



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DANIEL PABLO DANTAS DIÓGENES

**DESENVOLVIMENTO DE UM *SMART PLUG* PARA CONTROLE E  
MONITORAMENTO DE APARELHOS ELÉTRICOS UTILIZANDO O CONCEITO  
DE INTERNET DAS COISAS**

MOSSORÓ-RN

2019

DANIEL PABLO DANTAS DIÓGENES

**DESENVOLVIMENTO DE UM *SMART PLUG* PARA CONTROLE E  
MONITORAMENTO DE APARELHOS ELÉTRICOS UTILIZANDO O CONCEITO  
DE INTERNET DAS COISAS**

Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal Rural do Semi-Árido, como requisito para obtenção do título de Mestre em Engenharia Elétrica.

Linha de Pesquisa: Sistemas de Controle e Automação.

Orientadora: Prof.<sup>a</sup>. Dra. Danielle Simone da Silva Casillo - UFERSA.

MOSSORÓ-RN

2019

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

D591d Diógenes, Daniel Pablo Dantas.

Desenvolvimento de um *Smart Plug* para Controle e Monitoramento de Aparelhos Elétricos Utilizando o Conceito de Internet das Coisas / Daniel Pablo Dantas Diógenes. – 2019.

114 f. : il.

Orientadora: Danielle Simone da Silva Casillo.  
Dissertação (Mestrado) – Universidade Federal Rural do Semi-Árido, Programa de Pós-Graduação em Engenharia Elétrica, 2019.

1. Smart Plug. 2. Internet das Coisas. 3. Medidor de Energia Residencial. I. Casillo, Danielle Simone da Silva, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

DANIEL PABLO DANTAS DIÓGENES

**DESENVOLVIMENTO DE UM *SMART PLUG* PARA CONTROLE E  
MONITORAMENTO DE APARELHOS ELÉTRICOS UTILIZANDO O CONCEITO  
DE INTERNET DAS COISAS**

Dissertação de mestrado acadêmico  
apresentada ao Programa de Pós-Graduação em  
Engenharia Elétrica da Universidade Federal  
Rural do Semi-Árido como requisito para  
obtenção do título de Mestre em Engenharia  
Elétrica.

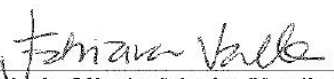
Linha de Pesquisa: Sistemas de  
Controle e Automação.


Defendida em: 18 / 02 / 2019.

**BANCA EXAMINADORA**

  
\_\_\_\_\_  
Profa. Dra. Danielle Simone da Silva Casillo – UFERSA  
Presidente

  
\_\_\_\_\_  
Prof. Dr. Alexandre Magnus Fernandes Guimarães – UFRN  
Membro Examinador Externo

  
\_\_\_\_\_  
Profa. Dra Fabiana Karla de Oliveira Martins Varella Guerra – UFERSA  
Membro Examinador Interno

  
\_\_\_\_\_  
Prof. Dr. Marcelo Roberto Bastos Guerra Vale – UFERSA  
Membro Examinador Interno

## EPÍGRAFE

*“O que passou, passou, mas o que passou luzindo, resplandecerá para sempre.”*

*Johann Goethe.*

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais Maria do Socorro e José Diógenes e meu irmão Diêgo, entre tantos motivos, pela formação moral, educação, carinho, apoio e dedicação ao longo de todos os anos, assim como auxílio em momentos cruciais desta qualificação, dando toda ajuda possível para a obtenção do meu sucesso.

A minha namorada Ana Emanuelle, pela paciência, companheirismo, ajuda e carinho repassados durante o tempo em que este trabalho foi desenvolvido, sempre me incentivando e dando forças para continuar nos momentos mais difíceis.

Aos meus familiares, que sempre estiveram dispostos a ajudar no que fosse possível e sempre demonstraram apoio as minhas decisões, em especial minhas tias e tios e meus avós Rita (*in memorian*) e Manoel, que além de participarem do meu crescimento pessoal sempre demonstraram orgulho pela pessoa que me tornei.

A minha orientadora Danielle Simone da Silva Casillo, por ter me acompanhado nestes dois anos de mestrado, sempre me estimulando e acreditando neste trabalho, disposta a ajudar sempre que necessário, sua paciência com minhas indecisões durante a pesquisa, dedicação e capacidade intelectual são características que sempre tentarei alcançar.

Aos meus professores do mestrado, Idalmir, Adriano Aron e Leonardo Casillo. Através de seus ensinamentos adquiri novos conhecimentos, tanto didáticos quanto técnicos. Aos meus colegas e amigos de pós-graduação, Thalita, Luan, Amanda, Josiran, Ana Luíza, Felliipe Bastos, Wigor, Rafael Alex e Érick, pela companhia no dia a dia e pelas diversas conversas ao longo destes 2 anos de curso.

## RESUMO

A melhoria na gestão dos recursos energéticos depende da quantidade de informações que são repassadas ao usuário, sendo de grande importância as medidas de potência, consumo e custo financeiro. Obrigatoriamente, as residências são dotadas de medidores de energia, no entanto estes pertencem a concessionária local, de modo que o consumidor não tem acesso aos dados medidos, apenas o consumo e faturamento ao final de cada mês. O controle de cargas se torna mais complicado, uma vez que já existem equipamentos com tal função, mas que não possuem a função de leitura de dados ou possuem uma instalação complexa. Considerando a forte tendência no uso de aplicativos e *smartphones*, e a própria expansão do conceito de internet das coisas, um equipamento que seja capaz de monitorar e controlar outros elementos, bem como enviar suas informações a internet torna-se uma necessidade ao consumidor. Diante disso, este trabalho tem como objetivo, desenvolver um plugue inteligente (*smart plug*) com capacidade de leitura e controle de dispositivos elétricos conectados a ele. O projeto foi motivado com o objetivo de facilitar o acesso as informações elétricas de qualquer eletrodoméstico conectado em tomadas monofásicas, bem como, permitir o controle da carga através de temporização ou acionamento manual. A proposta do *smart plug* foi realizada utilizando componentes de precisão, utilizados em equipamentos de uso comercial, como transformadores de corrente, transformadores de potencial e o circuito integrado de medição ADE7753, visando a uma boa precisão do produto final. As informações obtidas da rede elétrica e os comandos de acionamento e desligamento de equipamentos foram repassadas ao cliente em tempo real através de um aplicativo dedicado a esta finalidade, apresentando três ambientes de atividade, voltados para leitura de dados e comandos remotos. A comunicação foi realizada utilizando o protocolo MQTT em conjunto com o módulo Wi-Fi ESP8266-01. Os resultados dos testes mostraram que o *smart plug* apresentou um erro global de 2%, estando com um bom nível de exatidão se comparado a equipamentos comerciais, a comunicação através de aplicativo se mostrou efetiva, permitindo o acesso as informações através do celular. O controle de cargas e a temporização também se mostraram possíveis, permitindo assim, temporizar ou acionar um determinado equipamento de forma remota.

**Palavras-chaves:** *Smart Plug*, Internet das Coisas, Medidor de Energia Residencial.

## ABSTRACT

The improvement in the management of energy resources depends on the amount of information that is passed on to the user, being of great importance the measures of power, consumption and financial cost. The residences are equipped with energy meters, however these belong to the local concessionaire, so that the consumer does not have access to the measured data, only consumption and billing at the end of each month. The control of loads becomes more complicated, since there already exist devices with such function, but that do not have the function of reading of data or have a complex installation. Considering the growing trend in the use of applications and smartphones, and the very expansion in the concept of the internet of things, an equipment that is able to monitor and control other elements and send their information to the internet becomes a necessity to the consumer. The main objective of this work is to develop a smart plug with the ability to read and control electrical devices connected to it. The project was motivated with the objective of facilitating access to the electrical information of any appliance connected to single-phase sockets and to allow the control of the load through timing or manual activation. The smart plug proposal was made using precision components used in commercial equipment such as current transformers, potential transformers and the ADE7753 measurement integrated circuit, aiming at a good precision of the final product. The information obtained from the power grid and the equipment activation and shutdown commands were passed on to the customer in real time through an application dedicated to this purpose, presenting three activity environments, aimed at reading data and remote commands. The communication was performed using the MQTT protocol in conjunction with the Wi-Fi module ESP8266-01. The results of the tests showed that the smart plug presented an overall error of 2%, being with a good level of accuracy compared to commercial equipment, communication through the application was effective, allowing access to the information through the mobile phone. Load control and timing have also proved possible, allowing you to remotely trigger or trigger a specific device.

**Keywords:** Smart Plug, Internet of Things, Residential Energy Meter.



## LISTA DE FIGURAS

<b>Figura 1</b> – Medidor de Gardiner (a) e medidor de Fuller (b). .....	24
<b>Figura 2</b> – Diagrama de blocos de um multímetro digital.....	25
<b>Figura 3</b> – Diagrama de blocos de medidores de energia eletrônicos. ....	26
<b>Figura 4</b> – Circuito integrado ADE7758 em um medidor eletrônico.....	27
<b>Figura 5</b> – Resistor Shunt, construção física (a) e modelo teórico(b). ....	29
<b>Figura 6</b> – Bobina de Rogowski. ....	30
<b>Figura 7</b> – Transformador de corrente.....	31
<b>Figura 8</b> – Circuito com divisor de tensão resistivo. ....	32
<b>Figura 9</b> – Medidor convencional em comparação com um <i>smart meter</i> . ....	35
<b>Figura 10</b> – Diagrama de blocos genérico de um <i>smart meter</i> . ....	36
<b>Figura 11</b> – Kill a Watt, modelo P4400 (a) e 4480 (b).....	38
<b>Figura 12</b> – Smart Plug Power Meter. ....	39
<b>Figura 13</b> – TED 5000 Energy Detective (a) tela do sistema supervisório do aparelho (b)....	39
<b>Figura 14</b> – Tela de visualização do Snapgrid. ....	40
<b>Figura 15</b> – Informações apresentadas na tela de visualização do snapgrid. ....	41
<b>Figura 16</b> – Smart plug Sonoff (a) e forma de ligação (b). ....	41
<b>Figura 17</b> – Aplicativo eWeLink. ....	42
<b>Figura 18</b> – Paradigma da Internet das Coisas como um resultado de diferentes visões. ....	43
<b>Figura 19</b> – Blocos básicos da IoT. ....	44
<b>Figura 20</b> – Camadas da IoT. ....	46
<b>Figura 21</b> – O modelo de publish/subscribe para sensores em IoT.....	49
<b>Figura 22</b> – Arquitetura dos dispositivos baseados em Internet das Coisas.....	50
<b>Figura 23</b> – Tela do Node-RED. ....	52
<b>Figura 24</b> – Tela de desenvolvimento do Blynk.....	53
<b>Figura 25</b> – Diagrama de blocos do smart plug.....	54
<b>Figura 26</b> – Fonte de alimentação do smart plug. ....	55
<b>Figura 27</b> – Transformador de corrente DL-CT1005A (dimensões em milímetros). ....	56
<b>Figura 28</b> – Circuito para medição de corrente. ....	56
<b>Figura 29</b> – Transformador encapsulado T-211259-R. ....	57
<b>Figura 30</b> – Circuito para medição de tensão. ....	57
<b>Figura 31</b> – Circuito integrado de medição ADE7753.....	58

<b>Figura 32</b> – Circuito de medição do ADE7753. ....	59
<b>Figura 33</b> – Arduino Nano 3.0. ....	59
<b>Figura 34</b> – Conexões do Arduino Nano com o relé e o DS1307. ....	60
<b>Figura 35</b> – Circuito do ESP8266-01 para o módulo de comunicação. ....	60
<b>Figura 36</b> – Aba “RESUMO” do aplicativo para monitoramento e controle do smart plug. ...	61
<b>Figura 37</b> – Aba “DADOS ELÉTRICOS” do aplicativo. ....	62
<b>Figura 38</b> – Aba “TEMPORIZADOR” do aplicativo. ....	63
<b>Figura 39</b> – Circuito de medição do smart plug. ....	64
<b>Figura 40</b> – Smart plug desenvolvido. ....	64
<b>Figura 41</b> – Bancada de cargas resistivas, indutivas e capacitivas para o teste de precisão. ..	65
<b>Figura 42</b> – Analisador de energia AE-100 utilizado nos testes de precisão. ....	68
<b>Figura 43</b> – Telas de visualização do aplicativo para as medições com cargas resistivas. ....	76
<b>Figura 44</b> – Telas de visualização do aplicativo para as medições com cargas indutivas. ....	76
<b>Figura 45</b> – Telas de visualização do aplicativo para as medições com cargas capacitivas. ..	77
<b>Figura 46</b> – Telas de temporização do aplicativo para controle das cargas. ....	78

## LISTA DE TABELAS

<b>Tabela 1</b> – Classe de exatidão dos transformadores de corrente. ....	31
<b>Tabela 2</b> – Classe de exatidão dos transformadores de potencial.....	33
<b>Tabela 3</b> – Cargas utilizadas para os testes de leitura de potência. ....	65
<b>Tabela 4</b> – Parâmetros de teste para cargas resistivas. ....	66
<b>Tabela 5</b> – Parâmetros de testes para cargas capacitivas.....	66
<b>Tabela 6</b> – Parâmetros de testes para cargas indutivas. ....	66
<b>Tabela 7</b> – Potências utilizadas no teste de medição de consumo.....	67
<b>Tabela 8</b> – Faixas de precisão do analisador de energia AE-100. ....	67
<b>Tabela 9</b> – Custos dos componentes do módulo de aquisição de dados.....	69
<b>Tabela 10</b> – Custos dos componentes do módulo de processamento de dados.....	69
<b>Tabela 11</b> – Custos dos componentes do módulo de comunicação.....	69
<b>Tabela 12</b> – Custos total do smart plug. ....	69
<b>Tabela 13</b> – Resultado do teste de alcance do ESP8266-01. ....	75
<b>Tabela 14</b> – Erro médio percentual para as grandezas medidas. ....	79
<b>Tabela 15</b> – Erro teórico e experimental para o smart plug.....	80

## LISTA DE GRÁFICOS

<b>Gráfico 1</b> – Comparação de medição de tensão.....	70
<b>Gráfico 2</b> – Erro de medição entre o AE-100 e o smart plug para medidas de tensão. ....	71
<b>Gráfico 3</b> – Comparação de medição de corrente.....	71
<b>Gráfico 4</b> – Erro de medição entre o AE-100 e o smart plug para medidas de corrente. ....	72
<b>Gráfico 5</b> – Comparação de medição de potência ativa. ....	73
<b>Gráfico 6</b> – Erro de medição entre o AE-100 e o smart plug para medidas de potência.....	73
<b>Gráfico 7</b> – Comparação de medição de consumo. ....	74
<b>Gráfico 8</b> – Erro de medição entre o AE-100 e o smart plug para medidas de consumo.....	75

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AC	<i>Alternating Current</i> (Corrente alternada)
ADC	<i>Analogic to Digital Converter</i> (Conversor Analógico Digital)
ANEEL	Agência Nacional de Energia Elétrica
CI	Circuito Integrado
CoAP	<i>Constrained Application Protocol</i> (Protocolo de Aplicação Restrita)
DC	<i>Direct Current</i> (Corrente contínua)
DSP	<i>Digital Signal Processor</i> (Processador de Sinal Digital)
DTLS	<i>Datagram Transport Layer Security</i>
EF	Entidades Físicas
EV	Entidades Virtuais
FPGA	<i>Field Programmable Gate Array</i>
EXI	<i>Efficient XML Interchange</i>
GE	<i>General Eletric</i>
GPIO	<i>General Purpose Input/Output</i> (Portas Programáveis de Entrada e Saída)
GPRS	<i>General Packet Radio Services</i> (Serviços Gerais de Pacote por Rádio)
GSM	<i>Global System for Mobile Communication</i>
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
IoT	<i>Internet of Things</i> (Internet das Coisas)
IP	<i>Internet Protocol</i> (Protocolo da Internet)
IR	<i>Infrared</i> (Infravermelho)
LCD	<i>Liquid Crystal Display</i> (Display de Cristal Líquido)
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
MQTT	<i>Message Queue Telemetry Transport</i> (Transporte de Telemetria de
NFC	<i>Near Field Communication</i> (Comunicação por Campo de Proximidade)
OWF	<i>Web Ontology Language</i>
PLC	<i>Power Line Communication</i> (Comunicação Via Rede Elétrica)
PGA	Amplificador de Ganho Programável
PROCEL	Programa Nacional de Conservação de Energia Elétrica
RDF	<i>Resource Description Framework</i>
RFID	<i>Radio Frequency Identification</i> (Identificação por Rádio Frequência)

RMS	<i>Root Mean Square</i> (Valor Quadrático Médio)
RSSI	<i>Received Signal Strength Indication</i> (Indicação da intensidade do sinal recebido)
RTC	<i>Real Time Clock</i> (Relógio de Tempo Real)
SMD	<i>Surface Mounting Device</i> (Dispositivo de Montagem Superficial)
SPI	<i>Serial Peripheral Interface</i> (Interface Periférica Serial)
TC	Transformador de Corrente
TCP	<i>Transmission Control Protocol</i> (Protocolo de Controle de Transmissão)
TP	Transformador de Potencial
UDP	<i>User Datagram Protocol</i>
WSN	<i>Wireless sensor network</i> (Redes de Sensores Sem fio)

## LISTA DE SÍMBOLOS

A	Ampère
d	Dia
h	Hora
Hz	Hertz
kWh	Quilowatt-hora
kA	Quiloampère
kWh	Quilowatt-hora
k $\Omega$	Quilo-Ohm
M	Mês
mA	Miliampère
MHz	Mega Hertz
min	Minuto
mm	Milímetro
mV	Milivolts
nF	Nano Faraday
pF	Pico Faraday
R\$	Moeda em reais
s	Segundo
$\mu$ A	Microampère
$\mu$ F	Micro Faraday
US\$	Moeda em dólar
V	Volts
VAr	Volt ampère reativo
V <sub>ca</sub>	Voltagem em corrente alternada
V <sub>cc</sub>	Voltagem em corrente contínua
W	Watt
Y	Ano ( <i>Year</i> )
Z <sub>in</sub>	Impedância de entrada
Z <sub>out</sub>	Impedância de saída
$\Omega$	Ohm

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	18
1.1 JUSTIFICATIVAS .....	19
1.2 OBJETIVOS .....	20
<b>1.2.1 Objetivo Geral</b> .....	20
<b>1.2.2 Objetivos Específicos</b> .....	20
1.3 ESTRUTURA DO TRABALHO .....	20
<b>2. REFERENCIAL TEÓRICO</b> .....	22
2.1 ESTADO DA ARTE.....	22
2.2 MEDIDORES DE GRANDEZAS ELÉTRICAS .....	24
<b>2.2.1 Circuitos Integrados para Medição de Energia</b> .....	27
<b>2.2.2 Sensores para Medição de Energia Elétrica</b> .....	28
<b>2.2.3 Sensores de Corrente</b> .....	29
2.2.3.1 Resistor <i>Shunt</i> .....	29
2.2.3.2 Bobina de <i>Rogowski</i> .....	30
2.2.3.3 Transformador de corrente .....	30
<b>2.2.4 Sensores de Tensão</b> .....	31
2.2.4.1 Divisor Resistivo .....	32
2.2.4.2 Transformador de Potencial .....	32
2.3 <i>SMART GRIDS</i> .....	33
<b>2.3.1 Smart Meters</b> .....	34
<b>2.3.2 Smart Plugs</b> .....	36
2.4 INTERNET DAS COISAS .....	42
<b>2.4.1 Arquitetura e Organização</b> .....	45
<b>2.4.2 Protocolos de Comunicação</b> .....	47
2.4.2.1 MQTT - <i>Message Queue Telemetry Transport</i> .....	48
2.4.2.2 CoAP - <i>Constrained Application Protocol</i> .....	49
<b>2.4.3 Tecnologias Relacionadas</b> .....	50
2.4.3.1 <i>Node-RED</i> .....	52
2.4.3.2 <i>Blynk</i> .....	52
<b>3. DESENVOLVIMENTO DO SMART PLUG</b> .....	54
3.1 MODELO PROPOSTO .....	54
<b>3.1.1 Módulo de Aquisição de Dados</b> .....	55
<b>3.1.2 Módulo de Processamento de Dados</b> .....	57
<b>3.1.3 Módulo de Comunicação</b> .....	60



3.1.4 Módulo de Visualização e Controle.....	61
3.1.5 Montagem do <i>Smart Plug</i> e Circuito de Medição.....	64
3.2 PARÂMETROS DE TESTE .....	65
3.2.1 Teste de Precisão do Circuito.....	65
3.2.2 Teste de Comunicação e Acionamento Remoto .....	68
3.3 CUSTO DE DESENVOLVIMENTO DO <i>SMART PLUG</i> .....	68
<b>4. RESULTADOS E DISCUSSÕES .....</b>	<b>70</b>
4.1 RESULTADO DO TESTE DE PRECISÃO .....	70
4.1.1 Medição de Tensão.....	70
4.1.2 Medição de Corrente.....	71
4.1.3 Medição Potência Ativa .....	72
4.1.4 Medição de Consumo .....	74
4.2 RESULTADO DO TESTE DE COMUNICAÇÃO.....	75
4.3 ANÁLISE DOS DADOS COLETADOS .....	79
<b>5. CONCLUSÃO.....</b>	<b>82</b>
<b>REFERÊNCIAS .....</b>	<b>84</b>
<b>ANEXO A – Biblioteca ADE7753.h .....</b>	<b>88</b>
<b>ANEXO B – Biblioteca ADE7753.cpp .....</b>	<b>94</b>

## 1. INTRODUÇÃO

Segundo Caires (2012) atualmente se atribui importância fundamental ao crescimento sustentável, este é um argumento que deve ser considerado na evolução da filosofia de projeto das instalações elétricas do futuro conectadas às chamadas *smart grids* (redes inteligentes).

Uma boa gestão de recursos energéticos necessita de controle de informações, tais como a curva de carga de uma instalação. Os sistemas automatizados de monitoramento de energia elétrica podem ser aplicados desde um aparelho único isolado como também em consumidores industriais de pequeno e de grande porte. Alguns dos fatores de sua implementação são de extrema significância para o setor de energia elétrica, destacando-se um como o principal, o combate às perdas.

Mesmo com o aumento do fluxo de dados em diversos setores, consumidores residenciais e comerciais ainda participam de forma passiva em relação ao uso de energia elétrica. Os medidores inteligentes (*smart meters*) atuais mostram apenas o consumo total de energia elétrica, sem apresentar detalhes de como essa energia é consumida. A discriminação do uso individual de cargas elétricas permite ao consumidor conhecer seu padrão de consumo e este conhecimento pode ser utilizado para numerosos fins. (MONZANI, 2016).

Ferreira (2014) destaca que nas últimas décadas, ocorreu significativo aumento no poder computacional, na quantidade de dispositivos eletrônicos e nas tecnologias de comunicação de computadores disponíveis no mercado. Esse aumento de poder tornou possível o surgimento do conceito de Internet das Coisas, baseado no princípio que todos os aparelhos de uma residência podem estar conectados a rede mundial de computadores, enviando dados e recebendo solicitações.

A junção dos conceitos de *smart meter* e internet das coisas deu origem ao termo conhecido como *smart plug*, que consiste em um dispositivo de dimensões reduzidas, se comparado ao medidor convencional, dotado de capacidade de medição de grandezas elétricas e de controle de carga, acionando ou desligando remotamente qualquer aparelho doméstico a ele conectado, o acesso às informações são feitas por comandos remotos, através de aplicativos instalados em *smartphones*, permitindo que o usuário tenha total controle do dispositivo.

Diante dessas possibilidades de tecnologias, neste trabalho foi projetado, desenvolvido e implementado um *smart plug* que permita o monitoramento do consumo e controle de aparelhos eletrodomésticos em redes de baixa tensão, sendo este compatível com o padrão brasileiro de tomadas, facilitando a instalação e evitando interrupções nos circuitos. O

dispositivo foi desenvolvido utilizando o circuito integrado de medição ADE7753 e foi estabelecida a comunicação com usuário através de aplicativo.

Para verificação das funcionalidades do circuito de medição foram analisadas cargas indutivas, capacitivas e resistivas. Os resultados foram comparados com um analisador de energia comercial, verificando a precisão do plugue. A comunicação foi testada a partir da criação de um servidor local onde os valores medidos foram enviados e acessados através de um aplicativo para celular.

## 1.1 JUSTIFICATIVAS

As vantagens deste projeto podem ser observadas no que diz respeito a transparência em relação ao seu consumo de energia e praticidade, uma vez que o proprietário saberá exatamente o ponto onde o consumo é mais elevado e por meio dessa informação pode estabelecer metas de redução, tais como um controle de uso de aparelho de ar condicionado ou horário fixo para ligar/desligar lâmpadas. A praticidade é observada no acionamento de aparelhos através de um smartphone, podendo inclusive aplicar a função de temporização, ligando ou desligando um determinado aparelho em horário específico.

A facilidade no repasse e aquisição de informações é outro ponto positivo, já que não é necessário estar presente na residência para a visualização do consumo e sua instalação pode ser feita através de plugue compatível com o padrão brasileiro de tomadas. No que diz respeito ao aspecto técnico, o projeto foi realizado utilizando componentes de precisão que são usados em equipamentos comerciais e que possuem durabilidade compatível com ambientes industriais, portanto, uma vez instalados nos pontos de tomada os mesmos não necessitam de manutenção ou perturbações dos circuitos da instalação.

O trabalho foi desenvolvido visando a obtenção de um *smart plug* (plugue inteligente) que dará ao usuário final, no caso, o consumidor de baixa tensão, um controle exato do seu gasto de energia e capacidade de acionamento remoto de pontos de utilização, sem a necessidade de adquirir um equipamento mais robusto e de preço elevado. As informações repassadas estão disponíveis de forma remota, sendo possível verificar o consumo mesmo sem estar presente na instalação. O acionamento/desligamento dos pontos de uso também pode ser feito remotamente através de temporização, utilizando para isso um aplicativo móvel.

## 1.2 OBJETIVOS

Os objetivos do trabalho estão divididos em geral e específicos, sendo estes explicitados nas duas seções que seguem.

### 1.2.1 Objetivo Geral

Desenvolver um *smart plug* (plugue inteligente) com capacidade de medição de equipamentos elétricos de baixa tensão monofásicos, tendo como característica básica a facilidade de instalação nos pontos de utilização, controle de acionamento por temporização, interpretações dos dados de forma precisa e confiável, armazenamento de informações coletadas, bem como o acesso destas a partir de aplicativo, visando o acesso das informações a partir de um *smartphone*.

### 1.2.2 Objetivos Específicos

- Projetar o circuito do *smart plug* utilizando o módulo Wi-Fi ESP8266-01 em conjunto com o CI de medição ADE7753 e a plataforma de prototipagem Arduino;
- Projetar um dispositivo compacto e de fácil instalação nas tomadas residenciais, sem interferências na rede e sem distúrbios para o usuário final;
- Aplicar o conceito de “Internet das Coisas” para transmissão e armazenamento dos dados coletados através do módulo ESP8266-01 e o protocolo MQTT;
- Desenvolver um aplicativo para visualização e interpretação dos dados coletados;
- Realizar análise comparativa com instrumento de medição comercial com a finalidade de testar a precisão do protótipo desenvolvido;
- Realizar testes de comunicação para avaliar se os dados medidos estão disponíveis no aplicativo e teste de controle de cargas através de acionamento remoto para validação dos dados.

## 1.3 ESTRUTURA DO TRABALHO

No capítulo 1 foi apresentada a introdução do trabalho, a justificativa do mesmo e as apresentações dos objetivos gerais e específicos almejados durante sua elaboração.

No capítulo 2 será apresentado o referencial teórico contendo conceitos gerais de eletricidade e medição de energia, definição de *smart meters* e uma breve descrição a respeito da Internet das Coisas.

No capítulo 3 serão apresentados os materiais e métodos utilizados para o desenvolvimento do trabalho. Partindo da etapa de projeto até a esquematização do circuito do *smart plug* e testes de funcionalidade e comunicação.

No capítulo 4 serão analisados e discutidos os resultados obtidos após o processo de coleta de dados.

No capítulo 5 serão apresentadas as conclusões e sugestões de trabalhos futuros com a mesma temática do trabalho realizado.

## 2. REFERENCIAL TEÓRICO

Neste capítulo serão abordados os conceitos básicos necessários para a compreensão e execução desta dissertação, serão apresentados alguns trabalhos que tratam da construção de *smart plugs*, bem como de *smart meters*. Posteriormente detalham-se os princípios fundamentais da eletricidade, tais como os conceitos de Tensão, Corrente e Potência elétrica e as formas de medição utilizado por medidores eletrônicos, bem como os sensores utilizados. Por fim serão abordados os conceitos de *Smart Grids* (medidores inteligentes) e de internet das coisas.

### 2.1 ESTADO DA ARTE

Os trabalhos relacionados a *smart plugs* remontam do início dos anos 2000, com a expansão da comunicação sem fio e o nascimento da automação residencial. Potter (2002) em seu artigo “*Smart Plug and Play Sensors*”, descreve uma proposta de norma do grupo de trabalho IEEE (*Institute of Electrical and Electronics Engineers*) que define uma interface de modo misto que mantém o sinal analógico tradicional de um sensor, em conjunto com um sinal digital para acessar as informações do transdutor, teorizando um dos primeiros modelos de *smart plug*.

Com o surgimento do conceito de *smart meter* e *smart grids* iniciaram-se as pesquisas dedicadas aos dispositivos inteligentes, Cesário Júnior (2014) desenvolveu um protótipo de medidor inteligente para medição de grandezas elétricas em ambientes residenciais. Em seu trabalho foram usados transformadores de corrente e divisor resistivo para captação de tensão e corrente; o processamento foi realizado utilizando o circuito integrado de medição ADE7753 da Analog Devices, garantindo de 2% para as medições de tensão, corrente e potência ativa. A comunicação foi feita através de conexão Wi-Fi com um computador.

Lee (2011) desenvolveu um *smart plug* com rede de sensores sem fio para monitoramento do uso de energia elétrica, o dispositivo integrava um soquete de energia em corrente alternada, um relé para comutar a tomada (ligar e desligar), um sensor de transformador de corrente para detectar a carga atual do aparelho e uma interface de comunicação sem fio.

Dhond (2012) em seu trabalho “*Sub-metering made easy using Texas Instruments energy measurement ICs*”, descreve as características e benefícios de um modelo de *smart plug* aplicados na medição de energia elétrica individualizada de equipamentos a ele conectados. No artigo é utilizado um resistor *shunt* para medir a corrente e um divisor de tensão para a amostra

da tensão da rede. A energia é calculada em um microcontrolador utilizando conversores A/D (analógico/digital).

Brenkus et al. (2014) também trabalhou no desenvolvimento de um *smart meter* para cargas individuais, empregando *bluetooth* como meio de comunicação entre dispositivos e entre o usuário e o medidor, o trabalho destaca um comparativo entre as tecnologias de transmissão de dados.

Ahmed et al. (2015) propôs a criação de um *smart plug* para a medição de consumo de energia de aparelhos residenciais, o projeto aplica os conceitos de Internet das Coisas, com múltiplos plugues comunicando-se entre si através do protocolo ZigBee.

Zhang et al. (2015) apresentou um *smart plug* para acionamento de aparelhos elétricos. O plugue apresentava a função de acionar a carga remotamente, utilizando aplicativo móvel e comunicação Wi-Fi através do módulo ESP8266.

Lemos (2017) trabalhou com o desenvolvimento de um *smart meter* para análise de adesão de consumidores a tarifa branca. O medidor foi construído utilizando transformador de corrente e divisor resistivo, a leitura e interpretação dos dados foi realizada com o circuito integrado ADE7753, garantindo a precisão do dispositivo. A transmissão de dados foi feita através do módulo ESP8266, enviando os dados para um servidor remoto com acesso à rede.

Outros trabalhos abordam aplicações do uso de *smart plugs*, sejam eles modelos comerciais ou desenvolvidos previamente. Petrovic e Morikawa (2017) realizaram um estudo onde é utilizado um *smart plug* para detecção do tipo de carga conectada utilizando redes neurais artificiais. Estudo semelhante ao realizado por Zhai et al. (2018) que analisou a flexibilidade do uso de cargas residenciais para melhoria da gestão energética, utilizando um *smart plug*.

Fabricio (2018) aplicou os conceitos de Internet das Coisas para análise de equipamentos elétricos industriais, trabalhando com dispositivos comerciais de medição, porém integrando os mesmos em uma rede de compartilhamento unificada.

Bezerra (2018) desenvolveu *hardware* e *firmware* utilizando os conceitos de Internet das Coisas para aquisição e medição de dados de uma central condicionadora de ar, a comunicação com o servidor remoto, um computador Raspberry Pi, foi realizada por meio de um broker MQTT (servidor em comunicações).

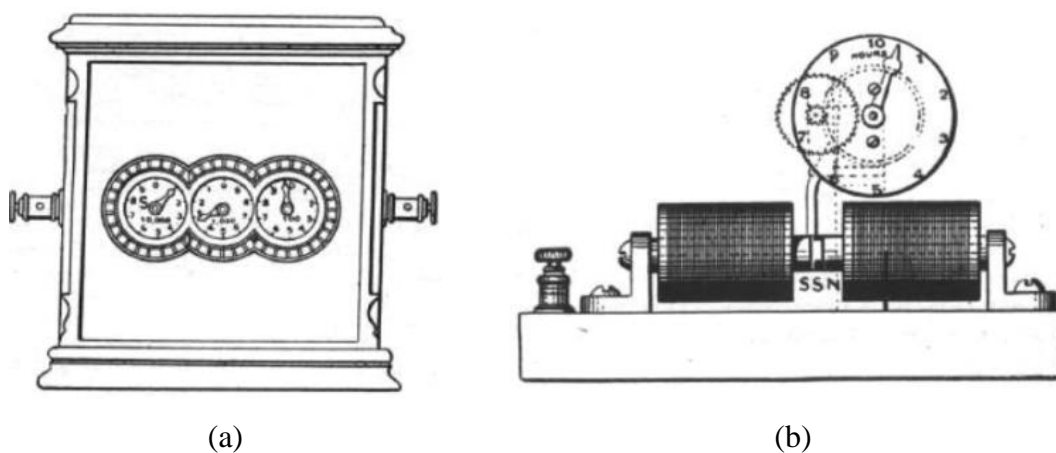
## 2.2 MEDIDORES DE GRANDEZAS ELÉTRICAS

É importante saber medir os níveis de corrente e de tensão de um sistema elétrico para verificar seu funcionamento, identificar defeitos e investigar efeitos que possam ocorrer e que seriam impossíveis de serem previstos em uma análise teórica. (BOYLESTAD, 2012).

Segundo Dahle (2010 apud Brito, 2016), os primeiros dispositivos que mediam o consumo de energia elétrica foram baseados em lâmpadas, como a tensão e corrente nessas lâmpadas eram constantes e que todas as lâmpadas eram ligadas a somente uma chave, era necessário medir apenas o tempo que essas lâmpadas ficavam acesas para obter o consumo final, nessa época criou-se a unidade lâmpada-hora.

Em 1872, Samuel Gardiner patenteou o primeiro medidor de energia elétrica, sendo este um mecanismo eletromagnético que marcava em um relógio a quantidade de tempo na qual a chave ficava ligada. Quase um século depois, em 1978, Fuller desenvolveu uma alternativa ao medidor patenteado, porém a única diferença foi que o medidor possuía uma armadura vibrando entre duas bobinas, que movimentavam o gerador para o registro de tempo. (BRITO, 2016). A Figura 1 mostra os medidores de Gardiner e o medidor de Fuller.

**Figura 1** – Medidor de Gardiner (a) e medidor de Fuller (b).



**Fonte:** Moro (2013).

Dahle (2010 apud Brito, 2016) afirma que a partir destas descobertas, a produção de novas alternativas de aparelhos que medissem o consumo de energia elétrica seguiu avançando em termos tecnológicos. Em intervalos de aproximadamente 10 anos, era inventado um modo mais moderno de medição do consumo de energia elétrica, até empresas de maior porte como a *General Electric* (GE) começaram a participar dessa corrida tecnológica na obtenção de um melhor meio. Somente nos anos 90, com os avanços na eletrônica que se permitiu a fabricação

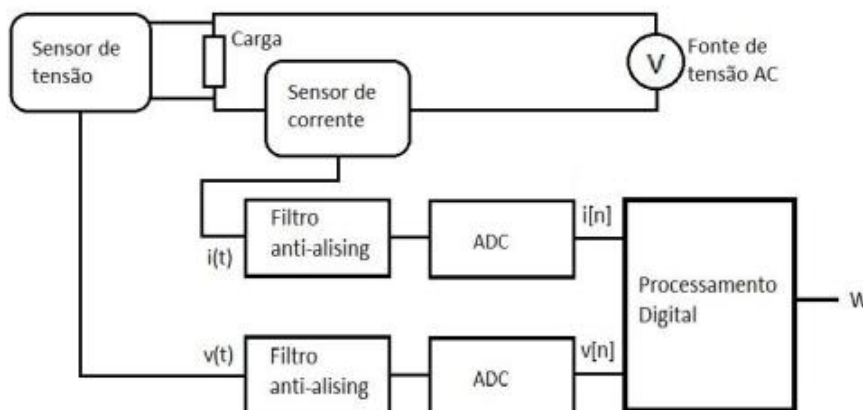


de medidores eletrônicos sem o uso de partes móveis, começando a aparecer modelos mais parecidos com os conhecidos em residências e estabelecimentos nos dias atuais.

De acordo com Moro (2013) os medidores eletrônicos proporcionam uma gama de informações úteis, permitindo a introdução de novos serviços, pois além de medir o consumo de energia, acrescenta mais informações do que um medidor convencional e que pode transmitir dados por meio de um formulário eletrônico de comunicação.

Esses medidores constituem a solução mais moderna atualmente. Nesse tipo de instrumento, os sinais de entrada de tensão e corrente são digitalizados e todo o processamento é feito por meio de um processador dedicado. O tipo de processador pode variar de simples microcontroladores dedicados providos de pequenos recursos de *hardware* até potentes DSPs (*digital signal processor*), rápidos o suficiente para prover processamento em tempo real, bem como transmissão via rede ou algum outro meio de comunicação. (BALBINOT E BRUSAMARELL, 2010). A Figura 2 mostra o diagrama de blocos básico de um multímetro eletrônico.

**Figura 2** – Diagrama de blocos de um multímetro digital.

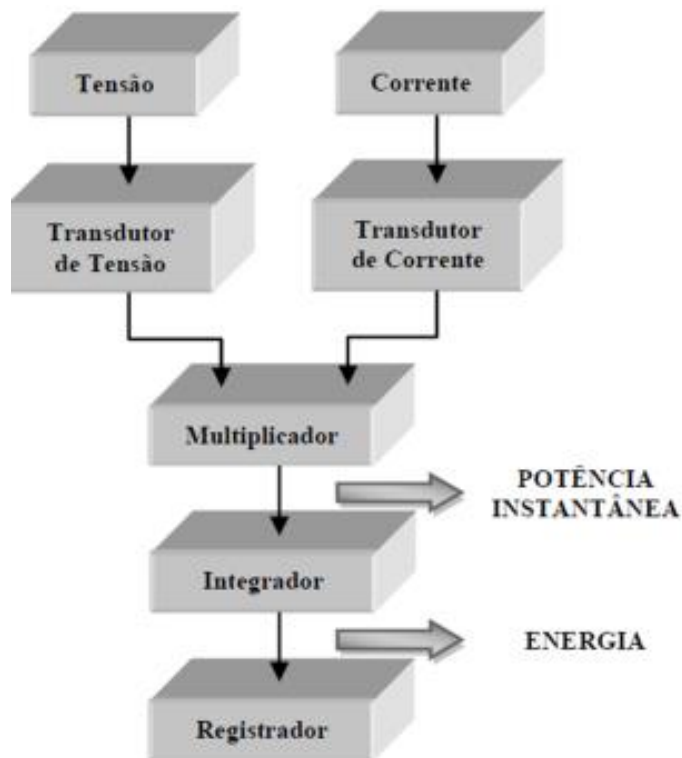


**Fonte:** Moro (2013).

De acordo com Cesário Júnior (2014) a norma NBR-14519 define medidor eletrônico de energia elétrica como: medidor estático no qual a corrente e a tensão agem sobre elementos de estado sólido (componentes eletrônicos) para produzir uma informação de saída proporcional à quantidade de energia elétrica medida. Basicamente, todos os medidores medem o consumo de energia elétrica. Um medidor pode ser equipado com um registrador e um display no qual o consumo acumulado pode ser lido. Leituras armazenadas em memória são registradas com a data e hora em que a leitura foi tomada. As leituras podem assumir a forma de consumo acumulado ou da própria energia consumida no intervalo das leituras especificadas. (COSTA, 2014)

Balthazar (2004 apud Costa, 2014) destaca que os medidores de energia eletrônicos podem ser descritos esquematicamente por um diagrama em blocos, conforme a Figura 3.

**Figura 3** – Diagrama de blocos de medidores de energia eletrônicos.

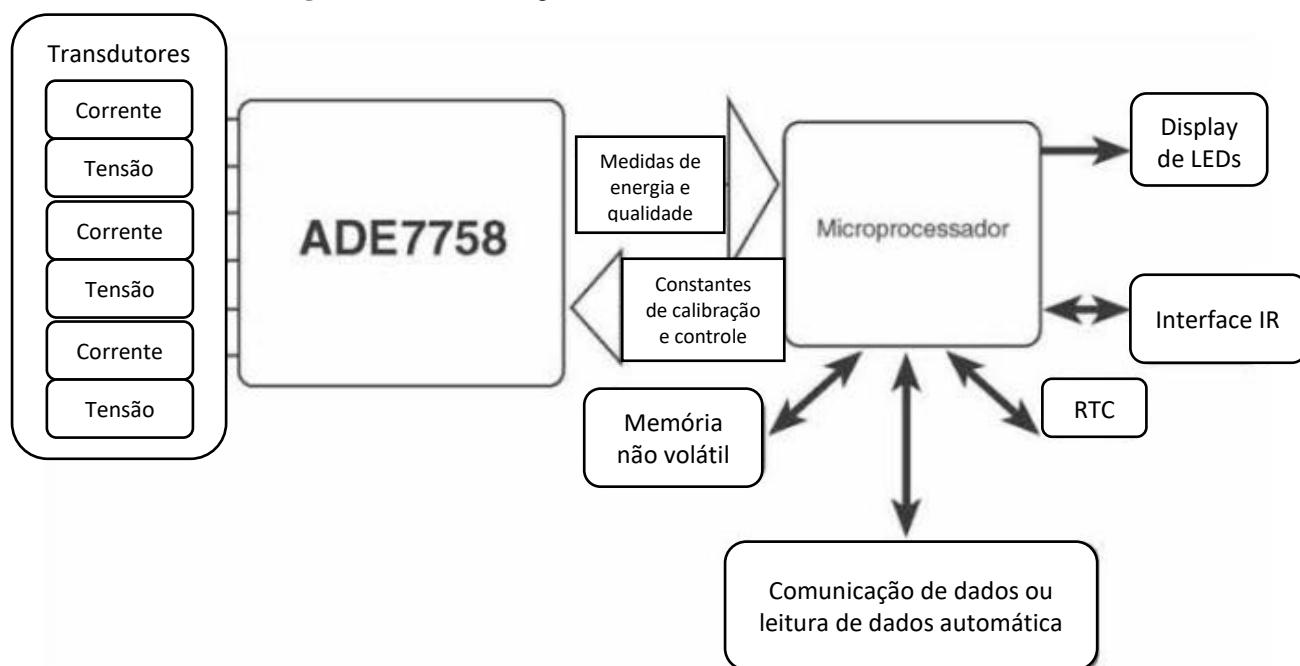


Fonte: Costa (2014).

A medição de energia é realizada usando-se três blocos principais: transdutor, multiplicador e integrador/armazenador. Os transdutores de tensão e corrente adquirem e adequam os sinais a serem multiplicados. A potência instantânea é determinada através do bloco multiplicador, realizando a multiplicação da medição da tensão e da corrente. A energia é obtida com o bloco integrador. Finalmente, esta informação de energia é armazenada e registrada no bloco registrador. (BALTHAZAR, 2004 apud COSTA, 2014)

Balbinot e Brusamarell (2010) destaca que os modernos medidores de energia também podem fazer o monitoramento e a análise de harmônicas, gerando informações importantes no que diz respeito à qualidade de energia. Exemplos de componentes dedicados para a medição de energia são o ADE7753 e ADE7758 da Analog Devices. Esses componentes podem medir energia ativa, energia reativa e aparente de sistemas monofásicos (ADE7753) e trifásicos (ADE7758). A Figura 4 mostra uma aplicação típica desse componente.

**Figura 4** – Circuito integrado ADE7758 em um medidor eletrônico.



Fonte: Balbinot e Brusamarell (2010).

### 2.2.1 Circuitos Integrados para Medição de Energia

O rápido avanço da eletrônica vem contribuindo para soluções cada vez mais integradas de componentes para medição de energia, com muitas funcionalidades e características diversas. (POPSIN, 2010)

Os circuitos integrados de medição são dispositivos construídos com a finalidade de reunir todo o conjunto de periféricos básicos para converter as grandezas elétricas medidas em sinais digitais, permitindo sua visualização. Equipamentos de medição precisos e com finalidades comerciais empregam estes circuitos integrados em sua construção, devido a precisão e o tamanho reduzido.

- STPM01 – ST Microelectronics

Segundo Popsin (2010) o circuito integrado STPM01 da ST Microelectronics, foi desenvolvido especialmente para medição de energia ativa, reativa e aparente, em sistemas monofásicos de distribuição, utilizando bobina de Rogowski (seção 2.2.3.2) e/ou transformador de corrente e/ou *shunt* (seção 2.2.3.1) como sensor de corrente, e transformador de potencial ou divisor de tensão resistivo como transdutores de tensão. O STPM01 consiste essencialmente de duas partes: parte analógica e parte digital. A parte analógica contém PGA's (Amplificadores

de ganho programável) e conversores A/D. A parte digital conta com um DSP encarregado de todos os cálculos necessários a obtenção dos valores das potências ativa, reativa e aparente, bem como, dos valores RMS (*root mean square*) de tensão e corrente.

O processo de calibração do medidor segue orientações do fabricante e se resume simplesmente ao ajuste de três registradores internos no medidor STPM01. Esses registradores atuam como compensadores de ganho, no caminho do processamento dos sinais de tensão e corrente no interior medidor. A compensação é capaz de corrigir desvios de até +/- 12 % que ocorrem em razão da tolerância dos componentes utilizados nos canais analógicos de entrada, formados pelos transdutores de tensão e corrente e filtros *anti-aliasing*. (POPSIN, 2010)

- ADE7753 – Analog Devices

Especialmente dedicado para medição de energia ativa, reativa e aparente, é possível medir os valores eficazes (RMS) de tensão e corrente elétrica para sistemas monofásicos a dois fios, seus canais de entrada permitem a utilização de bobinas de Rogowski, TC e/ou shunt como sensor de corrente e transformador de potencial ou divisor de tensão resistivo como transdutor de tensão.

As entradas analógicas possuem PGA's e conversores A/D 16 bits, com uma função selecionável para a integração digital no caso da utilização de sensores tipo Rogowski, eliminando a necessidade de um integrador externo, resultando em uma excelente estabilização e precisão entre a corrente e a tensão. (POPSIN, 2010)

Os resultados medidos são disponibilizados ao usuário através da Interface SPI (*serial peripheral interface*) em diversos registradores. O circuito possui funções para níveis de supervisão como tensão abaixo do recomendado (subtensão), frequência da linha, detecção de picos de tensão e um sensor de temperatura interno para verificar a temperatura e condição de operação. A calibração é efetuada via registradores pela interface serial com o envio de comando e valores a serem alterados para calibração de tensão, corrente e pulsos de energia.

### **2.2.2 Sensores para Medição de Energia Elétrica**

Segundo Thomazini (2004 apud Popsin, 2010) os sensores são dispositivos que convertem uma grandeza física em um sistema de medição e são geralmente definidos como um dispositivo que recebe e responde a um estímulo ou a um sinal. Muitas vezes um sensor é composto de um transdutor - dispositivo que converte um tipo de energia em outra e não

necessariamente em um sinal elétrico - e outra parte que converte a energia resultante em um sinal elétrico. São funções características de um elemento sensor:

- Função de transferência: estabelece a relação entre o sinal de entrada (grandeza a ser medida) e o sinal de saída, sendo sempre possível determinar uma função de transferência ideal para um sensor.
- Saturação: todos os sensores têm limites de funcionamento a partir dos quais perdem sua linearidade.
- Impedância: para uma saída em tensão, a impedância de saída ( $Z_{out}$ ) deve ser baixa e a impedância de entrada ( $Z_{in}$ ) deve ser alta. Por outro lado, para saída em corrente, a impedância de saída deve ser alta e a entrada deve ser baixa.
- Excitação: especifica quais as grandezas necessárias ao funcionamento de um sensor. Para alguns tipos de sensores é importante saber a especificação da resposta, frequência e qual a estabilidade necessária ao sinal de excitação. A resposta em frequência especifica qual a sensibilidade do sensor às variações de frequência do sinal de entrada.

### 2.2.3 Sensores de Corrente

Para realizar a medição de forma adequada, é necessário disponibilizar os níveis de corrente elétrica que circulam pelo circuito, conforme as características de entrada do circuito de medição. Para realizar a medição de corrente existem diversos dispositivos. (POPSIN, 2010)

#### 2.2.3.1 Resistor *Shunt*

Segundo Higashi (2006 apud Popsin, 2010) o resistor *Shunt* mede a intensidade de corrente obtida pela lei de *Ohm*. A Figura 5 mostra o resistor *Shunt* e o seu modelo teórico.

**Figura 5** – Resistor Shunt, construção física (a) e modelo teórico(b).



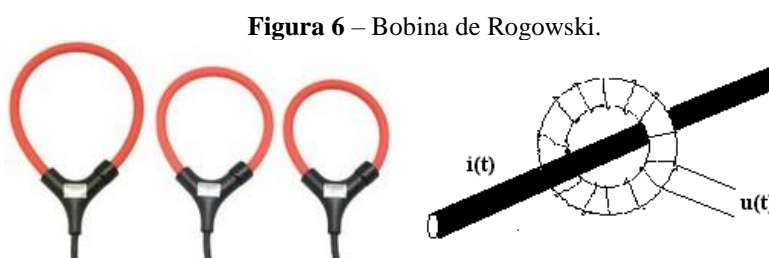
Fonte: Popsin (2010).

Apesar de ser um dos métodos de medição mais utilizados, o mesmo apresenta alguns inconvenientes:

- Em caso de alta corrente provoca um aquecimento do resistor;
- A tensão causa problemas de isolamento elétrica do sistema de medida;
- A alta frequência produz o efeito pelicular.

### 2.2.3.2 Bobina de *Rogowski*

A Bobina *Rogowski* consiste de um núcleo toroidal, que é colocado em torno do condutor. O campo magnético produzido pela corrente alternada no condutor induz uma tensão na bobina. (POPSIN, 2010) A Figura 6 mostra a bobina de Rogowski.

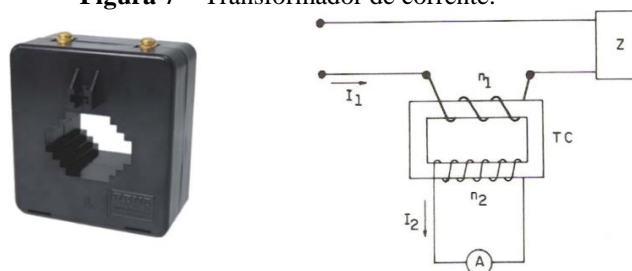


Fonte: Popsin (2010).

Higashi (2006 apud Popsin, 2010) ressalta que a faixa de leitura da Bobina *Rogowski* vai desde miliampères (mA) até alguns quiloampères (kA), ela não apresenta histerese, pois seu núcleo é de material não magnético. Possui boa linearidade, facilitando as medidas em lugares com acesso limitado, não possui contato físico com o circuito, baixo consumo e uma baixa variação do sinal da saída com a temperatura. Além disso, a indutância mútua não depende da corrente nem da frequência do sinal a medir. A única limitação em frequência vem determinada pela ressonância da bobina, a qual depende do projeto.

### 2.2.3.3 Transformador de corrente

O transformador de corrente ou simplesmente TC, transforma a corrente que circula em um enrolamento primário, através de um acoplamento magnético, em uma corrente induzida no secundário, conforme apresentado na Figura 7.

**Figura 7** – Transformador de corrente.

**Fonte:** Popsin (2010).

Popsin (2010) destaca que os tipos de transformadores de corrente são classificados em primeiro nível de acordo com o modelo do enrolamento primário - normalmente constituído por poucas espiras feitas por condutor de cobre de elevada secção - e em segundo nível pelo enrolamento secundário - constituído por uma bobina com derivações ou múltiplas bobinas ligadas em série e/ou paralelo.

Os transformadores de corrente são padronizados por classe de exatidão, de acordo com sua aplicação, conforme apresentado na Tabela 1.

**Tabela 1** – Classe de exatidão dos transformadores de corrente.

Classe de Exatidão	Aplicação
Menor do que 0,3	<ul style="list-style-type: none"> <li>• TC padrão;</li> <li>• Medições e laboratório;</li> <li>• Medições especiais;</li> </ul>
0,3	<ul style="list-style-type: none"> <li>• Medição de energia elétrica para faturamento ao consumidor.</li> </ul>
0,6 ou 1,2	<ul style="list-style-type: none"> <li>• Medição de energia elétrica sem finalidade de faturamento ao consumidor;</li> <li>• Alimentação de instrumentos de controle: voltímetro, Wattímetro, Varímetro, etc.</li> </ul>

**Fonte:** Filho (2007) apud Popsin (2010).

A classe de exatidão é o limite de erro, garantido pelo fabricante de um instrumento, que se pode cometer em qualquer medida efetuada pelo mesmo, sendo uma classificação do instrumento de medida para designar a sua exatidão.

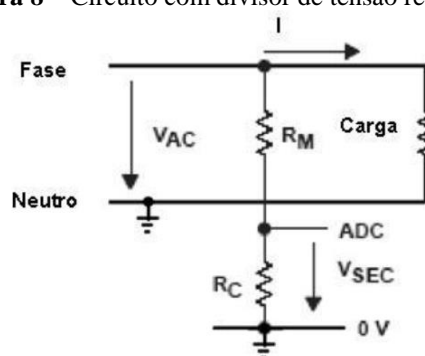
## 2.2.4 Sensores de Tensão

Os níveis de medida de tensão elétrica muitas vezes não podem ser aplicados diretamente aos circuitos que realizam o processamento desta grandeza. Neste caso se faz necessária a utilização de dispositivos ou métodos para adequar estes níveis de tensão que estão sendo medidos aos níveis que a instrumentação condiciona. (POPSIN, 2010)

### 2.2.4.1 Divisor Resistivo

Para Filho (2007 apud Popsin, 2010) a utilização de um divisor de tensão resistivo é uma alternativa para adequar os níveis do circuito a ser medido com o circuito de condicionamento. Este divisor tem como finalidade interpretar a tensão aplicada à entrada de um circuito, para que a mesma possa ser dividida entre os componentes do mesmo, indicando na saída uma tensão nos níveis adequados à entrada do circuito de medição. A Figura 8 mostra o circuito com divisor resistivo.

**Figura 8** – Circuito com divisor de tensão resistivo.



Fonte: Popsin (2010).

A Figura 8 representa um circuito onde a tensão da rede é aplicada ao resistor paralelo ( $R_M$ ), que possui maior valor, e a maior parte da tensão é aplicada sobre este resistor. Desta forma sobre o resistor série ( $R_C$ ) aplica-se a menor parcela da tensão da rede.

É importante salientar que os sistemas de medição possuem uma referência do sinal a ser medido. Como pode-se observar na Figura 8, é possível verificar que o neutro está aterrado, assim como 0 V da entrada de medição ADC, com isso se elimina pontos de referência flutuante, que permitem a entrada de interferências externas não desejadas. (POPSIN, 2010).

### 2.2.4.2 Transformador de Potencial

Para Filho (2007 apud Popsin, 2010), um transformador de potencial (TP) é um transformador para instrumentação cujo enrolamento primário é ligado em derivação com um circuito elétrico e cujo enrolamento secundário se destina a alimentar bobinas de potencial de instrumentos elétricos de medição, controle ou proteção. Na prática o TP é considerado um redutor de tensão, pois a tensão no seu circuito secundário é normalmente menor que a tensão no seu enrolamento primário. Assim como o TC, o TP também é padronizado por classe de exatidão, de acordo com sua aplicação, conforme a Tabela 2.



**Tabela 2** – Classe de exatidão dos transformadores de potencial.

<b>Classe de Exatidão</b>	<b>Aplicação</b>
Menor do que 0,3	<ul style="list-style-type: none"> <li>• TP padrão;</li> <li>• Medições e laboratório;</li> <li>• Medições especiais;</li> </ul>
0,3	<ul style="list-style-type: none"> <li>• Medição de energia elétrica para faturamento ao consumidor.</li> </ul>
0,6 ou 1,2	<ul style="list-style-type: none"> <li>• Medição de energia elétrica sem finalidade de faturamento ao consumidor;</li> <li>• Alimentação de instrumentos de controle: voltímetro, Wattímetro, Varímetro, etc.</li> </ul>

**Fonte:** Filho (2007) apud Popsin (2010).

### 2.3 SMART GRIDS

As *Smart Grids* podem ser traduzidas como redes elétricas inteligentes. Elas são capazes de integrar de forma inteligente o comportamento e as ações de todos os usuários conectados a ela. Utiliza-se tecnologias avançadas digitais para o monitoramento e gerenciamento do transporte de eletricidade em todas as fontes de geração, atendendo a variação de demanda de energia dos utilizadores finais. (BRITO, 2016)

No modelo atual de redes de energia o consumidor não possui informações sobre o sistema e não tem participação na geração. Além disso, essas redes possuem mecanismos limitados para a coordenação da operação, confiabilidade e qualidade do sistema. (FERREIRA, 2012).

De acordo com Caires (2012) espera-se que em um futuro próximo o consumidor seja envolvido na cadeia de controle pela chamada rede inteligente que tem a capacidade de trocar informações e energia de modo bidirecional. A rede inteligente teria então a capacidade de coordenar o intercâmbio de energia e informações de modo a otimizar o aproveitamento das fontes de energia, como parte dos recursos disponíveis. Além disso, ela teria a capacidade de otimizar o desempenho das funções de proteção minimizando a consequência das falhas ou atuando de maneira preventiva e dessa forma maximizaria a confiabilidade do sistema.

Torri (2012) destaca que as redes inteligentes não são derivadas de apenas uma tecnologia, mas sim da soma de várias partes. Cada uma destas partes é responsável por adicionar mais inteligência à rede. Apesar de uma pequena diferença no foco em cada um dos grupos existentes para padronização, os conceitos, em geral, são os mesmos para todos eles, enquanto as soluções empregadas serão relativas às necessidades e circunstâncias. Destaca-se que as tecnologias fundamentais das redes inteligentes são:

- Comunicação integrada, que interliga os diferentes componentes da arquitetura, permitindo um intercâmbio de informações em tempo real;
- Tecnologias de detecção e medição para permitir monitoramento e gestão remota, apoiando respostas mais rápidas e precisas, e para também permitir definição de preço da energia conforme a demanda;
- Componentes avançados permitindo aplicar as últimas pesquisas em supercondutividade, armazenamento, eletrônica de potência e diagnóstico;
- Métodos de controle avançado para monitorar os componentes essenciais, permitindo um diagnóstico rápido e soluções específicas adequadas a qualquer evento;
- Melhoria das interfaces e de apoio à decisão para amplificar a tomada de decisão humana.

### **2.3.1 Smart Meters**

A Medição Inteligente (*Smart Metering*) é um dos pilares das *Smart Grids*, pois além de medir corretamente o fluxo de energia, permite a transferência de informações do consumo de energia e o status da rede entre fornecedores e consumidores. (LEMOS, 2017).

Depuru et al. (2011 apud Rigodanzo, 2015) afirma que no futuro, as redes de distribuição e os medidores inteligentes terão um grande papel no monitoramento do desempenho e características de carga da rede. Neste cenário, as empresas poderão gerir de forma eficiente a demanda de energia uma vez que terão uma base confiável de consumo de todos os clientes, podendo orientá-los sobre maneiras eficientes de utilizar a energia residencial.

Apesar dos *smart meters* já estarem disponíveis no mercado global, a definição de conceitos e tecnologias do mesmo estão em pleno desenvolvimento, onde empresas de todos os tamanhos, institutos de pesquisa, universidades e mesmo desenvolvedores independentes estão trabalhando na construção e melhoria de dispositivos que serão parte do conceito amplo de *Smart Grid*. (CESÁRIO JÚNIOR, 2014).

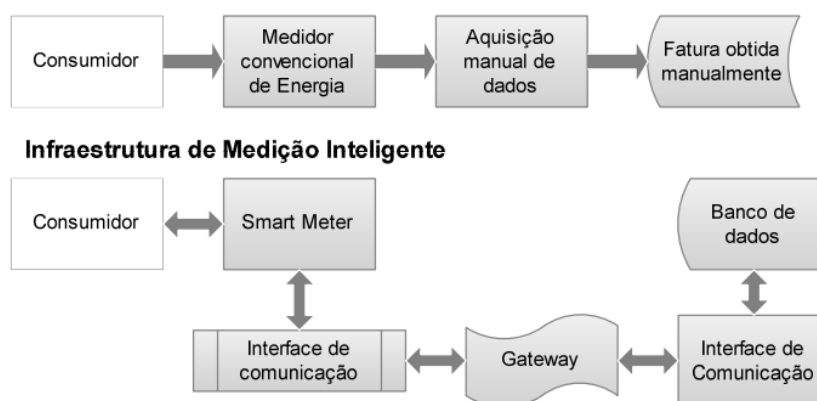
Cesário Júnior (2014) afirma que existem várias definições de medidores inteligentes e de sua constituição física, funcionalidades e meios de comunicação, onde pode-se destacar as seguintes características que o definem:

- Registrar em tempo real, ou próximo do real, o uso da eletricidade (potência) e consumo;
- Oferecer a possibilidade de ler as informações no local ou remotamente;

- Conectar com dispositivos pré-definidos;
- Limitar o consumo pelo medidor inteligente (cortar o fornecimento de energia).

A estrutura de medição dos *smart meters* difere da estrutura de medidores eletrônicos. Conforme citado anteriormente, as medições inteligentes trabalham com o armazenamento de informações e a disponibilidade das mesmas ao usuário, diferentemente dos medidores convencionais, onde as informações não são repassadas, apenas o montante final de consumo. A Figura 9 mostra a diferença entre as duas estruturas.

**Figura 9** – Medidor convencional em comparação com um *smart meter*.  
**Medidor de Energia Convencional**



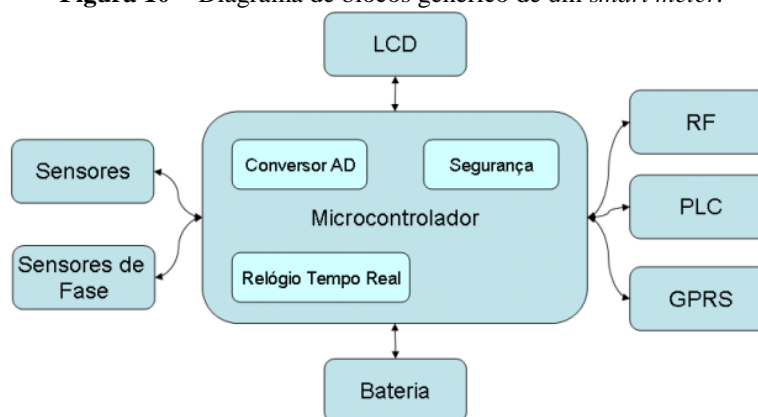
Fonte: Figueiró (2013) apud Rigodanzo (2015).

Com base na Figura 9 destacam-se as vantagens da estrutura de medição com *smart meters*:

- Economia de tempo: as leituras são enviadas eletronicamente, significando que não há gasto de logística quando se precisa de uma leitura;
- Maior controle do uso da energia: usando medidores inteligentes, é possível acompanhar o gasto de energia em reais e centavos, kilowatt horas, ou mesmo medir emissões de carbono. É possível descobrir o que acontece com o uso em horários de pico e programar-se melhor para medidas de eficiência;
- Redução das contas de energia: medidores inteligentes não reduzem diretamente a conta de energia, mas eles podem lhe dar muitas informações inteligentes para ajudar a reduzir custos. Pode-se definir orçamentos diários para que se saiba o quanto está sendo gasto e quando se atinge os limites diários.

Geralmente os medidores inteligentes possuem vários componentes com o objetivo de entregar uma ou mais características citadas anteriormente, tais componentes são uma mescla dos já utilizados nos medidores eletrônicos com o acréscimo de funções de comunicação e controle por meio de atuadores. A Figura 10 mostra o diagrama de blocos de um *smart meter*.

**Figura 10** – Diagrama de blocos genérico de um *smart meter*.



**Fonte:** Cesário Júnior (2014).

De acordo com a Figura 10 um microcontrolador fará o papel de coordenador dos seguintes blocos:

- Exibição de dados: através de uma tela (por exemplo LCD), serão exibidos os dados de consumo e demais parâmetros necessários;
- Comunicação: tanto as concessionárias quanto os consumidores devem ser capazes de ler os dados de consumo através de conexões cabeadas e sem cabos de forma automática e remota usando soluções de comunicação diferentes;
- Sensores de fase: algumas grandezas podem ser medidas (como temperatura) e deve ser possível medir uma, duas ou três fases, obtendo valores em tempo real de tensão e corrente a serem medidas.

### 2.3.2 Smart Plugs

Lee (2010 apud Ferreira, 2012) destaca que os medidores inteligentes são capazes de oferecer informações de consumo como tensão, corrente, potência, impedância, fator de potência, potência aparente, etc. No entanto, esse tipo de equipamento trabalha com a totalidade da instalação, normalmente servindo apenas como o gestor total dos dados elétricos, sem permitir, por parte do consumidor, ter o controle dos pontos de utilização ou de informações precisas acerca dos consumos de equipamentos.

Fabrício (2018) destaca que o avanço da Internet das Coisas possibilitou que esses eletrodomésticos comecem a se tornar inteligentes, com recursos de conectividade que lhes permitem ser automatizados, a fim de aproveitar as vantagens que a medição inteligente e tarifas variáveis podem trazer. O termo *smart plug* surge como uma ramificação dos *smart meters*, pois sua função é conectar dispositivos comuns (eletrodomésticos e máquinas de pequeno porte) em uma rede local e permitir que o usuário possa controlar o fluxo de energia e informações dos mesmos em tempo real.

Dependendo da sua sofisticação, podem ainda se comunicar com um computador ou com um dispositivo móvel, como celular ou *tablet* e enviar informações e dados pela internet permitindo assim o monitoramento à distância. (FERREIRA, 2012).

Os *smart plugs* passam a agir como pequenas unidades de supervisão e controle, semelhantes aos medidores inteligentes instalados pela concessionária de energia, no entanto, o tamanho é compacto e emprega o princípio do “*plug and play*”, normalmente adaptados para o uso em tomadas monofásicas. É importante destacar que estes plugues não representam uma alternativa aos medidores inteligentes, sendo um complemento no conceito dos mesmos, uma vez que os *smart meters* são de propriedade da distribuidora de energia para fins de faturamento.

Os componentes internos de um *smart plug* seguem o mesmo diagrama de blocos da Figura 10, como estes dispositivos contam com a capacidade de interrupção do fornecimento de energia é comum encontrar relés trabalhando com esta função além de contar com métodos de transmissão de dados e comandos por redes sem fio.

Em resumo, Josué (2010 apud Ferreira, 2012) define que existem *smart plugs* destinados a monitorar o consumo total da residência, que ficam localizados junto ao medidor da concessionária ou ao quadro elétrico, e existem os medidores chamados *plug-in*, cujo objetivo é medir o consumo de aparelhos individualmente, esses medidores ficam localizados entre a tomada de fornecimento de energia e o aparelho do qual se deseja medir ou controlar o fornecimento de energia. Entre os modelos comerciais de *smart plugs* destacam-se:

- *Kill a Watt*

O *Kill a Watt* é um dispositivo desenvolvido pela empresa P3 International, do tipo de sistema *plug-in*. Possui um display LCD no qual são mostradas medidas relativas ao consumo de energia do equipamento elétrico ao qual está conectado, entre elas: tensão, corrente, potência ativa, frequência, potência aparente ou energia. Essas medidas permitem, além de analisar o

consumo de um eletrodoméstico, fazer uma análise da qualidade da energia recebida. (WEEMS CREEK SOLUTIONS, 2012 apud FERREIRA, 2012).

Outro modelo do mesmo fabricante é o *Kill a Watt Graphic Timer P4480* que além de medir e monitorar os consumos de energia eléctrica, pode também ser programado para ligar/desligar o equipamento ao qual estiver ligado, o que ajuda a reduzir o consumo. (FERREIRA, 2012).

É importante destacar que o *Kill a Watt* não possui função de armazenamento de dados ou acesso a informações por aplicativo remotamente, os valores medidos são apresentados no display do equipamento. A Figura 11 mostra os dois modelos do aparelho.

**Figura 11** – Kill a Watt, modelo P4400 (a) e 4480 (b).



Fonte: Ferreira (2012).

- *Power Meter*

De acordo com Steplight (2012 apud Ferreira, 2012) o *Power Meter* é mais um dispositivo do tipo *plug-in* que disponibiliza informações sobre consumo de energia de forma precisa e fácil de entender. Entre suas funções estão:

- Medir o consumo de um aparelho em uma faixa de 1 a 2.400W;
- Informar o consumo em kWh;
- Informar o quanto está sendo gasto em dinheiro, a partir da tarifa da energia eléctrica;
- Medição de tensão, fator de potência atual e a frequência;
- Possui ainda um *timer* embutido que controla o tempo de utilização de um aparelho.

Assim como o *Kill a Watt* o *Power Meter* não apresenta a função de salvar os dados para análise posterior e não possui aplicativo de acesso. A Figura 12 mostra o equipamento.

**Figura 12** – *Smart Plug Power Meter*.



**Fonte:** Steplight (2012) apud Ferreira (2012).

- TED 5000 – *The Energy Detective*

Costa (2014) explica que o aparelho TED 5000 realiza as medições na rede elétrica e através de uma conexão sem fio *ZigBee*, disponibiliza os dados em outro dispositivo (módulo display) composto por um LCD (*liquid crystal display*). Através de um roteador os dados também podem ser visualizados em um computador ou smartphone. A Figura 13 mostra o TED 5000 e as telas do sistema supervisorio.

**Figura 13** – TED 5000 *Energy Detective* (a) tela do sistema supervisorio do aparelho (b).



(a)



(b)

**Fonte:** Costa (2014).

- *Snapgrid*

O *Snapgrid* é um gerenciador de consumo energético composto por um painel *touchscreen* e uma CPU de 16 canais, sendo que cada canal recebe pastilhas individuais de medição. Cada pastilha mede o consumo do disjuntor associado, informando à CPU principal (iHOUSE 2012 apud FERREIRA, 2012). A Figura 14 mostra o painel de visualização do *Snapgrid*.

**Figura 14** – Tela de visualização do Snapgrid.



**Fonte:** Ferreira (2012).

Segundo Ferreira (2012) uma vez instalado na porta do quadro de disjuntores da residência, o dispositivo monitora 24 horas por dia o gasto de energia elétrica de ambientes ou equipamentos pré-determinados pelo usuário que possuem um disjuntor independente. O consumo pode ser exibido para o consumidor em Quilowatts-hora ou em dinheiro de três formas diferentes:

- Modo instantâneo, que informa o consumo no exato momento da consulta, sendo atualizado a cada um minuto;
- Opção acumulado, que mostra soma do consumo, iniciado após a leitura pela empresa concessionária de energia elétrica até a data de consulta;
- Ou como previsão para o mês, com estimativa mensal baseada no consumo médio dos últimos dias.

Ferreira (2012) destaca que além disso é possível visualizar ainda a lista de ambientes ou equipamentos que mais consomem ou vêm consumindo energia elétrica. O produto ainda traz o opcional de acesso a relatórios via internet, exibidos em forma de gráficos coloridos e de fácil compreensão, os quais mostram o consumo por ambiente e por equipamento durante o mês, no dia a dia, ou destaca os pontos de maior consumo de energia elétrica. Os relatórios cobrem um período de até 12 meses.

O equipamento não possui acesso remoto através de aplicativo, porém os dados podem ser visualizados em um computador através da comunicação ethernet. A Figura 15 detalha as informações que são mostradas na tela do equipamento.



**Figura 15** – Informações apresentadas na tela de visualização do *snapgrid*.



Fonte: Ferreira (2012).

- *Sonoff Pow*

O Sonoff Pow é um interruptor Wi-Fi que permite controlar o fornecimento de energia elétrica através de um aplicativo móvel, além de monitorar o consumo de energia pelos equipamentos conectados a ele (ITEAD 2015 apud MAAS, 2017). A Figura 16 mostra o *Sonoff Pow* e o esquema de ligação do mesmo.

**Figura 16** – Smart plug *Sonoff* (a) e forma de ligação (b).



Fonte: Itead (2015) apud Maas (2017).

Conforme a Figura 16 os fios de alimentação do aparelho são parafusados nos soquetes de entrada do *Sonoff*. Uma vez instalado e ligado à rede de energia, é necessário conceder acesso a uma rede Wi-Fi, o que pode ser feito através de um smartphone. Segundo Maas (2017) o usuário precisa acessar, através do smartphone, a rede Wi-Fi disponibilizada pelo interruptor, e conceder as credencias de acesso à rede doméstica. Uma vez realizados esses primeiros passos,

o interruptor pode ser controlado através do aplicativo móvel eWeLink através de uma rede Wi-Fi, 2G, 3G ou 4G. As telas do aplicativo podem ser vistas na Figura 17.

**Figura 17** – Aplicativo eWeLink.



**Fonte:** Itead (2015) apud Maas (2017).

No aplicativo, o usuário pode então acompanhar o histórico de consumo de energia do equipamento conectado ao interruptor (B), assim como acompanhar este mesmo consumo em tempo real. Também é possível ligar e desligar o interruptor (A), além de agendar horários de funcionamento (C). O interruptor oferece suporte para tensões entre 110 e 250V AC, permitindo que seja utilizado na maioria dos mercados globais. Além disso, suporta correntes de até 16A e potência máxima de 3500W (ITEAD 2015 apud MAAS, 2017).

## 2.4 INTERNET DAS COISAS

Segundo Atzori et al. (2010 apud Antunes, 2016) a Internet das Coisas é um novo paradigma que surgiu e está rapidamente ganhando espaço nas telecomunicações. A ideia básica desse conceito é a possibilidade de interação, através de uma conexão com a Internet, entre diversos dispositivos chamados de “Coisas” - como Sensores, Celulares, Etiquetas, Lâmpadas, etc. - para alcançar objetivos comuns. Sintaticamente, o nome é composto por dois termos. O primeiro deles, “Internet”, sugere uma interpretação do ponto de vista orientado a rede. Já o segundo, “Coisas”, leva o foco para objetos genéricos conectados. (GREHS, 2016). A Figura 18 mostra a relação entre estes conceitos.

**Figura 18** – Paradigma da Internet das Coisas como um resultado de diferentes visões.



Fonte: Atzori (2010) apud Ferreira (2014).

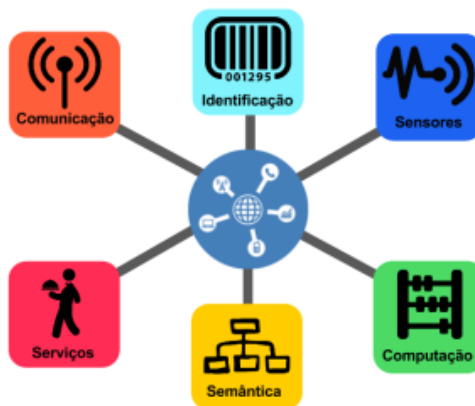
Grehs (2016) ressalta que essas visões podem ser definidas como:

- **Visões orientadas às Coisas:** A primeira definição de IoT deriva de orientada a Coisa. Nessa visão, as Coisas são muito simples, como as etiquetas de *Radio-Frequency IDentification* (RFID). O RFID, e diversas tecnologias semelhantes, como o *Wireless Sensor Network* (WSN). Mas, fica claro que a Internet das Coisas não pode ser apenas um sistema de endereçamento em que os únicos objetos são RFIDs.
- **Visão orientada à Internet:** Segundo Guinard et al. (2009 apud Grehs, 2016) a segunda definição de IoT deriva de orientada à Internet. Nessa visão, as Coisas podem comunicar-se automaticamente com computadores, provendo serviços para o benefício da raça humana. Essa visão propõe uma Internet das Coisas como uma infraestrutura global que conecta objetos físicos, através da estrutura de Internet que já existe, adicionada de captura de dados autônoma, conectividade e interoperabilidade. Essa visão mais próxima da Internet se relaciona com o conceito de *Web of Things*. De acordo com esse conceito, os padrões utilizados na Web são reutilizados para conectar os objetos/coisas do mundo real.

- **Visão orientada à semântica:** Por fim, a definição semântica de IoT diz que o número de dispositivos envolvidos na Internet do futuro é muito maior do que o esperado, o que trará problemas como armazenamento, interconexão, pesquisa e classificação das informações geradas por dispositivos IoT. Nesse cenário, o conhecimento semântico dos dados permitirá o tratamento e extração automatizada dessas informações, o que facilitará aspectos como escalabilidade de armazenamento e infraestrutura de comunicação.

Um dispositivo, mesmo conectado à nuvem, talvez não tire proveito do real potencial da Internet das Coisas. Para isso, precisamos considerar vários aspectos durante o projeto de um dispositivo IoT: entre eles a *hardware*, o *software* embarcado, o protocolo e o meio de comunicação – além do serviço que receberá os dados. (GREHS, 2016). A IoT pode ser vista como a combinação de diversas tecnologias, as quais são complementares no sentido de viabilizar a integração dos objetos no ambiente físico ao mundo virtual. (Santos et al., 2016). A Figura 19 apresenta os blocos básicos de construção da IoT.

**Figura 19** – Blocos básicos da IoT.



**Fonte:** Santos et al. (2016).

Com base na Figura 19 conforme apresentado por Santos et al. (2016) estes blocos podem ser definidos da seguinte forma:

- **Comunicação:** diz respeito às diversas técnicas usadas para conectar objetos inteligentes. Também desempenha papel importante no consumo de energia dos objetos sendo, portanto, um fator crítico. Algumas das tecnologias usadas são Wi-Fi, Bluetooth, IEEE 802.15.4 e RFID.

- **Identificação:** um dos blocos fundamentais, pois é primordial identificar os objetos unicamente para conectá-los à Internet. Tecnologias como RFID, NFC (*Near Field Communication*) e endereçamento IP podem ser empregados para identificar os objetos.
- **Sensores/Atuadores:** coletam informações sobre o contexto onde os objetos se encontram e, em seguida, armazenam/encaminham esses dados para *data warehouse*, *clouds* ou centros de armazenamento. Atuadores podem manipular o ambiente ou reagir de acordo com os dados lidos.
- **Computação:** inclui a unidade de processamento como, por exemplo, microcontroladores, processadores e FPGAs (*Field Programmable Gate Array*), responsáveis por executar algoritmos locais nos objetos inteligentes.
- **Serviços:** a IoT pode prover diversas classes de serviços, dentre elas, destacam-se os *Serviços de Identificação*, responsáveis por mapear Entidades Físicas (EF) (de interesse do usuário) em Entidades Virtuais (EV) como, por exemplo, a temperatura de um local físico em seu valor, coordenadas geográficas do sensor e instante da coleta; *Serviços de Agregação de Dados* que coletam e sumarizam dados homogêneos/heterogêneos obtidos dos objetos inteligentes; *Serviços de Colaboração e Inteligência* que agem sobre os serviços de agregação de dados para tomar decisões e reagir de modo adequado a um determinado cenário; e *Serviços de Ubiquidade* que visam prover serviços de colaboração e inteligência em qualquer momento e lugar em que eles sejam necessários.
- **Semântica:** refere-se à habilidade de extração de conhecimento dos objetos na IoT. Trata da descoberta de conhecimento e uso eficiente dos recursos existentes, a partir dos dados, com o objetivo de prover determinado serviço. Para tanto, podem ser usadas diversas técnicas como *Resource Description Framework (RDF)*, *Web Ontology Language (OWL)* e *Efficient XML Interchange (EXI)*.

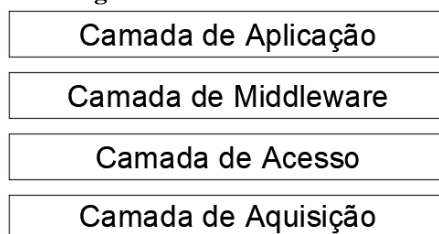
### 2.4.1 Arquitetura e Organização

Para conectar bilhões de objetos inteligentes à Internet, deve-se ter uma arquitetura flexível. Na literatura, existe uma variedade de propostas de arquiteturas, que se baseiam nas necessidades da academia e indústria (AL-FUQAHA et al. 2015 apud SANTOS et al., 2016).

Segundo Grehs (2016) uma das iniciativas para padronizar a arquitetura utilizada por soluções de IoT foi o projeto IoT-A, um consórcio europeu de empresas formado de 2010 a 2013, com o intuito de definir um modelo de referência. Apesar de existirem modelos de diferentes autores, todas metodologias são semelhantes e podem ser separadas em quatro

camadas genéricas: aquisição de dados, acesso, *middleware* e aplicação. De acordo com as características dos dispositivos envolvidos e sofisticação da aplicação, as camadas serão mais ou menos complexas. Em outro extremo, podem até ser suprimidas. A Figura 20 mostra as camadas que compõem o modelo de arquitetura da Internet das Coisas.

**Figura 20** – Camadas da IoT.



**Fonte:** Autoria Própria.

A camada de aquisição tem duas funções básicas, obtenção de dados físicos e comunicação. Dentro dessa camada, estão localizados os sensores, atuadores, sistemas embarcados e etiquetas RFID. Esses *hardwares* serão os responsáveis por prover identificação, coletar informações físicas do meio em que se encontram, atuar mecanicamente no meio e permitir o envio/recebimento de informações sobre tais sensores e identificadores para as camadas posteriores, vale salientar que a comunicação dessa camada é apenas com a camada de acesso, e não com o usuário final da aplicação. (FLEISCH 2010 apud GREHS, 2016).

A camada de acesso é responsável pelo primeiro tratamento dos dados recebidos, ou seja, executar atividades necessárias para que esses dados cheguem às próximas camadas de forma pertinente (YANG et al., 2011 apud GREHS). Entre essas atividades estão a comunicação com a Internet, normalmente utilizando Wi-Fi ou tecnologias de dados celular, como *Universal Mobile Telecommunications System* (UMTS), *Global System for Mobile communications* (GSM) e *Long Term Evolution* (LTE). (GREHS, 2016).

Zarghami (2013 apud Grehs, 2016) a camada de *middleware* possui como funcionalidades principais:

- Ser a interface com protocolos da camada de acesso;
- Oferecer abstração de dispositivo;
- Atuar sobre o contexto dos dados e oferecer abstração para a aplicação.

Ferreira (2014) destaca que para atingir a primeira funcionalidade, a camada de *middleware* deve ser capaz de definir os protocolos de troca de informação com as camadas inferiores, independente das diferenças físicas que podem existir entre os dispositivos. A

segunda funcionalidade, abstração de dispositivo, será obtida através da definição da sintaxe das mensagens que serão transmitidas pelos protocolos de comunicação. Essa camada também deve ser capaz de atuar sobre o contexto dos dados, ou seja, coletar as informações oferecidas pelos dispositivos, e, posteriormente, selecionar a informação que pode ter algum impacto sobre o processamento, como a necessidade de tomar uma decisão. A última função fundamental da camada de *middleware* é oferecer abstração para a aplicação IoT, ou seja, esse componente deve prover uma interface para comunicar-se com a aplicação e o cliente final.

Na camada de aplicação ocorre a entrega das funcionalidades aos usuários. Essa camada é responsável por apresentar os recursos oferecidos pelo sistema e permitir atuações e tarefas de acordo com as funcionalidades. Essa aplicação se encontra em aplicativos de celular ou websites, de fácil utilização pelos usuários. Com o aumento da maturidade de diversas tecnologias, tais aplicações também podem estar transparentes aos usuários, como casos de utilização de IoT em *Smart Grid*, logística e dispositivos inteligentes. (SONEGO, 2016).

Santos et al. (2016) destaca que as redes de objetos inteligentes serão parte de nossas vidas nos próximos anos. Por isso, entender as bases da IoT no que tange seus paradigmas de comunicação e modelos de conectividade é de grande importância, pois estes dois quesitos serão as chaves para construir novas aplicações para a IoT.

## 2.4.2 Protocolos de Comunicação

Grehs (2016) destaca que um dos aspectos mais importantes para o desenvolvimento da Internet das Coisas é a padronização da comunicação. Esse ponto tem sido muito discutido nos últimos anos, trazendo consigo uma gama de protocolos para atender aos requisitos das aplicações da melhor maneira possível. Como dito anteriormente, baseado na visão, em que se espera uma integração entre os dispositivos IoT e a Internet, a escolha do protocolo IP para endereçamento é a mais adequada, entretanto, é necessário definir qual o protocolo de nível de aplicação será utilizado, sobre o IP, e é nesse ponto que a gama de protocolos é importante.

Segundo Santos et al. (2016) o protocolo HTTP é usado na Internet para acessar informações seguindo a estratégia requisição/resposta no paradigma cliente/servidor. O HTTP foi desenvolvido para redes com computadores tipo PC. Diferentemente dos PCs, os dispositivos usados na IoT possuem poder computacional restrito, o que limita a utilização do protocolo HTTP nesses elementos. Entre os problemas no uso do HTTP podem-se destacar:

- O HTTP é um protocolo síncrono. O cliente espera que o servidor responda. Os navegadores da web têm esse requisito, mas o custo é a baixa escalabilidade. Os sensores podem enviar leituras e permitir que a rede descubra o caminho e a sincronização ideais para entregar aos dispositivos e serviços de destino;
- É unidirecional. O cliente precisa iniciar a conexão. Na IoT, os dispositivos e sensores geralmente são clientes, eles não podem receber comandos da rede passivamente;
- É um protocolo pesado com muitos cabeçalhos e regras. Inadequado para redes restritas.

Para resolver esse problema, foram desenvolvidos dois protocolos da camada de aplicação especificamente para recuperar informações de dispositivos com baixo poder computacional: MQTT e CoAP.

#### 2.4.2.1 MQTT - *Message Queue Telemetry Transport*

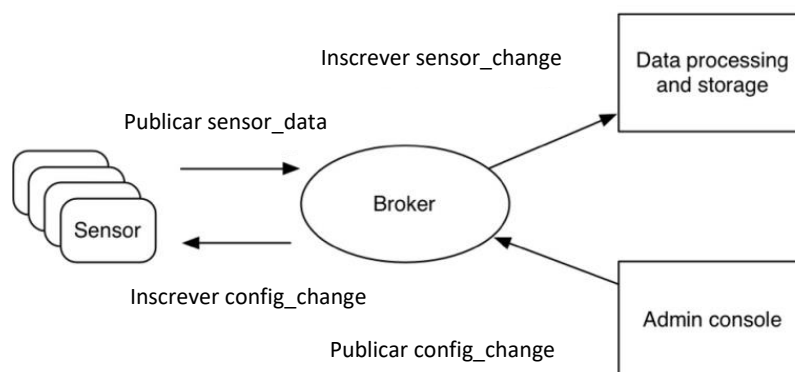
O MQTT é um protocolo de mensagem leve e simples baseado em *publish/subscribe* (publicar/inscrever). Os seus princípios são minimizar a largura de banda e os recursos dos dispositivos, enquanto também assegura confiabilidade e segurança de comunicação. Estes princípios o fazem ideal para a comunicação “*machine-to-machine*” (M2M) ou máquina a máquina, para a Internet das Coisas e para aplicações móveis (TOMAS, 2014).

Seguindo a arquitetura do MQTT, os dispositivos devem conectar-se a um servidor capaz de concentrar todas as informações. Comumente, o dispositivo é responsável por enviar (*publish*) as informações ao servidor, que opera como intermediário (*broker*), tendo conhecimento dos clientes que estão interessados nas informações enviadas (*subscribes*). (GREHS, 2016).

O protocolo define dois tipos de indivíduos na rede, um *message broker* e um número de clientes. O *message broker* é um servidor que recebe todas as mensagens dos clientes e encaminha estas mensagens aos clientes destinatários. Um cliente é qualquer dispositivo que pode interagir com o *broker* e receber ou enviar mensagens. Um cliente pode ser um sensor no campo ou uma aplicação na plataforma que processa informação (YUAN 2017 apud PITON, 2017). A Figura 21 ilustra o modelo do protocolo MQTT.



**Figura 21** – O modelo de *publish/subscribe* para sensores em IoT.



**Fonte:** Yuan (2017) apud Piton (2017).

Segundo Piton (2017) as mensagens MQTT são organizadas por *topics* (tópicos), assim existe a flexibilidade de especificar quais clientes tem acesso a certas mensagens. O cliente se conecta ao *broker* e pode se inscrever ou publicar para qualquer tópico do *broker*. Na Figura 25, sensores publicam suas mensagens no tópico de “*sensor\_data*” e se inscrevem no tópico “*config\_change*”, aplicações de processamento de dados se inscrevem no tópico “*sensor\_change*” e um “*Admin Console*” recebe comandos e ajusta as configurações dos sensores e publica no tópico “*config\_change*”.

A utilização do MQTT permite o desenvolvimento de diversas aplicações, desde o monitoramento de vazamentos no meio industrial até dispositivos de segurança residencial. Qualquer outra aplicação em que a necessidade fundamental é dividir informações coletadas de diversas fontes e torná-la disponível ao usuário pode se beneficiar do uso do MQTT. (GREHS, 2016).

#### 2.4.2.2 CoAP - *Constrained Application Protocol*

O CoAP é um protocolo de transferência de documentos que, ao contrário do HTTP, é desenhado para atender às necessidades de dispositivos restritos. O protocolo é especializado em atuar em nós e redes sujeitas a restrições no âmbito da IoT, sendo desenhado para desempenhar comunicação máquina-máquina em aplicações como automação residencial e energia inteligente (TOMAS, 2014).

De acordo com Jaffey (2014 apud Maas, 2017), para reduzir o consumo de banda, “mapas de bits e mapeamentos de strings para inteiros são usados extensivamente para salvar espaço”. Essa estratégia garante que a análise e geração dos pacotes sejam simples e exijam poucos recursos em sistemas restritos. O protocolo segue um modelo cliente/servidor e oferece

as operações GET, PUT, POST e DELETE entre cliente e servidor, e uma das suas principais características é a interoperabilidade com o protocolo HTTP, através do uso de proxies responsáveis por adaptar as requisições entre um formato e o outro.

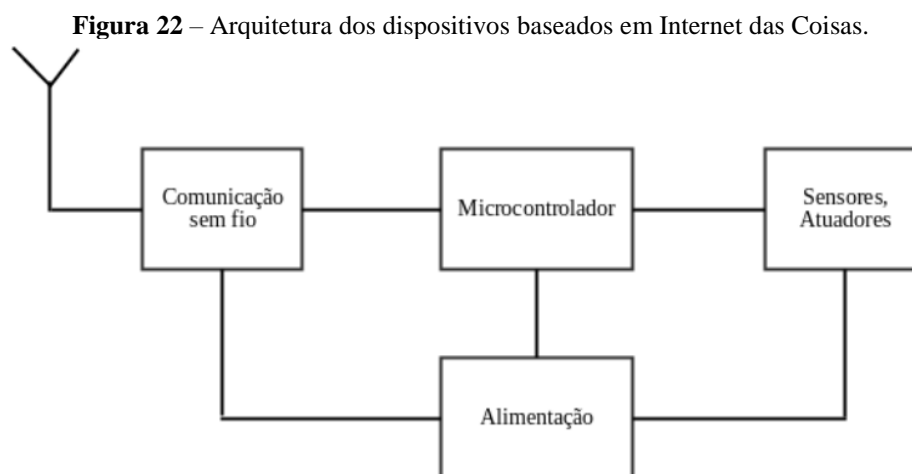
Jaffey (2014 apud Maas, 2017) destaca que ao contrário do protocolo HTTP, que opera em cima do protocolo TCP (*Transmission Control Protocol*), o CoAP opera com o protocolo UDP (*User Datagram Protocol*). Isso permite que cliente e servidor se comuniquem através de datagramas que não dependem de conexão. Assim, o protocolo não garante a entrega das mensagens, tratamentos de erro e estratégias de tolerância são de responsabilidade da aplicação. Em termos de segurança, o protocolo UDP permite o mesmo nível de segurança que o HTTPS através do emprego da *Datagram Transport Layer Security* (DTLS).

No protocolo CoAP um cliente pode observar um recurso do servidor. Desta forma, o servidor manterá uma conexão ativa com este cliente, informando-o de cada mudança de estado daquele recurso na medida em que elas acontecem (JAFNEY, 2014 apud Santos et al. 2016).

Neste trabalho será usado o protocolo MQTT devido a sua praticidade, considerando o tempo de resposta e o baixo consumo, além do suporte a múltiplas conexões e manutenção da confiabilidade e segurança no transporte de mensagens.

### 2.4.3 Tecnologias Relacionadas

A arquitetura básica dos objetos inteligentes e da Internet das Coisas é composta por quatro unidades: processamento/memória (microcontrolador), comunicação, alimentação (fornecimento de energia) e sensores/atuadores. A Figura 22 apresenta uma visão geral da arquitetura de um dispositivo e a interligação entre seus componentes.



Fonte: Grehs (2016).

A Unidades de processamento é composta de uma memória interna para armazenamento de dados e programas, um microcontrolador e um conversor analógico-digital para receber sinais dos sensores. As CPUs utilizadas nesses dispositivos são, em geral, as mesmas utilizadas em sistemas embarcados e comumente não apresentam alto poder computacional.

Frequentemente existe uma memória externa, para manter um “log” de dados. As características desejáveis para estas unidades são consumo reduzido de energia e ocupar o menor espaço possível, comumente, utiliza-se microcontroladores com a aparência de um microchip tradicional. Exemplos de tecnologias empregadas no ambiente da Internet das Coisas são as plataformas Arduino e Raspberry Pi, ou microcontroladores das famílias PIC e ATMEGA. Para a o desenvolvimento do *smart plug* deste trabalho foi escolhido o ATMEGA 328, mais especificamente a plataforma Arduino, em sua versão Nano.

A Unidade de comunicação consiste de pelo menos um canal de troca de informações utilizando ou não, uma transmissão por meio físico. Para comunicações Wi-Fi, a maioria das plataformas são de baixo custo e baixa potência. As formas de comunicação podem ser o Ethernet (IEEE 802.3), Wi-Fi (IEEE 802.11), Bluetooth (IEEE802.15.1) ou o ZigBee. (SANTOS et al. 2016). Para este trabalho foi utilizada a comunicação Wi-Fi para envio das informações entre os aparelhos medidos e o usuário.

A unidade de alimentação é responsável por fornecer energia aos componentes do dispositivo. Normalmente, a fonte de energia consiste de uma bateria (recarregável ou não) e um conversor CA-CC e tem a função de alimentar os componentes. Entretanto, existem outras fontes de alimentação como energia elétrica, solar e mesmo a captura de energia do ambiente através de técnicas de conversão, conhecidas como *energy harvesting*. (GREHS, 2016). Neste trabalho será utilizada uma fonte de tensão encapsulada de baixa potência para alimentação dos periféricos do *smart plug*.

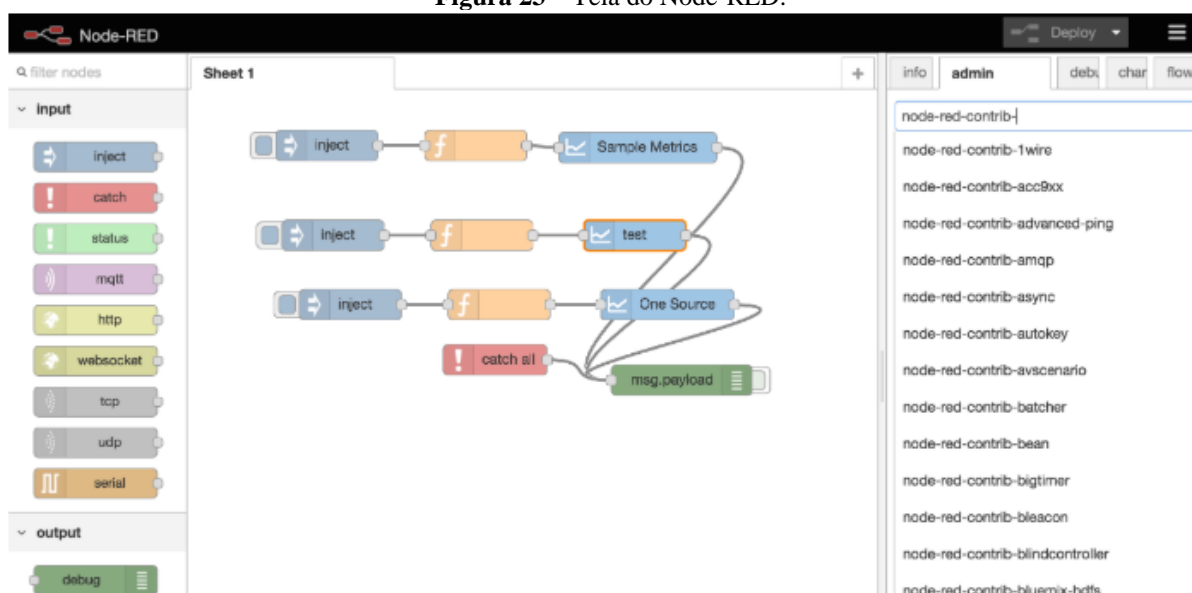
A unidade de sensores e atuadores realiza o monitoramento do ambiente no qual o objeto se encontra. Os sensores capturam valores de grandezas físicas como temperatura, umidade, pressão e presença. Atualmente, existem literalmente centenas de sensores diferentes que são capazes de capturar essas grandezas. Atuadores, como o nome indica, são dispositivos que produzem alguma ação, atendendo a comandos que podem ser manuais, elétricos ou mecânicos. (FERREIRA, 2012). Neste trabalho será usado um transformador de corrente como sensor de corrente e um transformador de potencial como sensor de tensão.

### 2.4.3.1 Node-RED

O *Node-RED* proporciona um ambiente gráfico de programação de fluxo de mensagens. Trata-se de uma plataforma web construída sobre a estrutura Node.js. O Node.js se utiliza da linguagem de programação *Javascript* para o desenvolvimento de aplicações do lado do servidor.

A plataforma permite construir um gráfico composto de nós (*nodes*), os quais podem ser mensagens de entrada e saída ou uma lógica. As mensagens são passadas pelos nós, podendo ser processadas ou transformadas. Alguns nós também podem ser usados para armazenar dados. A Figura 23 mostra uma visão da estrutura do *Node-RED*.

**Figura 23** – Tela do Node-RED.



**Fonte:** *nodered.org* (2019).

É no servidor onde se executam os nós que tratam os fluxos de mensagens. Por rodar sobre Node.js, ter acesso as suas bibliotecas, e permitir a criação de novos nós pela comunidade (compostos de arquivos HTML e *Javascript*), o *Node-RED* permite grande flexibilidade para integrar diferentes tecnologias. Essa flexibilidade e facilidade de uso o tornou uma opção interessante para implementar sistemas supervisórios.

### 2.4.3.2 Blynk

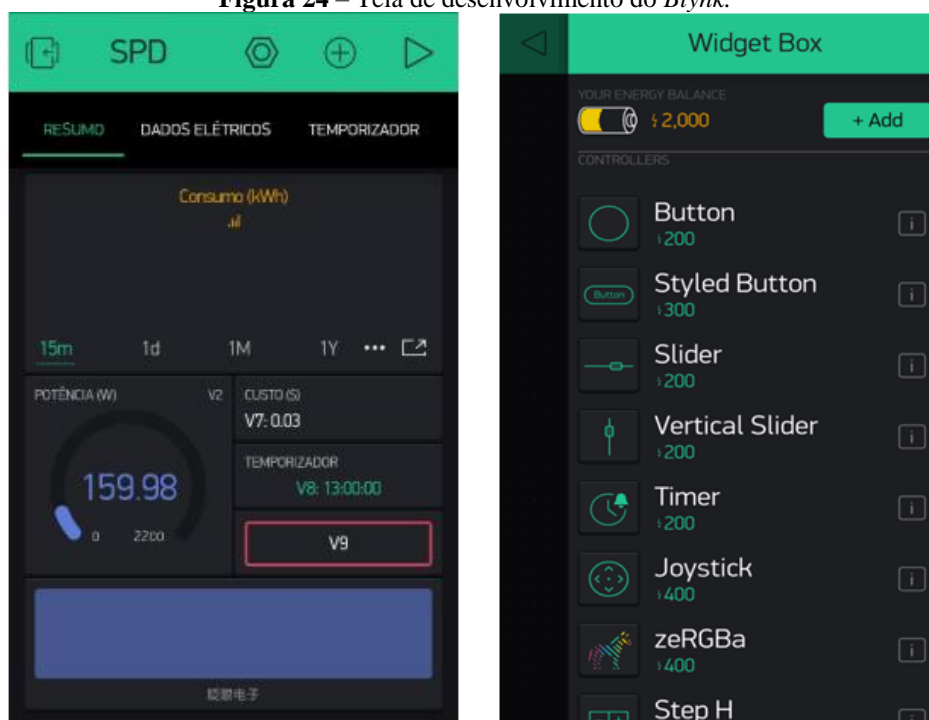
O *Blynk* é uma plataforma desenvolvida para iOS e Android para controlar Arduino, *Raspberry Pi* e outros dispositivos através da internet de uma forma simplificada. Consiste em

um painel digital onde é possível construir uma interface gráfica para projetos, simplesmente “arrastando e soltando” *widgets*, sem complicações com plataformas mais complexas.

O *Blynk* não está vinculado a algum dispositivo ou *shield* específico. Se um Arduino ou *Raspberry Pi* está ligado à internet através de Wi-Fi, Ethernet ou o chip ESP8266, o *Blynk* poderá apresentar todas as informações importantes processadas que o projetista desejar.

O *Blynk* foi um projeto desenvolvido pela *Massachusetts Institute of Technology* (MIT), com a finalidade de contribuir para o ensino e a aderência de mais pessoas para o conceito de Internet das Coisas. Na Figura 24, pode ser demonstrada sua interface. No lado direito da imagem, estão algumas opções de *widgets*, no lado esquerdo da imagem, os *widgets* já estão adicionados ao aplicativo em desenvolvimento.

**Figura 24** – Tela de desenvolvimento do *Blynk*.



**Fonte:** Autoria Própria.

Neste trabalho será utilizada a plataforma *Blynk* para a visualização e controle do dispositivo, sua escolha se deu à simplicidade no desenvolvimento do aplicativo, bem como a possibilidade de ser instalado em qualquer dispositivo móvel, *smartphone* ou *tablet*.

### 3. DESENVOLVIMENTO DO *SMART PLUG*

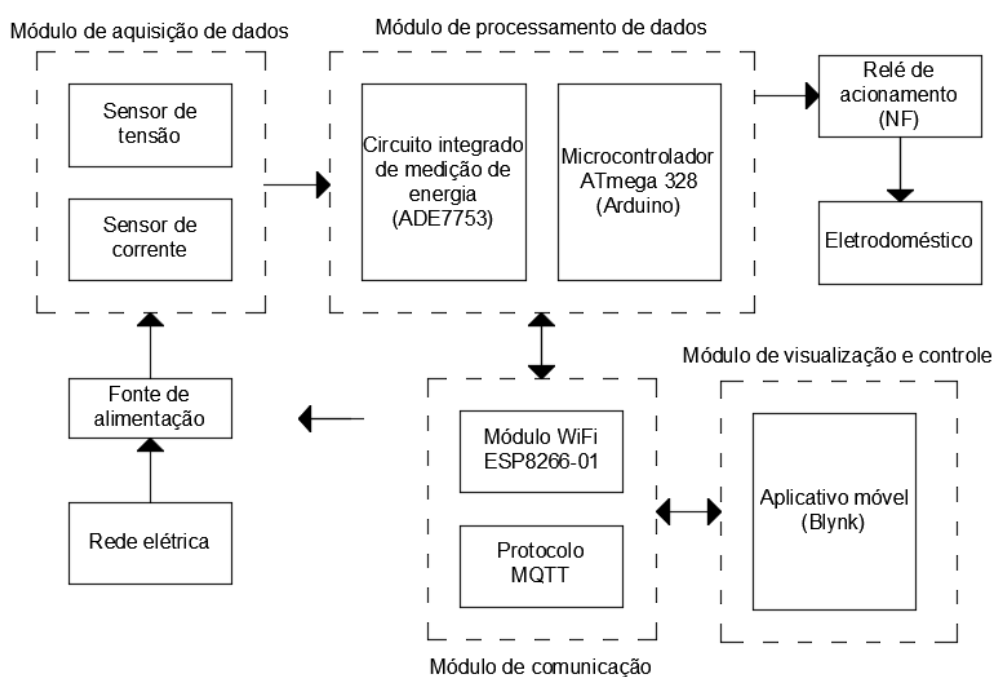
Neste capítulo serão apresentados os procedimentos técnicos e componentes utilizados para execução deste trabalho. Inicialmente será abordado o modelo do *smart plug*, dividido em módulos com funcionalidades específicas, nas seções subsequentes são apresentados os testes realizados para verificação das funcionalidades do plugue.

#### 3.1 MODELO PROPOSTO

A finalidade do *smart plug* é monitorar as grandezas elétricas de um equipamento que esteja diretamente conectado em seus terminais, bem como permitir o acionamento ou desligamento dos aparelhos remotamente, utilizando um aplicativo. O projeto foi dividido em quatro módulos, sendo estes: aquisição de dados, processamento de dados, comunicação e visualização e controle.

O primeiro consiste na captação dos valores de tensão e da corrente demandados pelo equipamento conectado ao plugue. No processamento serão realizadas a recepção dos dados e os cálculos de potência ativa, aparente, fator de potência e consumo. O módulo de comunicação é o responsável por enviar os dados remotamente do plugue até o dispositivo móvel ou computador. O módulo de visualização e controle é responsável pela exibição dos dados medidos, bem como o acionamento do aparelho conectado ao plugue. A Figura 25 mostra o diagrama de blocos do *smart plug* com os seus módulos e principais componentes.

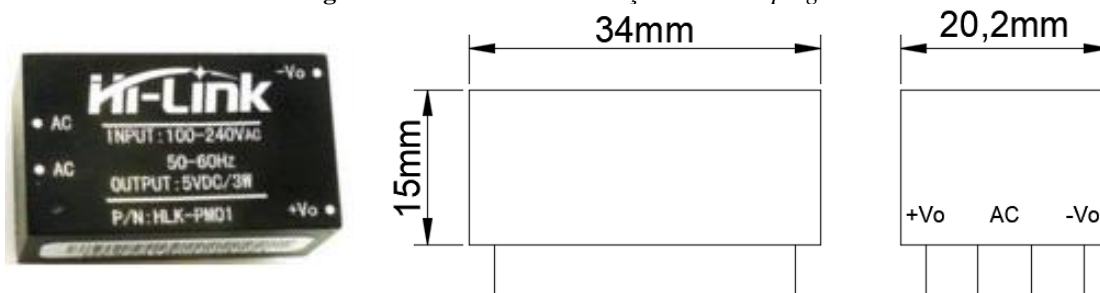
**Figura 25** – Diagrama de blocos do *smart plug*.



Fonte: Autoria Própria.

A alimentação do *smart plug* foi realizada por uma fonte encapsulada, recebendo até 240V da rede elétrica e fornecendo uma tensão em corrente contínua de 5V e potência de até 3W, necessária para o acionamento do microcontrolador, do CI de medição e demais periféricos. A fonte utilizada pode ser visualizada na Figura 26.

**Figura 26** – Fonte de alimentação do *smart plug*.



Fonte: Autoria Própria.

### 3.1.1 Módulo de Aquisição de Dados

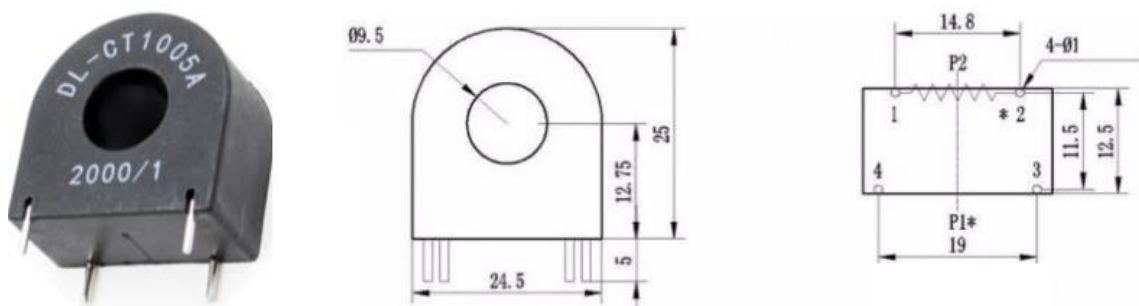
Este módulo é formado pelo transformador de corrente DL-CT1005A e pelo sensor de tensão, que neste projeto foi representado pelo transformador de potencial T-211259-R. Conforme destacado na seção anterior, o objetivo deste módulo é obter os valores de tensão e corrente do equipamento elétrico conectado ao plugue e desta forma obter as demais grandezas elétricas. Este módulo comunica-se diretamente com o módulo de processamento, sendo os dados dos sensores enviados ao circuito integrado de medição, o ADE7753.

- Medição de corrente

A leitura de corrente foi feita através de um sensor não invasivo, que recebeu em seu terminal primário a corrente nominal da carga, disponibilizando um valor proporcional no secundário. A conexão para medição foi realizada conectando o condutor fase através do orifício central do transformador de corrente (terminal primário).

O sensor escolhido, o transformador de precisão DL-CT1005A, suporta até 10A no terminal primário, em seu secundário a saída máxima é de 5mA, o que o adequa aos níveis de corrente das tomadas residenciais. A precisão do sensor é menor ou igual a 0,3%. O transformador utilizado e suas dimensões, pode ser visto na Figura 27.

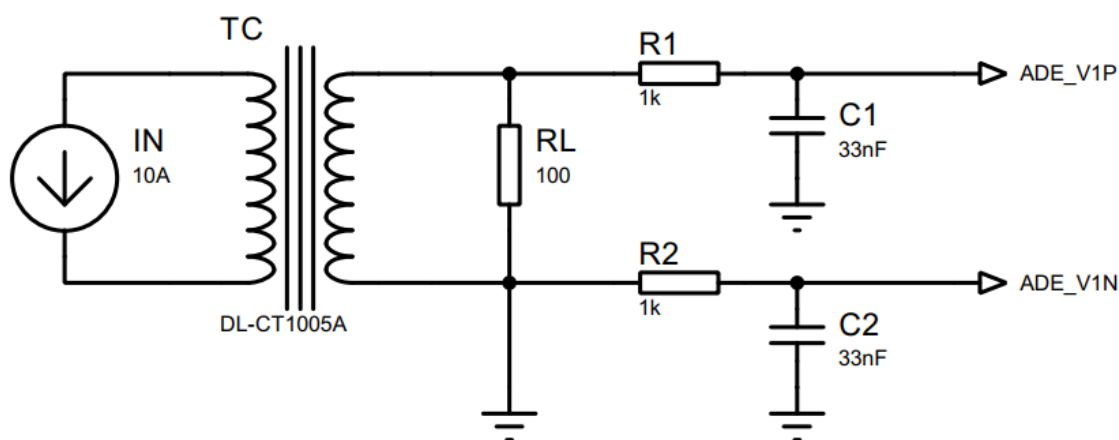
**Figura 27** – Transformador de corrente DL-CT1005A (dimensões em milímetros).



Fonte: Autoria própria.

O ADE7753 necessita que seu terminal de leitura de corrente receba um sinal máximo de 0,5V, para adequar-se aos níveis de tensão, utilizou-se um resistor de carga (RL) no valor de  $100\Omega$ , atendendo aos requisitos do CI de medição. A saída do transformador de corrente, após o resistor de carga, envia os dados para os pinos V1P e V1N do ADE7753. O circuito de medição de corrente está representado na Figura 28.

**Figura 28** – Circuito para medição de corrente.



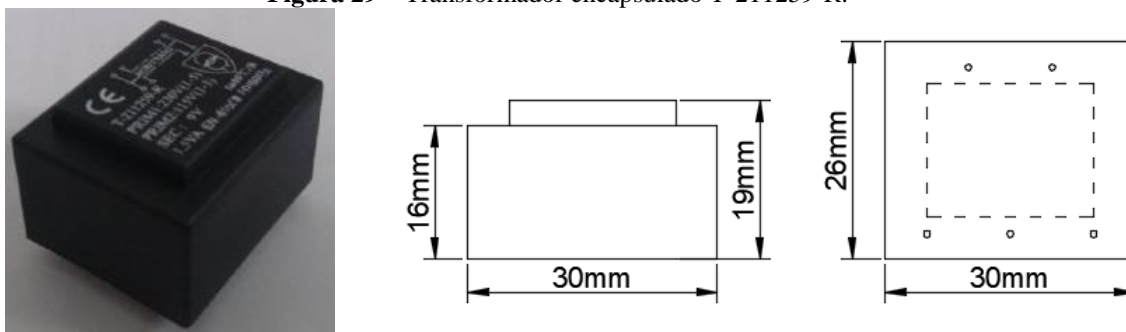
Fonte: Autoria própria.

- Medição de tensão

A leitura da tensão foi realizada conectando o transformador de potencial em paralelo com os condutores da rede elétrica. O sensor escolhido, o transformador de potencial T-211259-R, suporta até 230V em seu terminal primário, disponibilizando no secundário uma tensão de 13,5V (para uma tensão de 220V). A precisão do sensor é menor ou igual a 0,8%. O transformador utilizado e suas medidas pode ser visto na Figura 29.



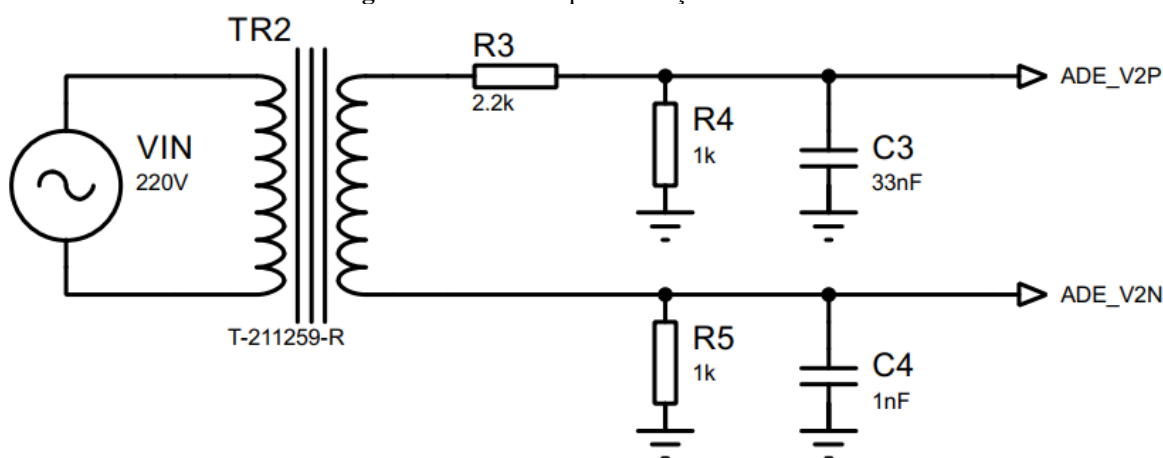
**Figura 29** – Transformador encapsulado T-211259-R.



Fonte: Autoria Própria.

O transformador atua reduzindo a tensão da rede de 220V (valor RMS) para um valor de 13,5V, ainda inadequado para a entrada do ADE7753. Visando condicionar o valor de saída do sensor para que o mesmo atenda ao requisito da entrada do ADE7753 foi usado um divisor de tensão conectado entre os terminais do transformador, reduzindo a tensão de saída para aproximadamente 0,35V, atendendo ao requisito de entrada do CI de medição, que tem uma máxima tensão de entrada de 0,5V. O circuito de medição pode ser visto na Figura 30.

**Figura 30** – Circuito para medição de tensão.



Fonte: Autoria Própria.

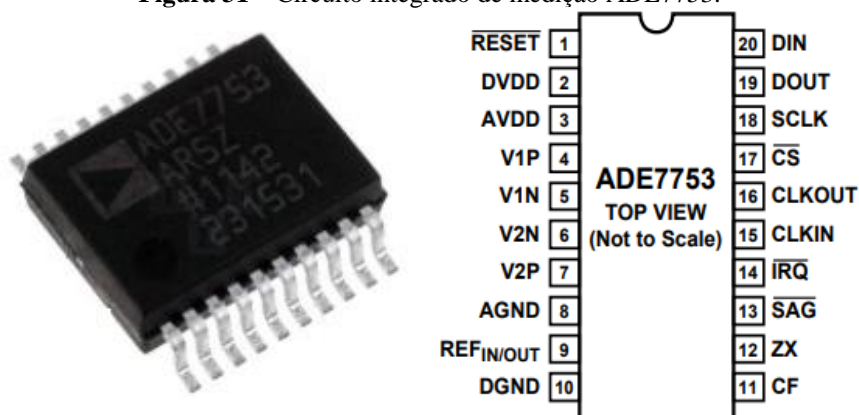
### 3.1.2 Módulo de Processamento de Dados

Este módulo é o responsável por processar as informações vindas do módulo de aquisição e dos demais blocos, funcionando como gerenciamento de dados do *smart plug*. O processamento das informações vindas dos sensores foi realizado por duas unidades, o circuito integrado de medição ADE7753 e o ATMEGA328 (Arduino Nano).

- Circuito integrado de medição ADE7753

Para medição de energia elétrica, foi utilizado o circuito integrado de medição ADE7753 da Analog Devices, mostrado no Capítulo 2. A escolha deste modelo, ocorreu pela disponibilidade de comunicação SPI, que possui calibração por *software* e acesso a outras informações relevantes, tais como picos (valor acima da referência de 220V) e “afundamentos” de tensão (muito abaixo da referência), bem como quedas de energia (tensão nula). O ADE7753 e a listagem dos pinos podem ser vistos na Figura 31.

**Figura 31** – Circuito integrado de medição ADE7753.

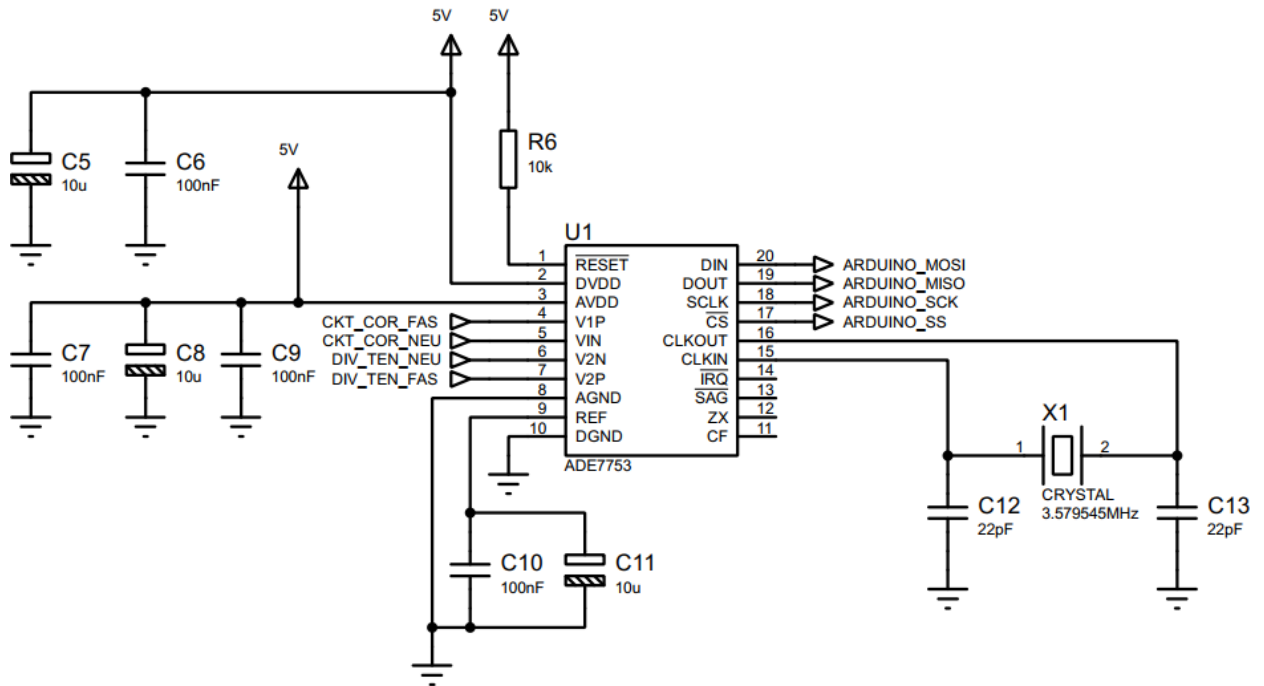


Fonte: Analog Devices (2018).

Conforme a Figura 31, o ADE7753 possui dois canais de entradas analógicas (V1 e V2) ambos limitados a  $\pm 0,5$  V em relação ao sinal de terra analógico (AGND). No canal 1 (V1P e V1N), foi realizada a medição de corrente utilizando o transformador DL-CT1005A (apresentado na seção 3.1.1), sendo utilizado o circuito da Figura 28 para condicionar uma entrada de no máximo 0,5V. A medição de tensão foi realizada através do canal 2 (V2P e V2N), utilizando o transformador T-211259-R e o circuito divisor de tensão da Figura 30.

De acordo como *datasheet* do ADE7753, foram acrescentados resistores e capacitores específicos para os canais de leitura de tensão e corrente e nos pinos AVDD, DVDD e RESET. Bem como um oscilador de 3,58 MHz. A comunicação do ADE7753 com o microcontrolador foi realizada através do protocolo SPI, por meio dos pinos DIN, DOUT, SCLK e CS, que por sua vez, conectaram-se aos pinos correspondentes do Arduino Nano: MISO, MOSI, SCK e SS, respectivamente. A Figura 32 ilustra as ligações do ADE7753.

**Figura 32** – Circuito de medição do ADE7753.

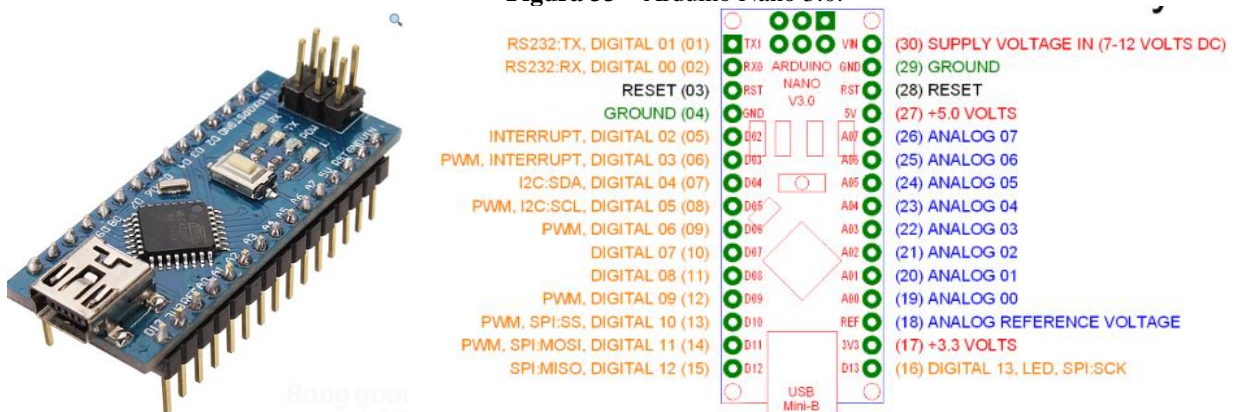


Fonte: Autoria Própria.

- Arduino Nano

O microcontrolador ATMEGA328, presente na placa de prototipagem Arduino Nano 3.0, foi o responsável por coletar as informações do ADE7753 através de comunicação SPI, bem como comunicar-se enviando e recebendo informações do módulo de comunicação. A escolha do modelo Arduino Nano deu-se por seu tamanho reduzido, tendo seu desempenho em *hardware* semelhante ao Arduino UNO. A Figura 33 mostra a plataforma de prototipagem e a identificação dos pinos.

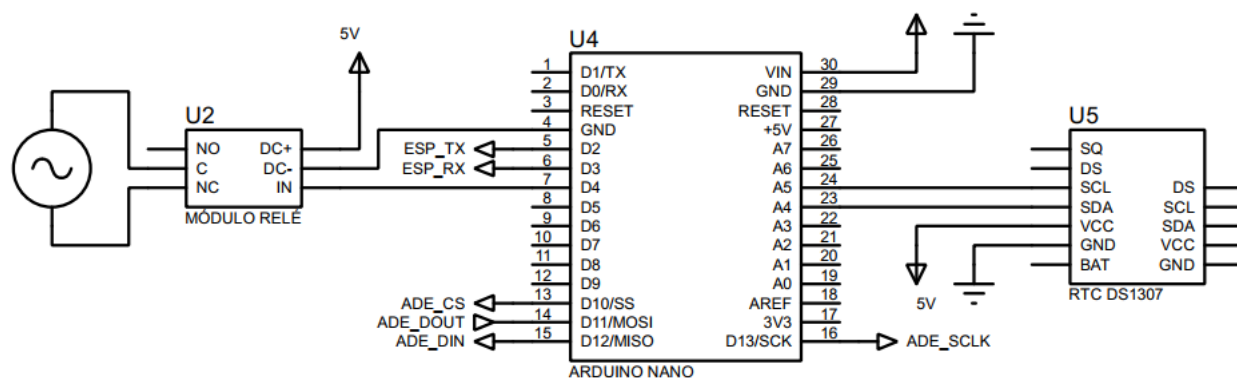
**Figura 33** – Arduino Nano 3.0.



Fonte: Usinainfo (2018).

O Arduino Nano comunicou-se com o ESP8266-01 através das portas seriais RX, TX, ao mesmo tempo em que comandou o acionamento do relé, que irá desligar ou acionar a carga, utilizando a porta digital D4. Para o controle de temporização e cálculo do consumo, foi utilizado o módulo RTC (*Real Time Clock*) DS1307, que também se comunica diretamente com o Arduino. O circuito de processamento pode ser visto na Figura 34.

**Figura 34** – Conexões do Arduino Nano com o relé e o DS1307.

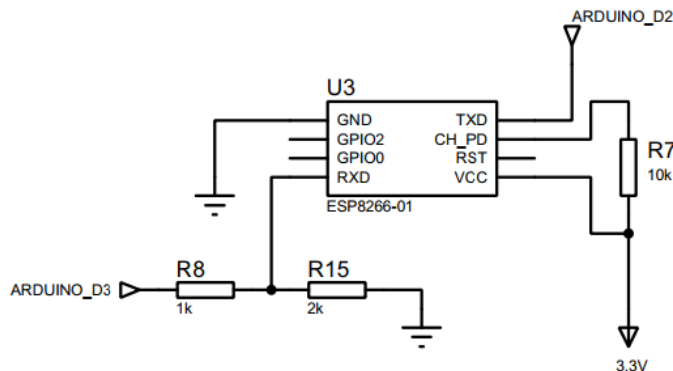


Fonte: Autoria Própria.

### 3.1.3 Módulo de Comunicação

Para envio dos dados coletados e solicitações através do aplicativo para o microcontrolador, foi utilizado o módulo Wi-Fi ESP8266-01. O modelo foi escolhido por se tratar de uma solução completa, para envio de dados *wireless* além de ser bastante utilizado em aplicações de internet das coisas, comunicando-se de forma eficaz com o Arduino. A alimentação do ESP8266-01, é em nível de 3.3V, diferentemente dos outros componentes, o que demandou uma redução na tensão de alimentação de 5V. O circuito de comunicação pode ser visto na Figura 35.

**Figura 35** – Circuito do ESP8266-01 para o módulo de comunicação.



Fonte: Autoria Própria.

O protocolo de comunicação utilizado foi o MQTT, apresentado no Capítulo 2, utilizou-se o broker *Eclipse Mosquitto*, que foi instalado em um *Raspberry Pi*, funcionando como servidor local para a troca de informações e comandos, a partir da interface criada utilizando o aplicativo blynk, foi possível monitorar e controlar os eletrodomésticos conectados.

A opção por um servidor local visou apenas testar a comunicação entre os dispositivos, sendo possível utilizar um broker público ou privado, disponibilizando acesso em qualquer lugar, para fins de projeto. O *smart plug* desenvolvido neste trabalho possui acesso restrito apenas à rede local onde está inserido.

### 3.1.4 Módulo de Visualização e Controle

Conforme definido no projeto, a visualização e controle de informações foram feitas através de um aplicativo para *smartphones*. Para a construção do aplicativo foi utilizada a plataforma Blynk, desenvolvida para iOS e Android para controlar placas de desenvolvimento como o Arduino, *Raspberry Pi*, *NodeMCU* e outros dispositivos através da Internet.

A plataforma consiste em um painel digital onde é possível construir uma interface gráfica para projetos, o Blynk possui um servidor gratuito para o envio e recebimento de informações, porém, neste projeto o servidor utilizado foi a rede local, funcionando a partir do *Raspberry Pi* citado na seção 3.1.3.

Foram desenvolvidas 3 abas no aplicativo, a primeira, chamada “RESUMO”, que apresenta as principais informações. A aba e seus componentes podem ser vistos na Figura 36.

**Figura 36** – Aba “RESUMO” do aplicativo para monitoramento e controle do *smart plug*.



Fonte: Autoria Própria.

Conforme a Figura 36, esta aba comporta um resumo de informações elétricas e o controle do *smart plug*, servindo como uma visualização rápida das principais funções, sendo também a aba de visualização padrão ao iniciar o aplicativo; seus componentes são:

- Consumo do aparelho conectado ao *smart plug* em forma de gráfico, exibindo os dados acumulados a cada 15 minutos, por dia (d), mês (M) ou ano (Y);
- Potência ativa do equipamento em tempo real;
- Custo gerado pelo equipamento, apresentando um valor acumulado;
- Botão Liga/Desliga, acionando o equipamento conectado ao plugue;
- Temporizador para ajustar o horário de desligamento, funcionando como *timer*;
- *Display* informando se a carga conectada ao plugue está ligada ou desligada.

A aba “DADOS ELÉTRICOS” mostra todas as grandezas elétricas do aparelho conectado ao *smart plug*, além de apresentar o gráfico do consumo em tempo real, nesta aba é possível ver detalhadamente cada uma das grandezas elétricas medidas, a Figura 37 mostra a aba e seus componentes.

**Figura 37** – Aba “DADOS ELÉTRICOS” do aplicativo.



**Fonte:** Autoria Própria.

Conforme a Figura 37 a aba apresenta todos os dados elétricos coletados do equipamento que está conectado ao *smart plug*, sendo estas informações:

- Consumo do equipamento conectado ao *smart plug* em forma de gráfico, exibindo os dados acumulados a cada 15 minutos, por dia (d), mês (M) ou ano (Y);
- Custo gerado pelo equipamento, apresentando um valor acumulado;
- Tensão fornecida ao aparelho;
- Corrente elétrica em valor instantâneo;
- Fator de potência da carga;
- Potência ativa, reativa e aparente.

A aba “TEMPORIZADOR” é a responsável por controlar o fornecimento de energia ao aparelho conectado ao *smart plug*, ela apresenta um *display* com a indicação de carga ligada ou desligada, apresentando a opção de acionar o *timer* (temporizador) e programar um horário de desligamento ou desligar através de um botão simples, de maneira instantânea. Também é possível ver o custo e o consumo acumulado. Um *display* gráfico mostra o histórico de acionamento e desligamento da carga, com opções em tempo real (live), dia (d), semana (w) ou mês (M). A Figura 38 mostra a aba.

**Figura 38** – Aba “TEMPORIZADOR” do aplicativo.



**Fonte:** Autoria Própria.



### 3.1.5 Montagem do *Smart Plug* e Circuito de Medição

O circuito do *smart plug* pode ser observado na Figura 39, a placa contendo o circuito foi armazenada em uma caixa plástica que possuía em suas faces a entrada e a saída dos pinos de acordo com o padrão brasileiro de tomadas, o *smart plug* pode ser visto na Figura 40, em vista frontal (a) e vista lateral (b).

**Figura 39** – Circuito de medição do *smart plug*.



Fonte: Autoria Própria.

**Figura 40** – *Smart plug* desenvolvido.



(a)

(b)

Fonte: Autoria Própria.



Conforme a Figura 40, o plugue atua como um meio de ligação entre o aparelho conectado e a tomada, o case do *smart plug* foi limitado pelo tamanho da placa de circuito, de modo que este foi o menor tamanho possível diante das possibilidades.

### 3.2 PARÂMETROS DE TESTE

Para estabelecer uma metodologia de testes padrão, foram utilizadas bancadas didáticas para testes de carga, contendo componentes resistivos, indutivos e capacitivos, onde foi permitido verificar a variação das medições de acordo com diferentes tipos de potência aplicada. Para o teste de comunicação, analisou-se o alcance máximo permitido pelo ESP8266-01 e o acionamento de cargas remotamente através do aplicativo. No teste de comunicação também foi avaliada a exibição dos dados medidos pelo *smart plug* em tempo real através da interface gráfica.

#### 3.2.1 Teste de Precisão do Circuito

As bancadas didáticas possuem alimentação em 220/380V e três canais de conexão, variando a potência de acordo com o tipo de carga, conforme a Tabela 3, desta forma foi possível verificar a precisão das medições de tensão, corrente, potência ativa, reativa e fator de potência do circuito do *smart plug*. A Figura 41 mostra o modelo das bancadas de testes para as 3 cargas.

**Tabela 3** – Cargas utilizadas para os testes de leitura de potência.

<b>Tipo de Carga</b>	<b>Potência Disponível por Conexão</b>
Resistiva	350 W
Capacitiva	840 VAr
Indutiva	200 VAr

Fonte: Autoria Própria.

**Figura 41** – Bancada de cargas resistivas, indutivas e capacitivas para o teste de precisão.



Fonte: Didatech (2018).

Os testes foram realizados em 3 grupos de carga, inicialmente resistivas, variando a potência em 350W. A tensão de fornecimento foi fixada em 220V, as informações pertinentes as cargas usadas nos testes resistivos podem ser observadas na Tabela 4.

**Tabela 4** – Parâmetros de teste para cargas resistivas.

<b>Potência (W)</b>	<b>Tensão (V)</b>	<b>Corrente (A)</b>	<b>Fator de Potência</b>
350	220	1,5909	1
700	220	3,1818	1
1050	220	4,7727	1

Fonte: Autoria Própria.

Os testes com cargas capacitivas foram realizados com duas faixas de potências predominante reativas (circuito RLC de 840 e 1680 VAr), variando o fator de potência e consequentemente as correntes para cada caso. A tensão foi mantida em 220V, destaca-se que o limite do sensor de corrente (DL-CT1005A), conforme apresentado na seção 3.1.1, é de 10A, portanto a máxima carga utilizada no teste foi de 1680 Var com fator de potência 0,65. Os dados usados nos testes estão listados na Tabela 5.

**Tabela 5** – Parâmetros de testes para cargas capacitivas.

<b>Potência (W)</b>	<b>Tensão (V)</b>	<b>Corrente (A)</b>	<b>Fator de potência</b>
1355,40	220	7,2481	0,85
942,50	220	5,7725	0,75
718,50	220	5,0243	0,65
553,20	220	4,5718	0,55
423,28	220	4,2755	0,45
1346,97	220	10,0490	0,65
1106,50	220	9,1435	0,55
846,56	220	8,5511	0,45

Fonte: Autoria Própria.

Para as cargas indutivas utilizou-se três valores de carga predominantemente indutivas (circuito RLC), com diferentes fatores de potência, os dados utilizados podem ser vistos na Tabela 6.

**Tabela 6** – Parâmetros de testes para cargas indutivas.

<b>Potência (W)</b>	<b>Tensão (V)</b>	<b>Corrente (A)</b>	<b>Fator de potência</b>
322,71	220	1,7257	0,85
171,07	220	1,1963	0,65
100,78	220	1,0180	0,45
645,73	220	3,4515	0,85
342,13	220	2,3925	0,65

201,56	220	2,0260	0,45
968,14	220	5,1772	0,85
513,20	220	3,5889	0,65
302,34	220	3,0539	0,45

Fonte: Autoria Própria.

Nos testes foram medidas as tensões, correntes, potências ativa, reativa e o fator de potência. Para testar a precisão na medição do consumo, utilizou-se como parâmetros as cargas apresentadas nas Tabelas 4, 5 e 6, considerando-se um tempo de uso para cada uma delas, conforme a Tabela 7.

**Tabela 7** – Potências utilizadas no teste de medição de consumo.

Potência da carga (W)	Tempo de utilização (h)	Consumo estimado (kWh)
100,78	12	1,2094
201,56	12	2,4187
350,00	12	4,2000
423,28	12	5,0794
513,20	12	6,1584
645,73	24	15,4975
718,50	24	17,2440
968,14	24	23,2354
1106,50	24	26,5560
1355,40	24	32,5296

Fonte: Autoria Própria.

Para verificar a precisão do *smart plug* foi utilizado um analisador de energia comercial. O modelo utilizado foi o AE-100, fabricado pela Instrutherm, com capacidade de medição de tensão, corrente, fator de potência, potências ativa, reativa e aparente e energia (consumo). O analisador possui uma capacidade 20.000 leituras, armazenando os dados em arquivo de texto para análise posterior. As precisões do aparelho para as grandezas medidas podem ser vistas na Tabela 8. O AE-100 pode ser visto na Figura 42.

**Tabela 8** – Faixas de precisão do analisador de energia AE-100.

Grandeza	Precisão (%)
Tensão	$\pm 0,3$
Corrente	$\pm 0,5$
Potência ativa	$\pm 1,0$
Potência reativa	$\pm 1,0$
Potência aparente	$\pm 1,0$
Fator de potência	$\pm 1,0$

Fonte: Autoria Própria.

**Figura 42** – Analisador de energia AE-100 utilizado nos testes de precisão.



Fonte: Autoria Própria.

### 3.2.2 Teste de Comunicação e Acionamento Remoto

Por se tratar de um equipamento que se comunica utilizando a rede Wi-Fi, dois testes foram realizados, sendo o primeiro baseado nos níveis de sinal recebidos pelo módulo ESP8266-01 com o aumento da distância, e um teste prático de recepção dos dados pelo aplicativo.

No primeiro teste, utilizou-se o comando de busca de redes Wi-Fi do ESP8266-01 e verificou-se o nível de sinal recebido de um roteador, o RSSI (*received signal strength indication*), à medida que se aumentava a distância entre ele e o módulo. Através do teste obteve-se a máxima faixa de operação do ESP8266-01.

No segundo teste, foi verificado o status de exibição de informações, utilizando a aba “TEMPORIZADOR” do aplicativo desenvolvido. Com a comunicação em perfeito estado o aplicativo exibirá se a carga conectada ao *smart plug* está ligada ou desligada, do mesmo modo que a aba “DADOS ELÉTRICOS” exibirá os dados que estão sendo obtidos pelo ADE7753.

### 3.3 CUSTO DE DESENVOLVIMENTO DO *SMART PLUG*

Os custos envolvidos no desenvolvimento do *smart plug* foram divididos de acordo com os módulos apresentados na seção 3.1. Na Tabela 9, são apresentadas as informações de custo financeiro para a construção do módulo de aquisição de dados. Na Tabela 10, são disponibilizados os custos para o desenvolvimento do módulo de processamento de dados. Na Tabela 11, são disponibilizados os custos para o desenvolvimento do módulo de comunicação.

**Tabela 9** – Custos dos componentes do módulo de aquisição de dados.

Item	Descrição	Quantidade	Preço Unitário (R\$)	Total (R\$)
1	Resistor de precisão 100 $\Omega$	1	0,22	0,22
2	Resistor de precisão 1 k $\Omega$	4	0,22	0,88
3	Resistor de precisão 2,2 k $\Omega$	1	0,22	0,22
4	Capacitor cerâmico 33 nF	4	0,08	0,32
5	Transformador DL-CT1005A	1	16,00	16,00
6	Transformador T-211259-R	1	18,00	18,00
<b>TOTAL</b>				35,64

Fonte: Autoria Própria.

**Tabela 10** – Custos dos componentes do módulo de processamento de dados.

Item	Descrição	Quantidade	Preço Unitário (R\$)	Total (R\$)
1	Resistor de precisão 10 k $\Omega$	1	0,22	0,22
2	Capacitor cerâmico 100 nF	4	0,08	0,32
3	Capacitor cerâmico 22 pF	2	0,08	0,16
4	Capacitor eletrolítico 10 uF	3	0,10	0,30
5	Cristal 3,579545 MHz	1	0,72	0,72
6	Módulo relé	1	9,90	9,90
7	RTC DS1307	1	2,61	2,61
8	ADE7753	1	20,00	20,00
9	Arduino Nano	1	34,80	34,80
<b>TOTAL</b>				69,03

Fonte: Autoria Própria.

**Tabela 11** – Custos dos componentes do módulo de comunicação.

Item	Descrição	Quantidade	Preço Unitário (R\$)	Total (R\$)
1	Resistor 1 k $\Omega$	1	0,08	0,08
2	Resistor 2 k $\Omega$	1	0,08	0,08
3	Resistor 10 k $\Omega$	1	0,08	0,08
4	Módulo Wi-Fi ESP8266-01	1	26,00	26,00
<b>TOTAL</b>				26,24

Fonte: Autoria Própria.

Na Tabela 12, são apresentados os custos totais para o desenvolvimento do protótipo, incluindo a fonte de alimentação e os demais componentes.

**Tabela 12** – Custos total do *smart plug*.

Item	Descrição	Total (R\$)
1	Módulo de aquisição de dados	35,64
2	Módulo de processamento de dados	69,03
3	Módulo de comunicação	26,00
4	Fonte de alimentação encapsulada	15,70
5	Case para montagem	19,90
6	Placa de circuito impresso ilhada	14,80
<b>TOTAL</b>		181,07

Fonte: Autoria Própria.

## 4. RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados obtidos após o teste operacional do *smart plug* apresentado no capítulo 3. Inicialmente serão expostos os valores obtidos nos testes de precisão, sendo estes comparados com um analisador de energia comercial. Por fim, serão analisados os dados e realizadas as explicações finais no que diz respeito a precisão e funcionamento do medidor desenvolvido.

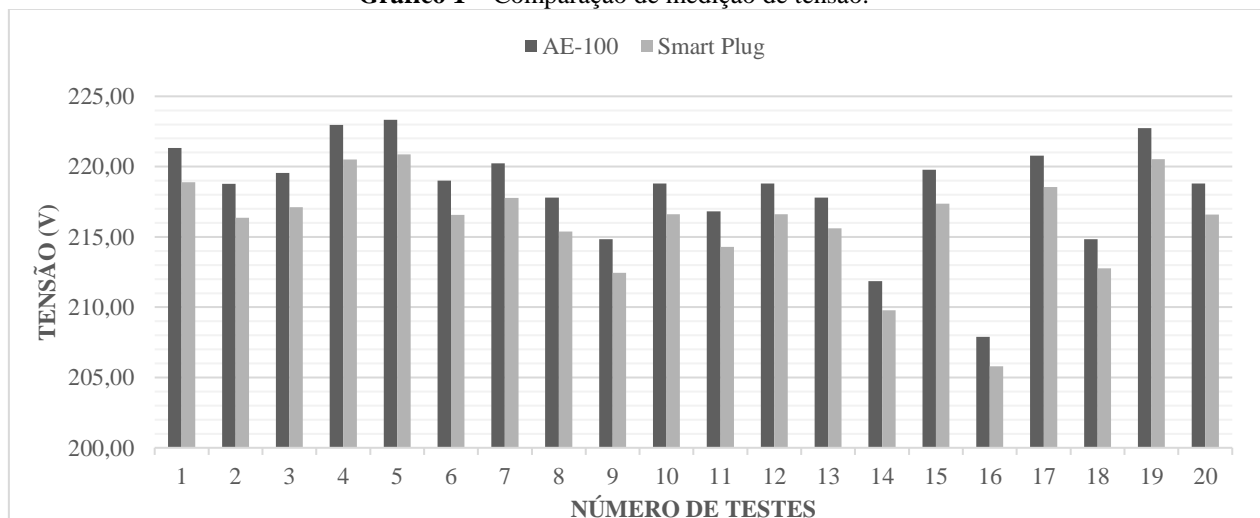
### 4.1 RESULTADO DO TESTE DE PRECISÃO

Nos resultados dos testes de precisão serão apresentadas as leituras de tensão, corrente, potência ativa e consumo. Apesar do *smart plug* fornecer a leitura das demais informações, para o usuário final as mais importantes constituem-se em consumo (kWh) e potência ativa (W). Como todas as demais grandezas (potência aparente, reativa e fator de potência) podem ser obtidas através da tensão e da corrente, optou-se por apresentar apenas os resultados do teste de precisão destas duas grandezas básicas. Conforme apresentado no Capítulo 3, foram feitas 20 análises, divididas em 3 cargas resistivas, 8 indutivas e 9 capacitivas.

#### 4.1.1 Medição de Tensão

A comparação entre os dados medidos com o AE-100 e o *smart plug* para tensão elétrica podem ser observados no Gráfico 1. Os erros obtidos nas leituras podem ser observados no Gráfico 2.

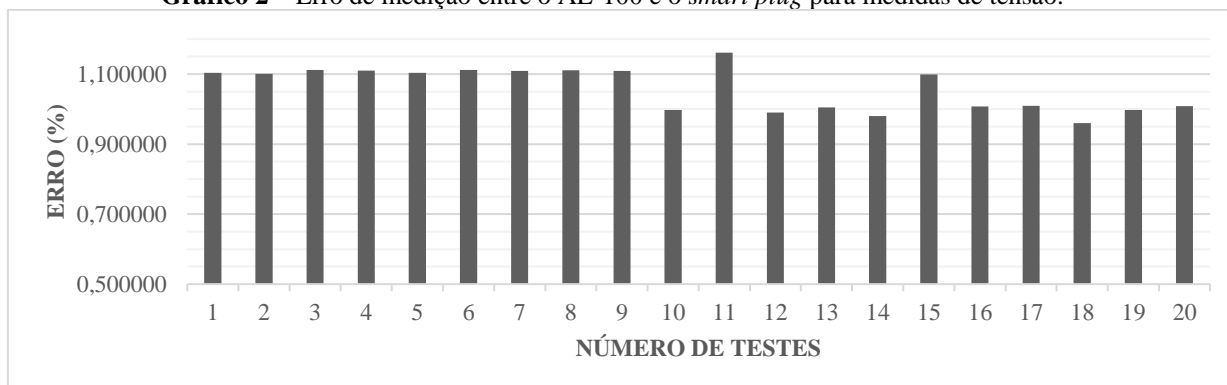
**Gráfico 1** – Comparação de medição de tensão.



Fonte: Autoria Própria.

No Gráfico 1 o eixo vertical representa a faixa de tensão obtida nas medições. Convém destacar que durante os testes a tensão de fornecimento foi variada, com máximo de 224V e mínimo de 208V. O eixo horizontal representa a quantidade de testes realizados. As 20 medições foram realizadas separadamente das leituras de potência e corrente, uma vez que os valores tensão não foram fixos.

**Gráfico 2** – Erro de medição entre o AE-100 e o *smart plug* para medidas de tensão.



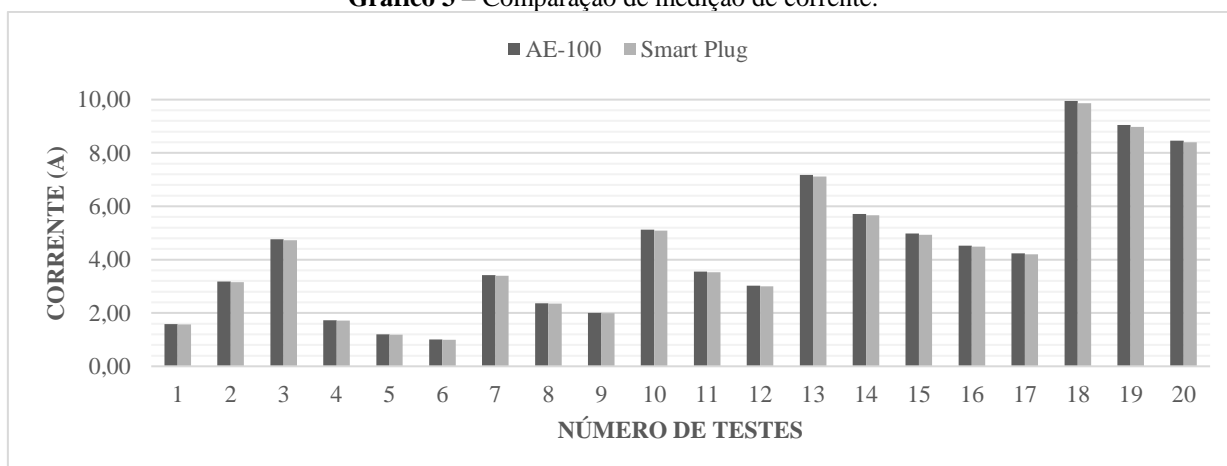
Fonte: Autoria Própria.

De acordo com o Gráfico 2, as oito primeiras medições apresentaram uma taxa de erro aproximada em 1,1%, nos demais testes verificou-se que o maior erro se estabeleceu na décima primeira medição, com 1,16%, o menor erro foi observado na décima oitava medição, com valor de 0,96%. O erro médio total foi de 1,01% para os testes de tensão.

#### 4.1.2 Medição de Corrente

A comparação entre as medidas de corrente pode ser vista no Gráfico 3 e o erro está ilustrado no Gráfico 4.

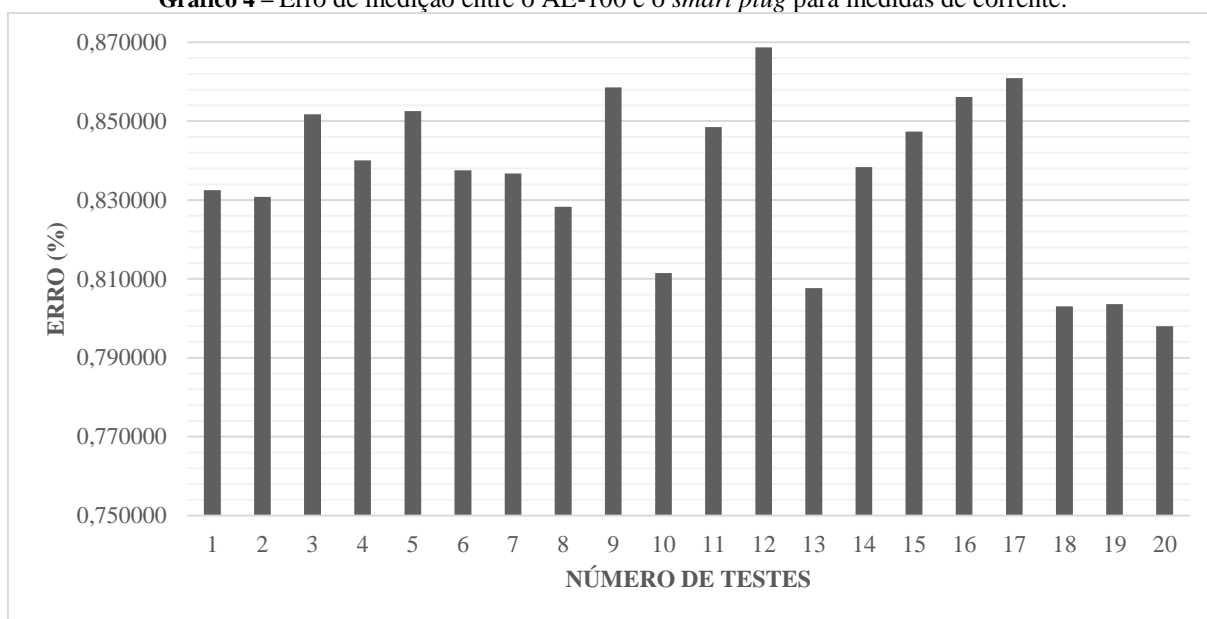
**Gráfico 3** – Comparação de medição de corrente.



Fonte: Autoria Própria.

No Gráfico 3 o eixo vertical representa a faixa de corrente obtida nas medições. Os testes de corrente foram realizados conforme apresentado no Capítulo 3, com três tipos de cargas distintas. O maior valor lido foi de 9,95A, na décima oitava medição, o menor valor foi registrado na décima sexta medição, com 1,01A. Destaca-se que o máximo suportado pelo sensor é de 10A, de modo que neste teste utilizou-se um valor de corrente próximo do limite. O eixo horizontal representa a quantidade de testes realizados.

**Gráfico 4** – Erro de medição entre o AE-100 e o *smart plug* para medidas de corrente.



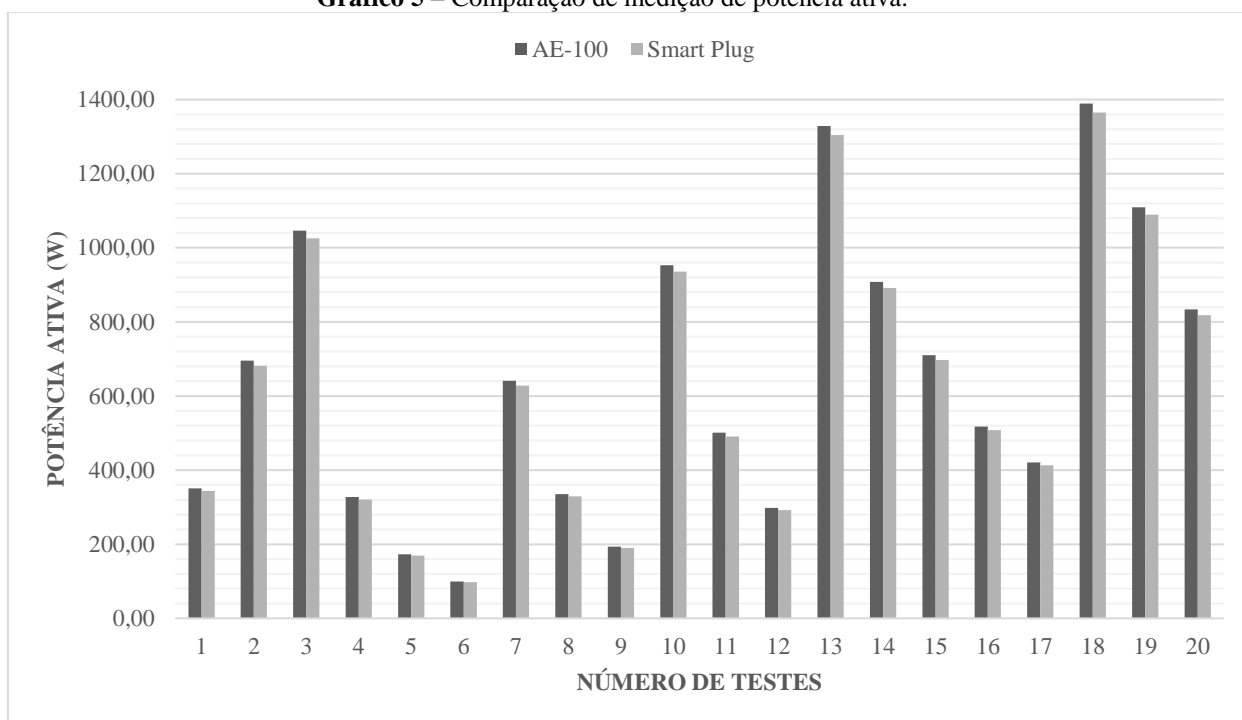
Fonte: Autoria Própria.

No Gráfico 4 observou-se que a taxa de erro sofreu variações de acordo com o valor da corrente, os menores erros foram observados nas três últimas medições, momento em que a intensidade da corrente foi maior e mais próxima do limite do sensor. A maior taxa de erro foi observada na décima segunda medição, com um valor de 0,86% e a menor taxa na vigésima amostra, registrando um valor de 0,80%. O erro médio observado foi de 0,83%.

### 4.1.3 Medição Potência Ativa

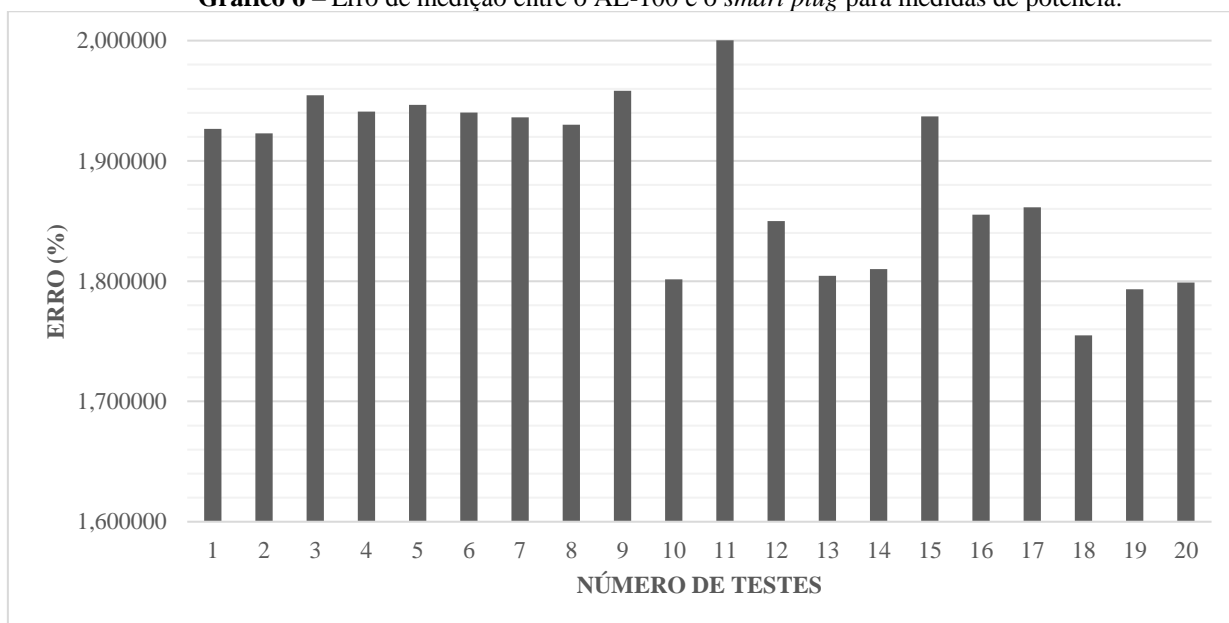
A comparação entre os dados medidos com o AE-100 e o *smart plug* para potência ativa podem ser observados no Gráfico 5. Os erros obtidos nas leituras podem ser observados no Gráfico 6.



**Gráfico 5** – Comparação de medição de potência ativa.

Fonte: Autoria Própria.

No Gráfico 5 o eixo vertical representa a faixa de potência medida. Os testes foram realizados conforme apresentado no Capítulo 3. A tensão foi fixada em 220V. O maior valor lido foi de 1389,20W, na décima oitava medição, o menor valor foi registrado na décima sexta medição, com 99,43W. O eixo horizontal representa a quantidade de testes realizados.

**Gráfico 6** – Erro de medição entre o AE-100 e o *smart plug* para medidas de potência.

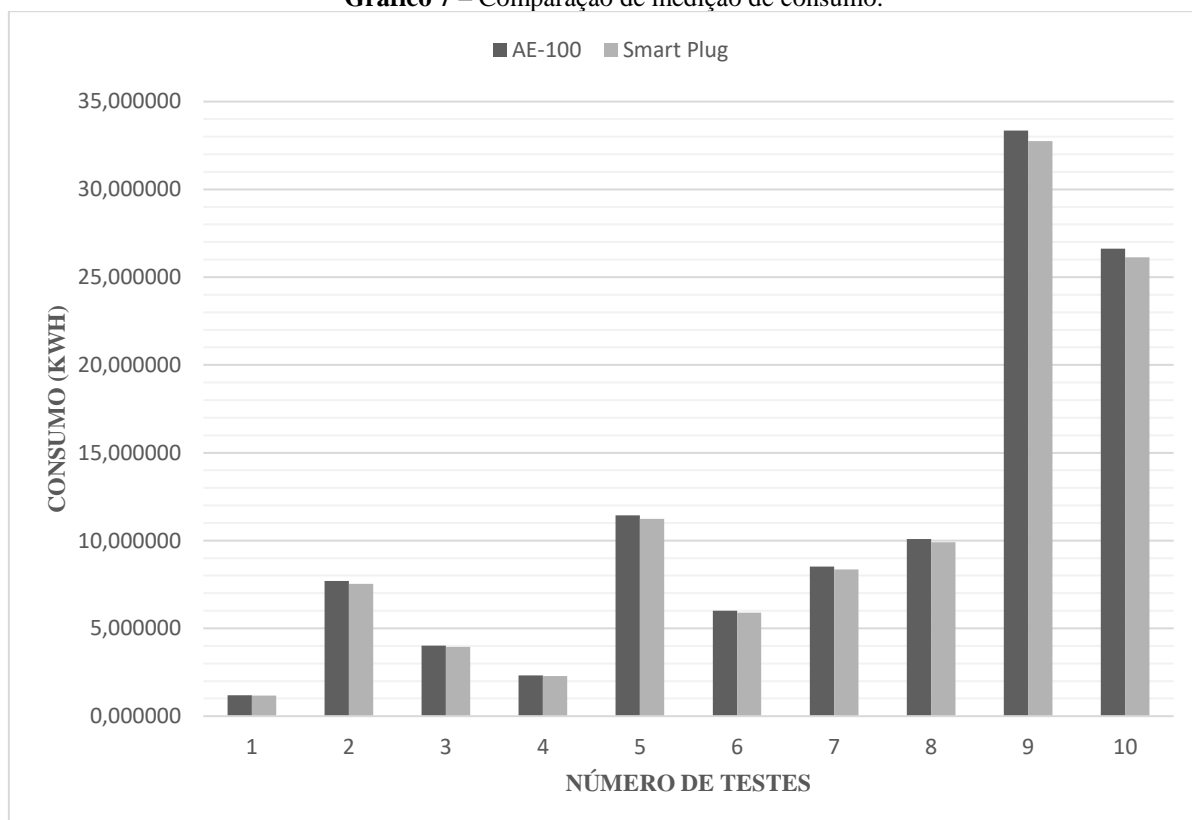
Fonte: Autoria Própria.

No Gráfico 6 observou-se que a taxa de erro sofreu variações de acordo com o valor da carga, os menores erros foram observados nas três últimas medições, semelhante ao ocorrido na medição de corrente. A maior taxa de erro foi observada na décima primeira medição, com um valor de 2% e a menor taxa na décima oitava amostra, registrando um valor de 1,75%. O erro médio observado foi de 1,88%.

#### 4.1.4 Medição de Consumo

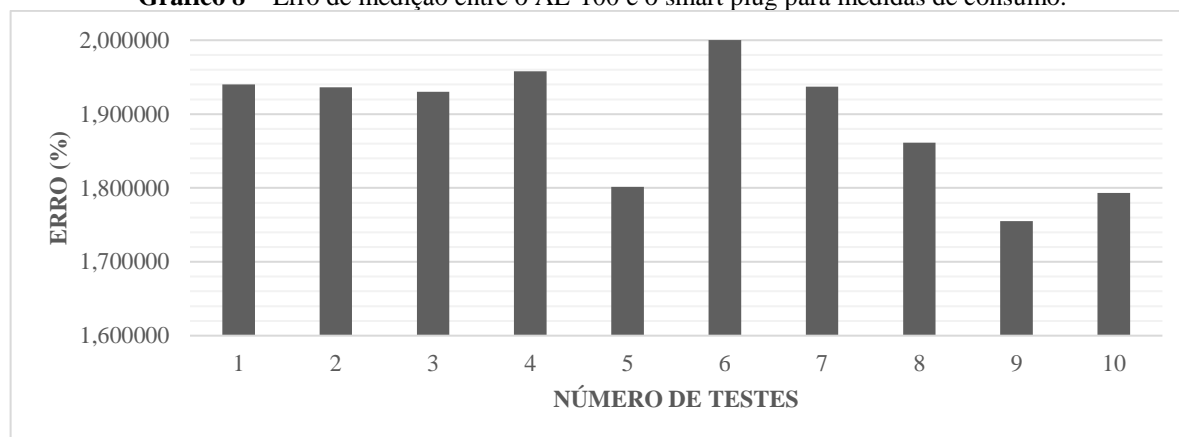
A medição do consumo foi realizada conforme os dados apresentados na Tabela 7, a comparação entre as medições do AE-100 e o *smart plug* podem ser observadas no Gráfico 7, o erro entre as medidas pode ser observado no Gráfico 8.

**Gráfico 7** – Comparação de medição de consumo.



Fonte: Autoria Própria.

No Gráfico 7 o eixo vertical representa o consumo registrado pelas cargas, por se tratarem de cargas distintas e terem horas de funcionamento diferenciadas, a faixa de medição variou de 1,21 kWh até 33,34 kWh, as medidas foram feitas em momentos diferentes, de acordo com o tipo de carga conectada. O eixo horizontal mostra a quantidade de medições que para este teste, foi reduzida para 10 amostras.

**Gráfico 8** – Erro de medição entre o AE-100 e o smart plug para medidas de consumo.

Fonte: Autoria Própria.

No Gráfico 6 observou-se que a taxa de erro sofreu variações de acordo com o valor da carga. A maior taxa de erro foi observada na sexta medição, com um valor de 1,93% e a menor taxa na nona amostra, registrando um valor de 1,79%. O erro médio observado foi de 1,88%.

## 4.2 RESULTADO DO TESTE DE COMUNICAÇÃO

Para o teste de alcance máximo do módulo ESP8266-01, os resultados podem ser vistos na Tabela 13.

**Tabela 13** – Resultado do teste de alcance do ESP8266-01.

Distância (m)	1	2	3	5	10	15	20
RSSI (dBm)	-43	-46	-48	-54	-76	-83	-

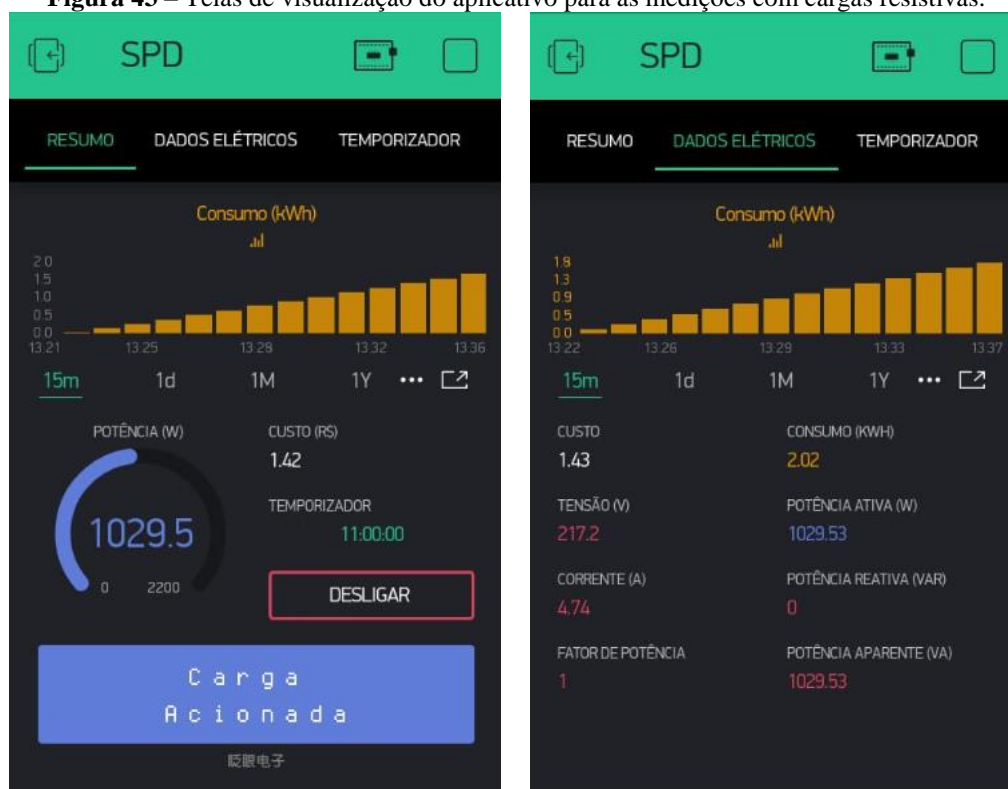
Fonte: Autoria Própria.

Os valores obtidos na Tabela 9 testaram o alcance de mais de um módulo, com a finalidade de verificar o valor padrão dos dispositivos, evitando conclusões equivocadas baseadas em um único ESP8266-01.

Conforme explicado no Capítulo 3, o teste foi realizado conectando o módulo Wi-Fi a um roteador e a medida que era feito um afastamento, fazia-se a leitura do *Received signal strength indication* (RSSI), o indicador de intensidade de sinal recebido. Os testes foram realizados até que o módulo perdesse a comunicação com o roteador, determinando os alcances vistos na Tabela.

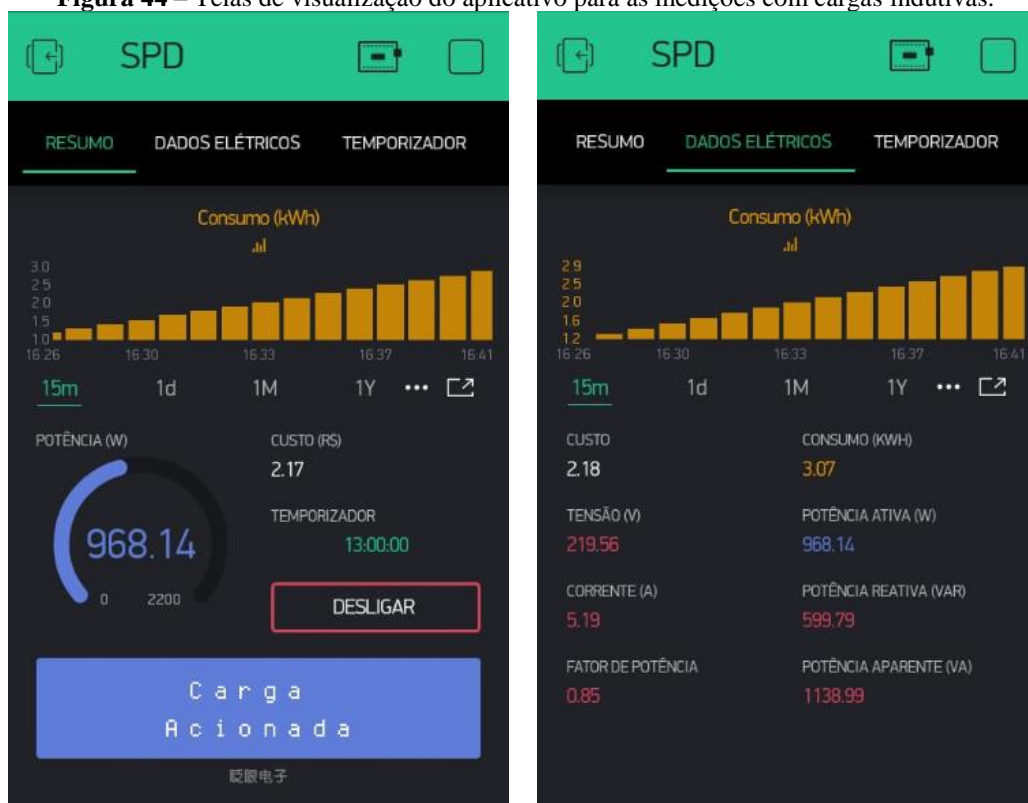
Os resultados do teste de comunicação através do aplicativo podem ser vistos nas Figura 42 (para cargas puramente resistivas), Figura 44 (cargas predominantemente indutivas) e Figura 45 (cargas predominantemente capacitivas).

**Figura 43** – Telas de visualização do aplicativo para as medições com cargas resistivas.



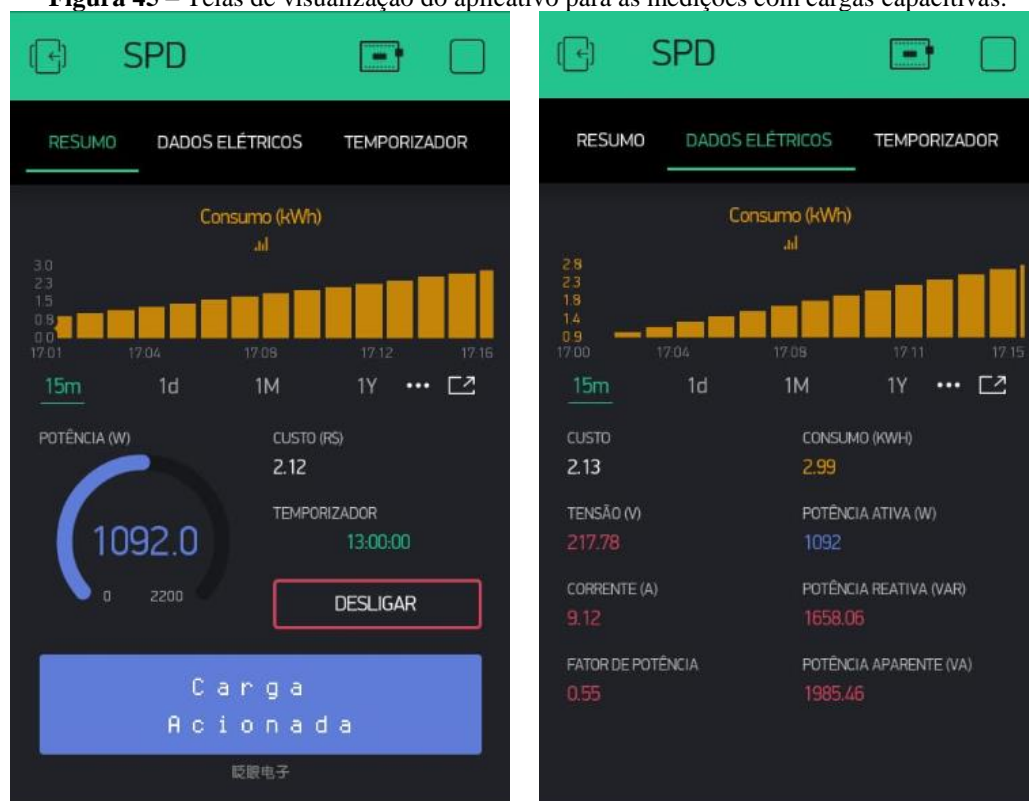
Fonte: Autoria Própria.

**Figura 44** – Telas de visualização do aplicativo para as medições com cargas indutivas.



Fonte: Autoria Própria.

**Figura 45** – Telas de visualização do aplicativo para as medições com cargas capacitivas.



Fonte: Autoria Própria.

Na aba “RESUMO” observa-se o indicador do consumo acumulado de forma crescente na forma de gráfico. O eixo vertical representa o consumo, variando instantaneamente e o eixo horizontal representa o tempo, que no caso das figuras, foi representado em intervalos de 15 minutos. Durante este tempo os valores lidos de consumo eram acumulados e ao fim da faixa de tempo o gráfico reiniciava sua forma, permitindo que o usuário possa voltar a qualquer momento e veja o resultado dos 15 minutos anteriores. Mudando a escala de tempo é possível ver a mesma forma gráfica (barras em ordem crescente) para dias, meses e anos.

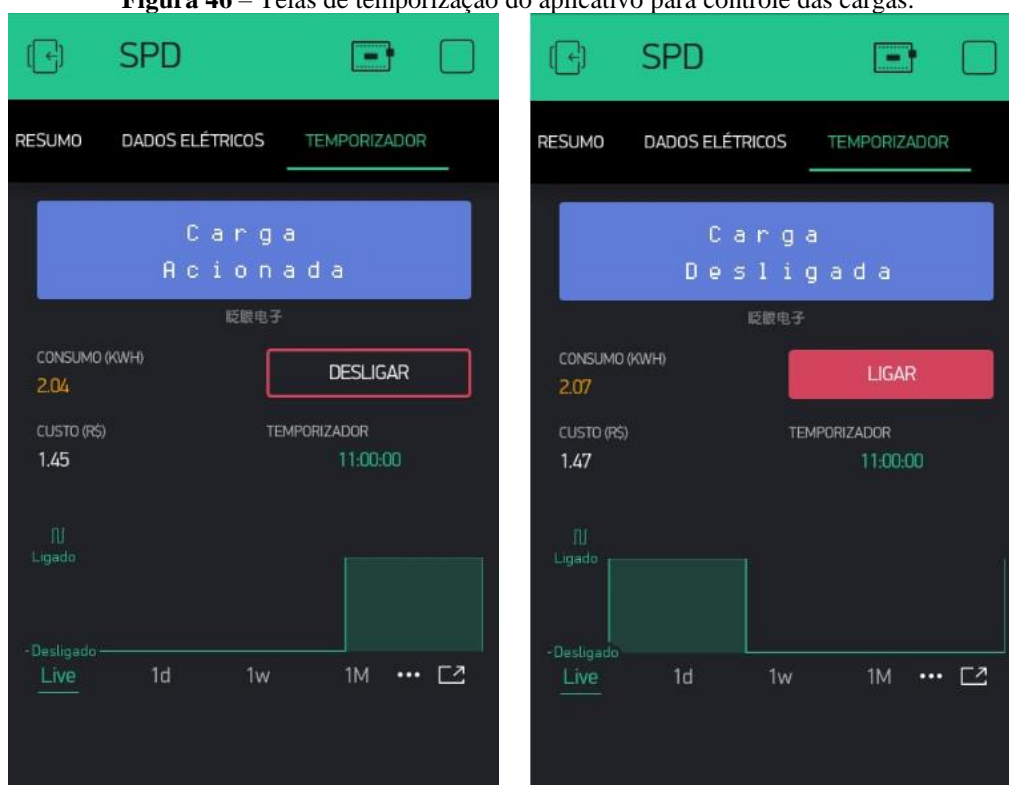
Ainda nesta aba observaram-se informações de custo financeiro, gerado a partir do consumo, e a potência ativa da carga conectada no *smart plug*. Por se tratar de uma aba de acesso rápido, as demais informações de grandezas elétricas não foram listadas, porém foi possível controlar o funcionamento da carga conectada através do botão “DESLIGAR”, que ao ser pressionado ativou o relé e isolou a carga da rede elétrica. O mesmo raciocínio pode ser aplicado ao botão “TEMPORIZADOR”, onde foi programado o horário de desligamento. O *display* na parte inferior da aba mostra em tempo real o estado da carga, que por padrão está sempre ligada, até que haja algum acionamento do relé.

Na aba “DADOS ELÉTRICOS” foi possível visualizar todas as informações para as grandezas elétricas, além de apresentar o gráfico de consumo. Ao conectar a carga ao *smart*

*plug* automaticamente as informações foram preenchidas, enviadas através do protocolo MQTT em conjunto com o ESP8266. Por se tratar de uma aba de visualização, não há componentes manipuláveis. Para os três tipos de carga utilizados no teste o funcionamento foi conforme o esperado.

A aba “TEMPORIZADOR” foi a responsável pelo teste de controle de carga, sendo utilizada para acionar ou desligar o relé e consequentemente promover o isolamento da rede elétrica, o resultado do teste pode ser visto na Figura 46.

**Figura 46** – Telas de temporização do aplicativo para controle das cargas.



**Fonte:** Autoria Própria.

O último teste, de controle de cargas, apresentado na Figuras 46, evidenciou que através do aplicativo foi possível controlar remotamente o fornecimento de energia para o aparelho elétrico conectado ao plugue. No acionamento manual, ao clicar no botão “DESLIGAR” a carga perdeu sua conexão com a rede elétrica, permanecendo desta forma até que o botão “LIGAR” fosse pressionado. No instante em que ocorreu o desligamento a *display* superior informou o estado atual da carga e o gráfico na região inferior do aplicativo informou o momento exato. Este gráfico funciona de forma semelhante ao do consumo, podendo exibir o acionamento em tempo real (Live) ou o histórico dos acionamentos por dia (d), semana (w) ou mês (M).

No outro modo de controle (TEMPORIZADOR) observou-se o mesmo funcionamento do acionamento manual, com a carga desligando após o ajuste de tempo estabelecido. Ao clicar na opção foi aberto um campo para a escolha do horário de desligamento o horário de acionamento, deixando a carga desconectada neste intervalo de tempo. No caso da Figura 46 o *timer* foi determinado para fazer o desligamento às 11:00 e restabelecer o fornecimento as 12:00, permanecendo desligado por 1h.

O *display* superior informou corretamente o estado da carga e o gráfico na extremidade inferior da tela mostrou o exato momento do acionamento e o histórico de uso do *timer*. Vale ressaltar que qualquer acionamento feito na aba “RESUMO”, automaticamente era registrado na aba “TEMPORIZADOR”, com a mesma lógica valendo na ordem inversa. O acionamento por meio manual automaticamente se sobrepunha ao *timer*, logo, caso o temporizador estivesse programado para desligar e fosse acionado o botão “DESLIGAR”, automaticamente o temporizador seria cancelado.

### 4.3 ANÁLISE DOS DADOS COLETADOS

No teste de precisão evidenciou-se que as medidas atingiram uma taxa de erro abaixo dos 2% (considerando o maior erro encontrado), o erro médio para cada uma das grandezas pode ser visto na Tabela 14.

**Tabela 14** – Erro médio percentual para as grandezas medidas.

<b>Grandeza</b>	<b>Tensão</b>	<b>Corrente</b>	<b>Potência ativa</b>	<b>Consumo</b>
<b>Erro médio (%)</b>	1,0594	0,8356	1,8861	1,8913

Fonte: Autoria Própria.

Considerando os valores dos erros e comparando-os com os valores apresentados na Tabela 8, observa-se que o *smart plug* possui um erro próximo de 1% em suas medições quando comparado ao AE-100. Dadas as limitações do circuito e dos componentes, destaca-se ainda que o erro, se manteve dentro de uma faixa estável, não havendo grandes discrepâncias ao utilizar cargas de potências muito distintas, algo atestado pela boa linearidade do sensor de corrente.

De acordo com a ANEEL (2018), o módulo 5 do PRODIST (Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional) destinado a estabelecer os requisitos mínimos para medição de grandezas elétricas, destaca na sua seção 5.2 o cálculo da incerteza através da Equação (1):

$$e(\%) = 0,05 \pm \sqrt{M^2 + Tc^2 + Tp^2} \quad (1)$$

Onde:

$e(\%)$  é o erro total do medidor;

$M$  é o erro do CI medidor;

$Tc$  é o erro do Transformador de corrente;

$Tp$  é o erro do Transformador de potencial;

A precisão teórica, de acordo com os *datasheets*, para o transformador de corrente e de potencial foi considerada em 0,3 e 0,8 respectivamente. A precisão do CI de medição considerada foi de 0,1. Aplicando estes valores na Equação (1) constata-se a precisão teórica do *smart plug*. A Tabela 15 mostra o comparativo entre o erro teórico e o experimental:

**Tabela 15** – Erro teórico e experimental para o *smart plug*.

<b>Erro teórico (%)</b>	<b>Erro experimental (%)</b>
0,9102	1,4030

Fonte: Autoria Própria.

A comparação apresentada demonstra uma diferença entre os valores dos erros, na comparação com o equipamento comercial (AE-100) a precisão se mostrou satisfatória, no entanto ao analisar o *smart plug* e sua precisão prevista, o mesmo se mostrou abaixo do esperado. Parte da perda de exatidão se deve a baixa precisão efetiva do transformador de potencial (T-211259-R), que no teste prático, sem estar diretamente conectado ao restante do circuito de medição, apresentou uma precisão de 1,1%, acima dos 0,8% apresentado no *datasheet*, destaca-se que foram testados 10 transformadores, que apresentaram taxas semelhantes, sempre acima do fornecido pelo fabricante. O transformador de corrente, também testado fora do circuito de medição, apresentou um erro de medição de 0,31%, dentro do esperado.

Outro ponto que afetou a precisão foi a calibração do CI de medição ADE7753. A comunicação entre o Arduino e o mesmo foi realizada, a nível de *software*, através da biblioteca ADE7753.h, de acesso público e desenvolvida para um equipamento de medição que também utiliza o mesmo circuito integrado. O processo de calibração foi realizado com um teste de carga semelhante ao teste de precisão, no entanto a quantidade de amostras foi menor, tomando-



se apenas três cargas (1 resistiva, 1 indutiva e 1 capacitiva) o que eventualmente pode ter gerado valores de constantes de calibração inadequadas, comprometendo a precisão final do circuito.

Os resultados do teste de comunicação, apresentados nas Figuras 43, 44 e 45, mostraram que o *smart plug* conseguiu estabelecer a conexão com o aplicativo criado, exibindo os dados elétricos em tempo real e acumulados. Todos os dados medidos se mostraram disponíveis através do próprio aplicativo e através de um arquivo .csv com as leituras realizadas, sendo possível utilizar as informações para análise posterior em um *software* de planilha eletrônica.

O alcance do ESP8266, conforme a Tabela 9, foi de 15 metros de distância entre o módulo e o ponto de acesso da rede, acima desta distância o módulo não conseguiu estabelecer a conexão e enviar os dados, da mesma forma que não era possível acessar ou controlar o *smart plug* e a carga a ele conectada.

O último teste, de controle de cargas, apresentado na Figura 46, evidenciou que através do aplicativo foi possível controlar remotamente o fornecimento de energia para o aparelho elétrico conectado ao plugue, no acionamento manual, ao clicar no botão (LIGAR ou DESLIGAR) ou na função de *timer* (TEMPORIZADOR), com a carga desligando após o ajuste de tempo estabelecido. O *display* superior informou corretamente o estado da carga e o gráfico na extremidade inferior da tela mostrou o exato momento do acionamento e o histórico de uso do *timer*.

## 5. CONCLUSÃO

Após as medições realizadas, o teste comparativo com um produto comercial e a verificação da comunicação através do aplicativo, ficou evidenciado que os principais objetivos definidos para este trabalho foram atingidos. Os componentes utilizados se mostraram satisfatórios, muito embora possam ser melhorados trocando por sensores mais precisos. O teste realizado com diferentes cargas mostrou que o *smart plug* possui uma precisão estimada em 1,40%, valor relativamente próximo ao encontrado em equipamentos como o analisador de energia AE-100.

Os erros de precisão acima do esperado podem ser corrigidos aplicando uma metodologia e calibração mais eficaz, bem como, desenvolvendo uma biblioteca própria de comunicação entre o ADE7753 e o microcontrolador. A troca do sensor de tensão também se faz necessária, diante dos problemas com a precisão do mesmo, destacam-se que dois equipamentos externos podem ser usados para a aquisição das constantes de calibração, o *energy shield*, que consiste em um módulo de medição para o Arduino e que utiliza o ADE7753, utilizando o mesmo é possível adquirir grandezas mais precisas, já que a biblioteca base de comunicação foi adaptada deste dispositivo. O outro equipamento é a placa de desenvolvimento fornecida pelo próprio fabricante do CI, com ela é possível calibrar o ADE7753 diretamente, muito embora seu custo seja elevado quando comparado ao *energy shield*.

No que diz respeito a construção física, o *smart plug* foi desenvolvido com a finalidade de ser encaixado nas tomadas monofásicas e conectar-se a qualquer equipamento que possua um cabo elétrico, o que se mostrou possível e dentro do esperado, as dimensões podem ser ainda menores se forem utilizados componentes SMD (*Surface Mounting Device*) e substituindo a fonte de alimentação por um circuito retificador simples. O custo de desenvolvimento também se mostrou relativamente baixo, dadas as funções disponíveis, sendo estimado em R\$ 190,00, ressaltando-se que esses custos podem ser parcialmente reduzidos com o emprego de um microcontrolador mais simples e produção em grandes quantidades.

Na comunicação observou-se que todas as informações foram disponibilizadas no aplicativo, tornando a visualização possível em qualquer lugar, uma vez que estivesse conectado a rede local, da mesma forma para o acionamento das cargas. Os limites de distância máxima do ESP8266-01 podem ser corrigidos trocando-se a placa por outra de maior alcance, já que sua função é apenas enviar as informações. O aplicativo desenvolvido além de se mostrar funcional ainda possui a capacidade de ser usado em tablets, de modo que o usuário pode utilizar uma tela maior para a visualização.

De modo geral o *smart plug* proposto cumpriu com seu objetivo de ser um equipamento portátil, de simples instalação e capaz de enviar dados e comandos através da rede, a precisão é aceitável, já que não se trata de um equipamento para faturamento ou medições críticas e todos os dados podem ser acessados pelo usuário de forma externa ao aplicativo. Ao utilizar outros plugues instalados nos principais pontos de tomada é possível mapear o custo de cada aparelho e analisar o perfil de uso de energia da instalação, com a função de temporização pode-se estabelecer metas imediatas de consumo, ajustando um equipamento elétrico para funcionar apenas em horas específicas, desligando-o em momentos sem uso e contribuindo para o uso consciente de energia.

Como sugestão de trabalhos futuros, recomenda-se:

- Aprimorar e miniaturizar o circuito, reduzindo o tamanho do *smart plug*.
- Projetar um plugue para tomadas trifásicas para cargas de pequenos motores ou outros equipamentos;
- A implementação de uma rede de *smart plugs* para geração de dados e perfis de uso de energia;
- Desenvolvimento de um ambiente de monitoramento dedicado, por computador ou na nuvem, facilitando o livre acesso aos dados;
- Aplicação do *smart plug* para implementação de medidas de eficiência energética em ambientes com alto consumo;
- Aplicação de Redes Neurais e técnicas de *Data Science* nos dados coletados.

## REFERÊNCIAS

AHMED, Maytham S. et al. Smart plug prototype for monitoring electrical appliances in Home Energy Management System. **2015 IEE Student Conference On Research And Development (scored)**, [s.l.], p.32-36, dez. 2015. IEEE. <http://dx.doi.org/10.1109/scored.2015.7449348>.

ANTUNES, Josué Batista. **Uma Plataforma para Gerenciamento e Aplicações em Internet das Coisas**. 2016. 111 f. Dissertação (Mestrado) - Mestrado em Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2016.

ANALOG DEVICES. **ADE7753: Single-Phase Multifunction Metering IC with di/dt Sensor Interface**. Disponível em: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7753.pdf>. Último acesso em 11/08/2018.

ANEEL - Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional – PRODIST, **Módulo 5 - Sistemas de Medição, revisão 5**. 2018. Disponível em [http://www.aneel.gov.br/documents/656827/14866914/Modulo5\\_Revisao5/4d9e298e-cbf6-4b09-a01a-2e55f05dc9c7](http://www.aneel.gov.br/documents/656827/14866914/Modulo5_Revisao5/4d9e298e-cbf6-4b09-a01a-2e55f05dc9c7). Último acesso em 25/10/2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14539: Medidores Eletrônicos de Energia Elétrica** - Especificações. Rio de Janeiro: Abnt, 2011. 18 p.

BALBINOT, Alexandre; BRUSAMARELL, Valner João. **Instrumentação e fundamentos de medidas - Volume 1**. 2. ed. Rio de Janeiro: Ltc, 2010. 398 p.

BARRETO, Guilherme Pimenta. **Sistema de Monitoramento e Controle de Consumo de Energia Elétrica para uma Residência**. 2015. 126 f. TCC (Graduação) - Curso de Engenharia da Computação, Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2015.

BEZERRA, Bemielison Gletson da Silva. **IoT Baseada em PHM Aplicada a Centrais Condicionadoras de Ar**. 2018. 90 f. Dissertação (Mestrado) - Curso de Mestrado em Engenharia Elétrica, Universidade Federal Rural do Semi-Árido, Mossoró, 2018.

BRENNKUS, Juraj et al. Power-efficient smart metering plug for intelligent households. **2015 25th International Conference Radioelektronika (radioelektronika)**, [s.l.], p.110-113, abr. 2015. IEEE. <http://dx.doi.org/10.1109/radioelek.2015.7129031>.

BRITO, João Luis Grizinsky de. **Sistema para monitoramento de consumo de energia elétrica particular, em tempo real e não invasivo utilizando a tecnologia Arduino**. 2016. 106 f. TCC (Graduação) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Estadual de Londrina, Londrina, 2016.

CAIRES, Luis Eduardo. **Aplicação de Redes Inteligentes nas Instalações Elétricas Residenciais**. 2012. 184 f. Dissertação (Mestrado) - Mestrado em Ciências, Programa de Pós-graduação em Energia, Universidade de São Paulo, São Paulo, 2012.

CESÁRIO JÚNIOR, José Maria. **Um Medidor de Energia Elétrica Integrado em Redes de Comunicações**. 2014. 95 f. Dissertação (Mestrado) - Curso de Mestrado em Tecnologia, Faculdade de Tecnologia, Universidade Estadual de Campinas, Limeira, 2014.

COSTA, Gabriel Rachadel. **Medidor Eletrônico de Consumo de Energia Elétrica Baseado em Sistema Embarcado e Android**. 2014. 112 f. TCC (Graduação) - Curso de Engenharia de Computação, Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, São José, 2014.

DIDATECH. **DT-EE001 - Componentes do Sistema de Estudo em Geração, Transmissão e Consumo de Energia**. Disponível em: < <http://www.didatech.com.br/dt-ee001-componentes/>>. Acesso em: 05 out. 2018.

DHOND, Prasad. **Sub-metering made easy using Texas Instruments energy measurement ICs**. 2012. Disponível em: <<http://www.ti.com/lit/wp/slay020/slay020.pdf>>. Acesso em: 10 ago. 2018.

FABRÍCIO, Marcos Aurélio. **Monitoramento de Equipamentos Elétricos Industriais Utilizando IoT**. 2018. 73 f. Dissertação (Mestrado) - Curso de Mestrado em Redes de Telecomunicações, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas, Campinas, 2018.

FERREIRA, Hiro Gabriel Cerqueira. **Arquitetura de Middleware para Internet das Coisas**. 2014. 125 f. Dissertação (Mestrado) - Mestrado em Engenharia Elétrica, Universidade de Brasília, Brasília, 2014.

FERREIRA, Josiane Bezerra. **Análise de Formas de Medição de Consumo de Energia Elétrica no Setor Residencial**. 2012. 74 f. TCC (Graduação) - Curso de Engenharia da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2012.

GREHS, Daniel Henrique. **Sistema de irrigação doméstico baseado em Internet das Coisas**. 2016. 55 f. TCC (Graduação) - Curso de Engenharia de Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

GUEDES, José Carlos de Souza. **Manual de Tarifação da Energia Elétrica**. Rio de Janeiro: Eletrobrás, 2011. 56 p.

LEE, Min Goo et al. **Wireless Electricity Monitoring System for Smart House using Smart Plug**. **Trans Tech Publications, USA**, 2011.

LEMOS, Ivan Pedrotti. **Medidor de Energia para Avaliação da Adesão à Tarifa Branca em Smart Grids**. 2017. 79 f. Dissertação (Mestrado) - Mestrado em Sistemas de Infraestrutura Urbana, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pontifícia Universidade Católica de Campinas, Campinas, 2017.

MAAS, Dyego Aleksander. **Protótipo de Plugue para Tomada Elétrica Controlado Remotamente**. 2017. 71 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2017.

MONZANI, Rafael Cuerda. **Monitoramento Não Invasivo e Identificação do Consumo de Energia Individual de Cargas Residenciais**. 2016. 148 f. Tese (Doutorado) - Doutorado em Engenharia Elétrica, Departamento de Sistemas e Energia, Universidade Estadual de Campinas, Campinas, 2016.

MORO, Jefferson Zortéa. **Medidor de Consumo de Energia Elétrica Descentralizado e com Interface Web**. 2013. 80 f. Dissertação (Mestrado) - Mestrado em Engenharia Elétrica, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2013.

PAULA, Gilberto José de. **Medidor de Demanda de Energia Elétrica Residencial com Acesso Remoto**. 2013. 66 f. TCC (Graduação) - Curso de Engenharia da Computação, Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2013.

PETROVIC, Tin; MORIKAWA, Hiroyuki. Active sensing approach to electrical load classification by smart plug. **2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (isgt)**, [s.l.], p.1-5, abr. 2017. IEEE. <http://dx.doi.org/10.1109/isgt.2017.8086053>.

PITON, Otávio Henrique Gotardo. **Automação Residencial Utilizando a Plataforma em Nuvem IBM Bluemix**. 2017. 65 f. TCC (Graduação) - Curso de Engenharia Elétrica, Escola de Engenharia, Universidade de São Paulo, São Carlos, 2017.

POPSIN, Jackson. **Desenvolvimento de um Analisador e Medidor Monofásico de Energia**. 2010. 85 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Universidade Luterana do Brasil, Canoas, 2010.

POTTER, D.. Smart plug and play sensors. **IEEE Instrumentation & Measurement Magazine**, [s.l.], v. 5, n. 1, p.28-30, mar. 2002. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/5289.988736>.

RIGODANZO, Jonas. **Instalação de Medidores Inteligentes no Brasil: uma Análise Econômica**. 2015. 120 f. Dissertação (Mestrado) - Mestrado em Engenharia Elétrica, Centro de Tecnologia, Universidade Federal de Santa Maria, Santa Maria, 2015.

SANTOS, Bruno P. et al. Internet das Coisas: da Teoria à Prática. **Minicursos / XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. Salvador: Sociedade Brasileira de Computação (SBC), 2016. Cap. 1. p. 1-50.

SÔNEGO, Arildo Antônio. **A Internet das Coisas Aplicada ao Conceito de Eficiência Energética**. 2016. 133 f. Dissertação (Mestrado) - Mestrado em Tecnologias da Informação e Comunicação, Programa de Pós-graduação em Tecnologias da Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2016.

TORRI, Lucas Bortolaso. **Uma Proposta de Arquitetura Extensível para Micro Medição em Smart Appliances**. 2012. 117 f. Dissertação (Mestrado) - Mestrado em Ciência da Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

TOMAS, Gustavo Henrique Rodrigues Pinto. **Uma Arquitetura para Cidades Inteligentes Baseada na Internet das Coisas**. 2014. 109 f. Dissertação (Mestrado) - Mestrado em Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2014.

USINAINFO. **Arduino Nano V3**. Disponível em: <<https://www.usinainfo.com.br/compativeis/arduino-nano-v3-3630.html>>. Acesso em: 12 jun. 2018.

ZHAI, Shaopeng et al. Appliance Flexibility Analysis Considering User Behavior in Home Energy Management System Using Smart Plugs. **IEEE Transactions On Industrial Electronics**, [s.l.], v. 66, n. 2, p.1391-1401, fev. 2019. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tie.2018.2815949>.

ZHANG, Yu Ping et al. Design of remote control plug. **2015 IEE International Conference On Applied Superconductivity And Electromagnetic Devices (asemd)**, [s.l.], p.25-30, nov. 2015. IEEE. <http://dx.doi.org/10.1109/asemd.2015.7453451>.

## ANEXO A – Biblioteca ADE7753.h

```

#ifndef ADE7753_H
#define ADE7753_H

//Register address
#define WAVEFORM 0x01
#define AENERGY 0x02
#define RAENERGY 0x03
#define LAENERGY 0x04
#define VAENERGY 0x05
#define RVAENERGY 0x06
#define LVAENERGY 0x07
#define LVARENERGY 0x08
#define MODE 0x09
#define IRQEN 0x0A // interrupts enable register
#define STATUS 0x0B // interrupts status register
#define RSTSTATUS 0x0C // interrupts status register but read will reset all interrupt flags
#define CH1OS 0x0D
#define CH2OS 0x0E
#define GAIN 0x0F
#define PHCAL 0x10
#define APOS 0x11
#define WGAIN 0x12
#define WDIV 0x13
#define CFNUM 0x14
#define CFDEN 0x15
#define IRMS 0x16
#define VRMS 0x17
#define IRMSOS 0x18
#define VRMSOS 0x19
#define VAGAIN 0x1A
#define VADIV 0x1B
#define LINECYC 0x1C
#define ZXTOUT 0x1D
#define SAGCYC 0x1E
#define SAGLVL 0x1F
#define IPKLVL 0x20
#define VPKLVL 0x21
#define IPEAK 0x22
#define RSTIPEAK 0x23
#define VPEAK 0x24
#define RSTVPEAK 0x25
#define TEMP 0x26
#define PERIOD 0x27
#define TMODE 0x3D
#define CHKSUM 0x3E
#define DIEREV 0x3F

```



//bits

// MODE register - Bit Location

#define DISHPF 0x0001 // bit 0 - HPF (high-pass filter) in Channel 1 is disabled when this bit is set.

#define DISLPF2 0x0002 // bit 1 - LPF (low-pass filter) after the multiplier (LPF2) is disabled when this bit is set.

#define DISCF 0x0004 // bit 2 - Frequency output CF is disabled when this bit is set.

#define DISSAG 0x0008 // bit 3 - Line voltage sag detection is disabled when this bit is set.

#define ASUSPEND 0x0010 // bit 4 - By setting this bit to Logic 1, both ADE7753 A/D converters can be turned off. In normal operation, this bit should be left at Logic 0. All digital functionality can be stopped by suspending the clock signal at CLKIN pin.

#define TEMPSEL 0x0020 // bit 5 - Temperature conversion starts when this bit is set to 1. This bit is automatically reset to 0 when the temperature conversion is finished.

#define SWRST 0x0040 // bit 6 - Software Chip Reset. A data transfer should not take place to the ADE7753 for at least 18 us after a software reset.

#define CYCMODE 0x0080 // bit 7 - Setting this bit to Logic 1 places the chip into line cycle energy accumulation mode.

#define DISCH1 0x0100 // bit 8 - ADC 1 (Channel 1) inputs are internally shorted together.

#define DISCH2 0x0200 // bit 9 - ADC 2 (Channel 2) inputs are internally shorted together.

#define SWAP 0x0400 // bit 10 - By setting this bit to Logic 1 the analog inputs V2P and V2N are connected to ADC 1 and the analog inputs V1P and V1N are connected to ADC 2.

#define DTRT1 0x0800 // bit 11 - These bits are used to select the waveform register update rate.

#define DTRT0 0x1000 // bit 12 - These bits are used to select the waveform register update rate.

#define WAVSEL1 0x2000 // bit 13 - These bits are used to select the source of the sampled data for the waveform register.

#define WAVSEL0 0x4000 // bit 14 - These bits are used to select the source of the sampled data for the waveform register.

#define POAM 0x8000 // bit 15 - Writing Logic 1 to this bit allows only positive active power to be accumulated in the ADE7753.

// bit 12, 11 DTRT1,0 0 0 These bits are used to select the waveform register update rate.

DTRT0	Update Rate	// *	DTRT 1
		// *	0 0
	27.9 kSPS (CLKIN/128)	// *	0 1
	14 kSPS (CLKIN/256)	// *	1 0
	7 kSPS (CLKIN/512)	// *	1 1
	3.5 kSPS (CLKIN/1024)	// *	

// bit 14, 13 WAVSEL1,0 0 0 These bits are used to select the source of the sampled data for the waveform register.

	// *	WAVSEL1, 0
Length Source	// *	0 0 24
bits active power signal (output of LPF2)	// *	0 1
Reserved	// *	1 0 24
bits Channel 1	// *	1 1 24
bits Channel 2		

/\*\* ==== INTERRUPT STATUS REGISTER (0x0B), RESET INTERRUPT STATUS REGISTER (0x0C), INTERRUPT ENABLE REGISTER (0x0A) ====

The status register is used by the MCU to determine the source of an interrupt request (IRQ). When an interrupt event occurs in the ADE7753, the corresponding flag in the interrupt status register is set to logic high. If the enable bit for this flag is Logic 1 in the interrupt enable register, the IRQ logic output goes active low. When the MCU services the interrupt, it must first carry out a read from the interrupt status register to determine the source of the interrupt.

\*\*/

// The next table summarizes the function of each bit for  
// the Interrupt Status Register, the Reset Interrupt Status Register, and the Interrupt Enable Register

```
//      Bit Mask // Bit Location / Description
#define AEHF    0x0001 // bit 0 - Indicates that an interrupt occurred because the active
energy register, AENERGY, is more than half full.
#define SAG     0x0002 // bit 1 - Indicates that an interrupt was caused by a SAG on the line
voltage.
#define CYCEND  0x0004 // bit 2 - Indicates the end of energy accumulation over an integer
number of half line cycles as defined by the content of the LINECYC register see the Line
Cycle Energy Accumulation Mode section.
#define WSMP    0x0008 // bit 3 - Indicates that new data is present in the waveform
register.
#define ZX      0x0010 // bit 4 - This status bit is set to Logic 0 on the rising and falling edge
of the the voltage waveform. See the Zero-Crossing Detection section.
#define TEMPREADY 0x0020 // bit 5 - Indicates that a temperature conversion result is
available in the temperature register.
#define RESET   0x0040 // bit 6 - Indicates the end of a reset (for both software or hardware
reset). The corresponding enable bit has no function in the interrupt enable register, i.e., this
status bit is set at the end of a reset, but it cannot be enabled to cause an interrupt.
#define AEOF    0x0080 // bit 7 - Indicates that the active energy register has overflowed.
#define PKV     0x0100 // bit 8 - Indicates that waveform sample from Channel 2 has
exceeded the VPKLVL value.
#define PKI     0x0200 // bit 9 - Indicates that waveform sample from Channel 1 has
exceeded the IPKLVL value.
#define VAEHF   0x0400 // bit 10 - Indicates that an interrupt occurred because the active
energy register, VAENERGY, is more than half full.
```

```

#define VAEOF    0x0800 // bit 11 - Indicates that the apparent energy register has
overflowed.
#define ZXTO     0x1000 // bit 12 - Indicates that an interrupt was caused by a missing zero
crossing on the line voltage for the specified number of line cycles see the Zero-Crossing
Timeout section.
#define PPOS     0x2000 // bit 13 - Indicates that the power has gone from negative to
positive.
#define PNEG     0x4000 // bit 14 - Indicates that the power has gone from positive to
negative.
#define RESERVED 0x8000 // bit 15 - Reserved.

```

```
//constants
```

```

#define GAIN_1   0x0
#define GAIN_2   0x1
#define GAIN_4   0x2
#define GAIN_8   0x3
#define GAIN_16  0x4
#define INTEGRATOR_ON 1
#define INTEGRATOR_OFF 0
#define FULLSCALESELECT_0_5V  0x00
#define FULLSCALESELECT_0_25V 0x01
#define FULLSCALESELECT_0_125V 0x02

```

```
// Class Attributes
```

```

#define CS 10 // Chip Select ADE7753 - Digital output pin nbr on Olimex Energy
Shield
#define WRITE 0x80 // WRITE bit BT7 to write to registers
#define CLKIN 4000000 // ADE7753 freq, 4.000000MHz
//The cristal associated to the for the ADE7753 the Olimex energy shield is 4.000000 MHz .
Therefore
//the CLKIN Frequency 3.579545 MHz mentionned in the specification is not applicable, and
all the
//calculations linked to the clock need to be reviewed.

```

```

class ADE7753 {
//public methods
public:
    void setSPI(void);
    void closeSPI(void);

    void setMode(int m);
    int  getMode(void);
    void setInterruptsMask(int i);
    int  getEnabledInterrupts(void);
    int  getInterruptStatus(void);
    int  getresetInterruptStatus(void);

    void printGetResetInterruptStatus(void);

```

```

void printGetMode(void);
void printAllRegisters(void);

long getActiveEnergyLineSync(void);
long getApparentEnergyLineSync(void);
long getReactiveEnergyLineSync(void);

long getVRMS(void);
long getIRMS(void);
long vrms();
long irms();
long getIpeak(void);
long getIpeakReset(void);
long getVpeak(void);
long getVpeakReset(void);

long getWaveform(void);
char getTemp(void);
int getPeriod(void);

void setPowerOffset(int d);
void setPhaseCalibration(char d);
void setEnergyGain(char d);
void setActivePowerGain(int d);
void setActiveEnergyDivider(char d);
void setApparentPowerGain(int d);
void setApparentEnergyDivider(char d);
void setZeroCrossingTimeout(int d);
void setSagVoltageLevel(char d);
void setSagCycles(char d);
void setIPeakLevel(char d);
void setVPeakLevel(char d);
void setLineCyc(int d);

void analogSetup(char gain_ch1, char gain_ch2, char os_ch1, char os_ch2, char
scale_ch1, char integrator_ch1);
void frequencySetup(int cfnun, int cfden);
void energySetup(int wgain, char wdiv, int apos, int vagain, char vdiv, char phcal);
void rmsSetup(int vrmsos, int irmsos);
void energyGain(int wgain, int vagain);
void miscSetup(int zxtout, char sagsyc, char saglvl, char ipklvl, char vpklvl, char tmode);
// ZXTOUT 12-bit (U) - Zero-Crossing Timeout
// SAGCYC 8-bit (U) - Sag Line Cycle Register.
// SAGLVL 8-bit (U) - Sag Voltage Level.
// IPKLVL 8-bit (U) - Channel 1 Peak Level Threshold
// VPKLVL 8-bit (U) - Channel 2 Peak Level Threshold
// TMODE 8-bit (U) - Test Mode Register

// //private methods

```

```

// private:
    unsigned char read8(char reg);
    unsigned int read16(char reg);
    unsigned long read24(char reg);
    void write16(char reg, unsigned int data);
    void write8(char reg, unsigned char data);
    void enableChip(void);
    void disableChip(void);
    long waitInterrupt(unsigned int interrupt);
};

#define NanodeReduceCodeSize 1
#ifndef NanodeReduceCodeSize

    long getActiveEnergy(void);
    long getActiveEnergyReset(void);
    long getApparentEnergy(void);
    long getApparentEnergyReset(void);
    char chkSum(void);
    char getCH1Offset(void);
    char getCH2Offset(void);
    char getPhaseCalibration(void);
    char getEnergyGain(void);
    char getActiveEnergyDivider(void);
    char getApparentEnergyDivider(void);
    char getSagCycles(void);
    char getSagVoltageLevel(void);
    char getIPeakLevel(void);
    char getVPeakLevel(void);

    int getWattGain(int load);
    int getVoltageOffset(void);
    int getCurrentOffset(void);
    int getActivePowerOffset(void);
    int getPowerOffset(void);
    int getActivePowerGain(void);
    int getFrequencyDividerNumerator(void);
    int getFrequencyDividerDenominator(void);
    int getZeroCrossingTimeout(void);
    int getApparentPowerGain(void);
    int getWattGain();
    int getLineCyc();

//};
#endif

#endif

```

## ANEXO B – Biblioteca ADE7753.cpp

```

#if ARDUINO >= 100
#include <Arduino.h> // Arduino 1.0
#else
#include <WProgram.h> // Arduino 0022+
#endif
#include <string.h>
#include <avr/pgmspace.h>
#include "SPI.h"
#include "ADE7753.h"
#include <avr/wdt.h> // Watchdog timer

/** ==== setSPI ====
 * Class constructor, sets chip select pin (CS define) and SPI communication with arduino.
 * @param none
 * @return void
 */

void ADE7753::setSPI(void) {
    pinMode(CS,OUTPUT); // Chip select by digital output on pin nbs CS
    digitalWrite(CS, HIGH);//is disabled by default, so need to set
    // SPI Init
    SPI.setDataMode(SPI_MODE2);
    SPI.setClockDivider(SPI_CLOCK_DIV32);
    SPI.setBitOrder(MSBFIRST);
    SPI.begin();
    delay(10);
}

void ADE7753::closeSPI(void) {
    SPI.end();
    delay(10);
}

/*****
 *
 * private functions
 *
 *****/

/** ==== enableChip ====
 * Enable chip, setting low ChipSelect pin (CS)
 * @param none
 *
 */
void ADE7753::enableChip(void){
    digitalWrite(CS,LOW);
}

```

```

}

/** ==== disableChip ====
 * Disable chip, setting high ChipSelect pin (CS)
 * @param none
 *
 */
void ADE7753::disableChip(void){
    digitalWrite(CS,HIGH);
}

/** ==== read8 ====
 * Read 8 bits from the device at specified register
 * @param char containing register direction
 * @return char with contents of register
 *
 */
unsigned char ADE7753::read8(char reg){
    enableChip();
    unsigned char b0;
    delayMicroseconds(50);
    SPI.transfer(reg);
    delayMicroseconds(50);
    b0=SPI.transfer(0x00);
    delayMicroseconds(50);
    disableChip();
    // return (unsigned long)SPI.transfer(0x00);
    return b0;
}

/** ==== read16 ====
 * Read 16 bits from the device at specified register
 * @param char containing register direction
 * @return int with contents of register
 *
 */
unsigned int ADE7753::read16(char reg){
    enableChip();
    unsigned char b1,b0;
    delayMicroseconds(50);
    SPI.transfer(reg);
    delayMicroseconds(50);
    b1=SPI.transfer(0x00);
    delayMicroseconds(50);
    b0=SPI.transfer(0x00);
    delayMicroseconds(50);
    disableChip();
}

```

```

        return (unsigned int)b1<<8 | (unsigned int)b0;
    }

/** ==== read24 ====
 * Read 24 bits from the device at specified register
 * @param: char containing register direction
 * @return: char with contents of register
 *
 */
unsigned long ADE7753::read24(char reg){
    enableChip();
    unsigned char b2,b1,b0;
    delayMicroseconds(50);
    SPI.transfer(reg);
    delayMicroseconds(50);
    b2=SPI.transfer(0x00);
    delayMicroseconds(50);
    b1=SPI.transfer(0x00);
    delayMicroseconds(50);
    b0=SPI.transfer(0x00);
    delayMicroseconds(50);
    disableChip();
    return (unsigned long)b2<<16 | (unsigned long)b1<<8 | (unsigned long)b0;
}

/** ==== write8 ====
 * Write 8 bits to the device at specified register
 * @param reg char containing register direction
 * @param data char, 8 bits of data to send
 *
 */
void ADE7753::write8(char reg, unsigned char data){
    enableChip();
    unsigned char data0 = 0;

    // 8th bit (DB7) of the register address controls the Read/Write mode (Refer to spec
page 55 table 13)
    // For Write -> DB7 = 1 / For Read -> DB7 = 0
    reg |= WRITE;
    data0 = (unsigned char)data;

    delayMicroseconds(50);
    SPI.transfer((unsigned char)reg);    //register selection
    delayMicroseconds(50);
    SPI.transfer((unsigned char)data0);
    delayMicroseconds(50);
    disableChip();
}

```



```

}

/** === write16 ===
 * Write 16 bits to the device at specified register
 * @param reg: char containing register direction
 * @param data: int, 16 bits of data to send
 *
 */
void ADE7753::write16(char reg, unsigned int data){
    enableChip();
    unsigned char data0=0,data1=0;
    // 8th bit (DB7) of the register address controls the Read/Write mode (Refer to spec
page 55 table 13)
    // For Write -> DB7 = 1 / For Read -> DB7 = 0
    reg |= WRITE;
    //split data
    data0 = (unsigned char)data;
    data1 = (unsigned char)(data>>8);

    //register selection, we have to send a 1 on the 8th bit to perform a write
    delayMicroseconds(50);
    SPI.transfer((unsigned char)reg);
    delayMicroseconds(50);
    //data send, MSB first
    SPI.transfer((unsigned char)data1);
    delayMicroseconds(50);
    SPI.transfer((unsigned char)data0);
    delayMicroseconds(50);
    disableChip();
}

/*****
 *
 *   public functions
 *
 *****/

/**
 * In general:
 * @params: void
 * @return: register content (measure) of the proper type depending on register width
 */

/** === setMode / getMode ===
 * This is a 16-bit register through which most of the ADE7753 functionality is accessed.

```

\* Signal sample rates, filter enabling, and calibration modes are selected by writing to this register.

\* The contents can be read at any time.

\*

\*

\* The next table describes the functionality of each bit in the register:

\*

* Bit	Location	Bit Mnemonic	Default Value	Description
* 0	DISHPF	0	0	HPF (high-pass filter) in Channel 1 is disabled when this bit is set.
* 1	DISLPF2	0	0	LPF (low-pass filter) after the multiplier (LPF2) is disabled when this bit is set.
* 2	DISCF	1	1	Frequency output CF is disabled when this bit is set.
* 3	DISSAG	1	1	Line voltage sag detection is disabled when this bit is set.
* 4	ASUSPEND	0	0	By setting this bit to Logic 1, both ADE7753 A/D converters can be turned off. In normal operation, this bit should be left at Logic 0. All digital functionality can be stopped by suspending the clock signal at CLKIN pin.
* 5	TEMPSEL	0	0	Temperature conversion starts when this bit is set to 1. This bit is automatically reset to 0 when the temperature conversion is finished.
* 6	SWRST	0	0	Software Chip Reset. A data transfer should not take place to the ADE7753 for at least 18 us after a software reset.
* 7	CYCMODE	0	0	Setting this bit to Logic 1 places the chip into line cycle energy accumulation mode.
* 8	DISCH1	0	0	ADC 1 (Channel 1) inputs are internally shorted together.
* 9	DISCH2	0	0	ADC 2 (Channel 2) inputs are internally shorted together.
* 10	SWAP	0	0	By setting this bit to Logic 1 the analog inputs V2P and V2N are connected to ADC 1 and the analog inputs V1P and V1N are connected to ADC 2.
* 12, 11	DTRT1,0	0	0	These bits are used to select the waveform register update rate.
		DTRT 1	DTRT0	Update Rate
		0	0	27.9 kSPS (CLKIN/128)
		0	1	14 kSPS (CLKIN/256)
		1	0	7 kSPS (CLKIN/512)
		1	1	3.5 kSPS (CLKIN/1024)
* 14, 13	WAVSEL1,0	0	0	These bits are used to select the source of the sampled data for the waveform register.
		WAVSEL1, 0	Length	Source
		0	0	24 bits active power signal (output of LPF2)
		0	1	Reserved
		1	0	24 bits Channel 1
		1	1	24 bits Channel 2
* 15	POAM	0	0	Writing Logic 1 to this bit allows only positive active power to be accumulated in the ADE7753.

\*

\*

```

* @param none
* @return int with the data (16 bits unsigned).
*/
void ADE7753::setMode(int m){
    write16(MODE, m);
}
int ADE7753::getMode(){
    return read16(MODE);
}

/* The IRQ Interrupt Request pin of the ADE7753 on the Olimex Energy Shield is not wired
to the Arduino board, so need to poll to service the interrupts.
See board spec and wiring diagram at:
http://www.olimex.cl/product\_info.php?products\_id=797&product\_\_name=Arduino\_Energy\_Shield
http://www.olimex.cl/pdf/Main\_Sch.pdf
*/

/** === getInterrupts / setInterrupts
* ADE7753 interrupts can be deactivated at any time by setting the corresponding
* bit in this 16-bit enable register to Logic 0. The status register continues
* to register an interrupt event even if disabled. However, the IRQ output is not activated.
* The next table summarizes the function of each bit in this register:
*
* BitLocation / Interrupt Flag / Description
* 0   AEHF   Indicates that an interrupt occurred because the active energy register,
AEENERGY, is more than half full.
* 1   SAG    Indicates that an interrupt was caused by a SAG on the line voltage.

* 2   CYCEND    Indicates the end of energy accumulation over an integer number of
half line cycles as defined by the content of the LINECYC register—see the Line Cycle
Energy Accumulation Mode section.
* 3   WSMP    Indicates that new data is present in the waveform register.

* 4   ZX      This status bit is set to Logic 0 on the rising and falling edge of the the
voltage waveform. See the Zero-Crossing Detection section.
* 5   TEMP    Indicates that a temperature conversion result is available in the temperature
register.
* 6   RESET   Indicates the end of a reset (for both software or hardware reset). The
corresponding enable bit has no function in the interrupt enable register, i.e., this status bit is
set at the end of a reset, but it cannot be enabled to cause an interrupt.
* 7   AEOF    Indicates that the active energy register has overflowed.
* 8   PKV     Indicates that waveform sample from Channel 2 has exceeded the
VPKLVL value.
* 9   PKI     Indicates that waveform sample from Channel 1 has exceeded the
IPKLVL value.
* 10  VAEHF   Indicates that an interrupt occurred because the active energy
register, VAENERGY, is more than half full.
* 11  VAEOF   Indicates that the apparent energy register has overflowed.

```

\* 12 ZXTO Indicates that an interrupt was caused by a missing zero crossing on the line voltage for the specified number of line cycles—see the Zero-Crossing Timeout section.

\* 13 PPOS Indicates that the power has gone from negative to positive.

\* 14 PNEG Indicates that the power has gone from positive to negative.

\* 15 RESERVED Reserved.

\*

//Register address

IRQEN 0x0A // interrupts enable register

STATUS 0x0B // interrupts status register

RSTSTATUS 0x0C // interrupts status register but read will reset all interrupt flags

\* @param none

\* @return int with the data (16 bits unsigned).

\*/

```
int ADE7753::getEnabledInterrupts(void){
```

```
    return read16(IRQEN);
```

```
}
```

/\*\* === getStatus ===

\* This is an 16-bit read-only register. The status register contains information regarding the source of ADE7753 interrupts

\* @param none

\* @return int with the data (16 bits unsigned).

\*/

```
int ADE7753::getInterruptStatus(void){
```

```
    return read16(STATUS);
```

```
}
```

/\*\* === resetStatus ==

\* Same as the interrupt status register except that the register contents are reset to 0 (all flags cleared) after a read operation.

\* @param none

\* @return int with the data (16 bits unsigned).

\*/

```
int ADE7753::getresetInterruptStatus(void){
```

```
    return read16(RSTSTATUS);
```

```
}
```

/\*\* (1) === getActiveEnergyLineSync ===

\* The instantaneous active power is accumulated in this read-only register over

\* the LINECYC number of half line cycles.

\* Used in combination with CYCEND Interrupt Flag and with LINECYC register (nbr of half-cycles)

\* >>> This is the preferred method The advantage of summing the active energy over an integer number

\* of line cycles is that the sinusoidal component in the active energy is reduced to 0.

```

* This eliminates any ripple in the energy calculation. Energy is calculated more
* accurately and in a shorter time because the integration period can be shortened.
* @param none
* @return long with the data (24 bits 2-complement signed).
*/
long ADE7753::getActiveEnergyLineSync(void){
    return read24(LAENERGY);
}

/** (2) === getApparentEnergyLineSync ===
* The instantaneous real power is accumulated in this read-only register over
* the LINECYC number of half line cycles.
* >>> This is the preferred method The advantage of summing the active energy over an
integer number
* of line cycles is that the sinusoidal component in the active energy is reduced to 0.
* This eliminates any ripple in the energy calculation. Energy is calculated more
* accurately and in a shorter time because the integration period can be shortened.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getApparentEnergyLineSync(void){
    return read24(LVAENERGY);
}

/** (3) === getReactiveEnergyLineSync ===
* The instantaneous reactive power is accumulated in this read-only register over
* the LINECYC number of half line cycles.
* >>> This is the preferred method The advantage of summing the active energy over an
integer number
* of line cycles is that the sinusoidal component in the active energy is reduced to 0.
* This eliminates any ripple in the energy calculation. Energy is calculated more
* accurately and in a shorter time because the integration period can be shortened.
* @param none
* @return long with the data (24 bits 2-complement signed).
*/
long ADE7753::getReactiveEnergyLineSync(void){
    return read24(LVARENERGY);
}

/** === getIRMS ===
* Channel 2 RMS Value (Current Channel).
* The update rate of the Channel 2 rms measurement is CLKIN/4.
* To minimize noise, synchronize the reading of the rms register with the zero crossing
* of the voltage input and take the average of a number of readings.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getIRMS(void){
    long lastupdate = 0;
    ADE7753::getresetInterruptStatus(); // Clear all interrupts

```

```

lastupdate = millis();
while( ! ( ADE7753::getInterruptStatus() & ZX ) ) // wait Zero-Crossing
{ // wait for the selected interrupt to occur
    if ( ( millis() - lastupdate ) > 100)
    {
        wdt_reset();
        Serial.println("\n--> getIRMS Timeout - no AC input");
        break;
    }
}
return read24(IRMS);
}

/** === getVRMS ===
* Channel 1 RMS Value (Voltage Channel).
* The update rate of the Channel 1 rms measurement is CLKIN/4.
* To minimize noise, synchronize the reading of the rms register with the zero crossing
* of the voltage input and take the average of a number of readings.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getVRMS(void){
    long lastupdate = 0;
    ADE7753::getresetInterruptStatus(); // Clear all interrupts
    lastupdate = millis();
    while( ! ( ADE7753::getInterruptStatus() & ZX ) ) // wait Zero-Crossing
    { // wait for the selected interrupt to occur
        if ( ( millis() - lastupdate ) > 100)
        {
            wdt_reset();
            Serial.println("\n--> getIRMS Timeout - no AC input");
            break;
        }
    }
    return read24(VRMS);
}

/** === vrms ===
* Returns the mean of last 100 readings of RMS voltage. Also suppress first reading to avoid
* corrupted data.
* rms measurement update rate is CLKIN/4.
* To minimize noise, synchronize the reading of the rms register with the zero crossing
* of the voltage input and take the average of a number of readings.
* @param none
* @return long with RMS voltage value
*/
long ADE7753::vrms(){
    char i=0;
    long v=0;
    getVRMS();//Ignore first reading to avoid garbage

```

```

    for(i=0;i<100;++i){
        v+=getVRMS();
    }
    return v/100;
}

/** ==== irms ====
 * Returns the mean of last 100 readings of RMS current. Also suppress first reading to avoid
 * corrupted data.
 * rms measurement update rate is CLKIN/4.
 * To minimize noise, synchronize the reading of the rms register with the zero crossing
 * of the voltage input and take the average of a number of readings.
 * @param none
 * @return long with RMS current value in hundreds of [mA], ie. 6709=67[mA]
 */
long ADE7753::irms(){
    char n=0;
    long i=0;
    getIRMS();//Ignore first reading to avoid garbage
    for(n=0;n<100;++n){
        i+=getIRMS();
    }
    return i/100;
}

/** ==== getWaveform ====
 * This read-only register contains the sampled waveform data from either Channel 1,
 * Channel 2, or the active power signal. The data source and the length of the waveform
 * registers are selected by data Bits 14 and 13 in the mode register.
 * - Max sampling CLKIN/128 = 3.579545 MHz / 128 = 27.9 kSPS
 * - Bandwidth 14 kHz
 * - one of four output sample rates can be chosen by using Bits 11 and 12
 * of the mode register (WAVSEL1,0). The output sample rate
 * can be 27.9 kSPS, 14 kSPS, 7 kSPS, or 3.5 kSPS
 * - arrivals of new waveform samples after each read is indicated by interrupt
 * request IRQ, but unfortunately IRQ is not wired in the Olimex Energy Shield
 * - The interrupt request output, IRQ, signals a new sample
 * availability by going active low.
 * - In waveform sampling mode, the WSMP bit (Bit 3) in the
 * interrupt enable register must also be set to Logic 1.
 * - The interrupt request output IRQ stays low until the interrupt
 * routine reads the reset status register.
 * - Interrupt Flag WSMP (bit location 3) in the Interrupt Status Register
 * indicates that new data is present in the waveform register.
 * Therefore arrival of new waveform samples may be indicated by polling
 * this flag and then reading the 24-bit waveform register
 * --- we use this polling method for Arduino/Nanode ---
 * - When acquiring waveform data, disable low pass filter in order to
 * obtain and view all the high harmonics
 * @param none

```

```

* @return long with the data (24 bits 2-complement signed).
*/
long ADE7753::getWaveform(void){ // this function will have to be rewritten for allowing
rapid polling
    // of WSMP flag in ISR and getting rapid Waveform data over 1 full cycle
    // and storing to Arduino tiny RAM
    return read24(WAVEFORM);
}

/** ==== getIpeakReset ====
* Same as Channel 1 Peak Register except that the register contents are reset to 0 after read.
* @param none
* @return long with the data (24 bits 24 bits unsigned).
*/
long ADE7753::getIpeakReset(void){
    return read24(RSTIPEAK);
}

/** ==== getVpeakReset ====
* Same as Channel 2 Peak Register except that the register contents are reset to 0 after a read.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getVpeakReset(void){
    return read24(RSTVPEAK);
}

/** ==== getPeriod ====
* Period of the Channel 2 (Voltage Channel) Input Estimated by Zero-Crossing Processing.
* The ADE7753 provides the period measurement of the grid line.
* The period register is an unsigned 16-bit register and is updated every period.
* The MSB of this register is always zero.
* The resolution of this register is 2.2  $\mu$ s/LSB when CLKIN = 3.579545 MHz,
* which represents 0.013% when the line frequency is 60 Hz.
* When the line frequency is 60 Hz, the value of the period register is
* approximately  $CLKIN/4/32/60 \text{ Hz} \times 16 = 7457d$ .
* The length of the register enables the measurement of line frequencies as low as 13.9 Hz.
* @param none
* @return int with the data (16 bits unsigned).
*/
int ADE7753::getPeriod(void){
    return read16(PERIOD);
}

/** ==== getTemp ====
* Force a temperature measure and then returns it. This is done by setting bit 5 HIGH in
MODE register.
* Temperature measuring can't be calibrated, the values used in this function are according to
the datasheet
* (register TEMP is 0x00 at -25 celsius degrees).

```



\* The contents of the temperature register are signed (twos complement) with a resolution of  
 \* approximately 1.5 LSB/°C. The temperature register produces a code of 0x00 when the  
 ambient  
 \* temperature is approximately -25°C. The temperature measurement is uncalibrated in the  
 ADE7753  
 \* and has an offset tolerance as high as ±25°C.  
 \* @param none  
 \* @return char with the temperature in celsius degrees.  
 \*/

```
char ADE7753::getTemp(){
    unsigned char r=0;
    long lastMode = 0;
    long lastupdate = 0;
    lastMode = getMode();
    //Temp measure
    setMode(TEMPSEL);
    ADE7753::getresetInterruptStatus(); // Clear all interrupts
    lastupdate = millis();
    while( ! ( ADE7753::getInterruptStatus() & TEMPREADY ) ) // wait for Temperature
    measurement to be ready
        { // wait for the selected interrupt to occur
            if ( ( millis() - lastupdate ) > 100)
                {
                    wdt_reset();
                    Serial.println("\n--> Temperature Timeout no AC input");
                    ADE7753::getresetInterruptStatus(); // Clear all interrupts
                    break;
                }
        }
    //Read register
    r= read8(TEMP);

    // // Do it twice to make sure
    // setMode(TEMPSEL);
    // ADE7753::getresetInterruptStatus(); // Clear all interrupts
    // lastupdate = millis();
    // while( ! ( ADE7753::getInterruptStatus() & TEMPREADY ) ) // wait for
    Temperature measurement to be ready
        { // wait for the selected interrupt to occur
            if ( ( millis() - lastupdate ) > 2500)
                {
                    Serial.println("--> Temperature Timeout no AC input");
                    ADE7753::getresetInterruptStatus(); // Clear all interrupts
                    break;
                }
        }
    //Read register
    r= read8(TEMP);

    // Set to the previous mode
```

```

        setMode (lastMode);
        return r;
    }

// Functions for manual setting of calibrations

/** === energySetup ===
 * @param
 * @param
 */
void ADE7753::energySetup(int wgain, char wdiv, int apos, int vagain, char vadiv, char
phcal){
    write16(WGAIN,wgain);
    write8(WDIV,wdiv);
    write16(APOS,apos);
    write16(VAGAIN,vagain);
    write8(VADIV,vadiv);
    write8(PHCAL,phcal);
}

/** === frequencySetup ===
 * The output frequency on the CF pin is adjusted by writing to this 12-bit
 * read/write register—see the Energy-to-Frequency Conversion section.
 * @param cfnum: integer containing number (12 bits available unsigned. ie range=[0,4095])
 * @param cfden: the same as cfnum
 */
void ADE7753::frequencySetup(int cfnum, int cfden){
    write16(CFNUM,cfnum);
    write16(CFDEN,cfden);
}

/** === analogSetup ===
 * This 8-bit register is used to adjust the gain selection for the PGA in Channels 1 and 2
 * @param gain_ch1 char set the PGA channel 1 gain, use constants GAIN_1, GAIN_2,
GAIN_4, GAIN_8 and gain_16
 * @param gain_ch2 char set the PGA channel 2 gain, use constants GAIN_1, GAIN_2,
GAIN_4, GAIN_8 and gain_16
 * @param os_ch1 char set channel 1 analog offset, range : [-32,32]
 * @param os_ch2 char set channel 1 analog offset, range : [-32,32]
 * @param scale_ch1 char
 * @param integrator_ch1 char
 * @return char with the data (8 bits unsigned).
 */
void ADE7753::analogSetup(char gain_ch1, char gain_ch2,char os_ch1,char os_ch2,char
scale_ch1,char integrator_ch1){
    unsigned char pga = (gain_ch2<<5) | (scale_ch1<<3) | (gain_ch1); //format is |3 bits
PGA2 gain|2 bits full scale|3 bits PGA1 gain
    char sign = 0;
    char ch1os = 0, ch2os = 0;

```

```

write8(GAIN,pga); //write GAIN register, format is |3 bits PGA2 gain|2 bits full scale|3
bits PGA1 gain

```

```

// CH1OS: ch1 offset 6-bit, sign magnitude on Bit-5 then integrator on Bit-7
// Refer to spec Page 58 Table 16
if(os_ch1<0){
    sign=1;
    os_ch1=-os_ch1;
} else{ sign=0; }
ch1os = (integrator_ch1<<7) | (sign<<5) | os_ch1;
write8(CH1OS,ch1os);

```

```

// CH2OS: ch2 offset, sign magnitude (ch2 doesn't have integrator) and the offset
applied is inverted (ie offset of -1 sums 1)

```

```

if(os_ch2<0){
    sign=1;
    os_ch2=-os_ch2;
} else{ sign=0; }
ch2os = (sign<<5) | os_ch2;
write8(CH2OS,ch2os);

```

```

}

```

```

/** ==== rmsSetup ====
**/

```

```

void ADE7753::rmsSetup(int irmsos, int vrmsos){
    write16(VRMSOS,vrmsos);
    write16(IRMSOS,irmsos);
}

```

```

void ADE7753::miscSetup(int zxtout, char sagsyc, char saglvl, char ipklvl, char vpklvl, char
tmode){

```

```

// ZXTOUT 12-bit (U) - Zero-Crossing Timeout
// SAGCYC 8-bit (U) - Sag Line Cycle Register.
// SAGLVL 8-bit (U) - Sag Voltage Level.
// IPKLVL 8-bit (U) - Channel 1 Peak Level Threshold
// VPKLVL 8-bit (U) - Channel 2 Peak Level Threshold
// TMODE 8-bit (U) - Test Mode Register
write16(ZXTOUT,zxtout);
write8(SAGCYC,sagsyc);
write8(SAGLVL,saglvl);
write8(IPKLVL,ipklvl);
write8(VPKLVL,vpklvl);
write8(TMODE,tmode);

```

```

}

```

```

void ADE7753::setInterruptsMask(int Mask16){
    write16(IRQEN, Mask16);
}

```

```

void ADE7753::setLineCyc(int d){
    write16(LINECYC,d);
}

// May have to remove this code because of code/RAM size restriction with Nanode

#define NanodeReduceCodeSize 1
#ifndef NanodeReduceCodeSize

/** === printGetResetInterruptStatus ===
 * used for code verification et debugging
 */
void ADE7753::printGetResetInterruptStatus(void){

    int InterruptMask;

    Serial.print("Interrupt Status (binary): ");
    InterruptMask = ADE7753::getresetInterruptStatus();
    Serial.println(InterruptMask,BIN);

    if ( InterruptMask & AEHF )    Serial.println("AEHF");
    if ( InterruptMask & SAG )    Serial.println("SAG");
    if ( InterruptMask & CYCEND ) Serial.println("CYCEND");
    if ( InterruptMask & WSMP )   Serial.println("WSMP");
    if ( InterruptMask & ZX )     Serial.println("ZX");
    if ( InterruptMask & TEMPREADY ) Serial.println("TEMPREADY");
    if ( InterruptMask & RESET )  Serial.println("RESET");
    if ( InterruptMask & AEOF )   Serial.println("AEOF");
    if ( InterruptMask & PKV )    Serial.println("PKV");
    if ( InterruptMask & PKI )    Serial.println("PKI");
    if ( InterruptMask & VAEHF )  Serial.println("VAEHF");
    if ( InterruptMask & VAEOF )  Serial.println("VAEOF");
    if ( InterruptMask & ZXTO )   Serial.println("ZXTO");
    if ( InterruptMask & PPOS )   Serial.println("PPOS");
    if ( InterruptMask & PNEG )   Serial.println("PNEG");
    if ( InterruptMask & RESERVED ) Serial.println("RESERVED");
}

/** === printGetMode ===
 * used for code verification et debugging
 */
void ADE7753::printGetMode(void){

    int ModeMask;

    Serial.print("Mode (binary): ");
    ModeMask = ADE7753::getMode();
    Serial.println(ModeMask,BIN);
}

```

```

if ( ModeMask & DISHPF ) Serial.println("DISHPF");
if ( ModeMask & DISLPF2 ) Serial.println("DISLPF2");
if ( ModeMask & DISCF ) Serial.println("DISCF");
if ( ModeMask & DISSAG ) Serial.println("DISSAG");
if ( ModeMask & ASUSPEND ) Serial.println("ASUSPEND");
if ( ModeMask & TEMPSEL ) Serial.println("TEMPSEL");
if ( ModeMask & SWRST ) Serial.println("SWRST");
if ( ModeMask & CYCMODE ) Serial.println("CYCMODE");
if ( ModeMask & DISCH1 ) Serial.println("DISCH1");
if ( ModeMask & DISCH2 ) Serial.println("DISCH2");
if ( ModeMask & SWAP ) Serial.println("SWAP");
if ( ModeMask & DTRT0 ) Serial.println("DTRT0");
if ( ModeMask & DTRT1 ) Serial.println("DTRT1");
if ( ModeMask & WAVSEL0 ) Serial.println("WAVSEL0");
if ( ModeMask & WAVSEL1 ) Serial.println("WAVSEL1");
if ( ModeMask & POAM ) Serial.println("POAM");
}

/** === printAllRegisters ===
 * used for code verification et debugging
 */
void ADE7753::printAllRegisters(void){

    unsigned char b = 0;
    int i = 0;
    unsigned char sign = 0;
    int integrator = 0;
    Serial.print("WAVEFORM "); Serial.println(read24(WAVEFORM),DEC);
    Serial.print("AENERGY "); Serial.println(read24(AENERGY),DEC);
    Serial.print("RAENERGY "); Serial.println(read24(RAENERGY),DEC);
    Serial.print("LAENERGY "); Serial.println(read24(LAENERGY),DEC);
    Serial.print("VAENERGY "); Serial.println(read24(VAENERGY),DEC);
    Serial.print("RVAENERGY "); Serial.println(read24(RVAENERGY),DEC);
    Serial.print("LVAENERGY "); Serial.println(read24(LVAENERGY),DEC);
    Serial.print("LVARENERGY "); Serial.println(read24(LVARENERGY),DEC);
    Serial.print("MODE "); Serial.println(read16(MODE),BIN);
    Serial.print("IRQEN "); Serial.println(read16(IRQEN),BIN);
    Serial.print("STATUS "); Serial.println(read16(STATUS),BIN);
    Serial.print("RSTSTATUS "); Serial.println(read16(RSTSTATUS),BIN);

    // CHIOS register - see spec Page 58 Table 16
    b = read8(CHIOS);
    sign = b & 0x20; // test sign on bit-6
    if ( sign > 1 ) {i = - int (b & 0x1F); } else {i = int (b & 0x1F);}
    Serial.print("CHIOS "); Serial.print(i, DEC); // convert signed 6-bit into signed 16-
bit
    integrator = b & 0x80; // test integrator flag on bit-8
    if ( integrator > 1 ) { Serial.println(" Integrator ON");} else { Serial.println(" Integrator
OFF");}
}

```

```

    b = read8(CH2OS);
    sign = b & 0x20; // test sign on bit-6
    if ( sign > 1 ) {i = - int (b & 0x1F) ;} else {i = int (b & 0x1F);}
    Serial.print("CH2OS "); Serial.println(i, DEC); // convert signed 6-bit into signed 16-
bit

    Serial.print("GAIN "); Serial.println(read8 (GAIN),BIN);
    Serial.print("PHCAL "); Serial.println(read8 (PHCAL),HEX);
    Serial.print("APOS "); Serial.println(int(read16(APOS)),DEC);
    Serial.print("WGAIN "); Serial.println(int (read16(WGAIN)),DEC);
    Serial.print("WDIV "); Serial.println(read8 (WDIV),DEC);
    Serial.print("CFNUM "); Serial.println(read16(CFNUM),DEC);
    Serial.print("CFDEN "); Serial.println(read16(CFDEN),DEC);
    Serial.print("->IRMS "); Serial.println(read24(IRMS),DEC);
    Serial.print("->VRMS "); Serial.println(read24(VRMS),DEC);
    Serial.print("--IRMSOS "); Serial.println(int(read16(IRMSOS)),DEC);
    Serial.print("--VRMSOS "); Serial.println(int(read16(VRMSOS)),DEC);
    Serial.print("VAGAIN "); Serial.println(int(read16(VAGAIN)),DEC);
    Serial.print("VADIV "); Serial.println(read8 (VADIV),DEC);
    Serial.print("LINECYC "); Serial.println(read16(LINECYC),DEC);
    Serial.print("ZXTOUT "); Serial.println(read16(ZXTOUT),DEC);
    Serial.print("SAGCYC "); Serial.println(read8 (SAGCYC),DEC);
    Serial.print("SAGLVL "); Serial.println(read8 (SAGLVL),DEC);
    Serial.print("IPKLVL "); Serial.println(read8 (IPKLVL),DEC);
    Serial.print("VPKLVL "); Serial.println(read8 (VPKLVL),DEC);
    Serial.print("IPEAK "); Serial.println(read24(IPEAK),DEC);
    Serial.print("RSTIPEAK "); Serial.println(read24(RSTIPEAK),DEC);
    Serial.print("VPEAK "); Serial.println(read24(VPEAK),DEC);
    Serial.print("RSTVPEAK "); Serial.println(read24(RSTVPEAK),DEC);
    Serial.print("TEMP "); Serial.println(read8 (TEMP),DEC);
    Serial.print("PERIOD "); Serial.println(read16(PERIOD),DEC);
    Serial.print("TMODE "); Serial.println(read8 (TMODE),DEC);
    Serial.print("CHKSUM "); Serial.println(read8 (CHKSUM),DEC);
    Serial.print("DIEREV "); Serial.println(read8 (DIEREV),DEC);
}

/** ==== chkSum ====
 * Checksum Register. This 6-bit read-only register is equal to the sum of all the ones in the
previous read.
 * see the ADE7753 Serial Read Operation section.
 * @param none
 * @return char with the data (6 bits unsigned).
 */
char ADE7753::chkSum(){
    return read8(CHKSUM);
}

/** ==== getActiveEnergy ====
 * Active power is accumulated (integrated) over time in this 24-bit, read-only register

```

```

* @param none
* @return long with the data (24 bits 2-complement signed).
*/
long ADE7753::getActiveEnergy(void){
    return read24(AENERGY);
}

/** ==== getActiveEnergyReset ====
* Same as the active energy register except that the register is reset to 0 following a read
operation.
* @param none
* @return long with the data (24 bits 2-complement signed).
*/
long ADE7753::getActiveEnergyReset(void){
    return read24(RAENERGY);
}

/** ==== getApparentEnergy ====
* Apparent Energy Register. Apparent power is accumulated over time in this read-only
register.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getApparentEnergy(void){
    return read24(VAENERGY);
}

/** ==== getApparentEnergyReset ====
* Same as the VAENERGY register except that the register is reset to 0 following a read
operation.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getApparentEnergyReset(void){
    return read24(RVAENERGY);
}

/** ==== getCurrentOffset ====
* Channel 2 RMS Offset Correction Register.
* @param none
* @return int with the data (12 bits 2-complement signed).
*/
int ADE7753::getCurrentOffset(){
    return read16(IRMSOS);
}

/** ==== getVoltageOffset ====
* Channel 2 RMS Offset Correction Register.
*
* @param none

```

```

* @return int with the data (12 bits 2-complement's signed).
*/
int ADE7753::getVoltageOffset(){
    return read16(VRMSOS);
}

/** === setZeroCrossingTimeout / getZeroCrossingTimeout ===
* Zero-Crossing Timeout. If no zero crossings are detected
* on Channel 2 within a time period specified by this 12-bit register,
* the interrupt request line (IRQ) is activated
* @param none
* @return int with the data (12 bits unsigned).
*/
void ADE7753::setZeroCrossingTimeout(int d){
    write16(ZXTOUT,d);
}
int ADE7753::getZeroCrossingTimeout(){
    return read16(ZXTOUT);
}

/** === getSagCycles / setSagCycles ===
* Sag Line Cycle Register. This 8-bit register specifies the number of
* consecutive line cycles the signal on Channel 2 must be below SAGLVL
* before the SAG output is activated.
* @param none
* @return char with the data (8 bits unsigned).
*/
char ADE7753::getSagCycles(){
    return read8(SAGCYC);
}
void ADE7753::setSagCycles(char d){
    write8(SAGCYC,d);
}

/** === getLineCyc / setLineCyc ===
* Line Cycle Energy Accumulation Mode Line-Cycle Register.
* This 16-bit register is used during line cycle energy accumulation mode
* to set the number of half line cycles for energy accumulation
* @param none
* @return int with the data (16 bits unsigned).
*/
int ADE7753::getLineCyc(){
    return read16(LINECYC);
}

/** === getSagVoltageLevel / setSagVoltageLevel ===
* Sag Voltage Level. An 8-bit write to this register determines at what peak
* signal level on Channel 2 the SAG pin becomes active. The signal must remain

```



```

* low for the number of cycles specified in the SAGCYC register before the SAG pin is
activated
* @param none
* @return char with the data (8 bits unsigned).
*/
char ADE7753::getSagVoltageLevel(){
    return read8(SAGLVL);
}
void ADE7753::setSagVoltageLevel(char d){
    write8(SAGLVL,d);
}

/** ==== getIPeakLevel / setIPeakLevel ====
* Channel 1 Peak Level Threshold (Current Channel). This register sets the level of the
current
* peak detection. If the Channel 1 input exceeds this level, the PKI flag in the status register is
set.
* @param none
* @return char with the data (8 bits unsigned).
*/
char ADE7753::getIPeakLevel(){
    return read8(IPKLVL);
}
void ADE7753::setIPeakLevel(char d){
    write8(IPKLVL,d);
}

/** ==== getVPeakLevel / setVPeakLevel ====
* Channel 2 Peak Level Threshold (Voltage Channel). This register sets the level of the
* voltage peak detection. If the Channel 2 input exceeds this level,
* the PKV flag in the status register is set.
* @param none
* @return char with the data (8bits unsigned).
*/
char ADE7753::getVPeakLevel(){
    return read8(VPKLVL);
}
void ADE7753::setVPeakLevel(char d){
    write8(VPKLVL,d);
}

/** ==== getVpeak ====
* Channel 2 Peak Register. The maximum input value of the voltage channel since the last
read of the register is stored in this register.
* @param none
* @return long with the data (24 bits unsigned).
*/
long ADE7753::getVpeak(void){
    return read24(VPEAK);
}

```

```
/** ==== getIpeak ====
 * Channel 1 Peak Register. The maximum input value of the current channel since the last
 * read
 * of the register is stored in this register.
 * @param none
 * @return long with the data (24 bits unsigned) .
 */
long ADE7753::getIpeak(void){
    return read24(IPEAK);
}

/** ==== energyGain ====
 * @param
 * @param
 */
void ADE7753::energyGain(int wgain, int vagain){
    write16(WGAIN,wgain);
    write16(VAGAIN,vagain);
}

#endif
```