reativa (VAr) Carga Resistiva.....	60
Tabela 4.4 - Resultados de Potência Ativa (W), Aparente (VA) e Reativa (VAr) Carga <i>RLC</i>	61
Tabela 4.5 - Resultados de Fator de Potência Carga Resistiva e <i>RLC</i>	62
Tabela 4.6 - Resultados de Medição de Consumo Carga Resistiva e <i>RLC</i>	63

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
BIT	Menor Parcela de Informação Processada por um Computador
BYTE	Conjunto Equivalente a Oito <i>Bits</i>
DEC	Duração Equivalente de Interrupção por Unidade Consumidora
DIC	Duração de Interrupção Individual por Unidade Consumidora
DICRI	Duração da Interrupção Individual Ocorrida em Dia Crítico por Unidade Consumidora ou Ponto de Conexão
DIMMER	Dispositivo para Controle Gradual de Luminosidade
DMIC	Duração Máxima de Interrupção por Unidade Consumidora
EEPROM	Memória Somente Leitura Programável e Apagável Eletricamente
EMONLIB.H	Biblioteca de Medição
FEC	Frequência Equivalente de Interrupção por Unidade Consumidora
FIC	Frequência de Interrupção Individual por Unidade Consumidora
Fm	Frequência Máxima
FP	Fator de Potência
Fs	Razão da Amostragem
GATEWAY	Ponto de Ligação
Hz	Hertz (frequência)
I	Corrente Eficaz
li	Corrente Instantânea
Im	Corrente de Pico
IP	Protocolo da Internet
kWh	Quilowatt-hora
LED	<i>Ligh Emitting Diode</i> (Diodo Emissor de Luz)
MAC	Controle de Acesso à Mídia
Modbus	Protocolo de Comunicação Serial da <i>Modicon</i> ®
NOTEBOOK	Computador Portátil
Ø	Fase
OFFSET	Local de Posição de uma Informação
Øi	Ângulo de Fase da Corrente
Open-Source	Código Aberto
Øv	Ângulo de Fase da Tensão

P	Potência Ativa
POWER BANK	Banco de Energia
PRODIST	Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional
PVC	Policloreto de Vinila
PWM	<i>Pulse-Width Modulation</i> (Modulação Por Largura de Pulso)
Q	Potência Reativa
RLC	Circuito Resistor, Indutor e Capacitor
RMS	<i>Root Medium Square</i> (Valor Médio Quadrático)
S	Potência Aparente
SCADA®	<i>Supervisory Control and Data Acquisition</i> (Controle Supervisório e Aquisição de Dados)
SHIELD	Escudo
TCP	<i>Transmission Control Protocol</i> (Protocolo de Controle de Transmissão)
TL	Tensão de Leitura
UDP	<i>User Datagram Protocol</i> (Protocolo de Datagramas de Utilizador)
USB	Porta Universal
V	Tensão Eficaz
VA	Volt-ampère
VAr	Volt-ampère reativo
Vi	Tensão Instantânea
Vm	Tensão de Pico
VOID LOOP ()	Função de Repetição do Programa
VOID SETUP ()	Função de Início/Configuração do Programa
Web	Rede
ω	Frequência angular em radianos/segundo

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	MEDIÇÃO DE GRANDEZAS ELÉTRICAS	16
2.1.1	Tensão e Corrente Elétrica	16
2.1.2	Potência Ativa	17
2.1.3	Potência Aparente	18
2.1.4	Potência Reativa	18
2.1.5	Fator de Potência	20
2.1.6	Medição de kWh	20
2.2	MICROCONTROLADOR	20
2.2.1	<i>Hardware</i>	21
2.2.1.1	<i>Arduino® UNO Rev3</i>	21
2.2.1.2	<i>Arduino® Ethernet Shield W5100</i>	22
2.2.1.3	Transdutores	22
2.2.2	<i>Software</i>	25
2.2.2.1	Plataforma <i>Open-Source</i> de Programação <i>Arduino®</i>	25
2.3	SUPERVISÓRIO SCADABR®	25
2.3.1	Estrutura	26
2.3.2	Banco de Dados	26
2.3.3	Nível de Segurança e Controle	27
2.4	FORMAS DE MEDIÇÃO E COMPARAÇÃO	27
2.4.1	Normas, Regulamentos e Critérios a Serem Adotados	27
2.4.2	Aquisição da Medição	28
3	METODOLOGIA	30
3.1	LISTA DE MATERIAIS	31
3.2	CUSTOS	32
3.3	<i>HARDWARE</i>	33
3.3.1	Medição de Corrente Elétrica	34
3.3.2	Medição de Tensão Elétrica	35
3.3.3	Microcontrolador <i>Arduino®</i> Uno Rev3	36
3.3.4	<i>Arduino® Ethernet Shield W5100</i>	38
3.4	<i>SOFTWARE</i> DO MICROCONTROLADOR	43

3.4.1	Microcontrolador <i>Arduino</i> [®] e <i>Arduino</i> [®] <i>Ethernet Shield W5100</i>	43
3.4.2	Tensão e Corrente Elétrica	45
3.4.3	Potência Ativa.....	47
3.4.4	Potência Aparente	48
3.4.5	Fator de Potência	48
3.4.6	Potência Reativa	49
3.4.7	Potência Ativa Acumulada kWh.....	49
3.4.8	Envio Grandezas para o Supervisório	51
3.5	<i>SOFTWARE DO SUPERVISÓRIO</i>	52
3.5.1	Fonte de Dados	52
3.5.2	Pontos de Dados	54
3.5.3	Segurança do Supervisório	56
4	RESULTADOS E DISCUSSÕES	57
4.1	CALIBRAÇÃO DOS TRANSDUTORES	57
4.2	MEDIÇÃO DE TENSÃO E CORRENTE.....	58
4.3	MEDIÇÃO DE POTÊNCIAS	60
4.4	MEDIÇÃO DE FATOR DE POTÊNCIA	61
4.5	MEDIÇÃO DE CONSUMO	62
4.6	TESTE DOS SUPERVISÓRIO	63
4.6.1	Simulação de Queda de Tensão	64
4.6.2	Simulação de Níveis de Tensão	65
4.7	TESTES DE COMUNICAÇÃO	67
4.8	VIABILIDADE DO PROTÓTIPO	68
4.9	PROPOSTA PARA TRABALHOS FUTUROS	69
5	CONCLUSÃO	70
	REFERÊNCIAS	73
	ANEXO A – CÓDIGO FONTE MICROCONTROLADOR <i>ARDUINO</i>[®]	75
	ANEXO B – TELA GRÁFICA DO SUPERVISÓRIO	80
	APÊNDICE A – CÓDIGO FONTE BIBLIOTECA <i>EMONLIB.H</i>	81

1 INTRODUÇÃO

Através da análise dos indicadores presentes nas faturas de energia elétrica de cada concessionária, DIC (Duração de Interrupção Individual por Unidade Consumidora), FIC (Frequência de Interrupção Individual por Unidade Consumidora), DMIC (Duração Máxima de Interrupção por Unidade Consumidora) e DICRI (Duração da interrupção individual ocorrida em dia crítico por unidade consumidora ou ponto de conexão), constata-se que apresentam limitações na medição de grandezas elétricas, registrando apenas a quantidade de vezes e horas que o consumidor permaneceu sem energia, o que dificulta o entendimento por parte dos usuários. Não é possível, também, a consulta detalhada sobre cada evento ocorrido na unidade consumidora, tão pouco a visualização de demais variáveis presentes no sistema elétrico. Desta forma, como realizar a comparação destes indicadores e/ou verificar se em outros momentos a tensão de fornecimento permaneceu fora dos limites aceitáveis sem que seja necessário recorrer à concessionária?

Sendo assim, esta pesquisa apresentou o desenvolvimento de um projeto de um medidor de grandezas elétricas a partir do microcontrolador na plataforma *Arduino*®, visando o monitoramento da qualidade da energia elétrica em residências, bem como realizar a visualização dos dados medidos, por meio de um supervisório na plataforma *SCADA*®.

Equipamentos com esta função são amplamente empregados em indústrias de várias áreas e até mesmo, em casos específicos, em residências, o que possibilita a geração e monitoração de dados sobre as grandezas elétricas medidas. Este projeto diferencia-se dos equipamentos existentes pela sua maneira construtiva e materiais empregados em sua constituição, possibilitando alterações conforme necessidade de cada aplicação, visto que será confeccionado em uma plataforma *open-source*.

Para coleta das informações da rede elétrica, utiliza-se de transdutores de tensão e corrente, a fim de adequar os valores das grandezas, possibilitando a leitura por parte do microcontrolador *Arduino*®. Já para realização da comunicação entre o microcontrolador e o supervisório, é necessário o emprego de uma *Ethernet Shield*, sendo esta responsável por realizar a conversão dos dados produzidos pelo microcontrolador *Arduino*® para o protocolo *Modbus IP*.

Ao final da pesquisa, espera-se obter um conjunto de equipamentos capaz de fornecer informações sobre as grandezas elétricas medidas ao usuário final, por meio

de um supervisor, bem como detectar os momentos em que a tensão de fornecimento está fora dos limites normais e, também, os momentos em que o consumidor permaneceu sem energia elétrica.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo apresenta a pesquisa bibliográfica necessária para embasamento teórico, destinado a prover meios para cumprimento do objetivo geral e objetivos específicos exibidos no item 2, solucionando o problema exposto no item 1.5, deste projeto de pesquisa.

Serão abordados tópicos apresentando as grandezas do estudo, bem como o *hardware* e *software* propostos para implementação do mesmo.

Ao final do capítulo, apresenta-se os parâmetros de comparação e parametrização dos padrões a serem visualizados pelo usuário fictício final do protótipo.

2.1 MEDIÇÃO DE GRANDEZAS ELÉTRICAS

A medição de grandezas elétricas é fundamental para análise de circuitos elétricos, desde uma simples manutenção residencial até o uso em laboratórios de desenvolvimento de produtos. Tais instrumentos são capazes de realizar leituras básicas como tensão, corrente e resistência e também incorporar outras funções, como a medição de potência, frequência, entre outros, sendo estes equipamentos mais completos denominados de multímetros (BALBINOT, 2011).

2.1.1 Tensão e Corrente Elétrica

Em sistemas de corrente alternada, os valores de tensão e corrente elétrica podem ser medidos de duas formas: os valores instantâneos ou os valores eficazes (RMS). Os valores instantâneos de tensão (V_i) e corrente (I_i) são obtidos através da tensão (V_m) e corrente (I_m) de pico, ambas utilizando a defasagem da grandeza (SADIKU, 2013), sendo V_i e V_m dados em Volts, I_i e I_m dados em Ampères, ωt dado em radianos por segundo e ϕ é adimensional. A Equação 2.1 expressa a tensão instantânea:

$$V_i = V_m * \cos(\omega t + \phi)$$

Equação (2.1)

Já a Equação 2.2, expressa a corrente instantânea:

$$i_i = I_m * \cos (\omega t + \varnothing)$$
Equação (2.2)

Já para obtenção dos valores eficazes ou RMS, basta apenas substituir os valores de tensão (V_m) e corrente (I_m) de pico pelos valores de tensão (V) e corrente (I) RMS (SADIKU, 2013), sendo V e V_m dados em Volts e, I e I_m dados em Ampères. A Equação 2.3 apresenta a tensão eficaz:

$$V = V_m / \sqrt{2}$$
Equação (2.3)

Para obtenção da corrente eficaz, utilizar a Equação 2.4:

$$I = I_m / \sqrt{2}$$
Equação (2.4)

Findada a determinação da tensão e corrente instantânea e eficaz, faz-se necessário verificar como são calculadas as potências do sistema: ativa, aparente e reativa (SADIKU, 2013).

2.1.2 Potência Ativa

A potência ativa (P) é dada em *watts* e é um dos principais valores dos sistemas elétricos, pois é ela que determina qual a potência necessária para realizar o trabalho efetivo de um determinado equipamento. Ela pode ser instantânea, que é determinada pelo produto da tensão (V_i) e corrente instantâneas (I_i), ou pode ser a média de um determinado período. Equipamentos com função específica para medição de potência ativa utilizam a potência média, ou seja, utilizam tensão (V) e corrente eficaz (I), para determinação dos valores a serem exibidos ao usuário e pode ser determinada pela Equação 2.5 (SADIKU, 2013), sendo P dado em *Watts*, V dado em Volts e, $\varnothing_v$ e $\varnothing_i$ são adimensionais:

$$P = \frac{1}{2} * V * I * \cos (\varnothing_v - \varnothing_i)$$
Equação (2.5)

Sendo a potência ativa determinada pela Equação 2.5, o item 2.1.3, determinará a potência total absorvida pela rede, ou a potência aparente.

2.1.3 Potência Aparente

A partir dos valores de tensão (V) e corrente (I) eficaz, também se obtém a potência aparente (S), que é medida em volt-ampère (VA). A potência aparente representa toda a quantia de potência absorvida pela carga da rede, incluindo a parcela ativa e a parcela reativa, necessária para manter os componentes reativos do sistema. Para determinação da potência aparente, é utilizada a Equação 2.6 (SADIKU, 2013), sendo S dado em Volt-Ampère, V dado em Volts e I dado Ampères:

$$S = V * I \quad \text{Equação (2.6)}$$

Realizada a determinação da potência ativa e aparente, resta então, a determinação da potência reativa absorvida pelo sistema.

2.1.4 Potência Reativa

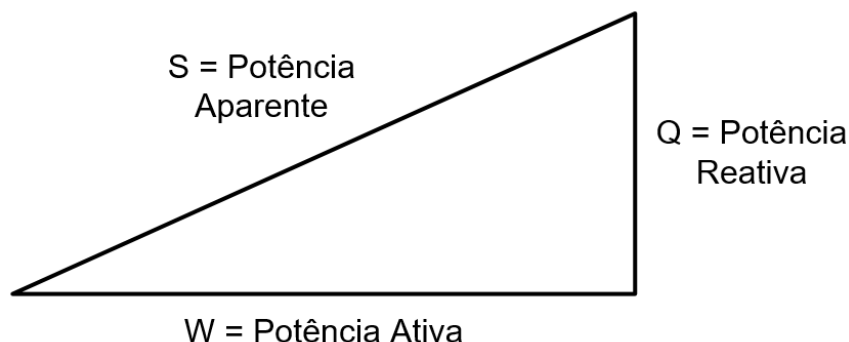
A potência reativa (Q) é a potência que expressa a energia desprendida para alimentação dos componentes reativos do sistema (indutores ou capacitores), ou seja, expressa a permuta de potência entre a fonte e a parcela reativa da carga, sendo medida em volt-ampère reativo (VAr). Para obtenção da potência reativa, utiliza-se também a tensão (V) e corrente (I) eficaz. A Equação 2.7 determina a potência reativa de um sistema (SADIKU, 2013), sendo Q dado em Volt-Ampère reativo, V dado em Volts, I dado em Ampère e, $\varnothing_v$ e $\varnothing_i$ são adimensionais:

$$Q = V * I * \cos (\varnothing_v - \varnothing_i) \quad \text{Equação (2.7)}$$

Outra forma para determinação da potência reativa (Q), é a interpretação das potências ativa, reativa e aparente utilizando sua forma geométrica equivalente, ou seja, podemos estabelecer uma relação e interpretar a potência ativa e reativa como sendo os catetos de um triângulo retângulo e a potência como a hipotenusa deste

mesmo triângulo retângulo (NILSSON, 2009). A Figura 2.1 mostra a representação gráfica deste triângulo, bem como o posicionamento das potências:

Figura 2.1 - Triângulo das Potências



Fonte: adaptado Nilsson (2009).

Segundo Lima (2012), para obtenção do valor de qualquer cateto ou da hipotenusa, para um triângulo retângulo, utiliza-se a Equação 2.8, que é referente ao Teorema de Pitágoras:

$$a^2 + b^2 = c^2 \quad \text{Equação (2.8)}$$

Utilizando-se da Equação 2.8, Nilsson (2009) afirma que, substituindo os catetos pela potência ativa (a) e reativa (b), e a hipotenusa pela potência aparente (c), mantém-se a igualdade da Equação 2.8, sendo que, conhecida duas variáveis, podemos obter qualquer uma das demais. Assim, realizando a manipulação matemática do Teorema de Pitágoras, obtém-se a potência reativa (Q), utilizando então, a potência aparente (S) e a potência ativa (W), através da Equação 2.9:

$$Q = \sqrt{S^2 - W^2} \quad \text{Equação (2.9)}$$

Após a determinação das potências ativa, aparente e reativa, permite-se obter um indicador do sistema elétrico em questão, que é o fator de potência.

2.1.5 Fator de Potência

O fator de potência é muito importante para os sistemas elétricos, pois representa a quantidade de potência aparente que está sendo consumida da rede, evidenciando a parcela de potência ativa que realmente está sendo transformada em trabalho. Quanto menor o fator de potência, menos eficiente o equipamento será quanto ao consumo de potência aparente e transformação em trabalho. Para cálculo do fator de potência, utiliza-se a Equação 2.10 (SADIKU, 2013), sendo FP adimensional, P dado em *Watts* e S dado em Volt-Ampère:

$$FP = P / S \quad \text{Equação (2.10)}$$

Segundo Johnson (1994), o fator de potência, que é adimensional, nunca poderá ser maior que 1 (um), ou seja, a potência ativa nunca poderá ser maior que a potência aparente. Quando em cargas puramente resistivas, o fator de potência será 1 (um), ou seja, não temos potência reativa dissipada no sistema, já para cargas puramente reativas, o fator de potência é zero, ou seja, não existe potência ativa dissipada no sistema.

2.1.6 Medição de kWh

Conforme Sadiku (2013), para obtenção do kWh, que é a potência média consumida durante um intervalo de tempo determinado, basta apenas obter o produto da potência média e a quantidade de horas em que determinado equipamento permaneceu em trabalho. A cobrança das concessionárias aos consumidores residenciais se dá basicamente na medição do kWh multiplicado com a tarifa vigente.

2.2 MICROCONTROLADOR

Aplicado para realização de diversas atividades, desde o simples acionamento de um *LED* até o controle de processos avançados, os microcontroladores são implementados em muitos ramos de tecnologia. Caracterizados pelo pequeno tamanho físico, são componentes eletrônicos, fabricados em um único encapsulamento, possuindo internamente todos os componentes indispensáveis para

o seu funcionamento, como: processador, memórias, entradas e saídas, conversor serial, contadores, entre outros. Sendo uma característica fundamental, a possibilidade de programação de uma lista de atividades que são realizadas seguindo a sequência lógica implementada pelo usuário, através de um *software* de programação específico de cada fabricante (SOUZA, 2003).

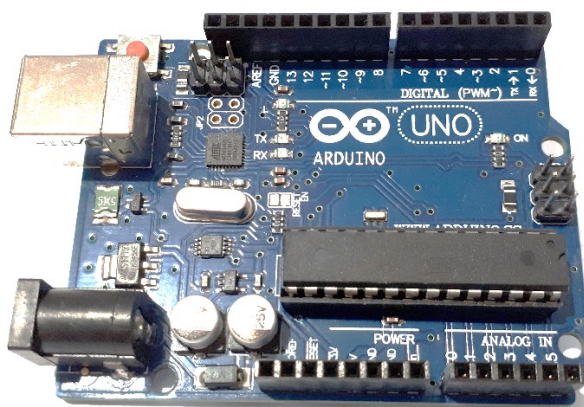
2.2.1 Hardware

De acordo com Crispim (2013), define-se por *hardware*, a parte física de um equipamento, ou seja, dispositivos eletrônicos que, quando conectados entre si, formam um determinado equipamento, como por exemplo: placa mãe, processador, entre outros, que formam um microcomputador.

2.2.1.1 Arduino® UNO Rev3

Arduino® Uno Rev3 é uma placa de processamento de dados de entradas e saídas, desenvolvida em uma plataforma *open-source* (código aberto), possibilitando ao usuário a inserção de um *software*, conforme a necessidade de cada aplicação. Fundamentada no microcontrolador *ATmega328P*, possui 14 entradas e/ou saídas digitais, sendo 6 destas disponíveis para utilização como saídas *PWM*, 6 entradas analógicas e velocidade de processamento de 16MHz. Tendo integrados a sua placa todos os componentes necessários para funcionamento do microcontrolador (MONK, 2010). Na Figura 2.2 há um exemplo da placa *Arduino®* UNO Rev3:

Figura 2.2 - *Arduino®* UNO Rev3



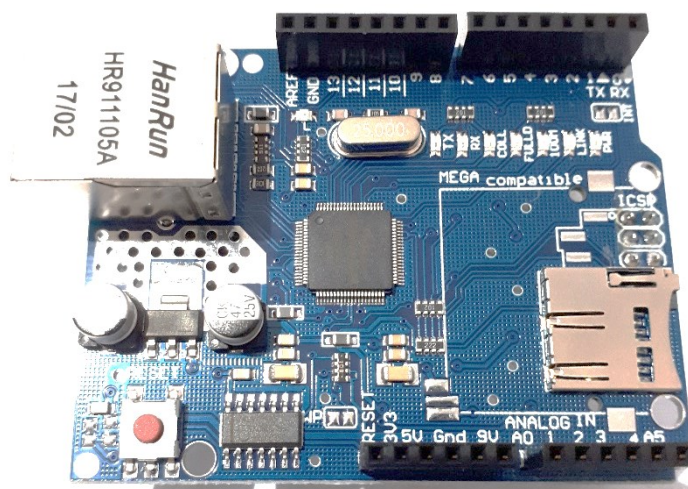
Fonte: Autor

Estas características possibilitam a utilização do *Arduino*® Uno Rev3 em vários tipos de aplicações, sendo estas, limitadas às características intrínsecas do *hardware*, apresentadas no início do item 5.2.1.1, conforme informações do fabricante.

2.2.1.2 *Arduino*® Ethernet Shield W5100

A placa *Arduino*® Ethernet Shield W5100, que é baseada no microcontrolador *WIZnet ethernet W5100*, permite que o *Arduino*® realize comunicação via rede *internet/ethernet*, nos protocolos *TCP* ou *UDP*, possui também um local para inserção de um cartão de memória, onde podem ser armazenados dados para serem utilizados pela rede. Através da rede, podem-se enviar e/ou receber informações e/ou comandos a serem interpretados pelo *Arduino*® (THOMSEN, 2014). Segue a Figura 2.3, um exemplo da placa *Ethernet Shield W5100*:

Figura 2.3 - *Ethernet Shield W5100*



Fonte: Autor

A utilização de uma *ethernet shield* permite a comunicação do *Arduino*® com *softwares* supervisórios e/ou aplicações desenvolvidas em ambiente *Web*, facilitando a aquisição e controle de dados.

2.2.1.3 Transdutores

Os transdutores são equipamentos que realizam a conversão de sinais, diferenciando-os seus tipos físicos entre a entrada e a saída, mas, mantendo a sua

correlação, ou seja, realizam a coleta de um sinal físico em uma determinada grandeza e a converte para outra grandeza, disponibilizando para utilizações diversas. Temos como exemplo: termopar, transformador de corrente, entre outros (BALBINOT, 2011).

Para possibilitar a medição de correntes e tensões elevadas com dispositivos eletrônicos é necessária realizar a conversão destes sinais, a níveis aceitáveis aos circuitos eletrônicos.

Os transdutores de corrente ou transformadores de corrente são utilizados para rebaixar correntes elétricas a níveis que possibilitem a medição, sem que o aparelho de medição possua a mesma característica de corrente que a carga na qual estão conectados. Realizam a transformação através do fenômeno de conversão eletromagnética, onde a corrente a ser medida, circula no primário gerando um fluxo magnético, sendo este, responsável pela indução de forças eletromotrizes no primário e secundário do transformador (MAMED, 2005).

O SCT-013 é um transformador de corrente não invasivo, com núcleo dividido, o que possibilita a instalação sem que seja necessário interromper o circuito a ser medido (THOMSEN, 2015). Segue Figura 2.4, exemplo do Transdutor de Corrente SCT-013-050:

Figura 2.4 - Transdutor de Corrente SCT-013-050



Fonte: Autor

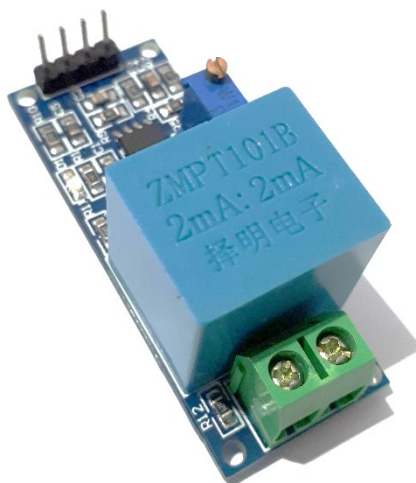
O primário do transformador acaba sendo o próprio circuito a ser medido, já o secundário é construído de ferrite. O modelo projetado para 50A possui em série com

o secundário, um resistor, disponibilizando saída de 1V, ou seja, para uma corrente de 50A no primário, a saída disponibilizará 1V (THOMSEN, 2015).

Já os transdutores de tensão ou transformadores de potencial, assim como os transdutores de corrente, possuem o mesmo objetivo, porém, neste caso, a tensão elétrica é o objetivo do rebaixamento, para que os equipamentos de medição sejam projetados para tensões menores, reduzindo assim seu isolamento e, conseqüentemente, o seu tamanho. Geralmente os transformadores de potencial possuem várias espiras no primário e um secundário dimensionado para a tensão na qual se deseja realizar a medição (MAMED, 2005).

O ZMPT101B é um transformador de potencial adaptado para placas de microprocessadores, assim como o *Arduino*®, que trabalham com tensões até 5V em sua saída (HUDSON, 2017). Segue Figura 2.5, com um exemplo do Transdutor de Tensão ZMPT101B:

Figura 2.5 - Transdutor de Tensão ZMPT101B



Fonte: Autor

Ele possui um transformador de potencial interno para redução da tensão da rede e um circuito secundário que possui um divisor de tensão, desta forma, possibilita que a senoide da tensão da rede seja medida por completo, pois ajusta o zero da senoide para 2,5V de saída, assim, os microcontroladores realizam a medição completa da tensão, visto que as entradas analógicas trabalham entre 0 e 5V, não aceitando valores negativos (HUDSON, 2017).

2.2.2 Software

Define-se por *software*, uma sequência de instruções, inseridas em algum tipo de linguagem de programação, para que sejam efetuados por uma determinada máquina, tendo a finalidade de realizar algum trabalho. Podem ser divididas em blocos de instruções, sendo denominadas aplicações. Cada aplicação é destinada a realizar uma tarefa, a qual foi previamente projetada (CRISPIM, 2013).

2.2.2.1 Plataforma *Open-Source* de Programação *Arduino*®

Para inserção das instruções na placa *Arduino*® UNO Rev3, ou qualquer placa da *Arduino*®, é utilizado um *software open-source* chamado *Arduino*® IDE, onde podem ser gravadas ou adquiridas informações na placa em questão. Sendo empregada a linguagem de programação C++, apenas adaptada com algumas modificações e particularidades (MONK, 2010).

2.3 SUPERVISÓRIO SCADABR®

Com larga utilização em indústrias, dos mais variados ramos de atividades, os sistemas supervisórios são implementados a fim de realizar o gerenciamento de processos e/ou equipamentos. A sigla *SCADA*® que dá nome ao sistema, traduz em poucas palavras o que o sistema deve oferecer ao usuário, ou seja, um sistema de controle supervisório e aquisição de dados. Do mais simples ao mais sofisticado, um sistema *SCADA*® deve fornecer ao usuário geração de gráficos, lista de alarmes, inclusão de lógicas de programação, entre outros. O sistema supervisório *ScadaBR*® também é uma plataforma *open-source* e com licença gratuita para utilização e distribuição, permitindo ainda customizações de *software* conforme necessidade de cada aplicação. Baseado na plataforma *Java*, após a inicialização do *software*, o sistema pode ser acessado através de um navegador de internet. De maneira simples e objetiva, o usuário poderá realizar a visualização e configuração de uma série de recursos para implementação da aplicação (GOMES, 2014).

2.3.1 Estrutura

Com o objetivo de possibilitar a aquisição de dados externos, o *ScadaBR*[®] possui um menu de configuração da Fonte de Dados e os Pontos de Dados, sendo estes utilizados para configuração do ambiente de leitura e do tipo de leitura dos dados, por exemplo, a fonte de dados indicará que os sensores estarão comunicando em protocolo *Modbus*, já o ponto de dados ficará responsável por dividir os tipos de leitura dentro de uma fonte de dados, ou seja, dentro de uma fonte de dados poderão estar alocados vários pontos de dados, como um sensor de temperatura, outro de umidade, entre outros (GOMES, 2014).

O monitoramento dos dados é realizado de duas formas: criando uma lista de observação, onde os dados serão visualizados em forma de lista, contendo suas informações e até mesmo gráficos em tempo real; e através de uma representação gráfica, onde os dados podem ser personalizados sobre uma imagem adicionada ao sistema, como por exemplo, uma imagem de uma máquina indicando o local físico de um sensor de temperatura (GOMES, 2014).

Intrínseco ao sistema ou personalizado pelo usuário, uma lista de alarmes e/ou eventos acompanha o *software*, possibilitando ao usuário realizar o gerenciamento de problemas referentes à comunicação com os pontos de dados, monitorar valores mínimos e máximos, entre outros. Atrelado a esta lista, podem ser definidos alarmes sonoros aos mais diversos tipos de níveis de alarmes, auxiliando a compreensão da criticidade do evento (GOMES, 2014).

2.3.2 Banco de Dados

Outra grande necessidade de usuários de sistemas supervisórios é a geração de relatórios, a fim de realizar inúmeras análises referente ao processo ou até mesmo para resolução de problemas ocorridos no decorrer do processo. Assim, o *ScadaBR*[®] permite a geração de relatórios dos inúmeros pontos de dados que podem ser inseridos pelo usuário. Estes dados permanecem gravados em seu banco de dados e podem ser recuperados pelo usuário através da interface do sistema. Além dos relatórios gerados manualmente, podem ser agendados relatórios com periodicidades definidas. Para compartilhamento dos relatórios, pode ser configurada uma lista de e-

mails, onde estes serão enviados após sua geração, independentemente da forma que foi gerado, manual ou automaticamente (GOMES, 2014).

2.3.3 Nível de Segurança e Controle

A segurança na operação dos processos é fundamental para a continuidade dos serviços e para preservação de vidas, sendo imprescindível a funcionalidade de controle dos níveis de usuário que acessam o *ScadaBR*[®], ou seja, para realizar o acesso à configuração de fontes de dados, pontos de dados, alarmes, acionamento de equipamentos, entre outros, é necessário possuir um usuário tipo administrador, assim, no momento da geração de cada usuário, o administrador poderá especificar o nível de acesso e, ainda, configurar quais as informações estarão disponíveis e habilitar ou não o envio de alarmes configurados pelo administrador (GOMES, 2014).

2.4 FORMAS DE MEDIÇÃO E COMPARAÇÃO

Desde os primórdios da humanidade, as necessidades de comparação de fatos através da observação permitiam a determinação de certos episódios, como por exemplo, a observação do sol permitia que os antigos concluíssem que os dias e noites eram fatos que ocorriam com uma certa periodicidade. Atualmente, a medição de grandezas é primordial para comprovação de algum experimento, sendo estas medições remetidas às leis que fundamentaram cientificamente aquele experimento (BALBINOT, 2011).

2.4.1 Normas, Regulamentos e Critérios a Serem Adotados

A Aneel é a agência reguladora do sistema de energia elétrica do Brasil, desta forma, tem a função de regular, fiscalizar, implementar políticas e diretrizes, estipular custos, entre outros.

A tensão secundária nominal de fornecimento do Brasil alterna de estado para estado, conforme critérios adotados por cada concessionária, desta forma, para análise dos limites inferiores e superiores serão utilizados os valores de tensão de linha 220V e tensão de fase 127V (ANEEL, 2018).

Segundo Aneel (2018), definem-se os níveis adequado, precário e crítico de fornecimento de tensão em regime permanente, para unidades consumidoras atendidas em 220/127V, conforme valores dos itens a até c:

- a) Adequado: $202V \leq TL \leq 231V$ e $117 \leq TL \leq 133$;
- b) Precário: $191V \leq TL < 202V$ ou $231V < TL \leq 233V$ e $110V \leq TL < 117V$ ou $133V < TL \leq 135V$;
- c) Crítico: $TL < 191V$ ou $TL > 233V$ e $TL < 110V$ ou $TL > 135V$.

Ainda para medição da qualidade do fornecimento de energia, cada concessionária deve manter a continuidade de abastecimento, sendo que para representar os indicadores coletivos são utilizados o DEC e FEC, e para representar os indicadores individuais são utilizados o DIC, FIC, DMIC e DICRI. DEC e FEC são utilizados para representar a duração e frequência, respectivamente, em que um conjunto de unidades consumidoras permaneceram sem energia, já o DIC e FIC são utilizados para representar a duração e frequência, respectivamente, em que uma unidade consumidora permaneceu sem energia. Outros indicadores como o DMIC e o DICRI, também representam valores individuais para uma unidade consumidora ou ponto de conexão, sendo utilizados para representar o tempo máximo de interrupção e a duração individual de cada interrupção do fornecimento de energia, ocorrido em dia crítico, respectivamente. Os valores de limite destes indicadores são definidos pela Aneel e repassados às concessionárias (ANEEL, 2018).

2.4.2 Aquisição da Medição

A utilização de microcontroladores ou microprocessadores digitais para leitura de sinais de tensão ou corrente analógicos não é possível sem a realização de um conversor analógico para digital, ou seja, a leitura analógica é realizada e, posteriormente, convertida para uma representação binária equivalente. Assim, a amostragem é a coleta de dados realizada em períodos discretos em função do tempo, ou seja, uma determinada leitura analógica é amostrada em períodos definidos, sendo os valores coletados, armazenados em uma memória, sendo estes, proporcionais à magnitude do sinal original (BALBINOT, 2011).

Devido a esta característica, as medições de tensão e corrente, por exemplo, precisam de um determinado número de amostras no tempo, para que o sinal representado após a conversão seja o mais fiel possível ao sinal original. O Teorema de Nyquist mostra que a razão da amostragem (f_s) de um sinal deve ser igual ou superior a duas vezes a frequência máxima (f_m) de operação do sistema, conforme Equação 2.11 (BALBINOT, 2011), sendo f_s e f_m dados em Hertz:

$$f_s \geq 2 * f_m$$

Equação (2.11)

Assim, por exemplo, para um sistema de frequência máxima (f_m) de 60Hz, a razão de amostragem deve ser igual ou superior a 120Hz, para cada sinal analógico medido.

3 METODOLOGIA

Para desenvolvimento do protótipo do medidor de grandezas elétricas, utilizou-se a pesquisa de referências apresentada no capítulo 2 deste trabalho, definindo assim, os requisitos de *hardware* e *software* necessários para cumprimento dos objetivos definidos.

Assim, definiu-se a utilização da placa microcontrolada *Arduino*® UNO Rev3, como a base operacional do protótipo, sistema na plataforma *Open Source*, na qual, todos os equipamentos periféricos serão conectados, e que receberá as informações dos transdutores de tensão e corrente, bem como realizará o processamento dos dados e envio para o supervisor na plataforma *SCADA*®.

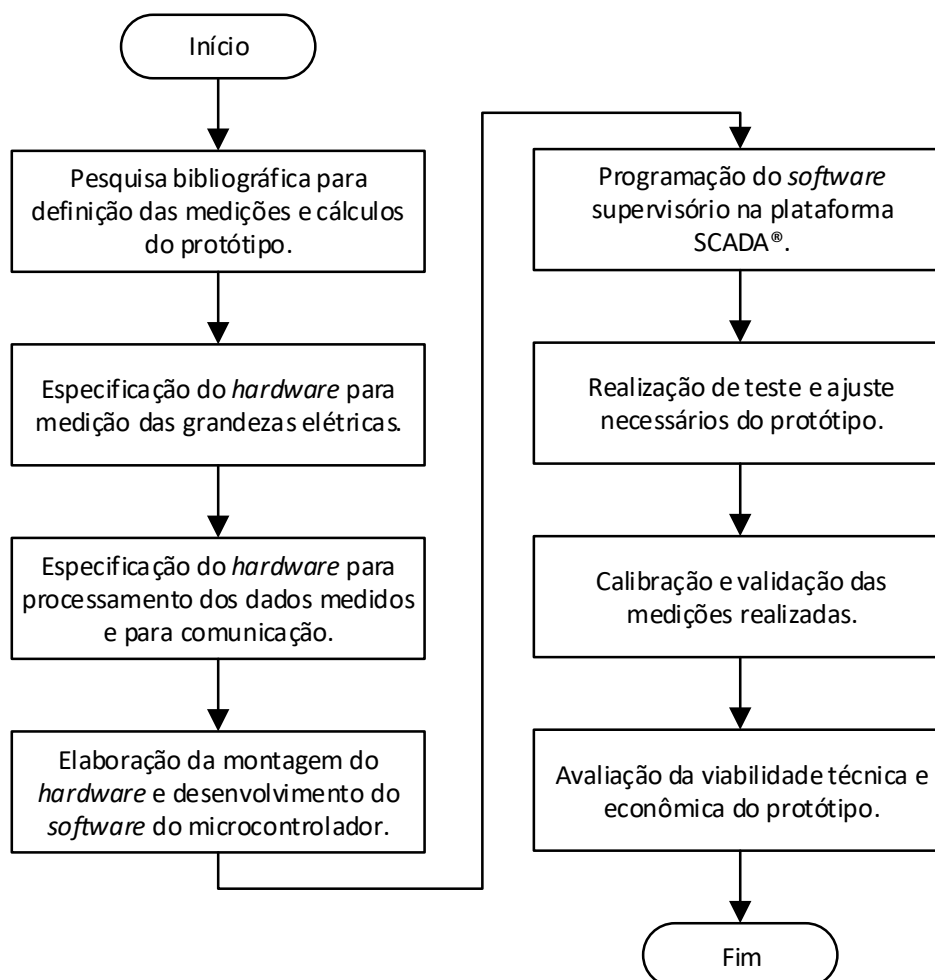
Posteriormente, será desenvolvido o algoritmo que realizará os cálculos, interpretação, tratamento e transferência dos dados coletados pelos transdutores ao *software* supervisor, que será desenvolvido na plataforma *SCADA*®. Sendo o *hardware* e *software* de processamento desenvolvidos com a plataforma *Arduino*®.

Após o tratamento dos dados e disponibilização ao *software* supervisor, será desenvolvida a etapa de personalização da interface com o usuário, sendo utilizado o *ScadaBR*®, que também é uma plataforma de código aberto, assim como o *Arduino*®, no qual serão definidos os níveis de acesso, formas de apresentação dos dados, indicadores, relatórios, gráficos em tempo real e o banco de dados. Assim, poderá ser desenvolvida a programação de aquisição dos dados transmitidos pelo microcontrolador ao supervisor.

Por fim, será feita a conferência dos dados coletados pelo projeto através da comparação com outro instrumento de medição. Comparando os dados de ambos os equipamentos, poderemos verificar a precisão das leituras do projeto.

A Figura 3.1 exibe o fluxograma macro de desenvolvimento das atividades para confecção do protótipo de medição de grandezas elétricas:

Figura 3.1 - Fluxograma de Atividades Confeção Protótipo



Fonte: Autor

Os parâmetros adotados para escolha dos materiais empregados neste protótipo foram: funcionalidade e baixo custo, permitindo assim, realizar as medições necessárias com o menor custo total possível. Com base nestas especificações, o item 3.1 apresenta a lista completa de materiais e *softwares* necessários para implementação do protótipo.

3.1 LISTA DE MATERIAIS

Para confecção do protótipo do medidor de grandezas elétricas para monitoramento da qualidade da energia em residências, será necessária a implementação dos seguintes materiais e *softwares* descritos nos itens a até o:

- a) Microcontrolador *Arduino*® UNO Rev3;

- b) *Arduino® Ethernet Shield W5100*;
- c) Transdutor de corrente elétrica SCT-013-050 50A;
- d) Transdutor de tensão elétrica ZMPT101B 250V;
- e) Caixa PVC sobrepor 190x140x70mm;
- f) Conector fêmea PJ-392 3,5mm estéreo;
- g) Conector barreira 4 vias 15A;
- h) Cabo extensão RJ45 macho para fêmea;
- i) Cabo extensão *USB* macho para fêmea;
- j) *Power Bank* 5600mAh EP-C897;
- k) Chave liga/desliga 10A;
- l) Porta fusível c/ fusível 1A;
- m) Cabos para interligação;
- n) *Software* supervisor *ScadaBR®*;
- o) *Software* de programação *Arduino®*.

Os custos e especificação dos modelos dos equipamentos descritos nos itens de *a* até *o*, serão apresentados no desenvolvimento do projeto.

3.2 CUSTOS

Apresentados os materiais e *softwares*, faz-se necessário realizar a apresentação dos custos unitários e total para implementação do projeto, conforme a Tabela 3.1:

Tabela 3.1 - Custos de Implementação do Protótipo

Item	Produto	Custo
1	Microcontrolador <i>Arduino</i> ® UNO Rev3	R\$ 65,50
2	<i>Arduino</i> ® <i>Ethernet Shield</i> W5100	R\$ 25,92
3	Transdutor de corrente elétrica SCT-013-050 50A	R\$ 20,46
4	Transdutor de tensão elétrica ZMPT101B 250V	R\$ 6,89
5	Caixa PVC sobrepor 190x140x70mm	R\$ 48,00
6	Conector fêmea PJ-392 3,5mm estéreo	R\$ 0,65
7	Conector barreira 4 vias 15A	R\$ 5,47
8	Cabo extensão RJ45 macho para fêmea	R\$ 8,16
9	Cabo extensão <i>USB</i> macho para fêmea	R\$ 6,13
10	<i>Power Bank</i> 5600mAh EP-C897	R\$ 25,00
11	Chave liga/desliga 10A	R\$ 3,00
12	Porta fusível c/ fusível	R\$ 8,00
13	Cabos para interligação, resistores, capacitor	R\$ 15,00
Total		R\$ 238,18

Fonte: Autor

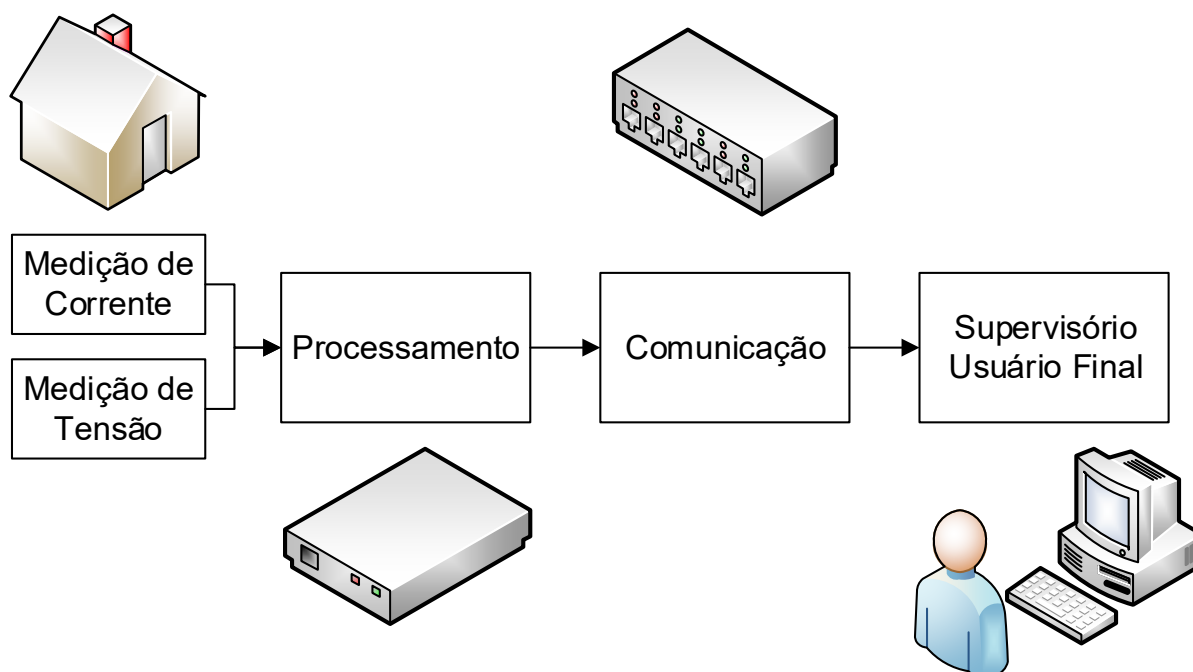
Findada a apresentação dos materiais e *softwares* que serão empregados no desenvolvimento deste protótipo, bem como o custo de aquisição geral para construção do mesmo, será realizada a explanação da metodologia de montagem e programação.

3.3 HARDWARE

Para desenvolvimento do *hardware* do protótipo, projetou-se um sistema interligado na qual exibe todo o sistema, de forma resumida, visando facilitar o entendimento e visualização de cada parte integrante da parte física do protótipo.

Sendo assim, a Figura 3.2 apresenta os componentes integrantes do *hardware* do protótipo, partindo da coleta das informações da carga a ser medida, passando pela parte de processamento, transmissão dos dados coletados/processados, até a disponibilização ao usuário final:

Figura 3.2 - Integração do *Hardware* do Protótipo



Fonte: Autor

Utilizando o mesmo critério da seleção dos materiais do protótipo, a topologia de montagem também deverá ser simples, conforme mostrada na Figura 3.2. Desta forma, o item seguinte apresentará a funcionalidade dos itens aplicados no *hardware* do protótipo.

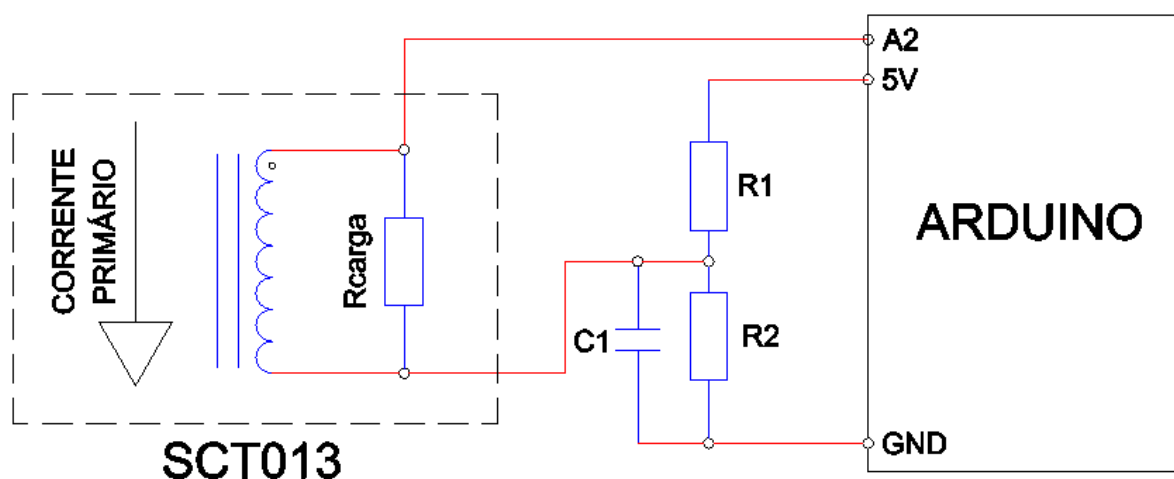
3.3.1 Medição de Corrente Elétrica

Para definir o *hardware* de medição de corrente elétrica que será aplicado no medidor, foi considerada uma corrente máxima de 50A, visto que a aplicação do medidor se dará em uma residência, sendo utilizado o SCT-013-050.

Conforme evidenciado no item 2.2.1.3.1, o transdutor é um transformador de corrente não invasivo, com núcleo dividido, o que possibilita a instalação sem que seja necessário interromper o circuito a ser medido. O primário do transformador acaba sendo o próprio circuito a ser medido, já o secundário é construído de ferrite. O modelo projetado para 50A possui em série com o secundário, um resistor, disponibilizando saída de 1V, ou seja, para uma corrente de 50A no primário, a saída disponibilizará 1V (THOMSEN, 2015).

Como o transdutor já fornece uma saída em tensão variável de 0 a 1V, ele pode ser conectado diretamente à entrada analógica do *Arduino*®, neste caso, utiliza-se a entrada A2. Será necessário apenas incluir um divisor de tensão para realizar alimentação do transdutor, desta forma, ele realizará a alteração do centro da senoide para 2,5V, da mesma maneira que o transdutor de tensão. Para confecção do divisor de tensão, será utilizado dois resistores de 10k Ω (R1 e R2) e um capacitor de 10 μ F (C1). O resistor denominado Rcarga é interno ao transdutor. A Figura 3.3 apresenta o circuito de ligação do transdutor ao *Arduino*®:

Figura 3.3 - Circuito de ligação do transdutor SCT-013-050



Fonte: Autor

Exemplificada a forma de medição de corrente elétrica, faz-se necessário definir a forma de medição da tensão elétrica.

3.3.2 Medição de Tensão Elétrica

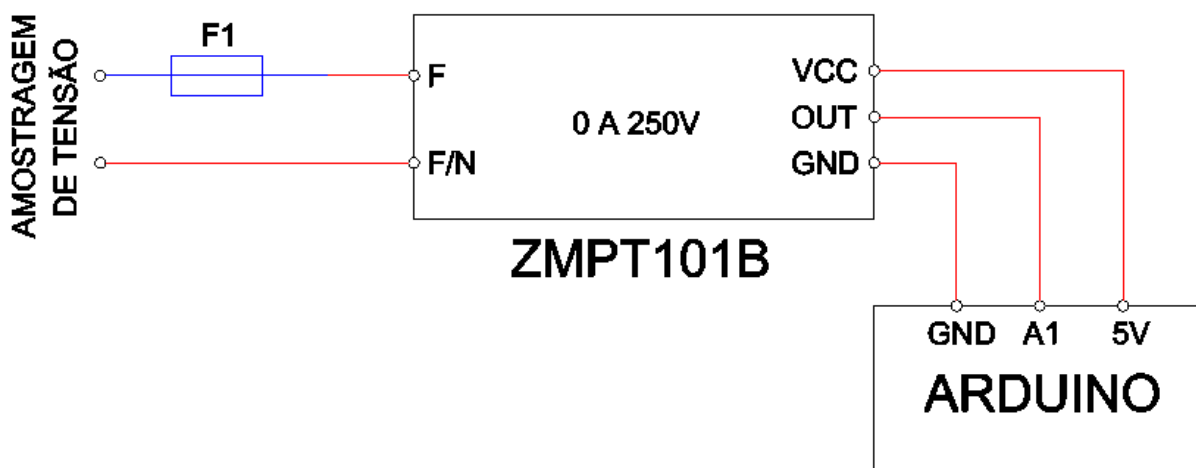
Para definir o *hardware* de medição de tensão elétrica que será aplicado no medidor, foi considerada uma alimentação monofásica de no máximo 250V entre fase e neutro, assim, será utilizado o transdutor de tensão ZMPT101B.

Conforme item 2.2.1.3.2, o transdutor possui um transformador de potencial interno para redução da tensão da rede e um circuito secundário que possui um divisor de tensão, desta forma, possibilita que a senoide da tensão da rede seja medida por completo, pois ajusta o zero da senoide para 2,5V de saída, assim, os

microcontroladores realizam a medição completa da tensão, visto que as entradas analógicas trabalham entre 0 e 5V, não aceitando valores negativos (HUDSON, 2017).

Com esta característica, o transdutor pode ser conectado diretamente à entrada analógica do *Arduino*®, neste caso será utilizada a entrada A1. Este transdutor precisa também de uma alimentação de 5V, a qual está sendo fornecida pelo próprio *Arduino*®. O componente F1 é um fusível de 1A, ele realizará a proteção elétrica do ponto de amostragem de tensão caso ocorra algum problema elétrico com o transdutor. A Figura 3.4 apresenta o circuito de ligação do transdutor ao *Arduino*®:

Figura 3.4 - Circuito de ligação do transdutor ZMPT101B



Fonte: Autor

Definidos os modelos e formas de ligação dos transdutores de tensão e corrente elétrica, verificamos que ambos estão conectados ao microprocessador *Arduino*®, desta forma, precisamos definir os parâmetros de ligação do próprio *Arduino*®, bem como definir os periféricos que serão necessários para o seu completo funcionamento.

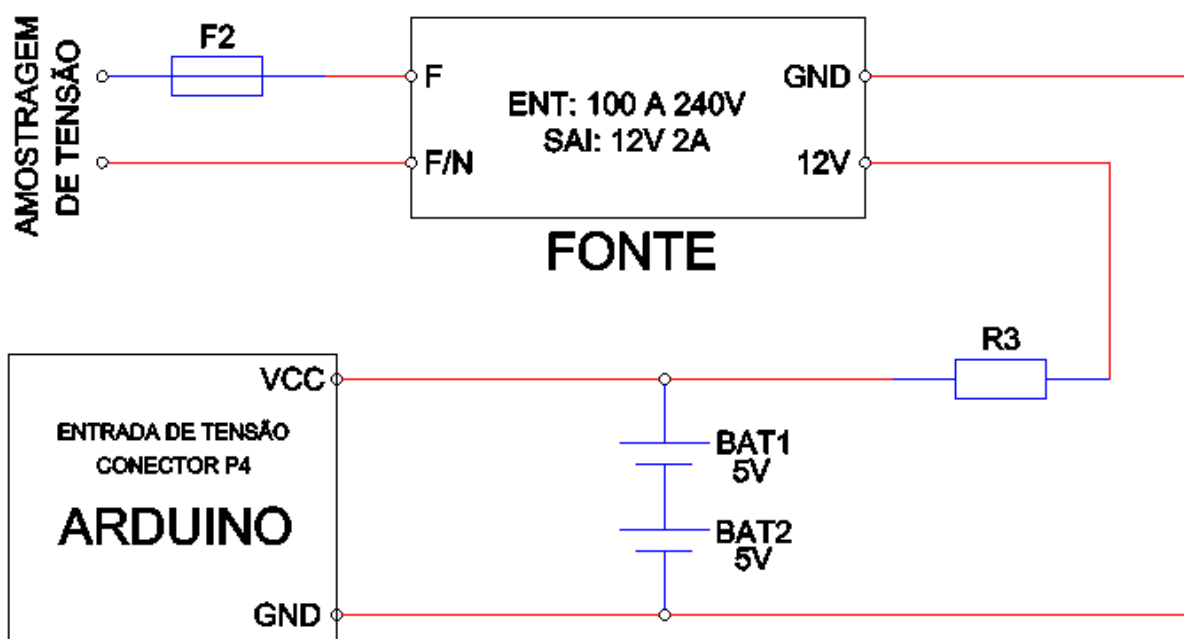
3.3.3 Microcontrolador *Arduino*® Uno Rev3

Conforme os itens 3.3.1 e 3.3.2, os transdutores de tensão e corrente estão conectados diretamente ao *Arduino*®, visto que estes já possuem circuitos internos que realizam a disponibilização dos sinais de corrente e tensão medidas inicialmente, em níveis aceitáveis pelo microcontrolador.

Para que o microcontrolador permaneça em funcionamento, é necessário que o mesmo esteja alimentado, o *Arduino*® possui dois locais para alimentação: via conector USB tipo B, no qual também é realizada a programação e comunicação serial; e via conector do tipo P4. Neste caso, será utilizada a segunda opção, visto que a primeira será utilizada exclusivamente para programação do protótipo.

A Figura 3.5 mostra o circuito de alimentação do microcontrolador:

Figura 3.5 - Alimentação do Microcontrolador *Arduino*® Uno Rev3



Fonte: Autor

Como será necessário coletar tensão para o transdutor de tensão, a fonte de alimentação do *Arduino*® será conectada a este mesmo ponto, também protegida eletricamente por um fusível (F2) de 1A. A fonte é compatível com as características do protótipo, podendo ser alimentada de 100 a 240V, tendo sua saída estável em 12V, suportando até 2A de corrente na saída.

Conforme Figura 3.5, o circuito também dota de um jogo de baterias, esta característica é essencial para o projeto, ele irá atuar quando o sistema não receber tensão da rede elétrica, mantendo o microcontrolador e todos os seus periféricos energizados. Isso se faz necessário para atender a característica do projeto que informará ao usuário final, quantas vezes e qual o tempo ele permaneceu sem energia.

Sem o sistema de baterias, o microcontrolador ficará desligado durante o período de falta de energia, não enviando informações ao supervisor durante este período. Outro ponto de grande importância é o computador em que o supervisor permanecerá em funcionamento, este também precisa permanecer energizado durante a falta de energia, garantindo assim, a comunicação entre o microcontrolador e o supervisor. Para este projeto, estamos considerando a utilização de um *notebook*.

A fonte de alimentação do microcontrolador também realizará o carregamento das baterias, ou seja, após um período de falta de energia, o sistema de baterias precisará ser recarregado para atuar novamente quando necessário, assim, foi incorporado ao sistema um resistor, que irá limitar a corrente do carregamento das baterias e do microcontrolador, e também, irá fornecer tensão dentro dos limites do conjunto de baterias.

O conjunto de baterias é especificado para trabalhar com até 5V, como as baterias estão em série, podem ser alimentadas com até 10V, desta forma, a queda de tensão (V_{lim}) esperada no resistor limitador (R_{lim}) é de 4V e a corrente máxima (I_{lim}) de 500mA, mantendo uma tensão de 8V no conjunto de baterias. A Equação 3.1 mostra o cálculo do resistor limitador, apresentado na Figura 3.5, considerando os dados iniciais:

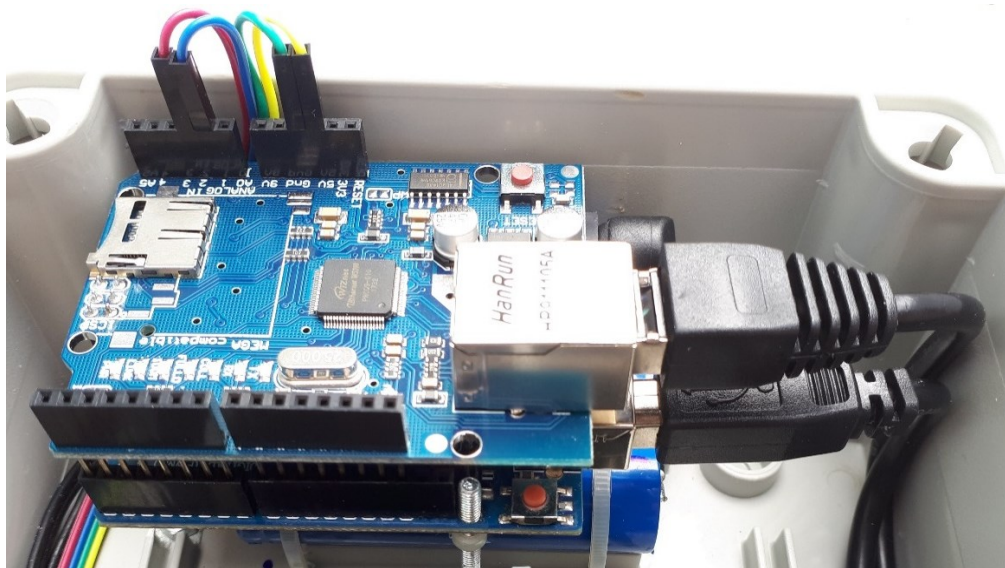
$$R_{lim} = V_{lim} / I_{lim} = 4 / 0,5 = 8\Omega \quad \text{Equação (3.1)}$$

Determinada a alimentação do microcontrolador e os transdutores, resta apenas definir o sistema de comunicação entre o microcontrolador e o supervisor.

3.3.4 *Arduino® Ethernet Shield W5100*

O módulo *Arduino® Ethernet Shield W5100* será utilizado para realizar a comunicação entre o microcontrolador *Arduino®* e supervisor. Este modelo de *shield* é específica para utilização em *Arduino®*, desta forma, possui encaixe projetado para os modelos UNO e MEGA, ou seja, não é necessário realizar alterações ou conexões externas, a placa vai conectada sobre os pinos do *Arduino®*, conforme Figura 3.6:

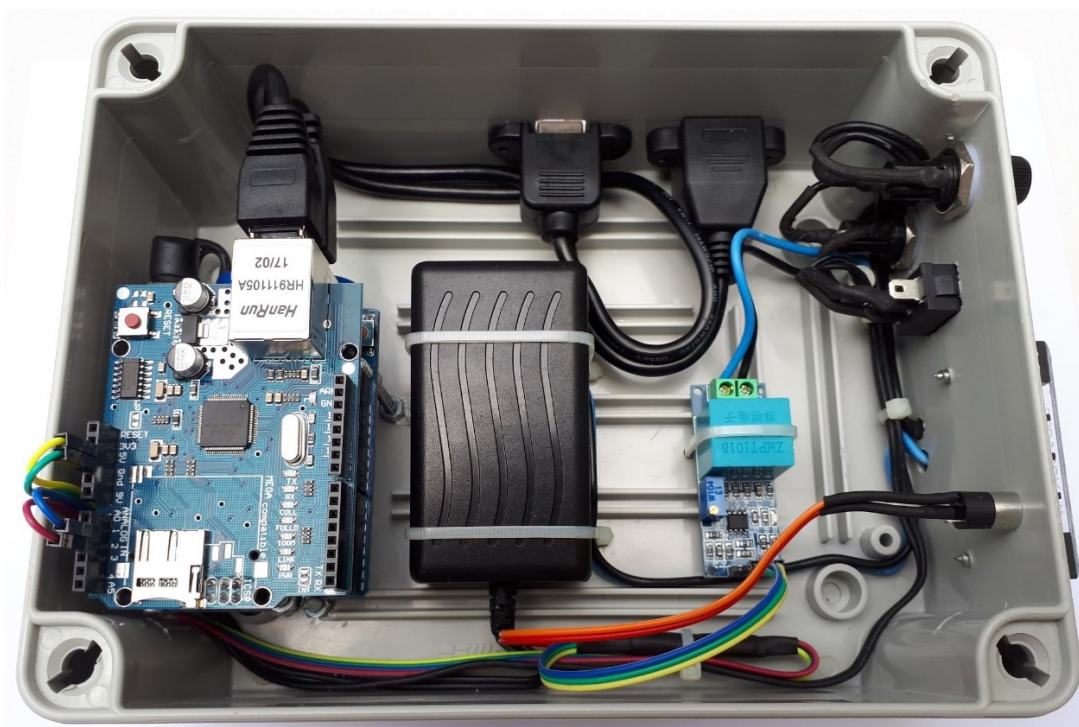
Figura 3.6 - Conexão *Arduino® Ethernet Shield W5100*



Fonte: Autor

Findada a especificação e projeto de montagem/ligação do tipo de *hardware* a ser utilizado no protótipo, realizou-se a montagem de todos os componentes especificado na lista de materiais, levando em consideração os circuitos de ligação apresentados nos itens anteriores. A Figura 3.7 mostra de forma geral o protótipo:

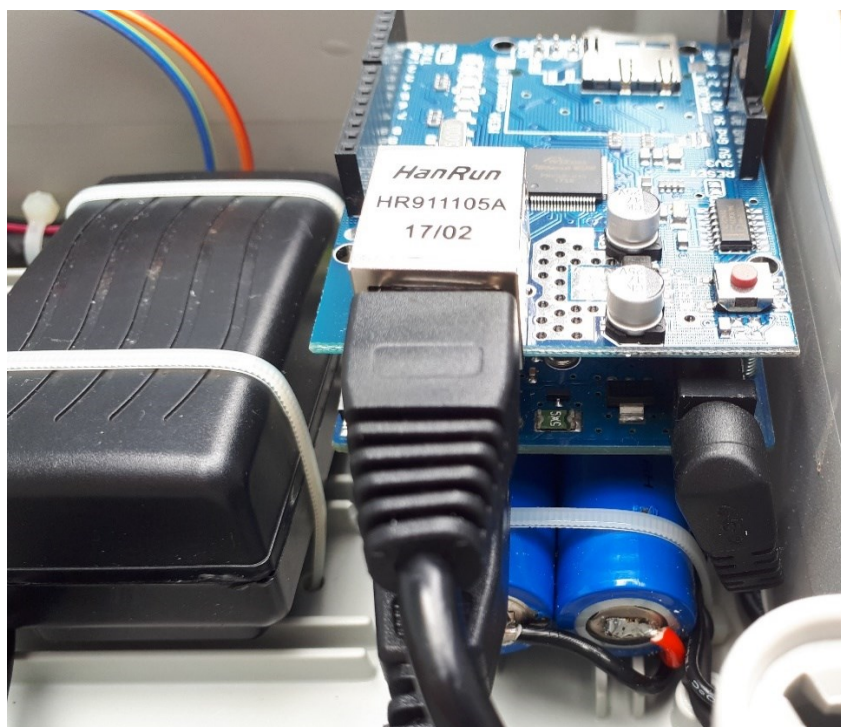
Figura 3.7 - Protótipo Medidor de Energia Elétrica



Fonte: Autor

Já a Figura 3.8 apresenta o microcontrolador *Arduino*®, juntamente com a *Ethernet Shield*, ou seja, o processamento e comunicação do protótipo e, também, o conjunto de baterias que está fixo abaixo do suporte das duas placas mencionadas anteriormente:

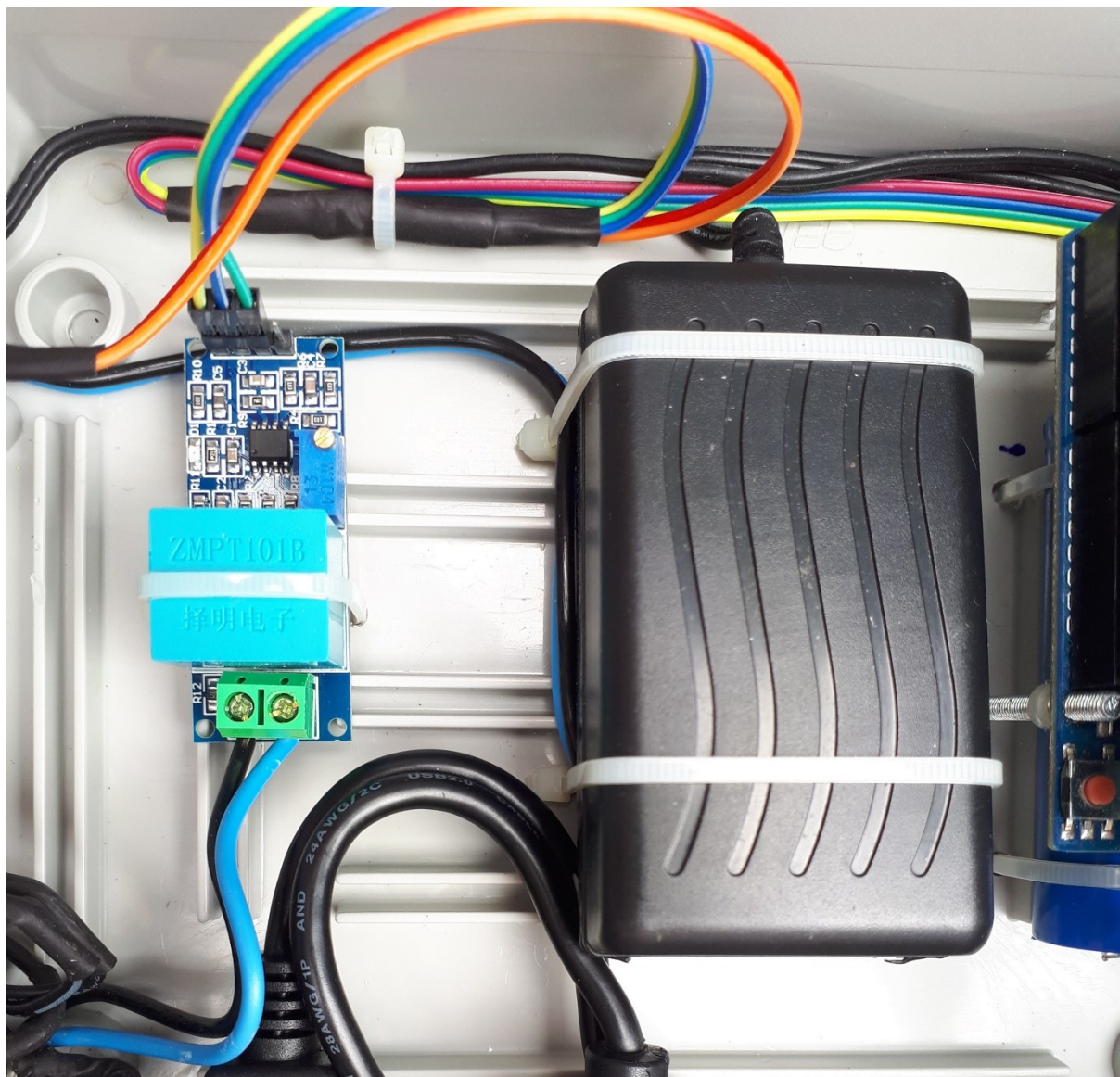
Figura 3.8 - Processamento e Comunicação do Protótipo



Fonte: Autor

Por sua vez, a Figura 3.9 apresenta a fonte de alimentação do protótipo, juntamente com o transdutor de tensão:

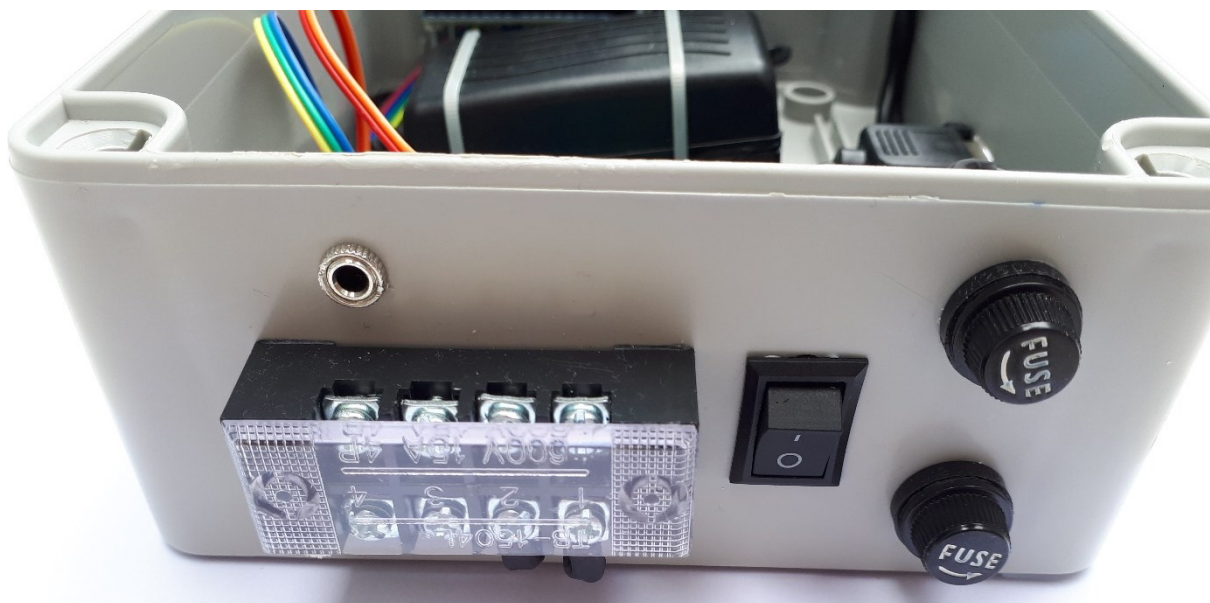
Figura 3.9 - Alimentação e Coleta de Tensão do Protótipo



Fonte: Autor

A Figura 3.10 traz a parte externa do protótipo, onde são conectados a amostragem de tensão no conector tipo barreira, a conexão do transdutor de corrente no conector P2, a chave liga e desliga do sistema e, por fim, os dois fusíveis de proteção do protótipo, um para a fonte de alimentação e outro para o transdutor de tensão:

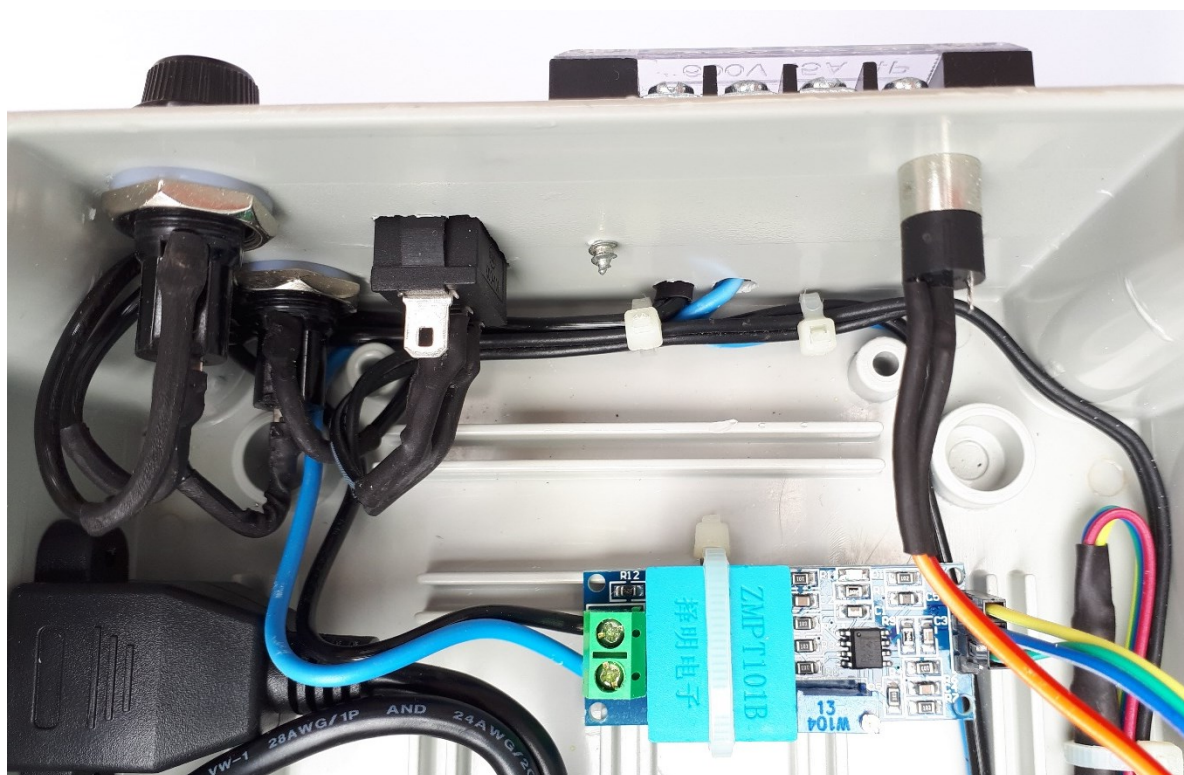
Figura 3.10 - Conexões de Entrada do Protótipo



Fonte: Autor

A Figura 3.11 mostra as conexões internas dos equipamentos mostrados na Figura 3.10:

Figura 3. 11 - Ligação dos Componentes de Entrada de Tensão e Corrente



Fonte: Autor

Por fim, a Figura 3.12 apresenta as conexões de programação e comunicação do protótipo, ambas são conectadas ao microprocessador e a *Ethernet Shield* por cabos extensores tipo *USB* tipo B e RJ45, respectivamente:

Figura 3.12 - Conexão de Programação e Comunicação Protótipo



Fonte: Autor

Concluída a etapa de definição e montagem do *hardware*, o próximo item apresentará as definições e programação do *software* do protótipo e do supervisor.

3.4 SOFTWARE DO MICROCONTROLADOR

Para realizar a leitura, processamento e envio ao *software* supervisor, é necessário que o microprocessador seja programado com uma série de instruções, chamada algoritmo. Este algoritmo que irá gerenciar o *hardware* tratado no item 3.3 deste trabalho. Portanto, para cada componente do *hardware*, será descrito agora, o algoritmo que será utilizado para que o conjunto possa desempenhar a função para qual foi projetado:

3.4.1 Microcontrolador *Arduino*® e *Arduino*® *Ethernet Shield* W5100

Para confecção do protótipo, estaremos trabalhando com seis bibliotecas de funções do *Arduino*®, uma delas é referente à medição de energia e as demais ao

Arduino® Ethernet Shield W5100, para comunicação entre o microcontrolador e o supervisor, todas as bibliotecas serão adicionadas no início do algoritmo, conforme mostra o código fonte:

```
#include <SPI.h> //Inclusão Biblioteca Comunicação Serial
#include <Ethernet.h> //Inclusão Biblioteca Comunicação Ethernet
#include <Modbus.h> //Inclusão Biblioteca Comunicação Modbus
#include <ModbusIP.h> //Inclusão Biblioteca Comunicação ModbusIP
#include "EmonLib.h" //Inclusão Biblioteca de Medição
#include <EEPROM.h> //Inclusão Biblioteca de Gravação kWh
EnergyMonitor emon1; //Criação de uma Instância de Medição

//Criação do Offset de comunicação para o ScadaBR
const float Ireg_Vrms = 0; //Tensão
const float Ireg_Irms = 1; //Corrente
const float Ireg_W = 2; //Potência Ativa
const float Ireg_VA = 3; //Potência Aparente
const float Ireg_VAR = 4; //Potência Reativa
const float Ireg_FP = 5; //Fator de Potência
const float Ireg_kWhtotal = 6; //Consumo em kWh
float amostra_Irms;
```

Neste código fonte, podemos verificar também a criação dos *offsets* para comunicação entre o microcontrolador e o supervisor. Estes *offsets* são necessários para informar ao supervisor qual o caminho, dentro da comunicação *Ethernet*, para que os dados sejam lidos pelo programa.

Prosseguindo com o desenvolvimento do algoritmo, faz-se necessário realizar a definição dos parâmetros do *Arduino® Ethernet Shield W5100*, desta forma, o código fonte a seguir, mostra as funções que configuram o *MAC*, *IP*, *Gateway* e *Máscara de Rede*, estas definições são necessárias para que o *Shield* possa se conectar à rede na qual está inserida, ou seja, como ele será conectado ao microcomputador, precisamos definir um *IP* válido, na mesma faixa do microcomputador, possibilitando assim a conexão.

Ainda neste código fonte, verificamos também a declaração da função que habilita a comunicação *Modbus IP*, que sem a qual não é possível realizar a comunicação com o supervisor. Finalmente, já dentro da função *void setup ()*,

realiza-se a habilitação da comunicação *Ethernet*, com os parâmetros de rede já estabelecidos e também, a comunicação Serial:

```
//Configuração do Ethernet Shield
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
byte ip[] = {192, 168, 1, 195};
byte gateway[] = {192, 168, 1, 1};
byte subnet[] = {255, 255, 255, 0};

//Configuração da Comunicação para o ScadaBR
ModbusIP mb;
void setup() {
  //Habilitando a comunicação Serial e Ethernet
  Serial.begin(9600);
  Ethernet.begin(mac, ip, gateway, subnet);
```

Realizada a inclusão das bibliotecas necessárias para o funcionamento do protótipo e também a definição das configurações iniciais, agora faz-se necessário declarar a parte do algoritmo que ficará em constante execução. Estas funções são declaradas dentro da estrutura *void loop ()*, sendo executadas até que o *Arduino*® seja desligado ou reiniciado.

Assim, dentro da estrutura *void loop ()* estarão contidas as instruções para coleta de informações dos transdutores e cálculo dos demais parâmetros de medição do protótipo, estas instruções serão tratadas nos itens 3.4.2 a 3.4.8.

3.4.2 Tensão e Corrente Elétrica

Inicialmente, o *Arduino*® realiza medição de tensão e corrente instantânea, através dos transdutores de tensão e corrente que retornam um valor de 0 a 5V e, 0 a 1V, respectivamente, estes sinais são entregues a duas entradas analógicas do *Arduino*®, possibilitando assim, a manipulação destes dados por parte do *Arduino*®.

A biblioteca para medição utilizando *Arduino*® é chamada de *EmonLib.h* (código fonte completo localizado no Apêndice A), nela estão descritos internamente as instruções necessárias para leitura de tensão e corrente e, posteriormente, o cálculo das demais grandezas fornecidas nesta biblioteca.

Desta forma, o código fonte mostra as duas configurações necessárias para a leitura de tensão e corrente:

```
//Definição dos pinos de entrada de tensão e corrente
// Tensão: pino de entrada, calibração, mudança de fase
emon1.voltage(1, 231.26, 0);
// Corrente: pino de entrada, calibração
emon1.current(2, 48.7);

//Habilitando a função de Registradores de Entrada para a
//comunicação com o ScadaBR para cada variável (cada grandeza).
mb.addIreg(Ireg_Vrms);
mb.addIreg(Ireg_Irms);
mb.addIreg(Ireg_W);
mb.addIreg(Ireg_VA);
mb.addIreg(Ireg_VAR);
mb.addIreg(Ireg_FP);
mb.addIreg(Ireg_kWhTotal);
}
```

A primeira linha de programação mostra a definição da entrada analógica que receberá os sinais de tensão, posteriormente mostra a calibração do transdutor, permitindo os ajustes necessários para realizar a leitura o mais próximo do valor real possível e, por fim, a calibração da mudança de fase. Já a segunda linha de programação mostra a definição da entrada analógica que receberá os sinais de corrente e, a seguir, mostra a calibração do transdutor.

A partir destas configurações já é possível realizar a medição de tensão e corrente instantânea e também a eficaz. Para o cálculo da tensão e corrente eficaz, a biblioteca *EmonLib.h* executa internamente três passos: primeiramente é realizada a leitura de tensão e corrente a partir das configurações mostradas anteriormente; na sequência, o algoritmo aplica um filtro passa baixa para realizar o deslocamento do centro da senoide (momento da passagem por 0V) para a tensão de 2,5V, desta forma é possível realizar a leitura de todo o sinal senoidal amostrado, bem como realizar a verificação do deslocamento do sinal de tensão e corrente. Com este segundo passo é possível extrair os sinais de tensão e corrente *RMS*; por último, é realizado o cálculo de tensão e corrente *RMS*, através da obtenção dos valores médio quadráticos dos sinais amostrados e filtrados.

Como estes passos são realizados internamente na biblioteca, apenas será realizada a inserção no algoritmo das funções que extraem os valores de tensão e corrente *RMS* da biblioteca, conforme mostra o código fonte:

```
//Função realiza o cálculo das grandezas da biblioteca Emonlib.h
//Número de cruzamento por zero, tempo limite
emon1.calcVI(20,2000);

//Extrai os valores de Tensão RMS da variável
float Vrms = emon1.Vrms;
//Extrai os valores de Corrente RMS da variável
float amostra_Irms = emon1.Irms;
```

Para realizar um tratamento nos dados fornecidos ao usuário final, evitando que valores baixos de tensão, corrente e potências, gerados pelo próprio erro dos transdutores, afetem a visualização dos dados, foi realizado a implementação de uma condição que, quando os valores de corrente forem menores ou iguais a 200mA, o algoritmo envia zero para todas as grandezas, mantendo apenas os valores de tensão, esta condição está sendo representada no código fonte:

```
//Exclui os valores de corrente, potências e fator de potência
//quando os valores de corrente são menores que 200mA
if(amostra_Irms <= 0.2){
float Irms = 0;
float VA = 0;
float FP = 0;
float W = 0;
float VAR = 0;
```

Após a obtenção dos valores *RMS* de tensão e corrente é possível realizar o cálculo das demais grandezas como: potência ativa, aparente e fator de potência.

3.4.3 Potência Ativa

Para obtenção da potência ativa, a biblioteca *EmonLib.h* realiza internamente o cálculo, utilizando os valores de tensão e corrente *RMS*, como já mostrado anteriormente. Juntamente aos valores *RMS* é necessário também o atraso do ângulo

da tensão e corrente, que também é obtido internamente, utilizando os passos descritos no 3.4.2.

Assim, os valores de potência ativa são extraídos realizando a inserção no algoritmo da função da biblioteca, conforme código fonte:

```
//Extrai os valores de Potência Ativa da variável  
float W = emon1.realPower;
```

A obtenção da potência aparente é realizada conforme explicação do item 3.4.4.

3.4.4 Potência Aparente

A potência aparente é extraída diretamente dos valores de tensão e corrente instantâneos, desta forma, a partir do produto dos valores *RMS* de tensão e corrente, podemos obter os valores de potência aparente. Realizando a inserção da função de potência aparente no algoritmo é possível realizar a obtenção dos valores dessa potência, conforme código fonte:

```
//Extrai os valores de Potência Aparente da variável  
float VA = emon1.apparentPower;
```

Como o algoritmo já realizou os cálculos de potência ativa e aparente, é possível realizar a obtenção do fator de potência do sistema a ser medido.

3.4.5 Fator de Potência

O fator de potência da carga é obtido através da divisão dos valores extraídos de potência ativa e aparente, assim, inserindo esta função do fator de potência no algoritmo, o microcontrolador realiza o cálculo, conforme código fonte:

```
//Extrai os valores de Fator de Potência da variável  
float FP = emon1.powerFactor;
```

Findada a extração dos valores de tensão e corrente *RMS*, potência ativa e aparente e, fator de potência, a biblioteca *EmonLib.h* não disponibiliza mais medições, desta forma, é necessário realizar a manipulação dos dados já extraídos para obtenção das demais grandezas a serem verificadas pelo protótipo.

3.4.6 Potência Reativa

Como já exemplificado, a potência reativa deve ser calculada externamente a biblioteca *EmonLib.h*, desta forma, serão utilizados os valores já extraídos de potência ativa e aparente para a obtenção da potência reativa. Assim, esta será obtida através do Teorema de Pitágoras, conforme código fonte:

```
//Extrai os valores de Potência Reativa da variável
float VAR = sqrt (sq(VA) - sq(W));
```

Após a obtenção da potência reativa, estão concluídos os cálculos e obtenção das potências consumidas instantaneamente pela carga, permanecendo pendente apenas o consumo acumulado, que será tratado no item 3.4.8.

3.4.7 Potência Ativa Acumulada kWh

A potência ativa acumulada, ou simplesmente kWh, é obtida pelos valores consumidos de potência ativa em um determinado espaço de tempo, ou seja, a potência ativa apresentada pelo protótipo é um valor para consumo em uma hora de utilização e, para realizar o cálculo de kWh foi efetuada uma medição do intervalo de tempo de um ciclo do microcontrolador, que neste caso é de aproximadamente 0,195 segundos. Assim, a potência ativa é dividida pela quantidade de vezes em que o valor é atualizado em uma hora (18.400), sendo estes valores somados, e obtendo assim o kWh.

Neste protótipo a medição de kWh é realizada pelo algoritmo apresentado no código fonte:

```
//Realiza o cálculo de Consumo em kWh
float kWh = (W / 18400);
```

```
//Realiza o somatório do Consumo em kWh
kWh = kWh + armazena_kWhTotal;
//Transfere o valor do Consumo em kWh para a variável
float kWhTotal = kWh;
armazena_kWhTotal = kWhTotal;
//Escreve a variável na memória EEPROM
EEPROMWriteInt(1, armazena_kWhTotal);
```

Assim, os valores de potência ativa são divididos pela média do tempo real entre cada leitura realizada pelo *Arduino*®, sendo estes valores somados e armazenados em um acumulador, podendo ser apresentado posteriormente ao usuário.

Demonstrado também neste código fonte anterior a gravação dos valores do kWh na memória *EEPROM* do microcontrolador, fazendo com que, caso o protótipo seja desligado, os valores de kWh ainda permaneçam em sua memória, tornando o valor mais próxima à realidade, já que esta medida é um acumulador. Para que seja possível realizar a gravação na memória dos valores de consumo, foi necessário realizar a criação de variáveis para receber os valores do acumulador e, para gravar os valores na memória do microcontrolador, o código fonte demonstra esta configuração:

```
//Criação de variáveis para receber o consumo em kWh
float kWh;
float kWhTotal;
float armazena_kWhTotal;

//Configuração do Armazenamento do Consumo em kWh
void EEPROMWriteInt(int address, int value); //2 Bytes
int EEPROMReadInt(int address);
```

Dando sequência no algoritmo de gravação do consumo na memória do microcontrolador, o código fonte a seguir mostra a repetição para realizar a escrita do valor de consumo em dois endereços de memória, pois, devido ao espaço ocupado pelo valor, é necessário utilizar dois *bytes*:

```
//Repetição para definição de leitura e gravação de
//dois espaços da memória EEPROM
```

```

void EEPROMWriteInt(int address, int value) {
    byte hiByte = highByte(value);
    byte loByte = lowByte(value);
    EEPROM.write(address, hiByte);
    EEPROM.write(address + 1, loByte);
}

int EEPROMReadInt(int address) {
    byte hiByte = EEPROM.read(address);
    byte loByte = EEPROM.read(address + 1);
    return word(hiByte, loByte);
}

```

3.4.8 Envio Grandezas para o Supervisório

Completada a obtenção dos dados das grandezas elétricas através das medições e cálculos realizados no microcontrolador, faz-se necessário realizar o envio destas informações ao sistema supervisório, para isso, o código fonte abaixo mostra o algoritmo que realiza o envio das informações através da *Arduino® Ethernet Shield W5100* ao *software* supervisório:

```

//Envio dos valores das variáveis para o ScadaBR
mb.Ireg(Ireg_Vrms, Vrms*100);
mb.Ireg(Ireg_Irms, Irms*100);
mb.Ireg(Ireg_W, W*10);
mb.Ireg(Ireg_VA, VA*10);
mb.Ireg(Ireg_VAR, VAR*10);
mb.Ireg(Ireg_FP, FP*1000);
mb.Ireg(Ireg_kWhTotal, armazena_kWhTotal*100);

```

Estas informações já tratadas, serão recebidas pelo supervisório e, posteriormente, apresentadas ao usuário final, para análise dos mesmos.

Os valores que estão sendo multiplicados nas variáveis estão realizando o ajuste das casas decimais dos valores recebidos pelo *software* supervisório, visto que este recebe apenas números inteiros, desta forma, os valores recebidos são divididos pelo mesmo valor multiplicado, fazendo com que as casas decimais possam ser verificadas pelo usuário.

O código fonte completo do algoritmo desenvolvido encontra-se no Anexo A deste projeto.

3.5 SOFTWARE DO SUPERVISÓRIO

Neste protótipo, o *software* supervisório tem a função de apresentar os dados provenientes do microcontrolador ao usuário final. O supervisório é confeccionado em uma plataforma *open-source*, possibilitando o nível de personalização necessário para desenvolvimento deste projeto.

O *software* escolhido foi o *ScadaBR*[®], assim, serão descritos no item 3.5.1 e 3.5.2 as configurações necessárias para recebimento das informações e geração de alarmes para informar ao usuário, caso algum parâmetro definido esteja fora do pré-estabelecido.

3.5.1 Fonte de Dados

Para aquisição dos dados provenientes do microcontrolador é necessário configurar a Fonte de Dados do supervisório, ou seja, inserir os dados necessários para que o microcontrolador possa enviar os dados ao supervisório e que ambos estejam trabalhando no mesmo protocolo de comunicação para que não ocorram falhas de comunicação.

Para que seja possível realizar a comunicação com o supervisório, precisamos garantir que alguns parâmetros sejam atendidos, visto que o não cumprimento destes acarretará em não funcionamento da comunicação. No item 3.4.1 foram realizadas algumas configurações no microcontrolador, desta forma, devemos inserir este em uma rede que comporte estas configurações, mesmo requisito deve ser verificado para inserção do microcomputador que receberá o supervisório.

Garantidos estes parâmetros, podemos realizar a configuração da Fonte de Dados, a Figura 3.13 mostra a configuração da Fonte de Dados, incluindo as informações solicitadas pelo *ScadaBR*[®], a fim de estabelecer a comunicação com o microcontrolador:

Figura 3.13 - Fonte de Dados



Fonte: Autor

Já a Figura 3.14 mostra todos os parâmetros que são necessários para inclusão de uma fonte de dados. Neste momento, precisamos incluir o tipo da comunicação do microcontrolador com o supervisor e o *IP* que foi definido anteriormente no algoritmo de programação do *Arduino*®, desta forma, o supervisor poderá realizar a comunicação:

Figura 3.14 - Programação da Fonte de Dados

Propriedades do modbus IP

Nome: Medidor

Export ID (XID): DS_210735

Período de atualização: 1 segundo(s)

Quantificação: ☐

Timeout (ms): 500

Tentativas: 2

Apenas quantidades contínuas: ☐

Criar pontos de monitor de escravo: ☐

Máxima contagem de leitura de bits: 2000

Máxima contagem de leitura de registradores: 125

Máxima contagem de escrita de registradores: 120

Tipo de transporte: TCP com manter-vivo

Host: 192.168.1.195

Porta: 502

Encapsulado: ☐

Níveis de alarme de eventos

Exceção de data source: Urgente

Exceção de leitura de data point: Urgente

Exceção de escrita em data point: Urgente

Fonte: Autor

Após a configuração da Fonte de Dados, existirá então um caminho de comunicação entre o microcontrolador e o supervisor, permitindo assim, a configuração dos Pontos de Dados.

3.5.2 Pontos de Dados

Criada então a forma e o caminho de comunicação entre microcontrolador e supervisor, pode-se adicionar os pontos de dados, que serão a lista de dados enviados entre os equipamentos, ou seja, a tensão, corrente, entre outros, serão chamados de Pontos de Dados, assim, cada informação será tratada por um Ponto de Dado.

A Figura 3.15 mostra a configuração realizada para obtenção dos valores através dos Pontos de Dados, neste caso, o Ponto de Dados de Tensão. Para as demais grandezas foi realizado o mesmo procedimento, alterando apenas o *offset* que foi definido no algoritmo de programação do *Arduino*®. A Figura 3.15 também mostra o multiplicador para visualização das casas decimais de cada dado, conforme descrição do item 3.4.8 deste projeto.

Figura 3.15 - Programação do Ponto de Dados

A imagem mostra uma janela de software intitulada "Detalhes do data point". Ela contém vários campos de entrada e controles para configurar um ponto de dados. Os campos e seus valores são os seguintes:

Etiqueta	Valor
Nome	Tensão
Export ID (XID)	DP_005116
Id do escravo	1
Faixa do registro	Registrador de entrada
Tipo de dados modbus	Inteiro de 2 bytes sem sinal
Offset (baseado em 0)	0
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	☐
Multiplicador	0.01
Aditivo	0

Fonte: Autor

No momento da configuração dos Pontos de Dados, também foram adicionadas as unidades de medida de cada grandeza, desta forma, os dados do supervisor já estarão formatados conforme necessidade, facilitando as demais manipulações dentro do supervisor.

Juntamente aos Pontos de Dados, foram adicionados os Detectores de Valores, esta ferramenta irá ser responsável por informar ao usuário sempre que um valor de leitura estiverem fora dos valores, por exemplo, a tensão deverá seguir os parâmetros apresentados no item 2.4.1 deste trabalho e, caso permaneça fora destes valores, o usuário será informado.

3.5.3 Segurança do Supervisório

Por fim, faz-se necessário realizar um controle de acesso ao sistema supervisório, assim, foram adicionadas informações de acesso fictícias apenas para teste do supervisório, bem como definido o nível de acesso deste usuário que, por exemplo, será apenas visualizador de dados, não podendo realizar alterações. A Figura 3.16 mostra a forma de acesso supervisório, bem como a tela de autenticação, onde o usuário deverá inserir as informações e posteriormente realizar o acesso.

Figura 3.16 - Acesso ao Supervisório



SCaDaBR
1.0 CE - Community Edition

User id

Password

Login ?

✓ Chrome 69 on Windows
Este browser é suportado.

Fonte: Autor

Desta forma, o sistema permanece com suas informações em segurança, não havendo problemas com reconfigurações devido ao mau uso do sistema e também repasse de informações errôneas devido à alteração de dados indevidos.

4 RESULTADOS E DISCUSSÕES

Findada a etapa de desenvolvimento do *hardware* do protótipo e programação do *software*, tanto do microcontrolador quanto do supervisor, faz-se necessário realizar os testes de validação do funcionamento do protótipo.

Para realizar esta validação, será utilizado como comparativo, um multimedidor de energia da marca *Schneider Electric*®, modelo PM5110, juntamente com um transformador de corrente da marca *Henz*®, modelo RH-78, relação de transformação de 75/5A, este equipamento realiza a leitura de grandezas elétricas como tensão, corrente, potências, distorção harmônica de tensão e corrente, e frequência.

4.1 CALIBRAÇÃO DOS TRANSDUTORES

O primeiro passo para verificação da funcionalidade do protótipo é a calibração dos transdutores, esta etapa é predecessora para as demais, visto que será realizada o ajuste dos medidores de tensão e corrente, a fim de representar o mais próximo possível os valores que estão sendo lidos da rede.

Para tanto, utilizaremos o multimedidor de energia PM5110, tanto o protótipo, quanto o analisador estarão realizando a leitura no mesmo ponto, ou seja, deverão apresentar os mesmos valores de tensão e corrente.

Inicialmente o protótipo apresentava uma leitura diferente do multimedidor de energia PM5110, como mostra a Figura 4.1, pois até o momento, nenhum ajuste foi implementado ao sistema:

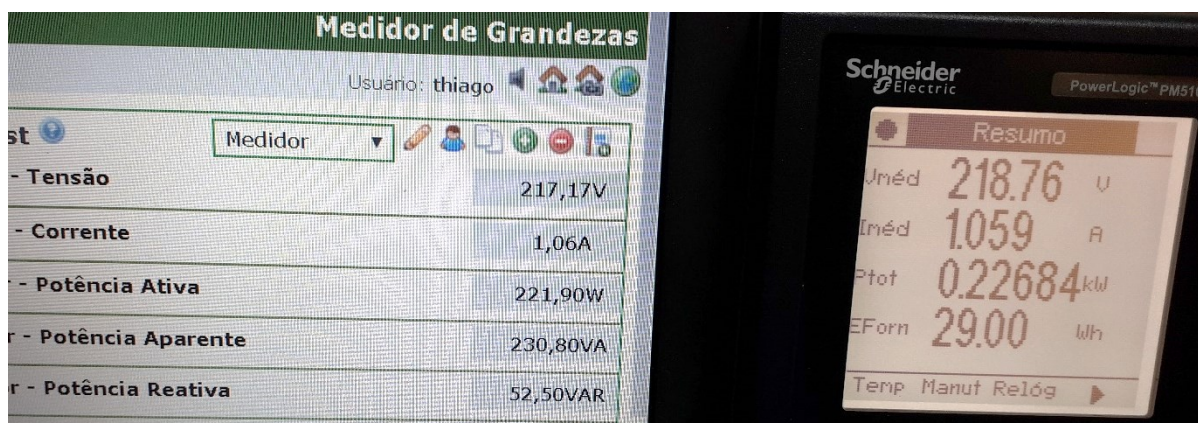
Figura 4.1 - Medição Anterior a Calibração



Fonte: Autor

Assim, foi realizado o ajuste dos transdutores no *software* do microcontrolador, alterando a programação da biblioteca *EmonLib.h*, a fim de obter o ajuste mais próximo do valor real da medição. A Figura 4.2 apresenta novamente a leitura do protótipo e do multimedidor de energia PM5110, já com os valores corrigidos:

Figura 4.2 - Medição Posterior a Calibração



Fonte: Autor

Após a realização da calibração dos transdutores, podemos iniciar os testes de medição de grandezas elétricas do protótipo. Esta segunda etapa foi dividida em duas gamas de medição: a primeira com uma carga puramente resistiva, utilizando um aquecedor de água da marca *E.B. Ind.*®, potência nominal de 830W e tensão de alimentação 220V; já a segunda com uma carga *RLC*, utilizando uma furadeira de bancada da marca *Somar*®, modelo FB 16, potência nominal 368W e tensão de alimentação de 220V.

Para padronização dos resultados, primeiramente será apresentada a medição com carga resistiva e, posteriormente, a medição com carga *RLC*, sendo amostrada dez valores de medição, durante um intervalo de até um minuto. O erro de medição será sempre mostrado tomando como base o multimedidor de energia PM5110.

4.2 MEDIÇÃO DE TENSÃO E CORRENTE

A validação da medição de tensão e corrente é bastante semelhante a calibração dos transdutores, visto que estamos coletando apenas tensão e corrente, sendo as demais grandezas calculadas internamente ao microcontrolador.

A primeira medição, com carga resistiva, resultou nos valores da Tabela 4.1:

Tabela 4.1 - Resultados de Medição Tensão e Corrente Carga Resistiva

Protótipo		Multimedidor de Energia PM5110		Erro Tensão (%)	Erro Corrente (%)
Tensão (V)	Corrente (A)	Tensão (V)	Corrente (A)		
222,48	3,59	221,70	3,61	0,35%	-0,55%
223,25	3,60	221,70	3,62	0,70%	-0,55%
223,31	3,59	221,64	3,62	0,75%	-0,83%
222,48	3,61	221,65	3,62	0,37%	-0,28%
222,47	3,58	221,61	3,62	0,39%	-1,10%
222,14	3,58	221,56	3,62	0,26%	-1,10%
221,84	3,59	221,49	3,62	0,16%	-0,83%
221,95	3,59	221,51	3,62	0,20%	-0,83%
222,12	3,61	221,45	3,62	0,30%	-0,28%
223,26	3,60	223,54	3,62	-0,13%	-0,55%

Fonte: Autor

A segunda medição, com carga *RLC*, resultou nos valores da Tabela 4.2:

Tabela 4.2 - Resultados de Medição Tensão e Corrente Carga *RLC*

Protótipo		Multimedidor de Energia PM5110		Erro Tensão (%)	Erro Corrente (%)
Tensão (V)	Corrente (A)	Tensão (V)	Corrente (A)		
215,80	1,01	216,37	1,02	-0,26%	-0,98%
215,62	1,02	216,31	1,02	-0,32%	0,00%
215,63	1,03	216,24	1,02	-0,28%	0,98%
215,62	1,02	216,24	1,02	-0,29%	0,00%
215,62	1,02	216,35	1,03	-0,34%	-0,97%
216,62	1,02	216,43	1,03	0,09%	-0,97%
216,97	1,03	216,46	1,03	0,24%	0,00%
216,03	1,01	216,58	1,03	-0,25%	-1,94%
217,15	1,02	216,86	1,03	0,13%	-0,97%
217,20	1,03	217,01	1,03	0,09%	0,00%

Fonte: Autor

Após a realização da medição de tensão e corrente, verifica-se que as leituras realizadas pelo protótipo estão bem próximas as do multimedidor de energia PM5110, apresentando um erro máximo de -0,34% para tensão e -1,94% para corrente, quando verificada a medição tanto com carga resistiva quanto com carga *RLC*. Assim, este item cumpriu o objetivo proposto.

4.3 MEDIÇÃO DE POTÊNCIAS

Dando sequência na validação dos parâmetros, será realizado neste item, a verificação da medição de potência ativa, aparente e reativa. Como as potências são obtidas através dos valores de tensão e corrente, estão sendo abordadas apenas neste momento, após a validação das medições.

Este item, 4.3, também está segmentado em dois momentos, um com medição de carga resistiva e outro com carga *RLC*. A Tabela 4.3 apresenta os valores da medição de carga resistiva:

Tabela 4.3 - Resultados de Potência Ativa (W), Aparente (VA) e Reativa (VAr) Carga Resistiva

Protótipo			Multimedidor de Energia PM5110			Erro W (%)	Erro VA (%)	Erro VAr (%)
W	VA	VAr	W	VA	VAr			
800,94	805,36	74,21	799,65	803,52	78,70	0,16%	0,23%	7,00%
802,24	805,84	76,09	799,96	803,85	79,00	0,29%	0,25%	-3,68%
801,05	805,74	76,83	800,46	804,34	78,90	0,07%	0,17%	10,05%
797,22	799,98	76,37	801,50	805,35	78,90	-0,53%	-0,67%	-15,88%
793,44	798,44	79,18	801,47	805,35	79,00	-1,00%	-0,86%	12,89%
801,11	805,37	82,69	801,62	805,51	79,10	-0,06%	-0,02%	4,54%
802,02	805,97	79,65	801,64	805,53	79,10	0,05%	0,05%	0,70%
795,73	799,16	73,99	801,43	805,32	79,10	-0,71%	-0,76%	-6,46%
803,91	807,87	79,88	801,34	805,22	78,90	0,32%	0,33%	1,24%
794,91	799,39	74,55	801,39	805,28	79,10	-0,81%	-0,73%	6,89%

Fonte: Autor

Já a Tabela 4.4 apresenta os valores de medição com carga *RLC*:

Tabela 4.4 - Resultados de Potência Ativa (W), Aparente (VA) e Reativa (VAr) Carga *RLC*

Protótipo			Multimedidor de Energia PM5110			Erro W (%)	Erro VA (%)	Erro VAr (%)
W	VA	VAr	W	VA	VAr			
216,60	225,20	50,80	221,14	225,03	41,70	-2,05%	0,08%	21,82%
215,40	223,10	47,20	220,73	224,60	41,52	-2,41%	-0,67%	13,68%
216,50	225,50	51,90	221,41	225,30	41,66	-2,22%	0,09%	24,58%
214,70	223,20	49,90	221,58	225,48	41,74	-3,10%	-1,01%	19,55%
216,20	224,70	50,10	221,38	225,26	41,66	-2,34%	-0,25%	20,26%
215,80	224,40	50,30	221,66	225,55	41,73	-2,64%	-0,51%	20,54%
217,10	225,00	48,30	221,64	225,51	41,61	-2,05%	-0,23%	16,08%
215,00	222,90	47,70	221,25	225,12	41,57	-2,82%	-0,99%	14,75%
213,10	220,80	47,00	219,31	223,05	40,69	-2,83%	-1,01%	15,51%
211,50	218,80	45,00	216,80	220,37	39,52	-2,44%	-0,71%	13,87%

Fonte: Autor

Após a realização da medição de potência ativa, aparente e reativa, verifica-se que as leituras realizadas pelo protótipo estão bem próximas as do analisador de energia, apresentando um erro máximo de -3.10% para potência ativa, -1,01% para potência aparente, exceto para a medição de potência aparente, apresentando um erro máximo de 24,58%, quando verificada a medição tanto com carga resistiva quanto com carga *RLC*. Assim, este item cumpriu com o objetivo proposto, porém, com restrição, devido aos valores de potência reativa.

4.4 MEDIÇÃO DE FATOR DE POTÊNCIA

Para a obtenção do fator de potência, estão sendo utilizados os valores de potência ativa e aparente, desta forma, após a validação destes valores, faz-se necessário realizar a validação do fator de potência. A tabela 4.5 apresenta os valores obtidos da medição de fator de potência com carga resistiva e também com carga *RLC*:

Tabela 4.5 - Resultados de Fator de Potência Carga Resistiva e *RLC*

Carga Resistiva			Carga <i>RLC</i>		
Protótipo	Multimedidor de Energia PM5110	Erro Fator de Pot. (%)	Protótipo	Multimedidor de Energia PM5110	Erro Fator de Pot. (%)
0,99	0,99	0,00%	0,96	0,98	-2,04%
0,99	0,99	0,00%	0,97	0,98	-1,02%
1,00	0,99	1,01%	0,96	0,98	-2,04%
1,00	0,99	1,01%	0,97	0,98	-1,02%
1,00	0,99	1,01%	0,97	0,98	-1,02%
0,99	0,99	0,00%	0,97	0,98	-1,02%
1,00	0,99	1,01%	0,96	0,98	-2,04%
1,00	0,99	1,01%	0,96	0,98	-2,04%
0,99	0,99	0,00%	0,96	0,98	-2,04%
1,00	0,99	1,01%	0,97	0,98	-1,02%

Fonte: Autor

Após a realização da medição de fator de potência, verifica-se que as leituras realizadas pelo protótipo estão bem próximas as do multimedidor de energia PM5110, apresentando um erro máximo de -2,04%, quando verificada a medição tanto com carga resistiva quanto com carga *RLC*. Assim, este item cumpriu o objetivo proposto.

4.5 MEDIÇÃO DE CONSUMO

Prosseguindo com os testes do protótipo, temos a medição de consumo, que é obtida através da medição de potência ativa e sendo apresentado em kWh, que é a unidade de medida utilizada para cobrança da energia elétrica dos consumidores. A tabela 4.6 apresenta os valores obtidos da medição de consumo com carga resistiva e também com carga *RLC*:

Tabela 4.6 - Resultados de Medição de Consumo Carga Resistiva e *RLC*

Carga Resistiva			Carga <i>RLC</i>		
Protótipo	Multimedidor de Energia PM5110	Erro Consumo (%)	Protótipo	Multimedidor de Energia PM5110	Erro Consumo (%)
9,88	10,00	-1,20%	10,12	10,00	1,20%
10,97	11,00	-0,27%	11,16	11,00	1,45%
11,86	12,00	-1,17%	12,22	12,00	1,83%
12,94	13,00	-0,46%	13,18	13,00	1,38%
13,89	14,00	-0,79%	14,20	14,00	1,43%
14,92	15,00	-0,53%	15,24	15,00	1,60%
15,81	16,00	-1,19%	16,24	16,00	1,50%
16,90	17,00	-0,59%	17,22	17,00	1,29%
17,84	18,00	-0,89%	18,22	18,00	1,22%
18,93	19,00	-0,37%	19,30	19,00	1,58%

Fonte: Autor

Após a realização da medição de consumo, verifica-se que as leituras realizadas pelo protótipo estão bem próximas as do multimedidor de energia PM5110, apresentando um erro máximo de 1,83%, quando verificada a medição tanto com carga resistiva quanto com carga *RLC*. Assim, este item cumpriu o objetivo proposto.

4.6 TESTE DOS SUPERVISÓRIO

O teste do supervisor se resume a verificação das funcionalidades descritas inicialmente neste projeto, assim, os itens descritos de *a* até *c* foram verificados com sucesso:

- a) Acesso ao supervisor por meio de usuário e senha de acesso;
- b) Acesso ao tela inicial e leitura das grandezas elétricas;
- c) Navegação pelos itens de medição e verificação da comunicação em tempo real.

A Figura 4.3 ilustra a tela inicial do supervisório, bem como a leitura das grandezas elétricas fornecidas pelo microcontrolador, já a tela gráfica do supervisório encontra-se no Anexo B deste projeto:

Figura 4.3 - Tela Inicial de Medição Supervisório

Watch list 

Medidor       

Medidor - Tensão	116,20V	23:08:29	   
 23:05:35 - Nível de Tensão Precário (Inferior)  			
 23:05:30 - Nível de Tensão Crítico (Inferior)  			
Medidor - Corrente	10,11A	23:08:29	   
Medidor - Potência Ativa	1172,00W	23:08:29	   
Medidor - Potência Aparente	1175,30VA	23:08:29	   
Medidor - Potência Reativa	76,40VAR	23:08:29	   
Medidor - Fator de Potência	1,00	23:08:29	   
Medidor - Consumo kWh	27,63kWh	23:08:29	   

Fonte: Autor

Findado os testes de acesso ao supervisório, verifica-se que o mesmo é funcional e disponibiliza os dados obtidos através do microcontrolador ao usuário final, conforme objetivo proposto. Para finalizar os testes do supervisório como um todo, resta realizar o teste que simula uma queda de tensão e a simulação de níveis de tensão fora do normal.

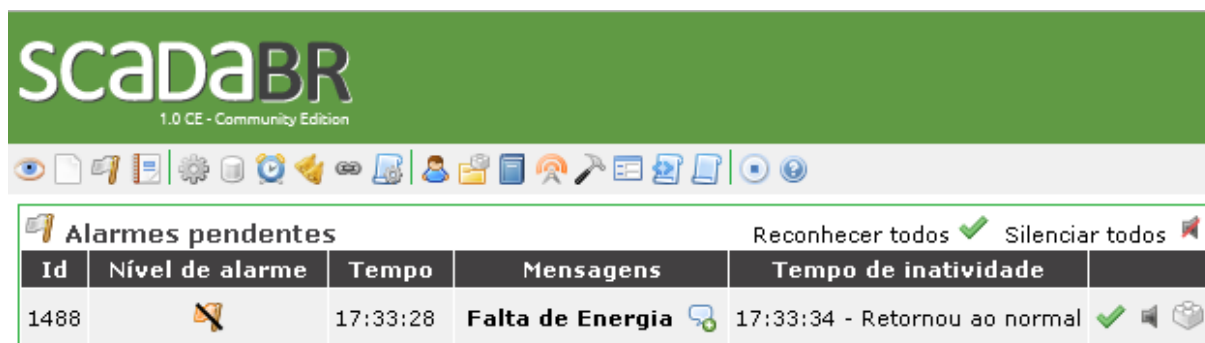
4.6.1 Simulação de Queda de Tensão

A simulação de queda de tensão consiste em informar ao usuário em que sua residência permaneceu com tensão abaixo de 50V por mais de 5 segundos, estes dados foram adotados como parâmetro para informação de queda de tensão. Caso ocorra algum momento sem energia, conforme parâmetros descritos anteriormente, o supervisório irá gerar um alarme com nível crítico, informando esta situação.

Para realização dos testes, utilizou-se um variador de tensão do tipo *dimmer* rotativo, tensão de alimentação 220V e saída variável de 0 a 220V, no qual alterou-se a tensão para valores abaixo de 50V. Passado o tempo definido no supervisório, um alarme é gerado informando ao usuário que ocorreu uma Falta de Energia, juntamente

com o horário do evento. A Figura 4.4 ilustra a tela de alarmes com o evento de falta de energia:

Figura 4.4 - Supervisório Alarme de Falta de Energia



Fonte: Autor

Verificou-se que, quando é retirada a tensão de alimentação, o protótipo apresenta uma lentidão no envio de informações, sendo os dados enviados com um atraso, ou seja, normalmente os dados são atualizados a cada um segundo pelo supervisório, neste momento são atualizados a cada dois segundos. Logo após a retomada da tensão, os valores voltam a ser enviados normalmente. Outro ponto é verificado quando é retirada a tensão de alimentação ou o valor de tensão permaneça menor que 100V, tensão limite para funcionamento da fonte de alimentação do protótipo, que é quando o sistema permanece em pleno funcionamento, registrando todos os valores, se existirem, porém, agora alimentado pelas baterias do protótipo. Esta condição garante que o sistema irá demonstrar ao usuário final, quando ocorram níveis de tensão inferiores ao normal e, até mesmo, uma falta de energia da concessionária. Desta forma, este item cumpriu com o objetivo, mesmo com o atraso na atualização dos valores, visto que esta situação não apresenta ônus ao usuário final.

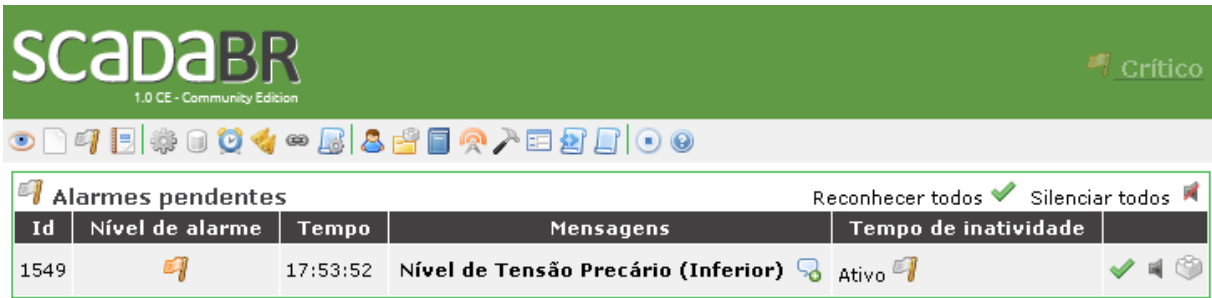
4.6.2 Simulação de Níveis de Tensão

A simulação dos níveis de tensão, visa atender o item 2.4.1 deste projeto, informando ao usuário quando os níveis de tensão de sua residência esteja fora dos níveis normais ou adequados. Desta forma, também com a utilização do variador de tensão do tipo *dimmer* rotativo, tensão de alimentação 220V e saída variável de 0 a

220V, no qual alterou-se a tensão para valores para simular os níveis de tensão precário e crítico, juntamente com um transformador potencial de 110/220V, para simular os valores de sobre tensão. Como estes indicadores apresentam valores tanto para 127V quanto para 220V, foram utilizados os valores para a tensão nominal de 220V, conforme descrito no item 3.5.2 deste projeto.

Semelhante ao teste de simulação de queda de tensão, o teste de níveis de tensão também irá gerar alarmes no supervisório, informando ao usuário que sua residência está com tensões fora dos valores adequados, juntamente com o horário do evento. A Figura 4.5 ilustra a tela de alarmes do supervisório com o evento de tensão em nível precário inferior:

Figura 4.5 - Supervisório Alarme de Nível Precário de Tensão Inferior



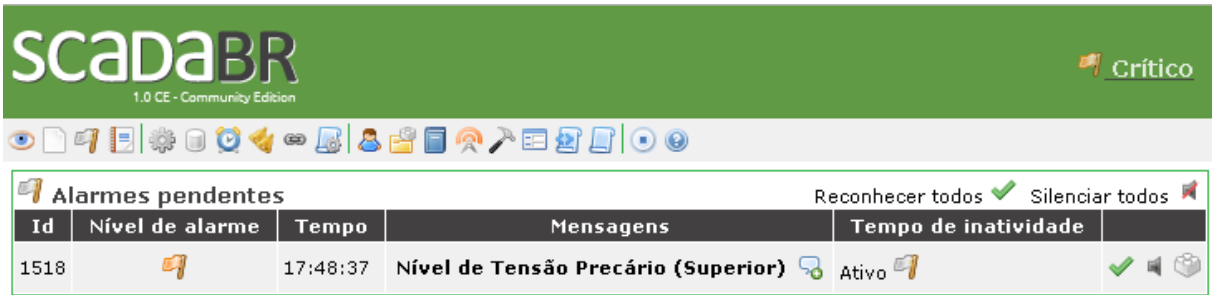
The screenshot shows the SCaDaBR 1.0 CE - Community Edition interface. At the top right, there is a 'Crítico' status indicator. Below the header, there is a table titled 'Alarmes pendentes'. The table has columns: Id, Nível de alarme, Tempo, Mensagens, Tempo de inatividade, and a final column with icons. The first row shows an active alarm with ID 1549, 'Nível de Tensão Precário (Inferior)', and a time of 17:53:52. The status is 'Ativo'.

Id	Nível de alarme	Tempo	Mensagens	Tempo de inatividade	
1549		17:53:52	Nível de Tensão Precário (Inferior)	Ativo	

Fonte: Autor

Já a Figura 4.6 ilustra a tela de alarmes do supervisório com o evento de tensão em nível precário superior:

Figura 4.6 - Supervisório Alarme de Nível Precário de Tensão Superior



The screenshot shows the SCaDaBR 1.0 CE - Community Edition interface. At the top right, there is a 'Crítico' status indicator. Below the header, there is a table titled 'Alarmes pendentes'. The table has columns: Id, Nível de alarme, Tempo, Mensagens, Tempo de inatividade, and a final column with icons. The first row shows an active alarm with ID 1518, 'Nível de Tensão Precário (Superior)', and a time of 17:48:37. The status is 'Ativo'.

Id	Nível de alarme	Tempo	Mensagens	Tempo de inatividade	
1518		17:48:37	Nível de Tensão Precário (Superior)	Ativo	

Fonte: Autor

Da mesma forma, a Figura 4.7 ilustra a tela de alarmes do supervisório com o evento de tensão em nível crítico inferior:

Figura 4.7 - Supervisório Alarme de Nível Crítico de Tensão Inferior

SCaDaBR

1.0 CE - Community Edition

Crítico

Alarmes pendentes

Reconhecer todos

Silenciar todos

Id	Nível de alarme	Tempo	Mensagens	Tempo de inatividade	
1539		17:52:22	Nível de Tensão Precário (Inferior)	Ativo	
1537		17:52:17	Nível de Tensão Crítico (Inferior)	Ativo	

Fonte: Autor

Já a Figura 4.8 ilustra a tela de alarmes do supervisório com o evento de tensão em nível crítico superior:

Figura 4.8 - Supervisório Alarme de Nível Crítico de Tensão Superior

SCaDaBR

1.0 CE - Community Edition

Alarmes pendentes

Reconhecer todos

Silenciar todos

Id	Nível de alarme	Tempo	Mensagens	Tempo de inatividade	
1505		17:45:30	Nível de Tensão Crítico (Superior)	Ativo	
1504		17:45:30	Nível de Tensão Precário (Superior)	Ativo	

Fonte: Autor

Em todos os outros momentos, o supervisório não apresentará alarmes referente ao nível de tensão, visto que os valores de tensão estarão enquadrados no parâmetro adequado. Para os demais níveis de tensão, foram confirmados a geração dos alarmes. Assim, este item cumpriu o objetivo proposto.

4.7 TESTES DE COMUNICAÇÃO

Por fim, resta realizar a validação de comunicação, tanto do supervisório, como do *Ethernet Shield*, sendo estes, desenvolvidas ao longo dos testes de medição e do supervisório, pois todos os dados coletados para realização do comparativo foram verificados através do supervisório. Assim, este item cumpriu o objetivo proposto.

4.8 VIABILIDADE DO PROTÓTIPO

Após a montagem e testes do protótipo, verificou-se que o mesmo apresenta um resultado geral que cumpriu com os objetivos definidos, visto que realizou medições de grandezas elétricas. Contudo, o protótipo não passou por um longo período de testes para que seja verificado sua funcionalidade ao longo do tempo.

No ponto de vista financeiro, o mesmo apresenta um custo de aproximadamente 25% de um medidor de porta de painel, por exemplo, o Multimedidor PM210MGCA, da marca *Schneider Electric*[®], que custa em torno de R\$ 1.000,00 reais, sendo que o PM210 apresenta várias características que superam o protótipo, sendo assim, este projeto é viável apenas para realização de estudos e não para comercialização.

Contudo, o desenvolvimento de medidores eletrônicos com microcontroladores não é descartada, apenas deve ser melhorada, a fim de obter confiabilidade no sistema de medição e implementação de demais funcionalidades.

4.9 PROPOSTA PARA TRABALHOS FUTUROS

Como proposta para trabalhos futuros, sugere-se o desenvolvimento de medidores de energia elétrica, utilizando microcontroladores com capacidade de processamento de dados simultâneos maior, desta forma, será possível o desenvolvimento de um analisador de energia e não somente um equipamento para leitura instantânea.

Outro ponto, é a melhora no *software* supervisor, implementando-se mais funcionalidades ao usuário, podendo este medidor ser implementado em sistema de automação residencial, onde o usuário poderá controlar o consumo de energia elétrica da residência à distância, por exemplo.

Outro grande desafio é a realização de medição trifásica, o que pode enriquecer o estudo e desenvolvimento de protótipo para medição de grandezas elétricas com microcontroladores.

5 CONCLUSÃO

O conceito de aquisição de dados de todos os tipos e níveis está sendo difundido pela humanidade, assim, este protótipo vai de encontro a esta tendência, disponibilizando ao usuário final uma nova maneira de verificar os dados presentes nas faturas de energia elétrica, acrescidos de informações que não constam nas faturas, bem como, monitorar a qualidade da energia elétrica de sua residência.

O *hardware* de processamento de dados do protótipo apresentou um desempenho aguardado, realizando a leitura dos dados de entrada e realizando os cálculos necessários para obtenção dos valores *RMS* de tensão e corrente, bem como a detecção da passagem por zero dos sinais de tensão e corrente, para que seja possível verificar o deslocamento dos sinais, sendo possível o cálculo de fator de potência da carga. Este ponto foi decisivo no projeto, pois, caso o processamento do microcontrolador *Arduino*® não fosse suficiente para tal, seria necessário substituir o hardware do protótipo, alterando completamente o projeto.

Os *hardwares* de medição de tensão e corrente também obtiveram um desempenho aguardado, possibilitando a leitura de tensão dentro de toda sua gama de tensão (0 a 250V) e de corrente (0 a 50A) com precisão, quando comparados ao Multimedidor de Energia PM5110, por meio de testes. Outro ponto positivo é que este transdutor de tensão é comercializado com todos os componentes necessários para deslocamento do zero da senoide de entrada para 2,5V em sua saída, ou seja, é um equipamento construído para trabalhar em conjunto com o microcontrolador *Arduino*®, facilitando a construção do protótipo. Já o transdutor de corrente não é comercializado com o circuito para interligação com o microcontrolador *Arduino*®, mas este fato não compromete a utilização deste equipamento, visto que é um circuito simples de ser construído.

Outro desafio demasiadamente crítico do projeto foi a comunicação entre o microcontrolador e o *software* supervisor, visto que *software* de programação do microcontrolador *Arduino*® não possui uma biblioteca específica para envio de informações aleatórias via rede *ethernet*, desta forma, foi necessário adaptar uma biblioteca de envio de informação de um sensor de temperatura e umidade para realizar a comunicação. Caso o envio dos dados via *ethernet* não fosse possível, seria necessário alterar o tipo de comunicação do projeto para serial, reduzindo a mobilidade do sistema, visto que com a comunicação *ethernet*, o supervisor não

necessitaria permanecer junto ao protótipo, apenas compartilhar da mesma rede *ethernet*, ao contrário da comunicação serial, contudo, o *hardware* de comunicação teve um funcionamento conforme o esperado.

A biblioteca *EmonLib.h*, utilizada no microcontrolador *Arduino*® foi de suma importância para o desenvolvimento do protótipo, visto que a medição e tratamento dos dados para obtenção das grandezas elétricas é de elevado grau de complexidade, assim, com a utilização da biblioteca disponível, foi possível realizar a obtenção das grandezas elétricas de forma assertiva.

Os testes comparativos com carga resistiva e *RLC* também foram essenciais para o desenvolvimento final do protótipo, pois foram realizados vários testes até obter um ajuste preciso de leitura dos transdutores de tensão e corrente. Os testes com carga resistiva foram demasiadamente mais rápidos e menos trabalhosos, pois o ajuste necessário, a princípio, apresentou-se simples. Já os testes com carga *RLC* exigiram maiores ajustes e teste mais prolongados, a fim de obter os valores de potências e fator de potência reais, conforme a carga aplicada. Ao fim dos ajustes com cada carga, verificou-se novamente os ajustes, aplicando as cargas resistiva e *RLC*. Mesmo com os últimos ajustes realizados, alguns parâmetros se demonstraram assertivos para uma carga e para outra não, desta forma, realizado novos ajustes, visando a melhor representação das grandezas medidas para ambos os tipos de carga.

Ao final dos ajustes, obteve-se um resultado esperado quanto ao erro de medição, visto que as leituras de tensão, corrente e potência aparente apresentaram os valores mais próximos aos reais, já a potência ativa, reativa e fator de potência apresentaram valores medidos demasiadamente mais distantes da realizada quando verificados com carga *RLC*, principalmente a potência reativa, porém, ainda representando aceitavelmente as grandezas elétricas medidas. Os valores das medições com carga resistiva apresentaram um erro pequeno para todas as grandezas, exceto para potência reativa. Acredita-se que esta diferença de medição está na detecção da passagem por zero da senoide de tensão e corrente, visto que os transdutores apresentam um atraso na medição, contribuindo para esta situação. Este erro não acontece tão significativamente com carga resistiva, pois, como não ocorre defasagem na senoide da tensão e da corrente, a leitura apresenta valores mais exatos. A biblioteca *EmonLib.h* possui dispositivos para realizar a calibração deste atraso, entretanto, mesmo realizando o melhor ajuste possível para

representação de ambas as cargas, o erro permaneceu maior que as grandezas de tensão, corrente e potência aparente. A incapacidade de medição do *Arduino*® UNO devido à velocidade de processamento do microcontrolador está descartada, pois caso essa velocidade estivesse impactando na leitura, os valores de potência ativa e fator de potência com carga resistiva também apresentariam um erro maior quando comparados com os valores de tensão, corrente e potência aparente.

O *software* supervisor apresentou um desempenho essencial no projeto, sendo este uma plataforma *open-source*, apresentando diversas funcionalidades, desde a programação da fonte de dados e pontos de dados até a tela gráfica para visualização do usuário final. Outro ponto positivo no *software* é que em cada item possui um local de ajuda, onde estão descritas todas as funcionalidades daquele item, desta forma, auxiliam no desenvolvimento do sistema supervisor como um todo. Para este projeto, o *software* foi subutilizado, visto que várias funcionalidades não foram empregadas, pois não estavam contempladas no escopo inicial, mas que podem ser utilizadas em trabalhos futuros, como o envio de alarmes críticos por e-mail, possibilidade de acesso remoto do supervisor, possibilitando o usuário monitorar o sistema de qualquer lugar, entre outros. Mais uma disponibilidade é o envio de informações do supervisor para o microcontrolador, possibilitando o acionamento de cargas via supervisor, como por exemplo, ligar uma lâmpada.

Dado o exposto, de forma geral, entende-se que o projeto teve um resultado esperado, pois cumpriu todos os objetivos definidos inicialmente, levando informação de leitura das grandezas elétricas e melhorando as informações contidas nos indicadores das faturas de energia elétrica, ao usuário final. Percebe-se também, que a aplicação de sistemas *open-source* é crescente e que deve ser cada vez mais desenvolvida e utilizada, possibilitando ao usuário, desfrutar de sistemas extremamente desenvolvidos e com custo de aquisição demasiadamente baixo, quando comparados com sistemas de grandes marcas.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: Informação e documentação - Trabalhos acadêmicos - Apresentação. Rio de Janeiro: Abnt Editora, 2011. 11 p.

AGÊNCIA NACIONAL DE ENERGIA ELÉTRICA. **PRODIST MÓDULO 8**: Qualidade da Energia Elétrica. 2018. Disponível em: http://www.aneel.gov.br/documents/656827/14866914/Módulo_8-Revisão_10/2f7cb862-e9d7-3295-729a-b619ac6baab9. Acesso em: 18 maio 2018.

ALEXANDER, Charles K.; SADIKU, Matthew N. O. **Fundamentos de Circuitos Elétricos**. 5. ed. Porto Alegre: AMGH, 2013.

BALBINOT, Alexandre; BRUSAMARELLO, Valner João. **Instrumentação e fundamentos de medidas vol 1**. 2. ed. Rio de Janeiro: LTC, 2011.

CRISPIM, José. **Computador vs Máquina e Hardware vs Software**. 2013. Disponível em: https://www.jose-crispim.pt/artigos/conceitos/conc_art/02_comp_maq_hard_soft.html. Acesso em: 12 maio 2018.

GOMES, José Ricardo Ribeiro. **Automação Industrial com recurso a ferramentas “Open Source”**. 2014. 135 f. Dissertação (Mestrado) - Curso de Engenharia Eletrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2014.

HUDSON, Glyn; REED, Paul; LEA, Trystan. **Emoncms: EmonLib Calibration Theory**. 2017. Disponível em: <https://github.com/openenergymonitor/learn/blob/master/view/electricity-monitoring/ctac/emonlib-calibration-theory.md#voltage-sensor>. Acesso em: 11 maio 2018.

JOHNSON, David E.; HILBURN, John L.; JOHNSON, Johnny R. **Fundamentos de Análise de Circuitos Elétricos**. 4. ed. Rio de Janeiro: Prentice-hall do Brasil Ltda, 1994.

LIMA, Elon Lages. **Meu Professor de Matemática: e Outras Histórias**. 6. ed. Rio de Janeiro: Sbm, 2012. 206 p.

MAMEDE FILHO, João. **Manual de Equipamentos Elétricos**. 3. ed. Rio de Janeiro: LTC, 2005.

MONK, Simon. **30 Arduino® Projects: for de evil Genius**. New York: Mcgraw-hill Companies, 2010.

NILSSON, James W.; REIDEL, Susan A. **Circuitos Elétricos**. 8. ed. São Paulo: Pearson Prenticehall, 2009. 578 p

SOUZA, David José de. **Desbravando o PIC**: Ampliado e Atualizado para PIC16F628A. 8. ed. São Paulo: Érica, 2003.

THOMSEN, Adilson. **Como comunicar com o Arduino® Ethernet Shield W5100**. 2014. Disponível em: <<https://www.filipeflop.com/blog/tutorial-ethernet-shield-w5100/>>. Acesso em: 14 abr. 2018.

THOMSEN, Adilson. **Medidor de corrente não invasivo com Arduino®**. 2015. Disponível em: <<https://www.filipeflop.com/blog/medidor-de-corrente-sct013-com-arduino/>>. Acesso em: 15 abr. 2018.

ANEXO A – CÓDIGO FONTE MICROCONTROLADOR ARDUINO®

// PROJETO E DESENVOLVIMENTO DE UM MEDIDOR DE GRANDEZAS
// ELÉTRICAS PARA ANÁLISE DA ENERGIA ELÉTRICA EM RESIDÊNCIAS.

// CENTRO UNIVERSITÁRIO ASSIS GURGACZ

// ALUNO: THIAGO FERNANDES DA COSTA

// PROFESSOR ORIENTADOR: EDERSON ZANCHET

```
#include <SPI.h>      //Inclusão Biblioteca Comunicação Serial
#include <Ethernet.h>  //Inclusão Biblioteca Comunicação Ethernet
#include <Modbus.h>    //Inclusão Biblioteca Comunicação Modbus
#include <ModbusIP.h>  //Inclusão Biblioteca Comunicação ModbusIP
#include "EmonLib.h"   //Inclusão Biblioteca de Medição
#include <EEPROM.h>    //Inclusão Biblioteca de Gravação kWh
EnergyMonitor emon1;  //Criação de uma Instância de Medição
```

//Criação do Offset de comunicação para o ScadaBR

```
const float Ireg_Vrms = 0;    //Tensão
const float Ireg_Irms = 1;    //Corrente
const float Ireg_W = 2;       //Potência Ativa
const float Ireg_VA = 3;      //Potência Aparente
const float Ireg_VAR = 4;     //Potência Reativa
const float Ireg_FP = 5;      //Fator de Potência
const float Ireg_kWhtotal = 6; //Consumo em kWh
float amostra_Irms;
```

//Criação de variáveis para receber o consumo em kWh

```
float kWh;
float kWhtotal;
float armazena_kWhtotal;
```

//Configuração do Armazenamento do Consumo em kWh

```

void EEPROMWriteInt(int address, int value); //2 Bytes
int EEPROMReadInt(int address);

//Configuração do Ethernet Shield
byte mac[] = {0xDE,0xAD,0xBE,0xEF,0xFE,0xED};
byte ip[] = {192,168,1,195};
byte gateway[] = {192,168,1,1};
byte subnet[] = {255,255,255,0};
//Configuração da Comunicação para o ScadaBR
ModbusIP mb;

void setup() {

//Habilitando a comunicação Serial e Ethernet
Serial.begin(9600);
Ethernet.begin(mac, ip, gateway, subnet);

//Define a leitura da EEPROM na posição 1 e 2 da memória
armazena_kWhTotal = EEPROMReadInt(1);

//Definição dos pinos de entrada de tensão e corrente
// Tensão: pino de entrada, calibração, mudança de fase
emon1.voltage(1, 231.26, 0);
// Corrente: pino de entrada, calibração
emon1.current(2, 48.7);

//Habilitando a função de Registradores de Entrada para a
//comunicação com o ScadaBR para cada variável (cada grandeza).
mb.addIreg(Ireg_Vrms);
mb.addIreg(Ireg_Irms);
mb.addIreg(Ireg_W);
mb.addIreg(Ireg_VA);
mb.addIreg(Ireg_VAR);
mb.addIreg(Ireg_FP);

```

```

    mb.addIreg(Ireg_kWhtotal);
}

void loop() {

    mb.task();

    //Função realiza o cálculo das grandezas da biblioteca Emonlib.h
    //Número de cruzamento por zero, tempo limite
    emon1.calcVI(20,2000);

    //Extrai os valores de Tensão RMS da variável
    float Vrms    = emon1.Vrms;
    //Extrai os valores de Corrente RMS da variável
    float amostra_Irms = emon1.Irms;

    //Exclui os valores de corrente, potências e fator de potência
    //quando os valores de corrente são menores que 200mA
    if(amostra_Irms <= 0.2){
        float Irms    = 0;
        float VA      = 0;
        float FP      = 0;
        float W       = 0;
        float VAR     = 0;

    //Envio dos valores das variáveis para o ScadaBR
    mb.Ireg(Ireg_Vrms, Vrms*100);
    mb.Ireg(Ireg_Irms, Irms*100);
    mb.Ireg(Ireg_W, W);
    mb.Ireg(Ireg_VA, VA);
    mb.Ireg(Ireg_VAR, VAR);
    mb.Ireg(Ireg_FP, FP);
    mb.Ireg(Ireg_kWhtotal, armazena_kWhtotal*100);

```

```

//Caso os valores sejam maiores que 200mA realiza os cálculos
}else{
//Extrai os valores de Corrente RMS da variável
float Irms    = emon1.Irms;
//Extrai os valores de Potência Aparente da variável
float VA      = emon1.apparentPower;
//Extrai os valores de Fator de Potência da variável
float FP      = emon1.powerFactor;
//Extrai os valores de Potência Ativa da variável
float W       = emon1.realPower;
//Extrai os valores de Potência Reativa da variável
float VAR     = sqrt(sq(VA) - sq(W));
//Realiza o cálculo de Consumo em kWh
float kWh     = (W / 18400);
//Realiza o somatório do Consumo em kWh
kWht         = kWh + armazena_kWhtotal;
//Transfere o valor do Consumo em kWh para a variável
float kWhtotal = kWht;
armazena_kWhtotal = kWhtotal;
//Escreve a variável na memória EEPROM
EEPROMWriteInt(1, armazena_kWhtotal);

//Envio dos valores das variáveis para o ScadaBR
mb.Ireg(Ireg_Vrms, Vrms*100);
mb.Ireg(Ireg_Irms, Irms*100);
mb.Ireg(Ireg_W, W*10);
mb.Ireg(Ireg_VA, VA*10);
mb.Ireg(Ireg_VAR, VAR*10);
mb.Ireg(Ireg_FP, FP*1000);
mb.Ireg(Ireg_kWhtotal, armazena_kWhtotal*100);

}}
//Repetição para definição de leitura e gravação de
//dois espaços da memória EEPROM

```

```
void EEPROMWriteInt(int address, int value) {  
    byte hiByte = highByte(value);  
    byte loByte = lowByte(value);  
  
    EEPROM.write(address, hiByte);  
    EEPROM.write(address + 1, loByte);  
}
```

```
int EEPROMReadInt(int address) {  
    byte hiByte = EEPROM.read(address);  
    byte loByte = EEPROM.read(address + 1);  
  
    return word(hiByte, loByte);  
}
```

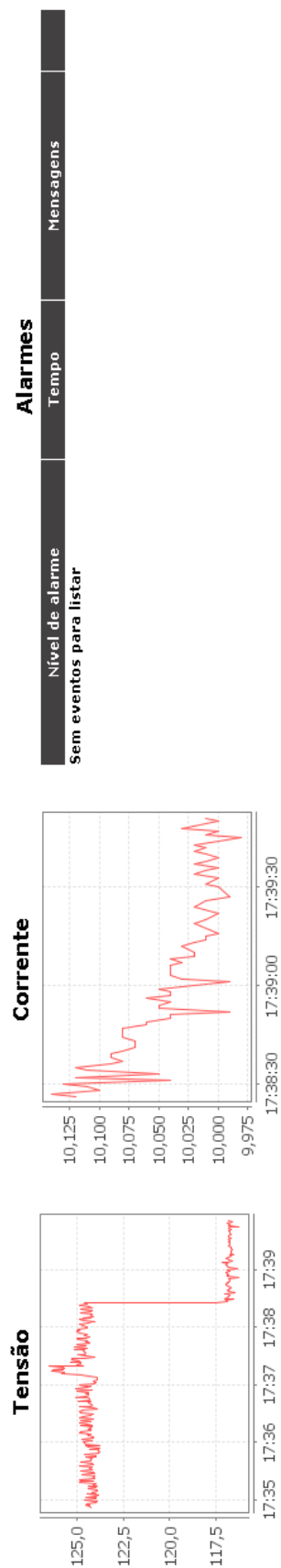
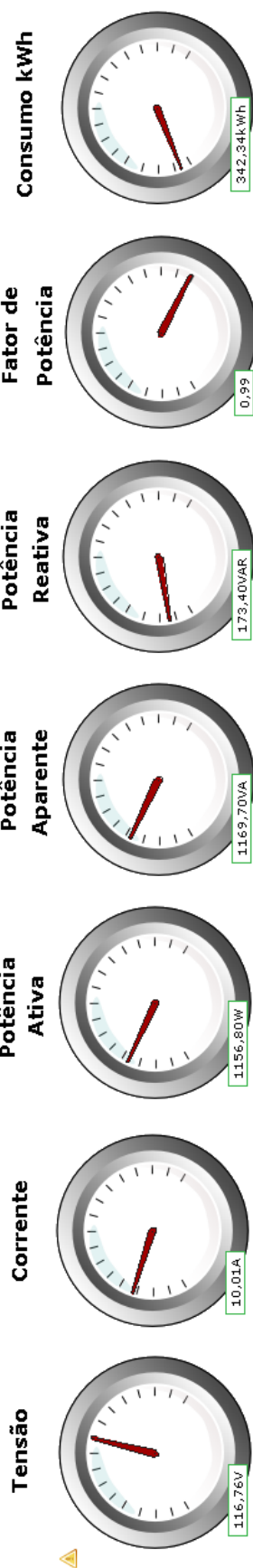
ANEXO B – TELA GRÁFICA DO SUPERVISÓRIO



PROJETO E DESENVOLVIMENTO DE UM MEDIDOR DE GRANDEZAS ELÉTRICAS, PARA ANÁLISE DA QUALIDADE DA ENERGIA ELÉTRICA EM RESIDÊNCIAS.



THIAGO FERNANDES DA COSTA



APÊNDICE A – CÓDIGO FONTE BIBLIOTECA EMONLIB.H

```

/*
Emon.cpp - Biblioteca para openenergymonitor
Criado por Trystan Lea, 27 de abril de 2010
GNU GPL
modificado para usar até 12 bits de resolução ADC (ex. Arduino Due)
por boredman@boredomprojects.net 26.12.2013
Filtro Low Pass para remoção de offset substitui o filtro HP 1/1/2015 – RW
*/

// Proboscide99 10/08/2016 - Configurações ADMUX adicionadas para ATmega1284
// e 1284P (644 / 644P também, mas não testadas) na função
// #include "WProgram.h" não comentar para uso em versões mais antigas do Arduino
// IDE

#include "EmonLib.h"
#ifdef ARDUINO && ARDUINO >= 100
#include "Arduino.h"
#else
#include "WProgram.h"
#endif

//-----
// Define os pinos a serem usados para sensores de tensão e corrente
//-----

void EnergyMonitor::voltage(unsigned int _inPinV, double _VCAL, double
_PHASECAL)
{
inPinV = _inPinV;
VCAL = _VCAL;
PHASECAL = _PHASECAL;
offsetV = ADC_COUNTS>>1;
}

void EnergyMonitor::current(unsigned int _inPinI, double _ICAL)
{

```

```

inPinI = _inPinI;
ICAL = _ICAL;
offsetI = ADC_COUNTS>>1;
}
//-----
// Define os pinos a serem usados para sensores de tensão e corrente baseados no
// mapa do pino emontx
//-----
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;
}
//-----
// procedimento emon_calc
// Calcula realPower, apparentPower, powerFactor, Vrms, Irms, incremento de kWh
// A partir de uma janela de amostra da tensão e corrente AC principais.
// O tamanho da janela Sample é definido pelo número de meias-ondas ou
// cruzamentos que escolhemos medir.
//-----
void EnergyMonitor::calcVI(unsigned int crossings, unsigned int timeout)
{
#ifdef 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) Espera que a forma de onda esteja próxima da parte 'zero' (mid-scale adc) na
curva sin.
//-----
boolean st=false; // um indicador para sair do loop while
unsigned long start = millis();// millis () - start garante que ele não fique preso no loop
se houver um erro.
while(st==false) // o loop while ...
{
startV = analogRead(inPinV); // usando a forma de onda de tensão
if ((startV < (ADC_COUNTS*0.55)) && (startV > (ADC_COUNTS*0.45))) st=true;
// verifique seu intervalo
if ((millis()-start)>timeout) st = true;
}
//-----
// 2) Malha de medição principal
//-----
start = millis();
while ((crossCount < crossings) && ((millis()-start)<timeout))
{
numberOfSamples++; // Contar o número de vezes em loop.
lastFilteredV = filteredV; // Usado para compensação de atraso / fase
//-----
// A) Ler em amostras brutas de tensão e corrente
//-----
sampleV = analogRead(inPinV); //Read in raw voltage signal
sampleI = analogRead(inPinI); //Read in raw current signal
//-----

```

```

// B) Aplique filtros de passa baixa digital para extrair o deslocamento de 2,5 V ou 1,65
V dc,
// então subtrai isso - o sinal está agora centrado em 0 contagens.
//-----
offsetV = offsetV + ((sampleV-offsetV)/1024);
filteredV = sampleV - offsetV;
offsetI = offsetI + ((sampleI-offsetI)/1024);
filteredI = sampleI - offsetI;
//-----
// C) Tensão do método raiz quadrada média
//-----
sqV= filteredV * filteredV; //1) square voltage values
sumV += sqV; //2) sum
//-----
// D) Método atual de raiz quadrada
//-----
sqI = filteredI * filteredI; //1) square current values
sumI += sqI; //2) sum
//-----
// E) Calibração de fase
//-----
phaseShiftedV = lastFilteredV + PHASECAL * (filteredV - lastFilteredV);
//-----
// F) Cálculo de Potência instantânea
//-----
instP = phaseShiftedV * filteredI; // Potência Instantânea
sumP +=instP; // Soma
//-----
// G) Encontre o número de vezes que a tensão cruzou a tensão inicial
// - a cada 2 cruzes teremos amostrados 1 comprimento de onda
// - então este método nos permite amostrar um número inteiro de meio comprimento
de onda que aumenta a precisão
//-----
lastVCross = checkVCross;

```

```

if (sampleV > startV) checkVCross = true;
else checkVCross = false;
if (numberOfSamples==1) lastVCross = checkVCross;
if (lastVCross != checkVCross) crossCount++;
}
//-----
3) Cálculos de pós-loop
//-----
// Cálculo da raiz da média da tensão e da corrente ao quadrado (rms)
// Coeficientes de calibração aplicados.

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);

// Valores de potência de cálculo
realPower = V_RATIO * I_RATIO * sumP / numberOfSamples;
apparentPower = Vrms * Irms;
powerFactor=realPower / apparentPower;

// Redefinir acumuladores
sumV = 0;
sumI = 0;
sumP = 0;
//-----
}
//-----

double EnergyMonitor::calcIrms(unsigned int Number_of_Samples)
{
#ifdef emonTxV3
int SupplyVoltage=3300;
#else

```

```

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

    // Filtro low-pass digital extrai o offset de 2,5 V ou 1,65 V dc,
    // então subtrai isso - o sinal está agora centrado em 0 contagens.

    offsetI = (offsetI + (sampleI - offsetI) / 1024);
    filteredI = sampleI - offsetI;

    // Corrente do método da raiz quadrada média
    // 1) valores atuais quadrados
    sqI = filteredI * filteredI;
    // 2) sum
    sumI += sqI;
}
double I_RATIO = ICAL * ((SupplyVoltage / 1000.0) / (ADC_COUNTS));
Irms = I_RATIO * sqrt(sumI / Number_of_Samples);

// Redefinir acumuladores
sumI = 0;
//-----
return Irms;
}

void EnergyMonitor::serialprint()
{
    Serial.print(realPower);
    Serial.print(' ');
    Serial.print(apparentPower);
    Serial.print(' ');
    Serial.print(Vrms);
    Serial.print(' ');

```

```

Serial.print(lrms);
Serial.print(' ');
Serial.print(powerFactor);
Serial.println(' ');
delay(100);
}

```

// graças a <http://hacking.majenko.co.uk/making-accurate-adc-readings-on-arduino>
 // e Jérôme que nos alertaram para <http://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/>

```

long EnergyMonitor::readVcc() {
long result;
// não usado no emonTx V3 - como o Vcc é sempre 3.3V - elimina o erro do bandgap
e a necessidade de calibração http://harizanov.com/2013/09

```

```

#if defined(__AVR_ATmega168__) || defined(__AVR_ATmega328__) || defined(
(__AVR_ATmega328P__)
ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
#elif defined(__AVR_ATmega644__) || defined(__AVR_ATmega644P__) ||
defined(__AVR_ATmega1284__) || defined(__AVR_ATmega1284P__)
ADMUX = _BV(REFS0) | _BV(MUX4) | _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); // Sem isso, a função sempre retorna -1 no ATmega2560
http://openenergymonitor.org/emon/node/
#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); // Aguarde a Vref liquidar
ADCSRA |= _BV(ADSC); // Convert
while (bit_is_set(ADCSRA,ADSC));
result = ADCL;
result |= ADCH<<8;
result = READVCC_CALIBRATION_CONST / result; // 1100mV * 1024 ADC etapas
http://openenergymonitor.org/emon/node/1186

return result;
#elif defined(__arm__)
return (3300); //Arduino Due
#else
return (3300); // Acho que outras arquiteturas não suportadas estarão executando um
3.3V!
#endif
}

```