

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
CURSO DE ENGENHARIA ELÉTRICA

IGOR LUIS COELHO SOUSA

**SISTEMA DE MEDIÇÃO LOCAL E REMOTA DE ENERGIA ELÉTRICA EM  
CONSUMIDORES DE BAIXA TENSÃO BASEADO EM ARDUINO.**

São Luís, MA  
2017

IGOR LUIS COELHO SOUSA

**SISTEMA DE MEDIÇÃO LOCAL E REMOTA DE ENERGIA ELÉTRICA EM  
CONSUMIDORES DE BAIXA TENSÃO BASEADO EM ARDUINO.**

Monografia apresentada à Coordenação do curso de Engenharia Elétrica, como requisito para obtenção do título de Bacharel em Engenharia Elétrica, na Universidade Federal do Maranhão - UFMA.

Orientador: Denivaldo Cicero Pavão Lopes

São Luís, MA

2017

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a)  
autor(a).  
Núcleo Integrado de Bibliotecas/UFMA

Sousa, Igor Luis Coelho.

Sistema de medição local e remota de energia elétrica em  
consumidores de baixa tensão baseado em arduino / Igor Luis Coelho  
Sousa. - 2017.  
57 f.

Orientador(a): Denivaldo Cicero Pavão Lopes. Monografia  
(Graduação) - Curso de Engenharia Elétrica,  
Universidade Federal do Maranhão, anexo do grupo de potência, 2017.

1. Arduino. 2. Consumo. 3. Energia. 4. Microcontrolador.  
5. Monitoramento. I. Lopes, Denivaldo Cicero Pavão. II. Título.

**SISTEMA DE MEDIÇÃO LOCAL E REMOTA DE ENERGIA ELÉTRICA EM  
CONSUMIDORES DE BAIXA TENSÃO BASEADO EM ARDUINO.**

**IGOR LUIS COELHO SOUSA**

Aprovada em: 25 / 07 / 2017.

**BANCA EXAMINADORA:**



Orientador Prof. Dr.: Denivaldo Cicero Pavao Lopes.  
Departamento de Engenharia Elétrica - UFMA



1º Examinador Prof. Dr.: Carlos Alberto Brandão Barbosa Leite  
Departamento de Engenharia Elétrica - UFMA



2º Examinador Prof. Dr.: Vicente Leonardo Paucar Casas  
Departamento de Engenharia Elétrica - UFMA

Dedico este trabalho aos meus pais que sonharam estes sonho junto a mim e me ajudaram a construí-lo com o suor de seus trabalhos.

## **AGRADECIMENTO**

A Deus pelo dom da vida e todas suas bênção que me ajudaram a superar todas as dificuldades.

Aos meus pais, Luis Antonio Cardoso Sousa e Luciana de Jesus Coelho Sousa, por todo seu amor e carinho que me ajudaram alcançar meus sonhos.

A Karla Virgínia, por todo seu amor e apoio nas horas mais difíceis.

Ao meu orientador Prof. Dr. Denivaldo Cicero Pavao Lopes, pelo apoio, por seus ensinamentos, conselhos e incentivos durante todo este trabalho.

Aos professores de engenharia elétrica da Universidade Federal do Maranhão, pelas horas intermináveis de estudo, trabalho e pelo incentivo.

Aos amigos da VALE, em especial Keiji Kanno, Alzira Silva e Josenilson Rocha, por todos seus ensinamentos e incentivo.

A todos meus amigos de graduação, em especial Max Alcobaça e Dennis Sodré pela sua amizade e apoio.

Ao meu amigo Rodrigo Nunes, por todo apoio e incentivo.

A todas as pessoas que não estão nesta lista, mas que de alguma forma me ajudaram para a realização deste trabalho.

Obrigado a todos!

"Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível."

**(Charles Chaplin)**

## RESUMO

O presente trabalho propõe um sistema de software/hardware alternativo para medição do consumo de energia elétrica residencial, a fim de que o consumidor consiga monitorar seus gastos de forma instantânea e eficiente. Tendo em vista que os sucessivos aumentos na tarifa de energia podem deixar o consumidor confuso e por não possuir uma forma prática para acompanhar a evolução do valor em reais de sua fatura mensal. Para alcançar tal objetivo, desenvolveu-se uma central de monitoramento com a placa Arduino, que interpreta sinais analógicos recebidos de um sensor de corrente não invasivo e de um sensor de tensão instalado na rede elétrica (ramal de entrada do consumidor de baixa tensão). Os resultados desse monitoramento são mostrados em um display LCD para visualização local. Os mesmos dados também são disponibilizados remotamente através do browser no site do ThingSpeak e de um aplicativo para smartphone. O sistema proposto também torna possível a exibição do valor em reais correspondente ao consumo de energia elétrica. Alguns experimentos servem para mostrar a viabilidade técnica do sistema de hardware/software proposto para medição de energia elétrica em unidades consumidoras residenciais.

**Palavras-chave:** Arduino, microcontrolador, consumo, energia, monitoramento.

## **ABSTRACT**

This Project aims to present an alternative software/hardware system for measurement of residential electricity consumption, so that the consumer can remote monitoring their expenses in an instantaneous and efficient way. The successive increases in energy costs may leave customers confused because they do not have a practical way to follow the evolution of the value in Reais (Brazilian money) of their monthly invoice. To reach that goal, a monitoring center with Arduino boards was developed, which interprets analog signals received from a non-invasive current sensor and a voltage sensor installed in the electrical network (input branch of the low-voltage consumer). The results of this monitoring are displayed on a LCD screen for local viewing. The same data is also available remotely at [thingspeak.com](https://thingspeak.com) and a smartphone application. The proposed system, also makes it possible to display the value in Reais corresponding to the monthly consumption of electricity. Some experiments demonstrate the technical feasibility of the proposed hardware/software system to measure electrical energy in residential consumption units.

**Keywords:** Arduino, microcontroller, consumption, energy, monitoring.

## LISTA DE FIGURAS

|   |    |
|---|----|
| <b>Figura 1:</b> Uma forma de onda de tensão ca senoidal.....                           | 21 |
| <b>Figura 2:</b> Ciclo de corrente alternada. ....                                      | 21 |
| <b>Figura 3:</b> Relação de fase entre tensão e corrente para uma carga resistiva. .... | 22 |
| <b>Figura 4:</b> Relações de fase de tensão e corrente numa carga reativa. ....         | 22 |
| <b>Figura 5:</b> Triângulo de potências. ....   | 23 |
| <b>Figura 6:</b> Arduino MEGA 2560. ....  | 24 |
| <b>Figura 7:</b> Tela inicial do IDE do Arduino. ....                                   | 26 |
| <b>Figura 8:</b> Ethernet shield W5100. ....  | 26 |
| <b>Figura 9:</b> Canal monitorado ThingSpeak.....                                       | 27 |
| <b>Figura 10:</b> Modelo proposto.....  | 28 |
| <b>Figura 11:</b> Esquema de montagem SCT-013-030.....                                  | 29 |
| <b>Figura 12:</b> Esquema de ligação do Trafo 220/9V. ....                              | 30 |
| <b>Figura 13:</b> Esquema de ligação: RTC e LCD 5110. ....                              | 31 |
| <b>Figura 14:</b> Montagem Final.....   | 31 |
| <b>Figura 15:</b> Parâmetros de calibração. ....  | 32 |
| <b>Figura 16:</b> Exemplo de chave de escrita. ....                                     | 33 |
| <b>Figura 17:</b> Estrutura de comunicação.....   | 34 |
| <b>Figura 18:</b> Código de envio para o servidor ThingSpeak. ....                      | 34 |
| <b>Figura 19:</b> IoT ThingSpeak Monitor Widget. ....                                   | 35 |
| <b>Figura 20:</b> Carga RL.....   | 36 |
| <b>Figura 21:</b> Capacitor para ligação RC. ....                                       | 37 |
| <b>Figura 22:</b> Ligação RLC (visão 360° da bancada).....                              | 37 |
| <b>Figura 23:</b> Teste de consumo de energia elétrica. ....                            | 38 |
| <b>Figura 24:</b> Medidor na entrada do quadro de distribuição. ....                    | 39 |
| <b>Figura 25:</b> Resultados RL.....  | 40 |
| <b>Figura 26:</b> Resultados RC .....   | 41 |
| <b>Figura 27:</b> Resultados RLC .....  | 41 |
| <b>Figura 28:</b> Comparativo medidor E22A x Arduino .....                              | 42 |
| <b>Figura 29:</b> Consumo parcial (CEMAR).....  | 43 |
| <b>Figura 30:</b> Consumo parcial (Arduino).....  | 43 |
| <b>Figura 31:</b> Widget, Potência (W) 1 hora e 24 horas. ....                          | 44 |

## LISTA DE TABELAS

|   |    |
|---|----|
| <b>Tabela 1:</b> Especificação técnica MEGA 2560..... | 25 |
| <b>Tabela 2:</b> Componentes do projeto.....          | 29 |
| <b>Tabela 3:</b> Comparativo Arduino x Fluke 345..... | 40 |

## LISTA DE ABREVIATURAS E SIGLAS

|       |                                      |
|-------|--------------------------------------|
| ANEEL | Agência Nacional de Energia Elétrica |
| CEMAR | Companhia Energética do Maranhão     |
| IDE   | Integrated Development Environment   |
| RTC   | Real Time Clock                      |
| AC    | Corrente Alternada                   |
| LCD   | Liquid Crystal Display               |
| Hz    | Hertz                                |
| UFMA  | Universidade Federal do Maranhão     |
| TC    | Transformador de Corrente            |
| TCP   | Transmission Control Protocol        |
| UDP   | User Datagram Protocol               |
| HTTP  | Hypertext Transfer Protocol          |
| IoT   | Internet of Things                   |

## LISTA DE EQUAÇÕES

|  |    |
|--|----|
| <b>Equação 1:</b> Equação para cálculo de energia elétrica .....       | 18 |
| <b>Equação 2:</b> Equação para cálculo da potência aparente .....      | 22 |
| <b>Equação 3:</b> Equação para cálculo da potência ativa.....          | 22 |
| <b>Equação 4:</b> Equação para cálculo da potência reativa.....        | 22 |
| <b>Equação 5:</b> Equação para cálculo do fator de potência .....      | 22 |
| <b>Equação 6:</b> Equação para cálculo da variável de calibração ..... | 29 |

## LISTA DE APÊNDICE

|  |    |
|--|----|
| <b>Apêndice - A:</b> Conta de energia CEMAR .....    | 46 |
| <b>Apêndice - B:</b> Especificação SCT-013-030 ..... | 47 |
| <b>Apêndice - C:</b> Código Final .....              | 48 |

## SUMÁRIO

|  |    |
|--|----|
| 1. INTRODUÇÃO .....                            | 16 |
| 1.1. Problemática .....                        | 16 |
| 1.2. Solução Proposta .....                    | 17 |
| 1.3. Objetivos .....                           | 17 |
| 1.3.1. Objetivo Geral .....                    | 17 |
| 1.3.2. Objetivos Específicos .....             | 17 |
| 1.4. Metodologia Aplicada.....                 | 18 |
| 1.5. Apresentação dos capítulos.....           | 18 |
| 2. FUNDAMENTAÇÃO TEÓRICA.....                  | 20 |
| 2.1. Energia Elétrica .....                    | 20 |
| 2.2. Tensão e Corrente CA .....                | 20 |
| 2.2.1. Cargas resistivas e reativas .....      | 21 |
| 2.3. Potência: Ativa, Reativa e Aparente ..... | 23 |
| 2.4. Arduino .....                             | 24 |
| 2.4.1. IDE Arduino .....                       | 25 |
| 2.4.2. Ethernet Shield.....                    | 26 |
| 3. METODOLOGIA .....                           | 28 |
| 3.1. Modelo Proposto.....                      | 28 |
| 3.2. Componentes do sistema .....              | 28 |
| 3.3. Montagem.....                             | 29 |
| 3.4. Calibração.....                           | 31 |
| 3.5. Conexão com o ThingSpeak.....             | 33 |
| 3.6. Processamento de dados .....              | 33 |
| 3.7. Testes .....                              | 35 |
| 3.7.1. Carga Indutiva .....                    | 36 |
| 3.7.2. Carga Capacitiva.....                   | 36 |
| 3.7.3. Carga Indutiva – Capacitiva .....       | 37 |

|  |    |
|--|----|
| 3.7.4. Comparativo com monofásico eletrônico Landys+Gyr E22A ..... | 38 |
| 3.8. Estudo de caso .....  | 38 |
| 4. RESULTADOS E ANÁLISE .....                                      | 40 |
| 4.1. Testes com as cargas.....                                     | 40 |
| 4.2. Landys+Gyr E22A x Arduino.....                                | 41 |
| 4.3. Estudo de caso .....  | 42 |
| 5. CONCLUSÃO .....   | 45 |
| REFERÊNCIAS.....   | 46 |
| APÊNDICE - A .....   | 48 |
| APÊNDICE - B .....   | 49 |
| APÊNDICE - C .....   | 50 |

## **1. INTRODUÇÃO**

Atualmente, o consumo de energia elétrica continua sendo um tema muito debatido pois o desenvolvimento socioeconômico tende sempre a criar demanda por mais energia e a sua geração ainda utiliza fontes não renováveis (FAPESP, 2010). Segundo o Ministério de Minas e Energia, no ano de 2015, as fontes renováveis da matriz elétrica brasileira representavam 24,5% contra 75,9% na média mundial, e 76,9% no bloco da OCDE (Organização para a Cooperação e Desenvolvimento Econômico).

Mesmo com a porcentagem do uso de fontes não renováveis do Brasil sendo pequena em relação à média mundial, as secas dos reservatórios que alimentam as hidroelétricas do país colaboraram para a criação da bandeira tarifária que entrou em vigor em 2015 no país. A bandeira vermelha fez os brasileiros desembolsarem quase R\$ 15 bilhões a mais pela energia elétrica, permaneceu em vigor quase o ano inteiro, as tarifas dispararam 50% em média (KAFRUNI, 2016).

Este ano a bandeira amarela vigorou durante o mês de março (cobrança extra de R\$ 2 para cada 100 kWh consumidos) e no mês de abril voltou ser vermelha, patamar 1, com custo de R\$ 3 a cada 100 kWh. Como o sinal é vermelho, os consumidores devem usar com eficiência e combater o desperdício de energia elétrica (ANEEL, 2017).

Diante deste cenário, resta ao consumidor obter formas de monitorar seu consumo para evitar surpresas desagradáveis no fim do mês com a conta de energia, visto que a tarifa cobrada no Maranhão é a terceira maior da região nordeste com aproximadamente 0,496 R\$/kWh (ANEEL, 2017).

### **1.1. Problemática**

Devido aos sucessivos aumentos na tarifa de energia, tanto quanto a inclusão das bandeiras tarifárias, o gasto mensal com a energia elétrica vem aumentando. Assim, os consumidores buscam maneiras de utilizar de forma efetiva a energia elétrica e evitar desperdícios.

Uma das maneiras dos consumidores evitarem desperdícios seria monitorar seu consumo de energia elétrica, acompanhando a própria medição da concessionária, que consiste na leitura constante do medidor instalado na entrada da instalação elétrica do consumidor. Porém esta forma não é prática, visto que o

consumidor terá que calcular a diferença de medição, multiplicando pela tarifa estabelecida pela ANEEL tornando-se um processo cansativo. Além disso, esse acompanhamento só pode ser feito enquanto o consumidor estiver na própria residência e não remotamente.

## **1.2. Solução Proposta**

No intuito de facilitar o monitoramento dos gastos com energia do consumidor residencial ao longo do mês, este projeto busca desenvolver um protótipo de medição de energia elétrica, utilizando o Arduino, para facilitar a visualização do consumo de energia elétrica. A concepção deste projeto tem por finalidade montar um perfil de consumo para visualização remota, coletando as informações através de sensores instalados no circuito geral do consumidor que enviam informações via internet. Para o mesmo foi utilizada uma placa ethernet para enviar as informações de consumo, para que estas pudessem ser vistas remotamente.

## **1.3. Objetivos**

Os objetivos a seguir constituem a finalidade deste projeto, assim como as metas que devem ser alcançadas.

### **1.3.1. Objetivo Geral**

Desenvolver um sistema de hardware/software para fazer a medição dos gastos com energia elétrica e disponibilizar essas informações para o consumidor de maneira prática e rápida.

### **1.3.2. Objetivos Específicos**

Os objetivos específicos são:

- Conceber, projetar e desenvolver um sistema de software/hardware para medição de energia elétrica, utilizado Arduino para leitura e processamento dos valores de consumo de energia elétrica;
- Desenvolver uma lógica para o cálculo da energia consumida, tendo como base o período praticado pela concessionária;

- Ajustar parâmetros dos sensores para melhor aproximação do valor real de medição;
- Realizar testes em laboratório para verificação de precisão dos componentes frente a marcas confiáveis do mercado;
- Instalar o sistema de software/hardware de medição de energia elétrica em uma residência (unidade consumidora), para fazer o comparativo entre o sistema proposto e o medidor da concessionária de energia elétrica.

#### **1.4. Metodologia Aplicada**

Procurou-se seguir a sequência de atividades descritas a seguir para execução desta monografia:

- Levantamento Bibliográfico para embasamento sobre cálculo de energia elétrica, sistemas com Arduino e demais temas aplicados no trabalho;
- Estruturação do modelo de software e hardware;
- Aquisição dos componentes utilizados na montagem do sistema de hardware/software;
- Montagem do sistema de hardware/software, consistindo em sensor de corrente, sensor de tensão, RTC e display LCD;
- Calibração dos sensores de tensão e corrente;
- Testes com cargas elétricas de diferentes características (RC, RL e RLC)
- Conexão com site *ThingSpeak.com* para envio dos dados obtidos na medição de consumo de energia elétrica, feito pelo sistema hardware/software;
- Aplicação do sistema de hardware/software em uma residência, para comparativo de medição frente ao medidor tradicional da concessionária.

#### **1.5. Apresentação dos capítulos**

Esta monografia está dividida em dividida em cinco capítulos, para facilitar o entendimento da mesma, sendo o primeiro capítulo uma introdução sobre o trabalho proposto. No capítulo 2 é introduzida toda a base teórica do que é energia elétrica, tensão e corrente CA, potência elétrica e a placa Arduino. No capítulo 3 é descrita de maneira aprofundada a metodologia aplicada no desenvolvimento do projeto. No capítulo 4 são expostos todos os resultados obtidos por meio dos testes e estudo de caso elaborados durante o projeto. No capítulo 5 é feita uma conclusão do trabalho

onde é apresentada a eficiência dos resultados obtidos. Há ainda os apêndices A, B e C que mostram, uma conta de energia da CEMAR, a especificação SCT-013-030 e o código final de programação do Arduino, respectivamente

## 2. FUNDAMENTAÇÃO TEÓRICA

Esta revisão tem o objetivo embasar e conceituar a metodologia utilizada na elaboração do projeto. Sendo assim, os conceitos foram ordenados de forma lógica bem como os componentes (hardware) empregados na concepção e desenvolvimento do protótipo do sistema de software/hardware de medição de energia elétrica.

### 2.1. Energia Elétrica

A energia é definida como “a capacidade de realizar trabalho e é medida em joules (J)” (ALEXANDER; SADIKU, 2013, p. 11). A energia elétrica é uma das formas de energia mais utilizadas no mundo e sua principal fonte de geração são as hidrelétricas que transformam a energia mecânica em elétrica, produzindo uma diferença de potencial elétrico que favorece a circulação de uma corrente elétrica.

Para o consumidor residencial, as formas mais comuns de transformação dessa energia são a luz, calor e movimento. Ela é medida em kWh (quilowatt-hora), e representa a quantidade de energia utilizada para alimentar uma carga pelo período de uma hora. Sua equação fica da seguinte maneira:

$$E_{el} = P \times \Delta t \quad (1)$$

Onde,

$E_{el}$  = Energia Elétrica

$P$  = Potência Ativa

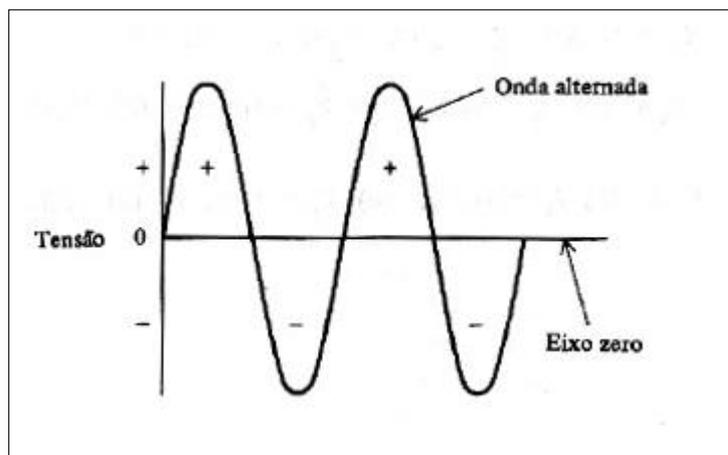
$\Delta t$  = Variação do tempo

### 2.2. Tensão e Corrente CA

Uma tensão CA (Figura 1) é aquela que sua polaridade não é constante, ou seja, sua polaridade varia em um intervalo de tempo (GUSSOW, 1997, p. 278). Existem diversas formas de tensões alternada, como: retangular, dente de serra, triangular e senoidal.

O formato entregue pela concessionária de energia elétrica é a senoidal, que possui uma frequência de 60 Hz, ou seja, varia 60 vezes por segundo.

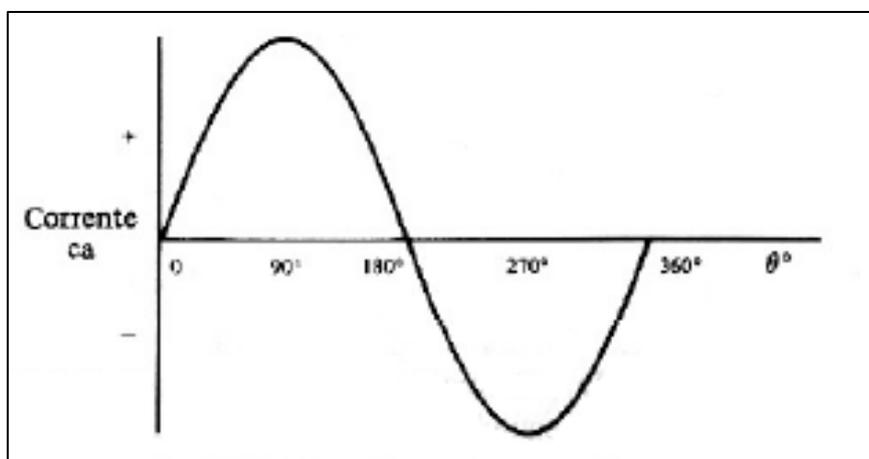
**Figura 1:** Uma forma de onda de tensão ca senoidal.



**Fonte:** GUSSOW (1997, p. 278).

O fluxo ordenado de cargas elétricas dentro de um meio (ex.: Condutor) é denominado corrente elétrica. Quando uma onda senoidal de tensão alternada é ligada aos terminais de uma resistência de carga, a corrente que passa pelo circuito também é uma onda senoidal como vemos no exemplo da Figura 2

**Figura 2:** Ciclo de corrente alternada.



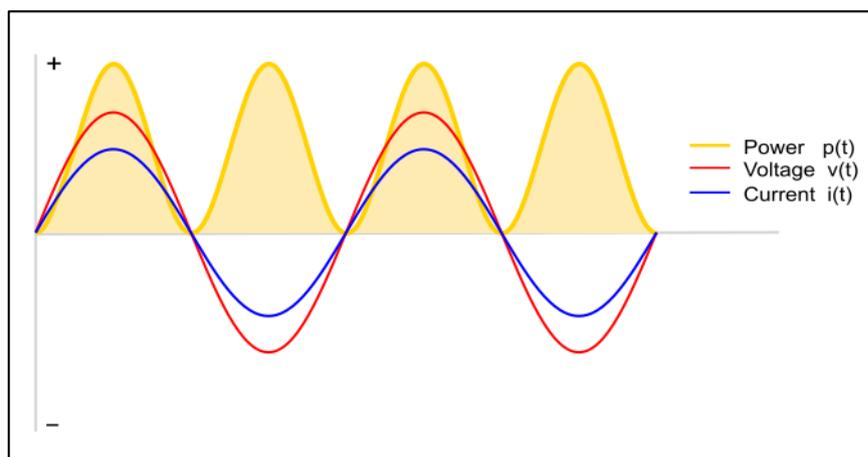
**Fonte:** GUSSOW(1997, p. 281)

### 2.2.1. Cargas resistivas e reativas

O sistema elétrico não se comporta da mesma forma para todos os tipos de carga. Quando falamos em cargas resistivas como chuveiro elétrico por exemplo, temos um comportamento da forma de onda da corrente similar ao da Figura 3, onde toda a potência flui para a carga, tendo a tensão (linha vermelha) em fase com a

corrente (linha azul). A linha amarela é a potência consumida num dado momento onde, para este caso, é resultado do produto entre a tensão e a corrente.

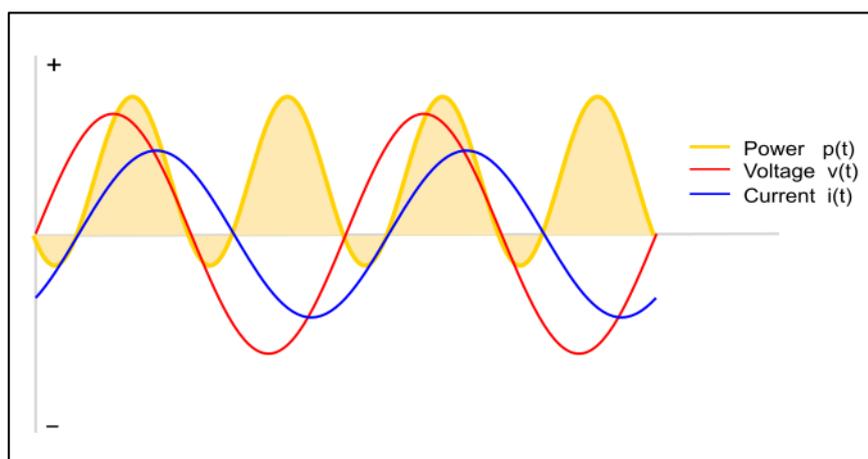
**Figura 3:** Relação de fase entre tensão e corrente para uma carga resistiva.



Fonte: [www.openenergymonitor.org](http://www.openenergymonitor.org)

Porém, segundo a EDP (2004, p. 2), a grande parte das cargas residenciais consomem energia reativa indutiva, como motores, lâmpadas de descarga, entre outros. Para essas cargas, necessitamos de um campo eletromagnético para manter seu funcionamento e essa potência utilizada para manter esse campo chamamos de potência reativa, potência essa que não transformada em trabalho e volta para a rede.

**Figura 4:** Relações de fase de tensão e corrente numa carga reativa.



Fonte: [www.openenergymonitor.org](http://www.openenergymonitor.org)

Como podemos ver na Figura 4, na linha amarela, determinado período de tempo a potência é positiva, fluindo para carga, e hora ela é negativa, fluindo para a fonte. Também vemos que a corrente está deslocada em relação a tensão, no caso

atrasada em relação à onda de tensão possuindo característica indutiva. Este deslocamento é chamado de defasagem entre a tensão e corrente e pode estar adiantada (carga capacitiva) ou atrasada (carga indutiva).

### 2.3. Potência: Ativa, Reativa e Aparente

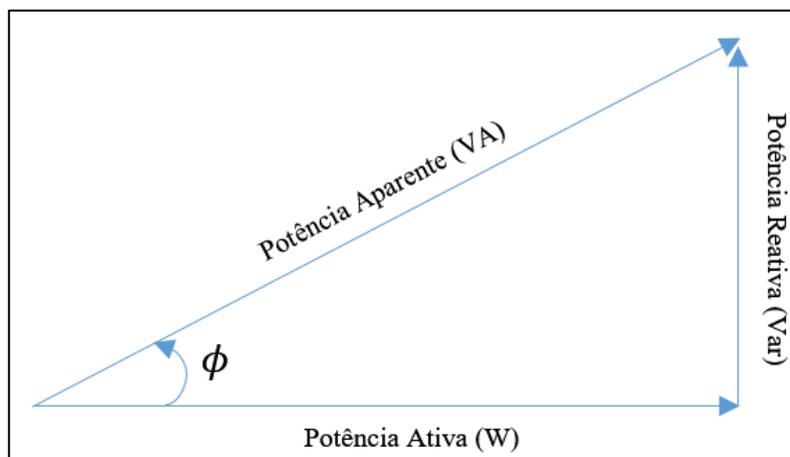
Potência é a velocidade com que se consome ou absorve energia (ALEXANDER; SADIKU, 2013). Ela representa o trabalho realizado por unidade de tempo, mas como vimos na Figura 4, temos a potência que é consumida pela carga, transformada em trabalho, e a potência que retorna para a fonte.

Potência ativa ( $P$ ), segundo Nilsson e Riedel (2009, p. 274), é a potência que efetivamente é convertida de uma forma elétrica para uma não elétrica. Ou seja, é a potência que realmente é transformada em trabalho. Ela é medida em watts (W) ou em quilowatt (kW).

Em circuitos onde precisamos criar ou manter um campo eletromagnético, como motores, reatores, lâmpadas fluorescentes, possuímos a potência reativa ( $Q$ ) que é medida em volt-ampere reativo (Var) ou quilovolt-ampere reativo (kVAr). Ela está presente em circuitos indutivos e capacitivos. Esta potência não é efetivamente transformada em trabalho e apenas circula entre a carga e a fonte.

Potência aparente ( $S$ ) é soma vetorial das potências ativa e reativa, ou seja, é a potência total absorvida pela instalação. Ela é medida em volt-ampere (VA) ou quilovolt-ampere (kVA) e é utilizada para o dimensionamento de condutores, transformadores, proteções, entre outros.

**Figura 5:** Triângulo de potências.



**Fonte:** Autor

A relação entre as potências pode ser demonstrada através da Figura 5, onde deduzimos as equações a seguir.

$$S = VI \quad (2)$$

$$P = S \times \cos \phi \quad (3)$$

$$Q = S \times \sin \phi \quad (4)$$

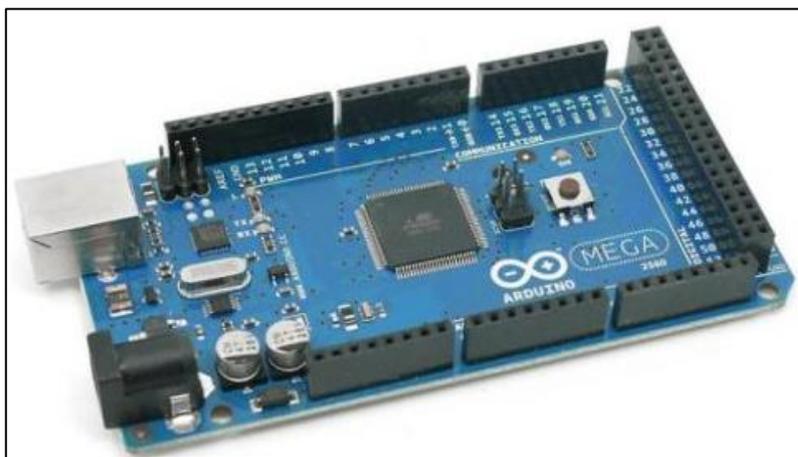
$$FP = \frac{P}{S} = \cos \phi \quad (5)$$

FP é o fator de potência e segundo GUSSOW (1997, p. 388), indica quanto da potência aparente é potência ativa e pode variar desde 1, quando o ângulo  $\phi$  de fase é  $0^\circ$ , até 0, quando  $\phi$  for  $90^\circ$ .

## 2.4. Arduino

O Arduino é basicamente um pequeno computador que você pode programar e controlar as entradas e saídas, ou seja, ele é uma plataforma móvel de computação física ou embarcada (MCROBERTS, 2011).

**Figura 6:** Arduino MEGA 2560.



**Fonte:** [www.embarcados.com.br](http://www.embarcados.com.br)

Em termos práticos, o Arduino é uma plataforma eletrônica de código aberto, composto de hardware (placa controladora) e software (ambiente de desenvolvimento) de fácil utilização (LEMOS, 2014). Com ela você pode ler entradas (luz em um sensor) e transformá-la em saída (relé) com um simples conjunto de instruções escrito em C.

A placa utilizada neste trabalho foi a MEGA 2560 (Figura 6). Esta placa possui o microcontrolador ATmega2560 que tem 256 KB de memória flash, 8KB de SRAM e 4KB de EEPROM. A mesma conta com 54 I/O digitais (com 14 que podem ser saída PWM) e 16 analógicos onde podemos utilizar diversos sensores, relés, botões, led's, entre outros. Suas principais características podem ser vistas na Tabela 1.

**Tabela 1:** Especificação técnica MEGA 2560.

| <b>Item</b>            | <b>Especificação</b>                       |
|------------------------|--|
| Microcontrolador       | ATmega2560                                 |
| Tensão de Operação     | 5 V  |
| Tensão de entrada      | 7-12 V                                     |
| Pinos Digitais I/O     | 54 (15 com saída PWM)                      |
| Pinos Analógicos       | 16   |
| Memória Flash          | 256 KB, com 8 KB utilizado pelo bootloader |
| SRAM                   | 8 KB                                       |
| EEPROM                 | 4 KB                                       |
| Clock Speed            | 16 MHz                                     |
| Corrente DC pino I/O   | 20 mA                                      |
| Corrente DC pino 3.3 V | 50 mA                                      |
| Comprimento            | 101,52 mm                                  |
| Largura                | 53,3 mm                                    |
| Peso                   | 37 g                                       |

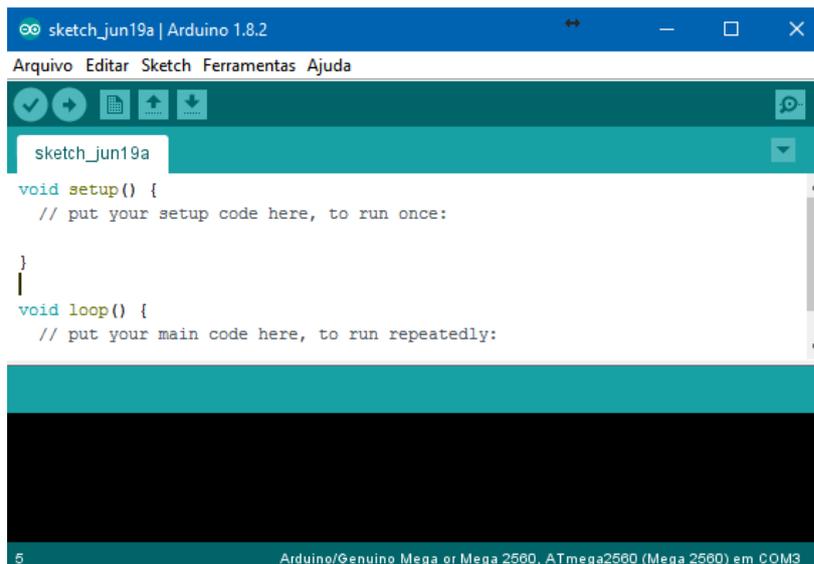
**Fonte:** Modificado de [www.arduino.cc](http://www.arduino.cc)

#### 2.4.1. IDE Arduino

O IDE (Integrated Development Environment) do Arduino (Figura 7) é uma aplicação em Java com o objetivo de facilitar o desenvolvimento da programação sem a necessidade de rodar programas em ambientes de linhas de comando, facilitando assim para pessoas leigas e sem conhecimento aprofundado em desenvolvimento de software.

A linguagem utilizada é baseada em C e possui recursos de realce de sintaxe, parênteses correspondentes e indentação automática.

**Figura 7:** Tela inicial do IDE do Arduino.



**Fonte:** Autor

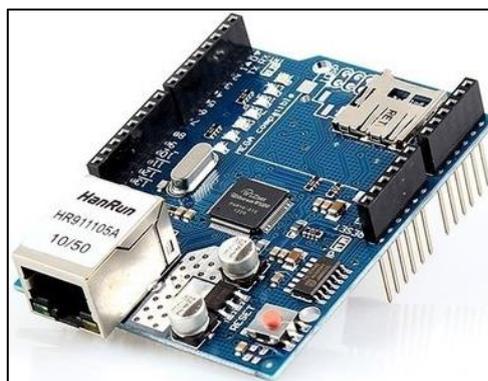
## 2.4.2. Ethernet Shield

Quando falamos em shields para Arduino, estamos falando em placas de expansão de hardware que ampliam as funções do Arduino. Como alguns exemplos podemos citar os shields para controle de motores, comunicação bluetooth e no caso deste projeto para comunicação ethernet.

Ao conectar o Arduino a uma rede por meio de cabo ethernet, podemos ler informações a partir de outro ponto da mesma rede ou enviar dados para sites (MCROBERTS, 2011).

Esta placa (Figura 8), utiliza um chip WIZnet ethernet W5100 e fornece acesso à rede (IP) utilizando os protocolos TCP e UDP.

**Figura 8:** Ethernet shield W5100.



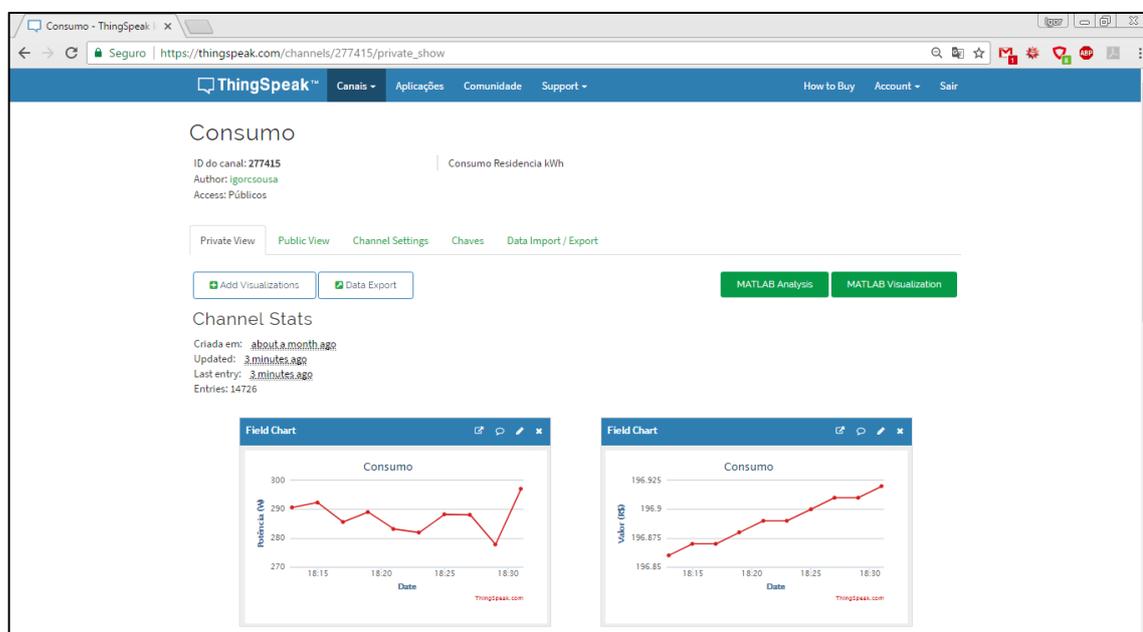
**Fonte:** <http://www.filipeflop.com>

### 3.4.2.1 ThingSpeak

Segundo seus criadores, o ThingSpeak é uma plataforma de IoT (internet of things) ou, internet das coisas em português, que permite agregar, visualizar e analisar fluxo de dados ao vivo na nuvem. Ele é um serviço gratuito que permite comunicação através de um protocolo baseado em HTTP.

Para enviar dados para o servidor do ThingSpeak, deve-se criar uma conta no seu site, que é mostrado na Figura 9, e criar um canal para receber as informações dos sensores de humidade, por exemplo. Os canais criados são personalizáveis e podem armazenar até 8 campos com informações.

**Figura 9:** Canal monitorado ThingSpeak.



**Fonte:** Autor

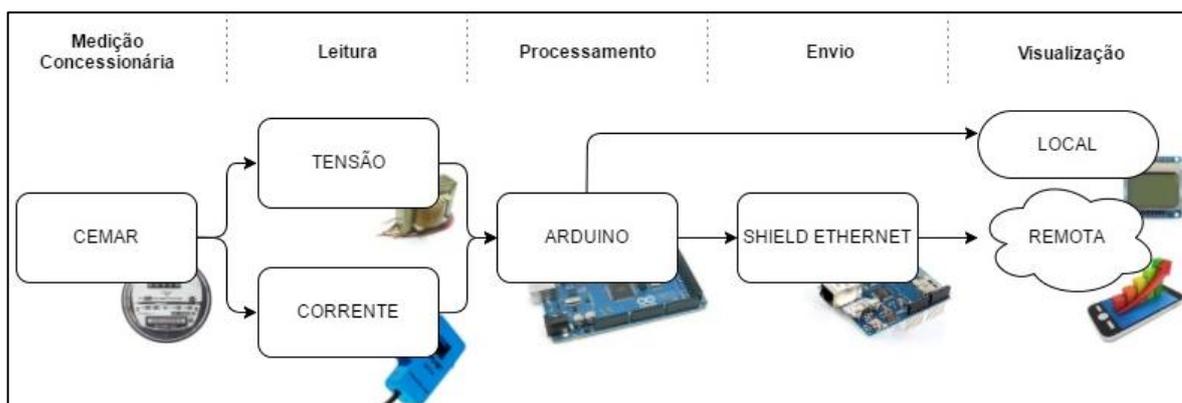
### 3. METODOLOGIA

Neste capítulo será apresentado o modelo proposto junto aos componentes utilizados na montagem. Também será apresentado os testes necessários para a calibração e precisão medidor.

#### 3.1. Modelo Proposto

O modelo proposto para o sistema de software/hardware de medição de energia elétrica residencial pode ser visto na Figura 10. Após o medidor convencional da concessionária, os componentes de leitura de tensão e de corrente foram instalados. O Arduino recebe as informações enviadas pelos sensores, processa e disponibiliza localmente através de um display e remotamente via internet.

**Figura 10:** Modelo proposto.



**Fonte:** Autor

#### 3.2. Componentes do sistema

A Tabela 2 mostra os componentes que foram necessários para a montagem do medidor, com o custo de aquisição. Com exceção dos capacitores e resistores, todos os outros componentes foram adquiridos via MercadoLivre.

**Tabela 2:** Componentes do projeto

| Componente               | Qtd | Valor      |
|--------------------------|-----|------------|
| Arduino Mega 2560 R3     | 1   | R\$ 44,00  |
| Ethernet Shield W5100    | 1   | R\$ 49,98  |
| Sensor Corrente Ac - 30a | 1   | R\$ 40,00  |
| Transformador 220V – 9V  | 1   | R\$ 11,00  |
| Real Time Clock DS1307   | 1   | R\$ 10,00  |
| Protoboard 830 pts       | 1   | R\$ 23,00  |
| Jumpers                  | 60  | R\$ 16,98  |
| Display Nokia 5110       | 1   | R\$ 18,00  |
| Capacitor 10 $\mu$ F     | 2   | R\$ 0,20   |
| Resistores               | 6   | R\$ 0,60   |
| Total                    |     | R\$ 213,76 |

Fonte: Autor

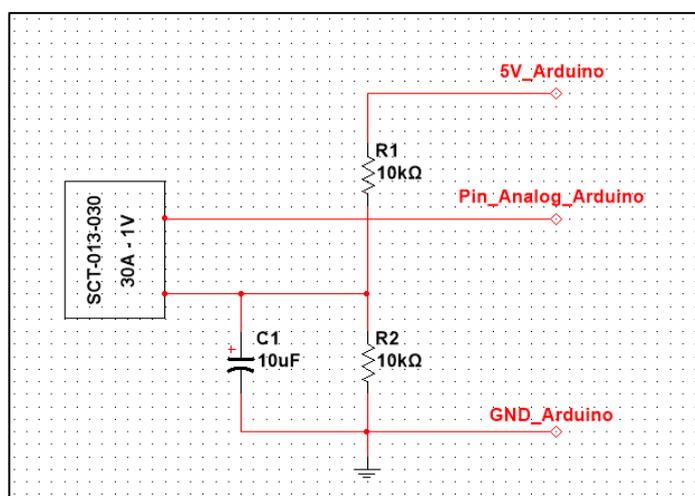
### 3.3. Montagem

O processo de montagem consistiu em três etapas, sendo a primeira para a medição de corrente onde foram utilizados os seguintes componentes:

- 1 x sensor SCT-013-030
- 2 x resistores de 10k $\Omega$
- 1 x capacitor 10uF

Tais componentes foram montados seguindo o esquema da Figura 11, permitindo a conexão do sensor com o Arduino.

**Figura 11:** Esquema de montagem SCT-013-030.



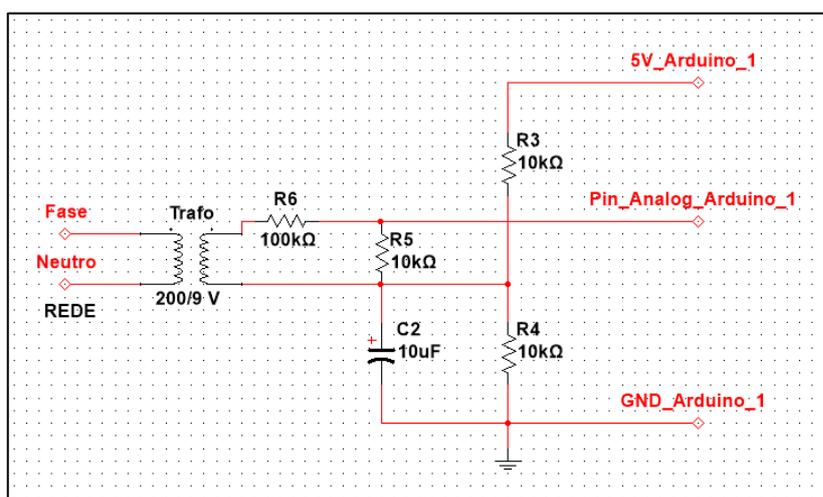
Fonte: Autor

A segunda etapa consistiu na montagem do transformador para medição de tensão da rede, onde foram necessários os seguintes componentes:

- 1 x transformador AC-AC 220/9V
- 3 x resistores de 10k $\Omega$
- 1 x resistores de 100k $\Omega$
- 1 x capacitor 10uF

Os mesmos seguiram o esquema de montagem da Figura 12, onde podemos ver as ligações entre o transformador e o Arduino.

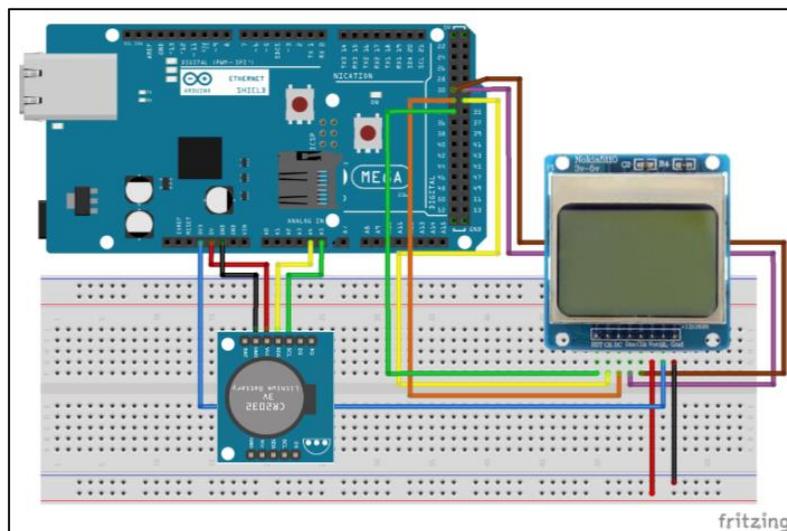
**Figura 12:** Esquema de ligação do Trafo 220/9V.



**Fonte:** Autor

A terceira e última etapa, consistiu na montagem dos componentes auxiliares como o display LCD e o RTC DS1307 que foram conectados ao Arduino conforme a Figura 13. O RTC auxilia na marcação do mês e dias de forma precisa, pois o mesmo contém as quantidades de dias de cada mês e ainda considera ano bissexto.

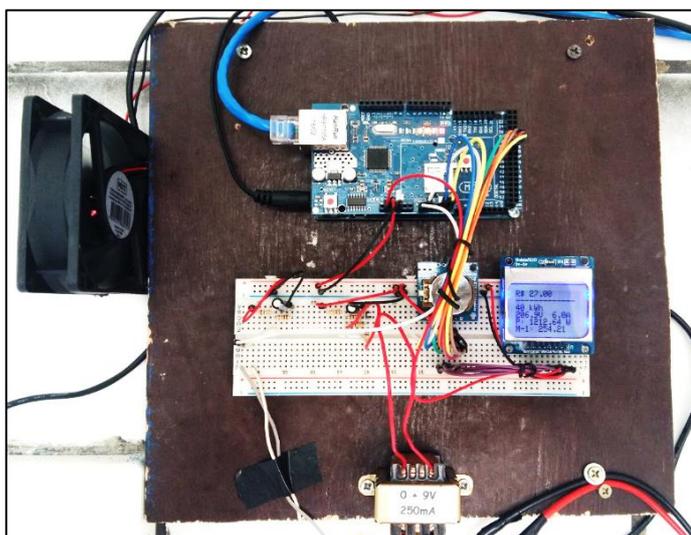
**Figura 13:** Esquema de ligação: RTC e LCD 5110.



**Fonte:** Autor

Todas essas ligações foram feitas na mesma protoboard e fixados todos os componentes num compensado 25x25cm com uma ventoinha para a ventilação dos mesmos. A organização final pode ser vista na Figura 14 a seguir:

**Figura 14:** Montagem Final.



**Fonte:** Autor

### 3.4. Calibração

Após a montagem, foi necessário a calibração dos sensores pois estes são influenciados por vários fatores que podem levar sua medição a um valor incorreto.

Caso não sejam feitos os procedimentos, os valores de potência ativa podem ter um erro de precisão de quase 40% (OPENENERGYMONITOR, 2017).

No caso do sensor de corrente, dois parâmetros foram utilizados. O primeiro foi o pino analógico do Arduino onde o sensor estava conectado e o segundo o valor de calibração que foi calculado de acordo com a Equação 6.

$$\text{Calibração} = \frac{\text{N}^\circ \text{ de espiras do sensor}}{\text{Resistor de carga}} = \frac{1800}{62} \approx 29 \quad (6)$$

Segundo o datasheet no Apêndice B, o TC possui 1800 espiras e um resistor de carga de 62  $\Omega$ , logo temos um valor de calibração 29. Este valor foi inserido comando: `emon1.current(pin_corrente, 29)`.

Para o circuito do transformador de tensão, três parâmetros foram utilizados. O primeiro foi pino de conexão no Arduino, o segundo a calibração de tensão, onde o procedimento foi medir a tensão da rede com um amperímetro confiável e ajustar o valor de calibração até o Arduino retornar um valor aproximado ao do amperímetro e terceiro e último, foi utilizado o valor de 1,7 para o deslocamento de fase. Estas instruções representadas na Figura 15, são encontradas na função SETUP no Apêndice C.

**Figura 15:** Parâmetros de calibração.

```
void setup()
{
  Serial.begin(9600);

  rtc.halt(false); //Aciona o relógio

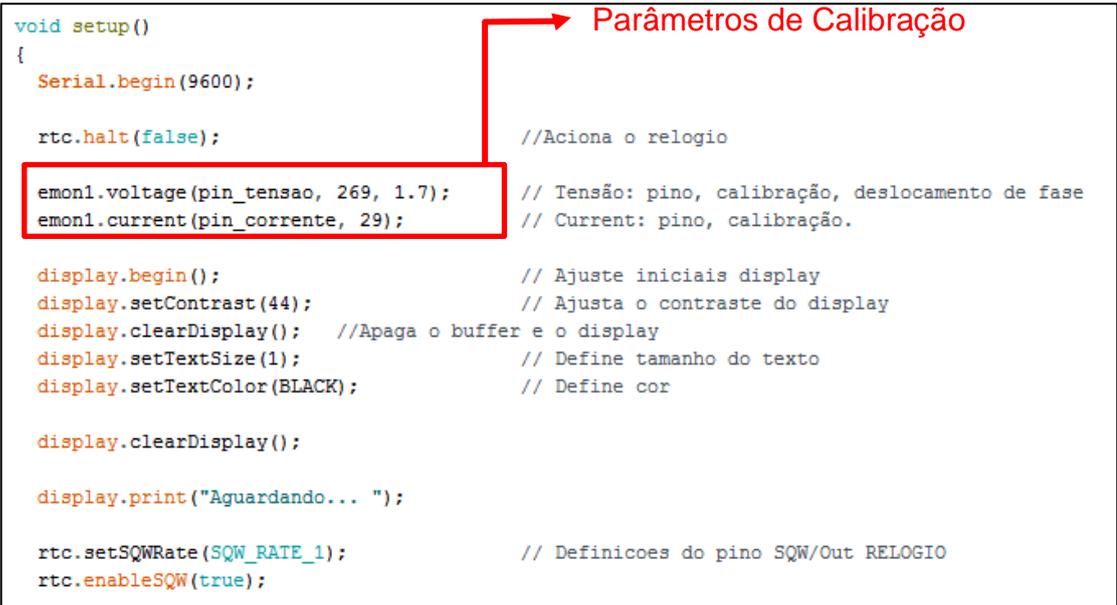
  emon1.voltage(pin_tensao, 269, 1.7); // Tensão: pino, calibração, deslocamento de fase
  emon1.current(pin_corrente, 29); // Current: pino, calibração.

  display.begin(); // Ajuste iniciais display
  display.setContrast(44); // Ajusta o contraste do display
  display.clearDisplay(); //Apaga o buffer e o display
  display.setTextSize(1); // Define tamanho do texto
  display.setTextColor(BLACK); // Define cor

  display.clearDisplay();

  display.print("Aguardando... ");

  rtc.setSQWRate(SQW_RATE_1); // Definições do pino SQW/Out RELOGIO
  rtc.enableSQW(true);
}
```



Fonte: Autor

Os comandos de calibração, `emon1.current` e `emon1.voltage`, pertencem a biblioteca `emonlib.h`. Esta biblioteca foi criada pelo projeto `openenergymonitor` que tem como objetivo desenvolver ferramentas open-source que nos ajudam a entender o consumo de energia. Fazendo uso dessa biblioteca, simplificou-se os cálculos de consumo de energia elétrica, tornando o programa mais simples.

### 3.5. Conexão com o ThingSpeak

Para enviar dados a nuvem do Thingspeak, foi necessário criar um canal para receber e exibir as informações de consumo da residência. Para isso, um canal com 5 campos foi criado: Potência (W), Valor\_Conta (R\$), Consumo (kWh), Tensão (V) e Corrente (A). Após a criação, este canal gera uma chave de escrita, vista na Figura 16, que é enviada pelo código do Arduino para que o ThingSpeak “entenda” para onde direcionar os dados enviados.

**Figura 16:** Exemplo de chave de escrita.

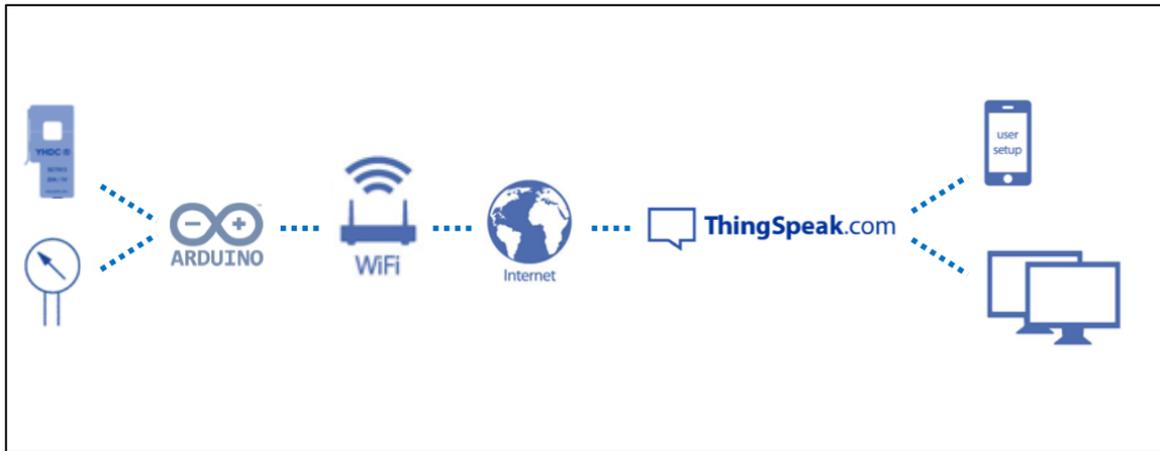


**Fonte:** Autor

### 3.6. Processamento de dados

Após o processamento dos dados obtidos dos sensores e calculado o consumo e kWh, foi necessário disponibilizar essas informações de consumo ao consumidor. Uma solução com custo zero ao usuário final, foi enviar esses dados via placa ethernet ao servidor do ThingSpeak seguindo a estrutura da Figura 17.

**Figura 17:** Estrutura de comunicação.



**Fonte:** Autor

O código da Figura 18 mostra como foram enviadas as informações ao servidor. No código devemos passar algumas informações junto ao dado a ser salvo, como a chave de escrita e o campo que a mesma deva ser salva. Para isso, foi feita uma requisição HTTP ao servidor do ThingSpeak, enviando uma string com as informações de requisição através da porta 80.

**Figura 18:** Código de envio para o servidor ThingSpeak.

```
void updateThingSpeak(String tsData)
{
  if (client.connect(thingSpeakAddress, 80))
  {
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(tsData.length());
    client.print("\n\n");

    client.print(tsData);
  }
}
```

**Fonte:** Autor

Com isso, as informações de consumo passaram a estar disponíveis *online*, podendo ser acessadas via web browser ou aplicativo para celular. No caso do aplicativo do celular, o *IoT ThingSpeak Monitor Widget* foi utilizado (veja Figura 19) por possuir dois recursos muito úteis, como: widget para área de trabalho e o recurso

de plotar os resultados num gráfico, podendo ser configurado para período de dia, semana, mês e customizado.

**Figura 19:** IoT ThingSpeak Monitor Widget.



Fonte: [www.googleplay.com](http://www.googleplay.com)

### 3.7. Testes

Com o auxílio do Prof. Dr. José Gomes de Matos e do Prof. Me. Antonio Dantas Maniçoba, foi montado no laboratório de máquinas elétricas da UFMA um ambiente de testes para o protótipo do medidor.

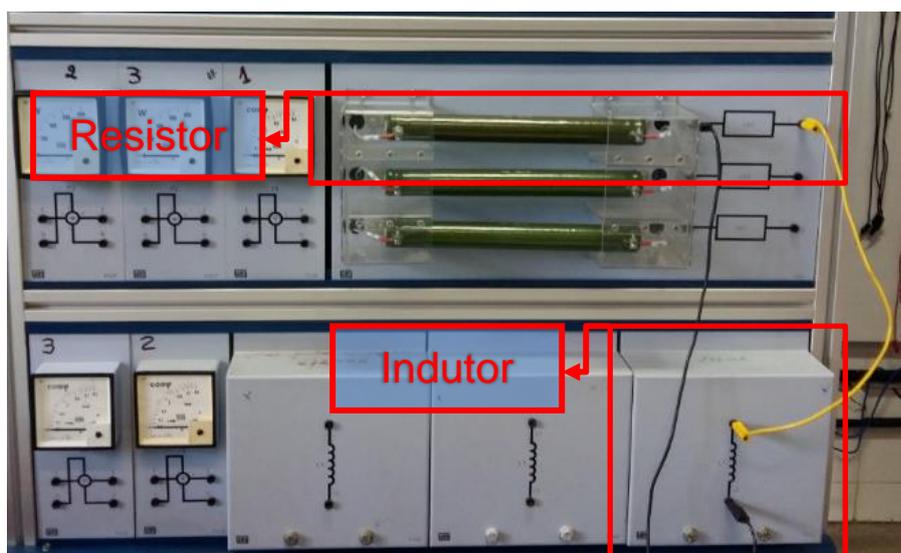
Os testes consistiram em utilizar cargas resistivas, indutivas e capacitivas para estudar a precisão do protótipo. Os materiais utilizados foram:

- Bancada WEG: Bancada composta de Resistores, Indutores e Capacitores;
- Lâmpadas halógenas: Lâmpadas tipo palito de 1000 e 300 watts;
- Fonte de alimentação variável Stelltrentrafo 0 - 250V;
- Medidor de qualidade de energia Fluke 345;
- Medidor monofásico eletrônico Landys+Gyr E22A.

### 3.7.1. Carga Indutiva

Para este teste foram utilizados o resistor e o indutor (Figura 20) da bancada WEG, que simula nestes caso uma carga com característica indutiva, mostrados na Figura 20:

**Figura 20:** Carga RL.



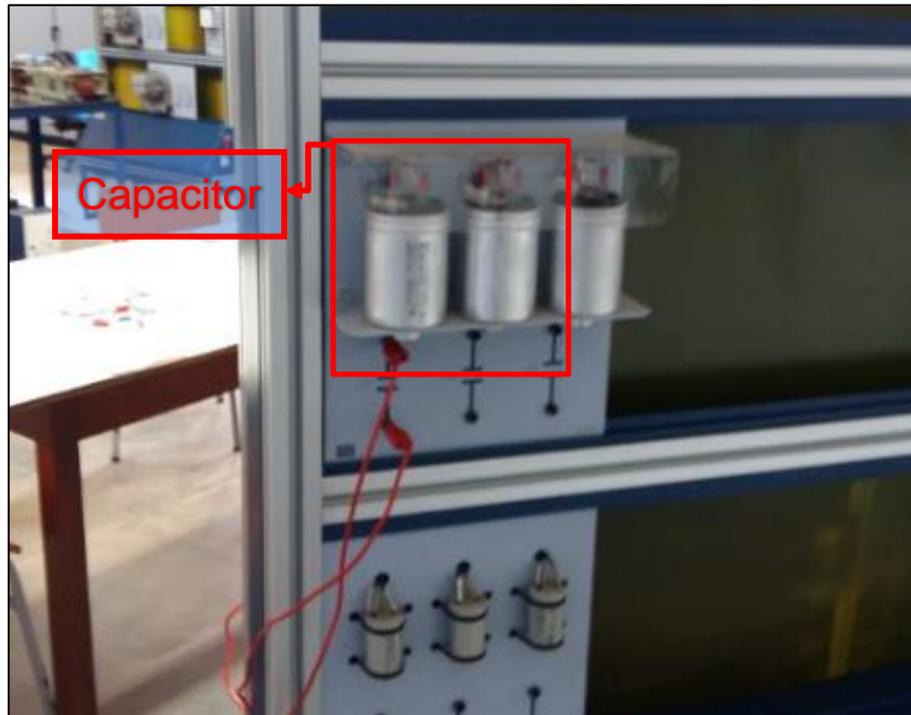
**Fonte:** Autor

O teste foi realizado, fazendo uma ligação RL (ligação entre resistor e indutor em série) e alimentando o circuito com uma tensão de aproximadamente 162 volts. O medidor Fluke 345 foi instalado na saída da fonte de alimentação.

### 3.7.2. Carga Capacitiva

Neste teste, o mesmo resistor foi utilizado, trocando apenas o indutor pelo capacitor (Figura 21). Nesta ligação temos um circuito RC (ligação entre resistor e capacitor em série).

**Figura 21:** Capacitor para ligação RC.



Fonte: Autor

### 3.7.3. Carga Indutiva – Capacitiva

Como forma de simular o circuito mais próximo do encontrado nas residências, uma ligação RLC foi feita. Esta ligação consiste em um resistor, capacitor e indutor ligados em série.

**Figura 22:** Ligação RLC (visão 360° da bancada).



Fonte: Autor

### 3.7.4. Comparativo com monofásico eletrônico Landys+Gyr E22A

O medidor monofásico eletrônico Landys+Gyr E22A foi utilizado para marcar o consumo de energia de 4 lâmpadas halógenas, sendo 3 lâmpadas de 300 W e uma de 1000W (Figura 23). O teste teve como objetivo marcar o consumo de 1kWh, comparando o medidor E22A com o marcado pelo sistema de software/hardware de medição de energia elétrica proposto.

No preparativo do teste foi observado que o medidor E22A estava marcando 3 kWh e o mesmo não pode ser zerado. Para resolver este problema, as cargas foram ligadas e aguardou-se o medidor avançar para 4 kWh. No momento que o medidor E22A avançou para 4kWh, ligou-se o sistema proposto, iniciando assim a medição de consumo de energia elétrica.

No início do teste, uma das lâmpadas de 300 W apagou, ficando apenas 3 lâmpadas consumindo energia elétrica.

**Figura 23:** Teste de consumo de energia elétrica.



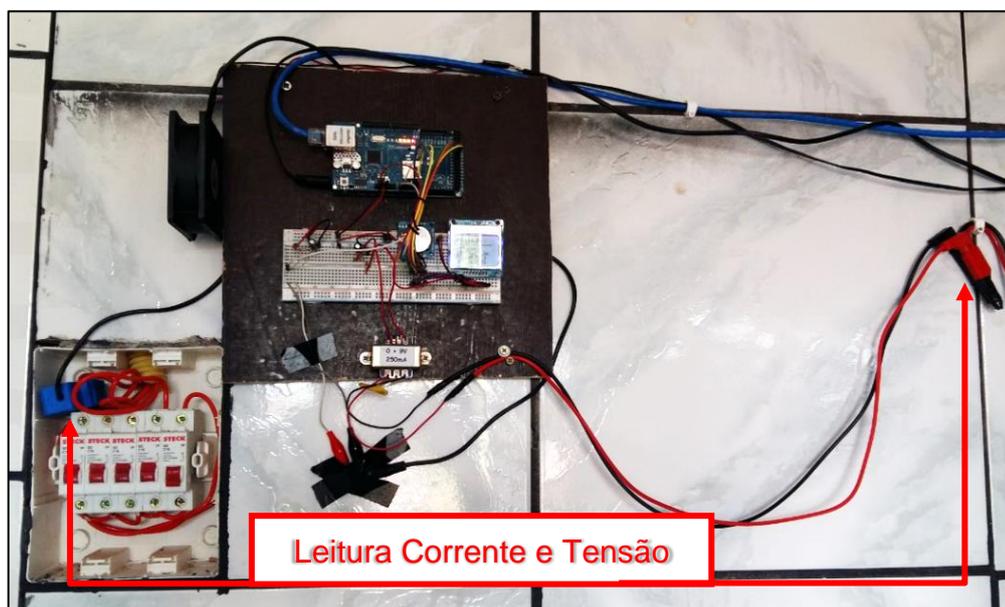
**Fonte:** Autor

### 3.8. Estudo de caso

Para comprovar a eficiência do sistema de software/hardware de medição de energia elétrica foi elaborado um estudo de caso numa residência com característica de gastos comum, sendo suas maiores cargas um ar-condicionado de 9000 BTU e um micro-ondas de 20 L. O sistema de software/hardware de medição de

energia elétrica foi instalado na entrada do quadro de distribuição da residência (veja Figura 24) com o objetivo de medir o consumo geral da mesma.

**Figura 24:** Medidor na entrada do quadro de distribuição.



**Fonte:** Autor

O período de medição foi iniciado no dia 9 de julho de 2017, mesmo dia que foi feita a leitura pela CEMAR. Na fatura impressa pela concessionária (Apêndice - A), o medidor apontava 13 901 kWh, e esse foi o valor tomado com referência de início da medição bastando apenas subtrair o novo de consumo pelo de referência para obter o consumo do mês (julho).

## 4. RESULTADOS E ANÁLISE

Neste capítulo serão apresentados os resultados dos testes e estudo de caso com suas respectivas análises.

### 4.1. Testes com as cargas

Para as cargas da bancada WEG, dos itens 4.7.1 a 4.7.3, que simularam os diversos tipos de cargas que podemos ter nas residências, tivemos os resultados mostrados na Figura 25, Figura 26 e Figura 27.

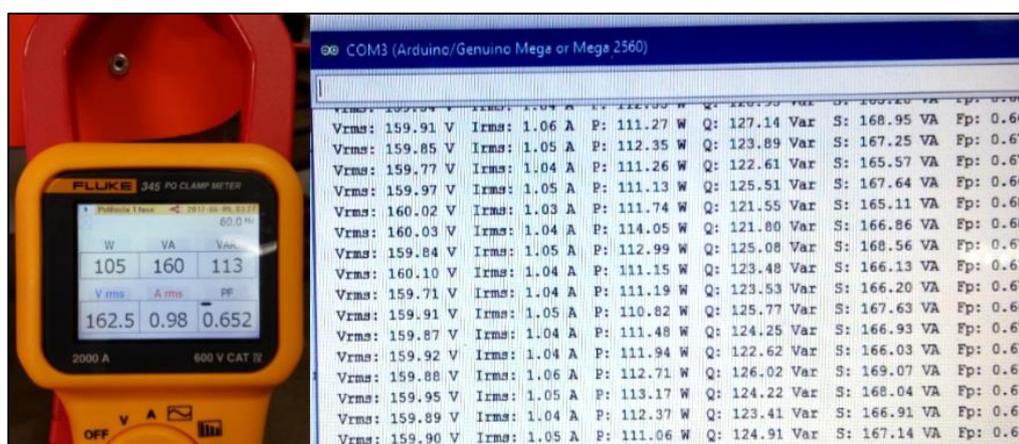
Com os valores das cargas gerados pelo sistema de software/hardware de medição de energia elétrica, foram calculadas as médias para as principais grandezas envolvidas no cálculo de energia elétrica. Após isso, os valores foram comparados com o de referência (medidor Fluke 345) e calculado o erro na medição como pode ser visto na Tabela 3.

**Tabela 3:** Comparativo Arduino x Fluke 345

|                 | Circuito RL |         |        | Circuito RC |         |        | Circuito RLC |         |        |
|-----------------|-------------|---------|--------|-------------|---------|--------|--------------|---------|--------|
|                 | Fluke 345   | Arduino | % erro | Fluke 345   | Arduino | % erro | Fluke 345    | Arduino | % erro |
| <b>Vrms (V)</b> | 162,50      | 159,89  | 1,61%  | 161,90      | 158,42  | 2,15%  | 162,10       | 158,97  | 1,93%  |
| <b>Irms (A)</b> | 0,98        | 1,05    | 6,68%  | 1,27        | 1,28    | 0,82%  | 1,45         | 1,47    | 1,63%  |
| <b>P (W)</b>    | 105,00      | 112,06  | 6,72%  | 158,00      | 157,32  | 0,43%  | 221,00       | 222,17  | 0,53%  |
| <b>Fp</b>       | 0,65        | 0,67    | 2,83%  | 0,76        | 0,78    | 1,77%  | 0,94         | 0,95    | 0,69%  |

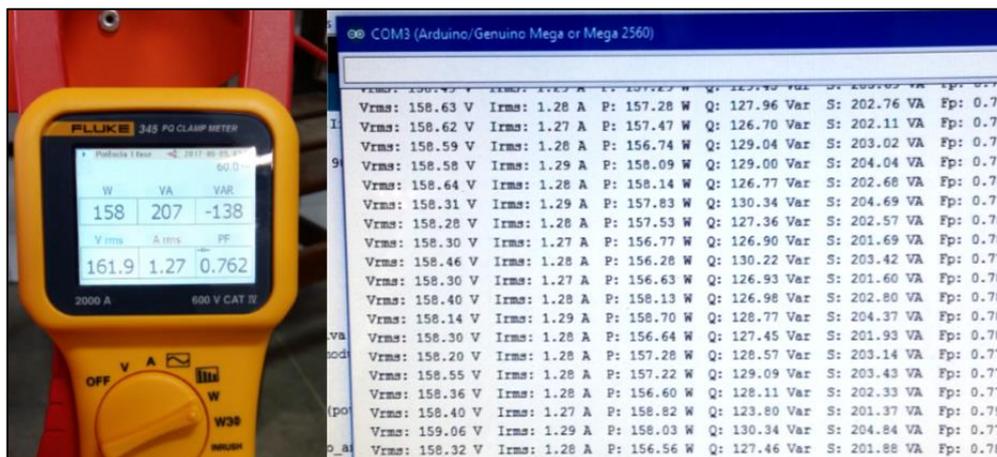
Fonte: Autor

**Figura 25:** Resultados RL



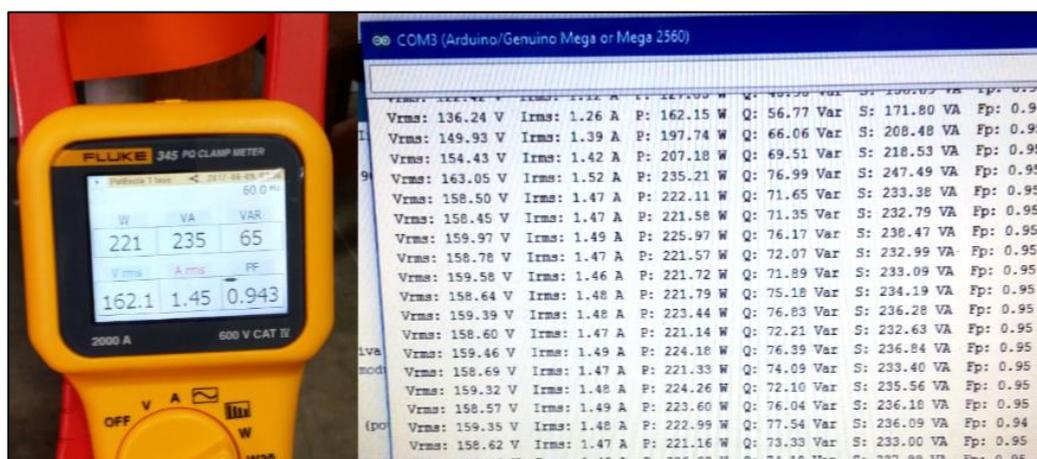
Fonte: Autor

Figura 26: Resultados RC



Fonte: Autor

Figura 27: Resultados RLC



Fonte: Autor

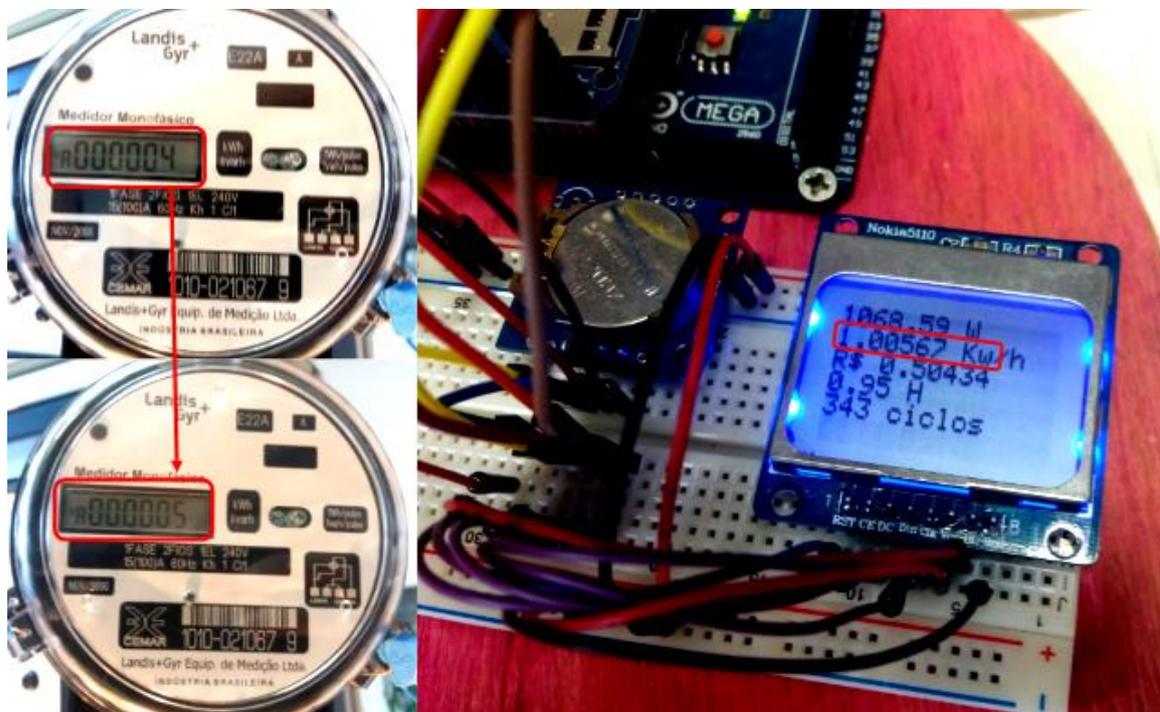
## 4.2. Landys+Gyr E22A x Arduino

Seguindo o teste descrito no item 4.7.4, foi testada a medição do sistema de software/hardware de medição de energia elétrica frente a do medidor monofásico eletrônico Landys+Gyr E22A. Durante o mesmo ocorreu outro problema diferente do relatado no item 4.6.4, que foi a atuação da proteção de sobrecorrente da fonte durante os primeiros minutos do teste. Este problema foi contornado reduzindo o valor de tensão fornecido pela fonte, que antes era de 220 V, para 181 V para não ultrapassar a corrente máxima de 6 A da fonte.

Estes problemas ocorridos não afetaram negativamente no resultado, pois os mesmos simularam tanto um “apagar de lâmpada” quanto uma queda de energia fornecida pela rede.

Na Figura 28, podemos ver o momento em que o medidor E22A passou de 4kWh para 5kWh (consumo de 1kWh), junto com o Arduino que também marcou o consumo de aproximadamente 1kWh.

**Figura 28:** Comparativo medidor E22A x Arduino



Fonte: Autor

### 4.3. Estudo de caso

Durante o estudo de caso descrito no item 5.3, não foi necessário aguardar a medição seguinte da concessionária. No caso, como o medidor da concessionária é eletrônico, podemos facilmente verificar o valor parcial de consumo ao longo do mês. Assim, foi verificado o valor mostrado pelo medidor da concessionária no dia 28 de junho de 2017, contabilizando 19 dias de medição, e observado o valor de 14 195 kWh de consumo parcial (Figura 29).

Com o valor parcial de consumo em mãos, bastou-se subtrair-lo do valor de referência do início do período de medição (13 901 kWh), obtendo-se o consumo de 294 kWh.

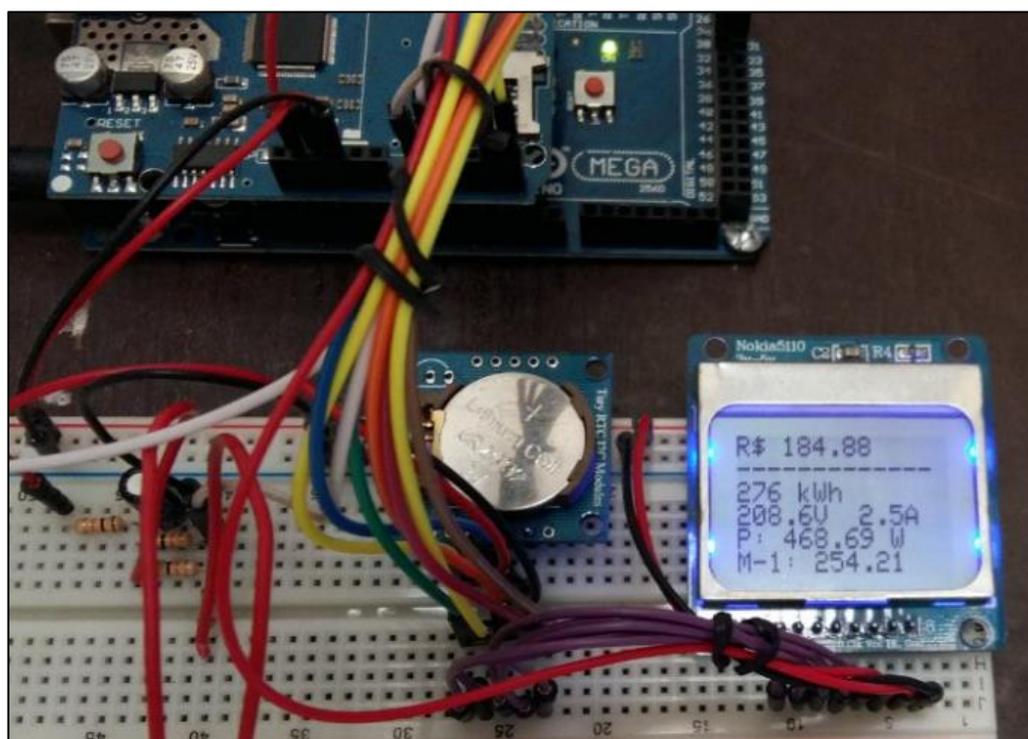
**Figura 29:** Consumo parcial (CEMAR)



**Fonte:** Autor

Como a Figura 30 mostra, a medição parcial do Arduino estava em 276 kWh no dia 28 de junho de 2017. Calculando o erro relativo dele em relação a medição da CEMAR, temos o valor de aproximadamente 6,12 %.

**Figura 30:** Consumo parcial (Arduino)

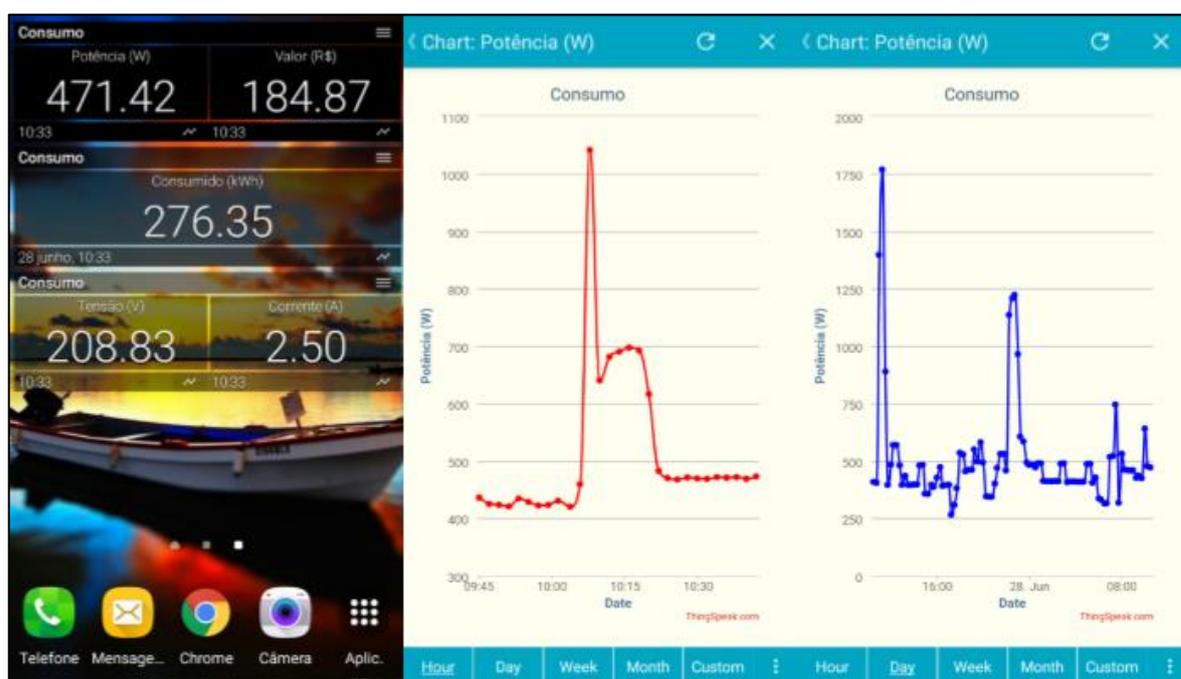


**Fonte:** Autor

No aplicativo IoT ThingSpeak Monitor Widget, instalado no smartphone, pode-se verificar as informações de consumo de maneira mais detalhada como mostra a Figura 31. Na tela inicial do smartphone pode ser visto os principais valores de consumo da residência, como: potência (W) que é o valor consumido naquele instante, valor (R\$) que representa o valor aproximado da fatura naquele momento, consumo (kWh) que é o valor de energia consumido, tensão (V) e corrente (A).

Para mais detalhes, bastou-se clicar no valor mostrado na tela do smartphone para exibição do gráfico de consumo num período estabelecido.

**Figura 31:** Widget, Potência (W) 1 hora e 24 horas.



**Fonte:** Autor

## 5. CONCLUSÃO

Durante o desenvolvimento do estudo, ocorreram alguns imprevistos que dificultaram a realização dos testes no medidor proposto, porém os mesmos não interferiram no resultado principal do trabalho. Como visto nos resultados, os testes indicaram que o medidor proposto é capaz de realizar medições próximas ao um medidor convencional, com taxa de erro menor que 7%. Assim sendo, o Arduino se mostrou eficaz para o objetivo a ele atribuído, comprovando mais uma entre tantas atividades que ele pode desenvolver.

O processamento de dados também foi feito de maneira eficaz durante o estudo, possibilitando armazenar os dados da medição em nuvem e disponibiliza-los via internet, seja por aplicativo para smartphone ou computador, o que facilita a consulta dos mesmos pelo consumidor.

Por fim, o trabalho realizado atingiu seus objetivos, pois o monitoramento do custo e consumo de energia foi feito de forma prática e instantânea, assim o consumidor será capaz de conhecer antes seus gastos com energia e tentar economiza-la, tendo em vista que os custos com energia equivalem a grande parte do orçamento mensal dos consumidores. Porém, estudos relacionados ao desenvolvimento de *software* e melhoramento do *hardware* ainda se faz necessário, levando em conta que a taxa de erro e os custos podem ser reduzidos, através de montagem e materiais mais apropriados.

## REFERÊNCIAS

Agência Nacional de Energia Elétrica - ANEEL. **Abril tem bandeira vermelha em seu menor patamar.** Disponível em <<http://www.aneel.gov.br/>>. Acesso em 31 março 2017.

Agência Nacional de Energia Elétrica - ANEEL. **Ranking das Tarifas.** Disponível em <<http://www.aneel.gov.br/ranking-das-tarifas>>. Acesso em: 01 abril 2017.

ALEXANDER, C. K.; SADIKU, M. N. O. **Fundamentos de Circuitos Elétricos.** 2ª ed. Porto Alegre: Editora Bookman, 2003.

**Arduino.** Arduino Mega. Disponível em <<https://www.arduino.cc/en/Main/arduinoBoardMega>>. Acesso em: 03 abril 2017.

**Arduino.** Arduino Mega. Disponível em <<https://www.arduino.cc/en/Main/arduinoBoardMega2560>>. Acesso em: 10 junho 2017.

BRASIL. ANEEL. **Resolução Normativa nº 414, de 9 de setembro de 2010.** Estabelece as Condições Gerais de Fornecimento de Energia Elétrica de forma atualizada e consolidada. Disponível em <<http://www2.aneel.gov.br/cedoc/ren2010414.pdf>>. Acesso em 22 abril 2017.

Companhia Energética do Maranhão - CEMAR. **Cobrança de tarifas.** Disponível em <<http://www.cemar116.com.br/residencial/informacoes/cobranca-de-tarifas>>. Acesso em: 01 abril 2017.

**EDP.** Manual de orientação aos consumidores. Disponível em <<http://www.edp.com.br/distribuicao/edp-bandeirante/informacoes>>. Acesso em 05 abril 2017.

**Engelétrica.** Manual de Correção do Fator de Potência. Disponível em <<http://www.engeletrica.com.br/fatordepotencia-manual-fatordepotencia.html>>. Acesso em 10 junho 2017.

**Filipeflop.** Ethernet Shield W5100 para Arduino. Disponível em <<http://www.filipeflop.com/pd-6b62c-ethernet-shield-w5100-para-arduino.html>> Acesso em 14 junho 2017.

Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP. **Um futuro com energia sustentável: iluminando o caminho.** São Paulo: FAPESP, 2010.

GUSSOW, M. **Eletricidade básica.** 2ª ed. São Paulo: Editora Makron Books, 2004.

HADDAD, J.; AGUIAR, S. C.; MARTINS, A.R.S. **Eficiência energética: integrando usos e reduzindo desperdícios.** Agência Nacional de Energia Elétrica, 1999.

KAFRUNI, SIMONE; **Seca esvazia reservatórios de hidrelétricas no Norte e Nordeste, conta de luz deve subir.** Disponível em <

[www.em.com.br/app/noticia/economia/2016/08/20/internas\\_economia,795671/seca-aumenta-a-tensao.shtml](http://www.em.com.br/app/noticia/economia/2016/08/20/internas_economia,795671/seca-aumenta-a-tensao.shtml)>. Acesso em 05 abril 2017

LEMOS, M. **Arduino: Conheça esta plataforma de hardware livre e suas aplicações.** Disponível em <<https://imasters.com.br/desenvolvimento/arduino-conheca-esta-plataforma-de-hardware-livre-e-suas-aplicacoes/?trace=1519021197&source=single>>. Acesso em 22 janeiro 2014.

MAXWELL, J. C.; RAYLEIGH, J. W. S. *Theory of Heat*. Londres: Longsmans Green, 1908.

McROBERTS, M., **Arduino Básico**, 1ª ed. (Zanolli, R, Trad.). São Paulo: Editora Novatec; 2012.

NILSSON, J. W.; RIEDEL, S. A. **Circuitos Elétricos**. 8ª.ed. São Paulo: Editora Pearson Prentice Hall, 2009.

**Openenergymonitor.** Calibration Procedure. Disponível em <<https://learn.openenergymonitor.org/electricity-monitoring/ctac/calibration>>. Acesso em 12 março 2017.

**OpenEnergyMonitor.** Open source monitoring for understanding energy. Disponível em <<https://learn.openenergymonitor.org/>>. Acesso em: 04 abril 2017.

SOUZA, F. **Arduino MEGA 2560**. Disponível em <[www.embarcados.com.br/arduino-mega-2560/](http://www.embarcados.com.br/arduino-mega-2560/)>. Acesso em 08 junho 2017

**ThingSpeak.** Learn More about ThingSpeak. Disponível em <[https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more)>. Acesso em 26 junho 2017.

WILLICE, K. IoT ThingSpeak Monitor Widget. Disponível em <[https://play.google.com/store/apps/details?id=ua.livi.thingspeakmonitor&hl=pt\\_BR](https://play.google.com/store/apps/details?id=ua.livi.thingspeakmonitor&hl=pt_BR)>. Acesso 27 junho 2017.

# APÊNDICE - A

Nota Fiscal | Fatura de Energia Elétrica | Série B: 000935679 | CFOP: 5258/AA  
 Nº da Fatura: 0201707000935679  
 Instalação: 2248522

**CEMAR**  
**Companhia Energética do Maranhão**  
 Alameda A, Qd SQS, nº 100, Loteamento Quitandinha,  
 Altos do Calhau - São Luis - MA CEP: 65.070-900  
 Insc. Estadual: 120.515.11-3 CNPJ: 06.272.793/0001-84

Para atendimento,  
 informe este número.  
**Conta Contrato**

**Conta do Mês** 07/2017 **Vencimento** 18/07/2017

**Dados do cliente**

Tipo de Tarifa: CONVENCIONAL MONOFASICA Tensão Nominal: 220 V  
 Classificação: Residencial Pleno - MONOFASICO UL/Seq: SL07B001-3000  
 Perdas de Transformação(%): 0 Nr Medidor: 12521252934  
 Fator de Potência: 0

**Demonstrativo de Faturamento**

| Descrição               | Quantidade | Preço    | Valor (R\$) |
|-------------------------|------------|----------|-------------|
| Consumo                 | 459        | 0,686073 | 314,91      |
| Adicional Band. Amarela |            |          | 4,36        |
| Cip-Ilum Pub Pref Munic |            |          | 13,07       |

**Total a pagar: R\$ 337,34**

**Composição do consumo (R\$)**

| Compra de Energia | Transmissão | Distribuição (Cemar) | Encargos Setoriais | Tributos | Total (R\$) |
|-------------------|-------------|----------------------|--------------------|----------|-------------|
| 38,41             | 5,57        | 97,85                | 29,18              | 38,26    | 319,27      |

**Histórico do Consumo (kWh)**

| Mês | Consumo (kWh) |
|-----|---------------|
| JUL | 459           |
| JUN | 388           |
| MAI | 408           |
| ABR | 432           |
| MAR | 392           |
| FEV | 389           |

**Informações de tributos**

| Tributos | Base de cálculo | Alíquota (%) | Valor (R\$) |
|----------|-----------------|--------------|-------------|
| ICMS     | 319,27          | 20,0000      | 63,85       |
| PIS      | 319,27          | 1,3639       | 4,36        |
| COFINS   | 319,27          | 6,2823       | 20,05       |

**Reservado ao Fisco** Período Fiscal 11/07/2017  
 55404569E4E0C7F5EC8952C954E883B6

**Informações do consumo do mês**

| Nº Medidor  | Leitura Anterior | Leitura Atual | Consumo | Qtde. de dias | Constante | Tarifa sem tributos (R\$)         |
|-------------|------------------|---------------|---------|---------------|-----------|-----------------------------------|
| 12521252934 | 13.901           | 14.360        | 459     | 32            | 01        | R\$ ANEEL 2214/17<br>459 0,496410 |

**Reaviso de Vencimento** **Número do Programa Social**

**Indicadores de continuidade**

| MAI/2017    | DIC   | FIC   | DMIC |
|-------------|-------|-------|------|
| Meta Men    | 6,31  | 3,23  | 3,03 |
| Meta Tri    | 10,82 | 6,47  | 0,00 |
| Meta Anu    | 21,25 | 12,95 | 0,00 |
| Apurado Men | 0,00  | 0,00  | 0,00 |

**Datas**

| Emissão    | Apresentação | Previsão próxima leitura: |
|------------|--------------|---------------------------|
| 11/07/2017 | 11/07/2017   | 11/08/2017                |

Conjunto: FORQUILHA  
 EUSD(R\$): 117,20

**Informações para o cliente**  
 Períodos: Band. Tarif. Verde: 10/06 - 30/06 Amarela: 01/07 - 11/07

## APÊNDICE - B

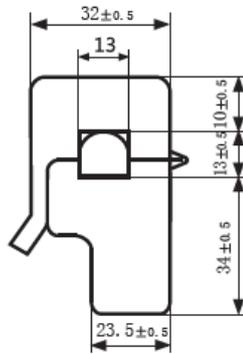
# SPECIFICATION

Customer Title : XiDi Technology Product Name: \_\_\_\_\_  
 Manufacture Model : SCT-013-030 \_\_\_\_\_

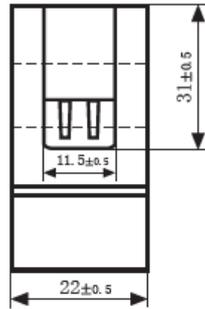
Charateristics: open size:13mm×13mm  
 1m leading wire  
 Core material:Ferrite  
 Fire resistance property:in accordance with  
 UL 94-V0  
 Dielectric strength: 1500V AC/1min 5mA  
 (between shell and output)



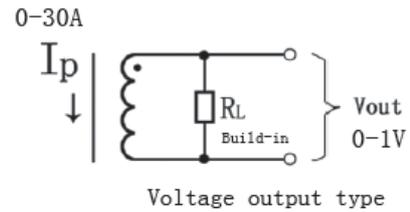
Outline size diagram:(in mm)



Front View



Side View



Voltage output type

Schematic Diagram

Typical table of technical parameters:

| input current | output voltage   | non-linearity    | build-in sampling resistance (RL)             |
|---------------|------------------|------------------|---|
| 0-30A         | 0-1V             | ±1%              | 62 Ω  |
| turn ratio    | resistance grade | work temperature | dielectric strength(between shell and output) |
| 1800:1        | Grade B          | -25°C ~+70°C     | 1500V AC/1min 5mA                             |

Customer Sign:

Phone: 0355-7929499-803  
 Cell: 13693334514  
 Contact Name: Engineer Chen

2011-7-21

## APÊNDICE - C

```
#include "EmonLib.h"           // Biblioteca EMON
#include <Adafruit_GFX.h>      // Display
#include <Adafruit_PCD8544.h> // Display
#include <Ethernet.h>         // Ethernet
#include <SPI.h>               // Utilizada pela Ethernet
#include <DS1307.h>           // RTC DS1307

EnergyMonitor emon1;

Adafruit_PCD8544 display = Adafruit_PCD8544(30, 31, 32, 33, 34);
//(Clk,Din,DC,CE,RST) Configura display

//Configuração do pinos do Arduino
int pin_tensao = A1;
int pin_corrente = A0;
DS1307 rtc(A4, A5);

// Variaveis para calculo de consumo
double kWh = 0;
double soma_kWh = 0;
double valor_conta = 0;
double valor_conta_ant = 0;
unsigned long tempo = 0;
unsigned long tempo_anterior = 0;
float tarifa = 0.668974; //valor da tarifa
int i = 0;
int ciclos = 0;
String Hora, Data;

// Variaveis para armazenar potencias
float pot_ativa = 0;
float Vrms = 0;
```

```

float Irms          = 0;

// Configuracao ETHERNET

byte mac[]          = { 0xD4, 0x28, 0xB2, 0xFF, 0xA0, 0xA1 }; // MAC
char thingSpeakAddress[] = "api.thingspeak.com";           //
Configurações THingSpeak
String writeAPIKey   = "8BLBTBPQQ1H9WG4E";                // Chave de
escrita ThingSpeak
const int updateThingSpeakInterval = 16 * 1000;            // Intervalo de
envio para o servidor

// Variaveis de setup
long lastConnectionTime = 0;
boolean lastConnected   = false;
int failedCounter      = 0;

// Iniciar Client Arduino
EthernetClient client;

void setup()
{
  Serial.begin(9600);          //Velocidade de comunicação serial

  rtc.halt(false);            //Aciona o relógio

  emon1.voltage(pin_tensao, 269, 1.7); // Tensão: pino, calibração, deslocamento
de fase
  emon1.current(pin_corrente, 29);     // Current: pino, calibração.

  display.begin();            // Ajuste iniciais display
  display.setContrast(44);     // Ajusta o contraste do display
  display.clearDisplay();      // Apaga o buffer e o display
  display.setTextSize(1);     // Define tamanho do texto
  display.setTextColor(BLACK); // Define cor

```

```

display.print("Aguardando... ");

rtc.setSQWRate(SQW_RATE_1);           // Definições do pino SQW/Out RELOGIO
rtc.enableSQW(true);

startEthernet();

display.display();
}

void loop()
{
do {
    ciclos = ciclos + 1;
    emon1.calcVI(20, 2000);           //time-out

    pot_ativa    = emon1.realPower;    //Recebe Potência Ativa
    Vrms         = emon1.Vrms;        //Recebe Tensão rms
    Irms         = emon1.Irms;        //Recebe corrente rms

    //Ignora valores pequenos que podem representar ruído
    if (Irms < 0.1 || Vrms < 10) {
        Vrms      = 0;
        Irms      = 0;
        pot_ativa = 0;
    }

    pot_ativa = sqrt(pot_ativa * pot_ativa);           // Módulo da potência ativa

    soma_kWh = soma_kWh + (pot_ativa / (60000));       // 60000 = intervalo
    medicao * kW

    tempo = millis() - tempo_anterior;
}

```

```
    enviar_valores (pot_ativa, valor_conta, kWh, Irms, Vrms); // Função de envio para  
ThingSpeak
```

```
} while (tempo < 60000);
```

```
tempo_anterior = millis();
```

```
kWh      = kWh + (soma_kWh / ciclos);
```

```
valor_conta = kWh * tarifa;
```

```
soma_kWh = 0;
```

```
//Informações display
```

```
display.clearDisplay();
```

```
display.print("R$ ");
```

```
display.println(valor_conta, 2);
```

```
display.println("-----");
```

```
display.print(kWh, 0);
```

```
display.println(" kWh");
```

```
display.print(Vrms, 1);
```

```
display.print("V ");
```

```
display.print(Irms, 1);
```

```
display.println("A");
```

```
display.print("P: ");
```

```
display.print(pot_ativa); //
```

```
display.println(" W");
```

```
display.print("M-1: ");
```

```
display.println(valor_conta_ant);
```

```
display.display();
```

```
ciclos = 0;
```

```
Hora = rtc.getTimeStr();
```

```
Data = rtc.getDateStr();
```

```
//Armazenar medição na virada do mês
```

```

if (Hora == "09:00:00" && Data == "11/07/2017") {

    kWh      = 0;
    valor_conta_ant = valor_conta;
    valor_conta = 0;
}
}

void enviar_valores (float pot_ativa, float valor_conta, double kWh, float Irms, float
Vrms) {

    // ##### ETHERNET #####

    // Printa resposta no serial monitor
    if (client.available())
    {
        char c = client.read();
        Serial.print(c);
    }

    // Desconectar do ThingSpeak
    if (!client.connected() && lastConnected)
    {
        Serial.println("...disconnected");
        Serial.println();

        client.stop();
    }

    // Chamar função para enviar para ThingSpeak
    if (!client.connected() && (millis() - lastConnectionTime >
updateThingSpeakInterval))
    {
        updateThingSpeak("field1=" + String(pot_ativa) + "&field2=" + String(valor_conta)
+ "&field3=" + String(kWh) + "&field4=" + String(Irms) + "&field5=" + String(Vrms));
    }
}

```

```

}

// Reinicia caso falha conexão 3 vezes
if (failedCounter > 3 ) {
    startEthernet();
}

lastConnected = client.connected();
}

void updateThingSpeak(String tsData) // Função de envio para o servidor
{
    if (client.connect(thingSpeakAddress, 80))
    {
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: " + writeAPIKey + "\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(tsData.length());
        client.print("\n\n");

        client.print(tsData);

        lastConnectionTime = millis();

        if (client.connected())
        {
            Serial.println("Connecting to ThingSpeak...");
            Serial.println();

            failedCounter = 0;
        }
        else

```

```

    {
        failedCounter++;

        Serial.println("Connection to ThingSpeak failed (" + String(failedCounter, DEC) +
        ");");
        Serial.println();
    }
}
else
{
    failedCounter++;

    Serial.println("Connection to ThingSpeak Failed (" + String(failedCounter, DEC) +
    ");");
    Serial.println();

    if (failedCounter >= 4 ) {
        startEthernet();
    }

    lastConnectionTime = millis();
}
}

void startEthernet()
{
    client.stop();

    Serial.println("Connecting Arduino to network...");
    Serial.println();

    delay(1000);

    // Connect to network amd obtain an IP address using DHCP
    if (Ethernet.begin(mac) == 0)

```

```
{  
  Serial.println("DHCP Failed, reset Arduino to try again");  
  Serial.println();  
}  
else  
{  
  Serial.println("Arduino connected to network using DHCP");  
  Serial.println();  
}  
}
```