



UNIVERSIDADE FEDERAL DE MATO GROSSO  
FACULDADE DE ARQUITETURA, ENGENHARIA E TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL,  
UTILIZANDO A PLATAFORMA DE HARDWARE LIVRE ARDUINO

JACKSON DOS SANTOS ANDRADE

ORIENTADOR: PROF. DR. FABRICIO PARRA SANTILIO

Cuiabá – MT

2017



UNIVERSIDADE FEDERAL DE MATO GROSSO  
FACULDADE DE ARQUITETURA, ENGENHARIA E TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL,  
UTILIZANDO A PLATAFORMA DE HARDWARE LIVRE ARDUINO

JACKSON DOS SANTOS ANDRADE

Trabalho Final de Curso apresentado ao  
Departamento de Engenharia Elétrica  
da Universidade Federal de Mato  
Grosso, como requisito parcial para a  
obtenção do título de bacharel em  
Engenharia Elétrica.

ORIENTADOR: PROF. DR. FABRICIO PARRA SANTILIO

Cuiabá – MT

2017

UNIVERSIDADE FEDERAL DE MATO GROSSO  
FACULDADE DE ENGENHARIA, ARQUITETURA E TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**CERTIFICADO DE APROVAÇÃO**

MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL,  
UTILIZANDO A PLATAFORMA DE HARDWARE LIVRE ARDUINO.

Discente: Jackson dos Santos Andrade.

Orientador: Fabricio Parra Santilio.

Aprovado em \_\_\_\_\_ de \_\_\_\_\_ de 2017.

Comissão Examinadora:

---

Prof. Msc. André Luiz Amorim da Fonseca  
Instituto Federal de Mato Grosso – IFMT

---

Profa. Dr. Camila dos Anjos Fantin

---

Prof. Dr. Fabricio Parra Santilio  
Universidade Federal de Mato Grosso – DENE / FAET / UFMT  
Orientador

---

Prof. Dr. Roberto Apolônio  
Universidade Federal de Mato Grosso – DENE / FAET / UFMT

Eu, Jackson dos Santos Andrade, autorizo o Departamento de Engenharia Elétrica, assim como a Universidade Federal de Mato Grosso a utilizar e reproduzir meu trabalho, apenas para fins de estudo e pesquisa, sendo expressamente vedado qualquer tipo de reprodução para fins comerciais sem prévia autorização específica do autor.

---

Discente: Jackson dos Santos Andrade

# AGRADECIMENTOS

Agradeço aos meus pais, Claudio e Jacira, por sempre me incentivar a estudar desde criança. A eles também por serem exemplos vivos de que com dedicação e trabalho duro é possível alcançar os nossos objetivos pessoais.

Ao meu orientador Prof. Dr. Fabricio Parra Santilio, pelo comprometimento, esforço e dedicação a este trabalho, pela enorme paciência durante este longo período de desenvolvimento e por acreditar no potencial deste trabalho.

Aos professores Roberto Perillo e Rodolfo Quadros, pelos conselhos, pela amizade, pelo apoio à minha formação acadêmica e pelo empenho notório à melhoria do curso de engenharia elétrica da UFMT.

Ao Prof. Dr. Dorival Gonçalves Júnior pela minha formação crítica às questões políticas, técnicas e sociais.

À instituição SENAI, pela minha formação técnica na área, que melhorou a minha formação como futuro engenheiro. Além do mais, forneceu-me a qualificação profissional que me possibilitou, junto a recursos pessoais dos meus pais, financiar a minha estadia na universidade.

À Barbara Moraes GIANESINI, colega de turma, pelas contribuições dadas ao trabalho, pois sem elas, o tempo de desenvolvimento teria se tornado muito maior.

À minha turma 2012/1, em especial aos membros do “conselho”, pelos melhores momentos da minha vida acadêmica na UFMT e pelos momentos de descontração durante as correrias e estresses em períodos de provas e trabalhos.

Às políticas públicas do Brasil, que me forneceu toda a minha educação de forma gratuita desde o pré-escolar até a minha graduação.

Aos membros da banca examinadora, pela disposição em auxiliar na conclusão deste trabalho.

Por fim, agradeço a todas as pessoas que contribuíram de alguma forma na minha graduação.

“Feliz é aquele que faz pequenas boas  
ações das quais não se lembra, pois  
não se corrompe esperando  
gratificações pelo que fez.”

Sólon Otomura

# RESUMO

**ANDRADE, J. Monitoramento de consumo de energia elétrica residencial, utilizando a plataforma de hardware livre Arduino.** Cuiabá – MT, 2017. 79 p. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Faculdade de Arquitetura, Engenharia e Tecnologia, Universidade Federal de Mato Grosso.

Atualmente, o conflito de interesses dos diferentes grupos que compõe a sociedade sobre o uso das diferentes fontes energéticas e os impactos sobre o meio-ambiente e na própria sociedade se torna cada vez maior. Diante disto, afim de realizar o uso racional de energia elétrica, os meios que um consumidor residencial tem como monitorar o seu consumo de maneira dinâmica são desmotivadores para a maioria. Nesse contexto, este trabalho apresenta o desenvolvimento de um protótipo de um medidor de energia que permita ao usuário monitorar o seu consumo de maneira dinâmica, por meio de tela ou via um servidor WEB, além do mais, o dispositivo também realizada registro de grandezas elétricas e demanda em cartão de memória, por fim, os dados de desenvolvimento do dispositivo são de código-aberto e possui custo de material relativamente baixo. Como resultado, espera-se que o protótipo facilite o monitoramento de consumo de energia, assim, motivando o usuário ser mais atuante no gerenciamento do uso da eletricidade.

Palavras-Chave – Arduino, protótipo de medidor de energia, projeto de código aberto.

# LISTA DE FIGURAS

Figura 2.1 Interface do software IDE .....	17
Figura 2.2 Arduino (a) MEGA, (b) PRO, (c) YÚN e (d) GEMMA.....	18
Figura 2.3 <i>Keypad Shield LCD</i> acoplado a um Arduino Uno .....	18
Figura 3.1 Tela inicial do Excel 2013 .....	21
Figura 3.2 Tela inicial do programa ANAWIN.....	21
Figura 3.3 Mega 2560.....	22
Figura 3.4 Arduino Uno .....	23
Figura 3.5 <i>Ethernet Shield V1</i> acoplado ao Arduino Mega.....	23
Figura 3.6 <i>Ethernet Shield V1</i> .....	24
Figura 3.7 Fonte de Alimentação .....	24
Figura 3.8 Módulo de Transformação .....	25
Figura 3.9 Módulo de Conversão .....	25
Figura 3.10 RTC DS1307 .....	26
Figura 3.11 <i>Shield Sd Card</i> .....	26
Figura 3.12 <i>MicroSD</i> e adaptador de cartão .....	26
Figura 3.13 <i>Display LCD</i> .....	27
Figura 3.14 Placa de controle .....	27
Figura 3.15 <i>Protoboard</i> .....	28
Figura 3.16 Sensor de Corrente .....	28
Figura 3.17 Placa de Conexão .....	29
Figura 3.18 Cabo V .....	29
Figura 3.19 Saga 3000 .....	30
Figura 3.20 Analisador Fluke 434 .....	31
Figura 3.21 Fluxograma do desenvolvimento do protótipo .....	32
Figura 3.22 Visão geral do protótipo.....	33
Figura 3.23 Conexões da fonte de alimentação .....	34
Figura 3.24 <i>Ethernet Shield V1</i> acoplado ao Mega 2560 .....	35
Figura 3.25 Conector ICSP do MEGA .....	35
Figura 3.26 Conexões do Mega 2560 .....	36
Figura 3.27 Conexões do Uno .....	37

Figura 3.28 Conexões do <i>LCD</i> e do RTC DS 1307 .....	37
Figura 3.29 Conexões do módulo SD.....	38
Figura 3.30 Conexões da placa de controle.....	39
Figura 3.31 Conexões da placa de conexão.....	39
Figura 3.32 Conexão de um cabo V .....	41
Figura 3.33 Conexão de um sensor de corrente.....	41
Figura 3.34 Rotina de funcionamento do primeiro programa .....	42
Figura 3.35 Linha 46 do primeiro programa .....	43
Figura 3.36 Exemplo de sinal de entrada do Uno .....	43
Figura 3.37 Local de calibração de medição .....	45
Figura 3.38 Exemplo de inserção de calibração nova .....	46
Figura 3.39 Fluxograma de comunicação entre o UNO e o MEGA .....	47
Figura 3.40 Endereço Slave do Uno .....	48
Figura 3.41 Inserção do endereço do Uno no Mega.....	48
Figura 3.42 Fluxograma do segundo programa.....	49
Figura 3.43 Fluxograma da rotina principal .....	51
Figura 3.44 Definição do tempo de demanda.....	52
Figura 3.45 primeira configuração do ThingSpeak .....	53
Figura 3.46 segunda configuração do ThingSpeak .....	53
Figura 3.47 Exemplo de segunda configuração .....	54
Figura 3.48 Terceira Configuração do ThingSpeak .....	54
Figura 3.49 Exemplo de canal do ThingSpeak .....	55
Figura 3.50 Circuitos de medição laboratorial .....	56
Figura 4.1 Gráfico de energia ao longo do tempo no primeiro circuito .....	61
Figura 4.2 Consumo total no primeiro circuito .....	61
Figura 4.3 Gráfico de energia ao longo do tempo no segundo circuito.....	62
Figura 4.4 Consumo total no segundo circuito.....	62
Figura 4.5 Gráfico de energia ao longo do tempo no primeiro circuito .....	63
Figura 4.6 Consumo total do primeiro circuito .....	63
Figura 4.7 Telas de informação do protótipo .....	64
Figura 4.8 Segmento de dados armazenado no cartão SD .....	65
Figura 4.9 gráfico no canal do ThingSpeak.....	66

# LISTA DE TABELAS

Tabela 3.1 Denominação dos módulos do protótipo .....	34
Tabela 3.2 Valor e significado da variável Y a cada ciclo .....	47
Tabela 4.1 Dados obtidos no primeiro circuito antes da calibração – Carga 1 .....	58
Tabela 4.2 Dados obtidos no primeiro circuito após calibração – Carga 1 .....	59
Tabela 4.3 Dados obtidos no segundo circuito antes da calibração – Carga 2.....	59
Tabela 4.4 Dados obtidos no segundo circuito após calibração – Carga 2 .....	59
Tabela 4.5 Dados obtidos no terceiro circuito antes da calibração – Carga 3.....	60
Tabela 4.6 Custo material de uma unidade do protótipo.....	66

# SUMÁRIO

## CAPÍTULO I

---

INTRODUÇÃO.....	13
1.1      Considerações iniciais .....	13
1.1.1      Monitoramento de Consumo.....	13
1.1.2      Justificativa .....	14
1.2      Objetivos .....	14
1.2.1      Objetivo Geral.....	14
1.2.2      Objetivos Específicos.....	14
1.2.3      Estruturação do TFC .....	14

## CAPÍTULO II

---

ARDUINO.....	16
2.1      Considerações iniciais .....	16
2.2      Plataforma Arduino .....	16
2.2.1      O <i>Software</i> do Arduino .....	16
2.2.2      O <i>Hardware</i> do Arduino .....	17
2.2.3      Arduino como plataforma de desenvolvimento do Protótipo .....	18

## CAPÍTULO III

---

MATERIAIS E MÉTODOS.....	20
3.1      Considerações iniciais .....	20
3.2      Materiais.....	20
3.2.1      Recursos Computacionais .....	20
3.2.2      Recursos Materiais .....	22
3.3      Métodos .....	31

3.3.1	Etapas seguidas para o desenvolvimento do protótipo .....	31
3.3.2	Desenvolvimento e construção do protótipo.....	32
3.3.3	Desenvolvimento do Software .....	41
3.3.4	Medições laboratoriais .....	56
<b>CAPÍTULO IV</b>		
RESULTADOS LABORATORIAS E DISCUSSÕES .....		58
4.1	Calibração do Protótipo.....	58
4.1.1	Primeiro Circuito – Carga 1 .....	58
4.1.2	Segundo Circuito – Carga 2 .....	59
4.1.3	Terceiro Circuito – Carga 3 .....	59
4.2	Validação do Protótipo.....	60
4.2.1	Primeiro Circuito – Carga 1 .....	60
4.2.2	Segundo Circuito – Carga 2 .....	62
4.2.3	Terceiro Circuito – Carga 3 .....	63
4.3	Demais Resultados .....	64
<b>CAPÍTULO V</b>		
CONCLUSÕES .....		67
REFERÊNCIAS .....		69
ANEXOS .....		72
ANEXO A.1.....		72
ANEXO A.2.....		73
ANEXO A.3.....		74
APÊNDICES .....		78
APÊNDICE A.1.....		78
APÊNDICE A.2.....		79

# CAPÍTULO I

---

## INTRODUÇÃO

### 1.1 Considerações iniciais

Este capítulo inicia-se com a temática da pesquisa, em seguida, apresenta a justificativa e os objetivos motivadores e, por fim, a estruturação deste Trabalho Final de Curso - TFC.

#### 1.1.1 Monitoramento de Consumo

Atualmente, o conflito de interesses dos diferentes grupos que compõe a sociedade sobre o uso das diferentes fontes energéticas e os impactos sobre o meio-ambiente e na própria sociedade se torna cada vez maior. Especificamente, no Brasil, este conflito tende aumentar no setor elétrico já que o consumo de energia elétrica para o horizonte de 2024 é de 786 TWh, um aumento de 4,1 % em relação a 2014 [1].

Diante disto, afim de realizar o uso racional de energia elétrica, quais são os meios que um consumidor residencial tem para monitorar o seu consumo, de forma a poder agir de maneira dinâmica no gerenciamento do uso de energia elétrica?

A primeira é por meio da fatura gerada pela concessionaria local, porém, devido a ela ser mensal, não permite ao consumidor monitorar o seu consumo de maneira instantânea.

A segunda é por meio de anotações dos valores indicados nos medidores de energia instalados nas residências, em seguida, a realização de determinados cálculos para a obtenção do consumo. Maneira esta que é pouco prática, pois o indivíduo tem que manter registros pontuais das medições, realizar cálculos e dependendo da região, a obtenção destas medições se tornam complicadas, a exemplo, tem-se a cidade de Cuiabá-MT, onde há medidores instalados há 3 metros do nível do chão.

A terceira é por meio de medidores eletrônicos disponíveis no mercado, porém a sua instalação e uso necessita de conhecimentos em eletrotécnica e de tratamentos de dados. Sendo que, quanto maior o número de recursos o medidor possui, maior será o valor deste. Assim, podendo desmotivar a compra pelo consumidor.

A quarta seria construir o próprio medidor, porém é necessário um conjunto de conhecimentos específicos, como de eletrotécnica, eletrônica, programação e outros, ademais, é necessário tempo disponível, pois o desenvolvimento de um aparelho demanda uma

quantidade elevada de tempo e também pode-se considerar o investimento em equipamentos para a construção e calibração do medidor e na obtenção de um local adequado de trabalho.

A partir deste questionamento demonstra-se que as alternativas disponíveis são desmotivadoras para a maioria dos consumidores residenciais que desejam monitorar o seu consumo de eletricidade de maneira dinâmica.

### 1.1.2 Justificativa

Com o intuito de facilitar o monitoramento de consumo de eletricidade pelos consumidores residências, o presente trabalho propõe-se a projetar um protótipo que permita ao usuário, de forma prática, monitorar o seu consumo de maneira instantânea por via local ou servidor web e manter registros dos dados de medições. Além do mais, o protótipo será de código-aberto, ou seja, permitirá a qualquer um construir a sua própria versão ou modificá-la.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Desenvolver um protótipo para medição de energia elétrica de código aberto para os consumidores residências.

### 1.2.2 Objetivos Específicos

- Desenvolver circuito eletrônico para a coleta, armazenamento e processamento de dados;
- Desenvolver rotina de processamento de dados e armazena-los em sistema WEB;
- Realizar medições de grandezas elétricas com o protótipo e medidores comerciais;
- Calibrar e validar o protótipo;
- Comparar o protótipo com um medidor comercial;

### 1.2.3 Estruturação do TFC

Incluindo este capítulo **Introdutório**, o presente trabalho está disposto em mais quatro capítulos, sendo eles:

Capítulo II – **Arduino**, é apresentado a plataforma Arduino, que é a base para construção do protótipo.

Capítulo III - **Materiais e Métodos**, são apresentados os materiais e métodos empregados no desenvolvimento deste trabalho.

Capítulo IV - **Resultados e discussões**, são apresentados os resultados de calibração e validação do protótipo, com base em medições de equipamentos comerciais.

Capítulo V - **Considerações finais e trabalhos futuros**, são apresentadas as considerações sintetizadas deste trabalho e encera-se com as sugestões de trabalhos futuros.

# CAPÍTULO II

---

## ARDUINO

### 2.1 Considerações iniciais

Este capítulo destina-se a apresentar a plataforma Arduino, plataforma esta que foi utilizada como base para a construção do protótipo.

### 2.2 Plataforma Arduino

Arduino é uma plataforma de licença livre de prototipagem eletrônica, baseado em hardware e software de fácil utilização [2]. Esta plataforma, ao logo dos anos, tem sido usada para diversos tipos de projetos, desde acender um simples LED (*Light Emitting Diode – Diodo Emissor de Luz*) até projetos complexos, como o desenvolvimento de impressoras 3D.

Foi desenvolvido em 2005 na Itália, com o objetivo de criar uma plataforma de rápida prototipagem para estudantes com pouco ou nenhum conhecimento de programação ou eletrônica. A placa eletrônica possui um preço relativamente baixo, sendo que o modelo mais utilizado, o UNO, pode ser encontrado no Brasil com preços abaixo de 30 reais. Ela também pode ser construída pelo usuário, já que a desenvolvedora também disponibiliza em seu site [3] o diagrama eletrônico das placas.

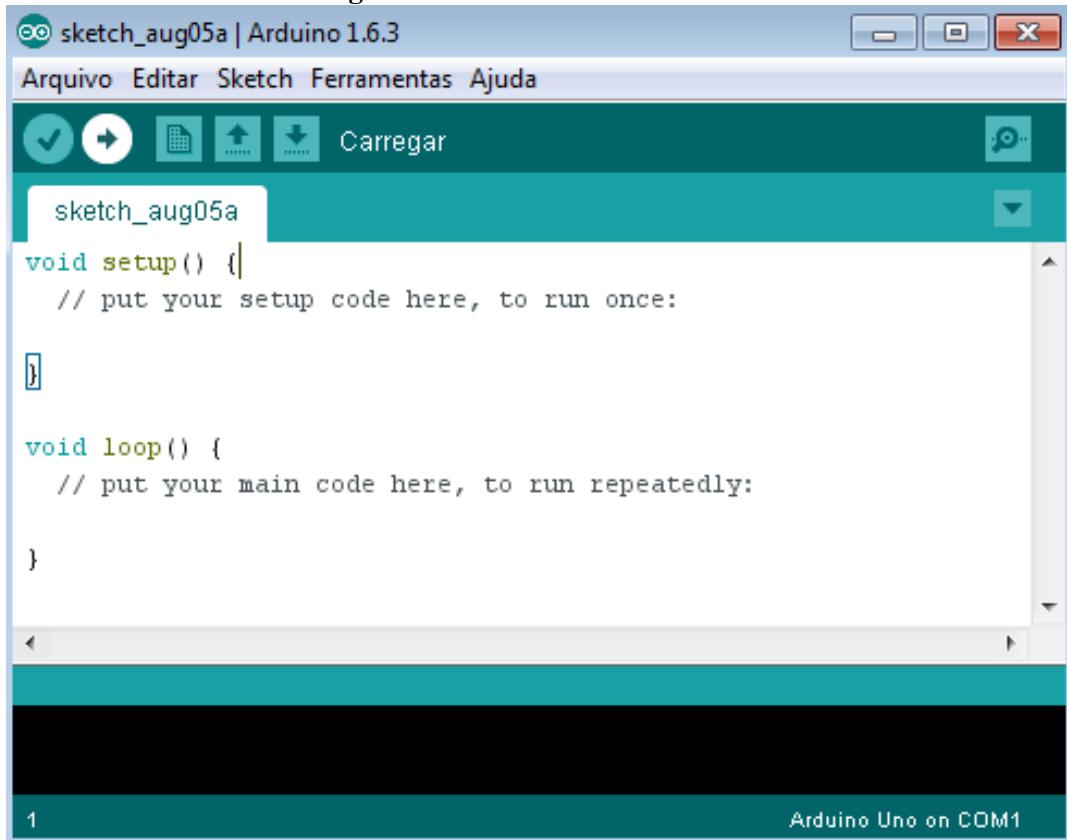
O software da plataforma, denominado de IDE (*Integrated Development Environment*), é disponibilizado de forma gratuita e funciona no Windows, Macintosh, Linux e OSX. É utilizado uma linguagem de programação própria, porém é essencialmente C/C++ [4]. No momento da programação, o programador pode encontrar na internet diversas bibliotecas contendo códigos já prontos, assim, tornando mais rápido e fácil o desenvolvimento do código final.

#### 2.2.1 O Software do Arduino

O IDE possui uma interface, figura 2.1, bastante simples e autoexplicativa, podendo ser escrita em várias linguagens como a *Processing*, *Max/MSP*, *Pure Data*, *SuperCollider*, *ActionScript*, *Java* e *C/C++* [5]. Além do mais, possui um editor de texto que oferece ao usuário

um conjunto de recursos para agilizar a programação, como realçador de texto, identificador de palavras especiais e outros.

**Figura 2.1** Interface do software IDE

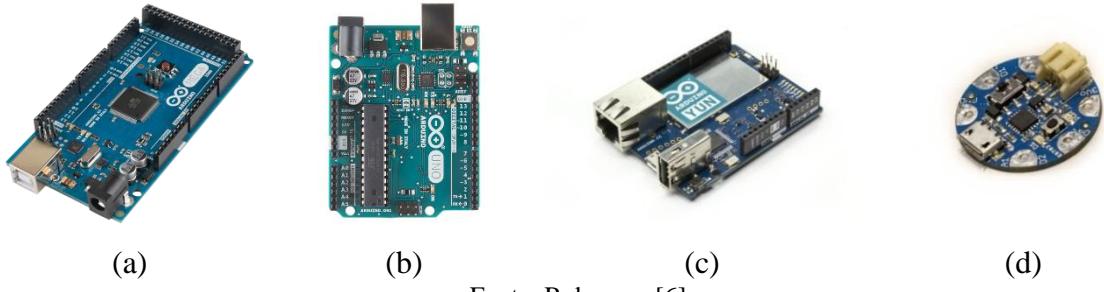


Fonte: Autoria Própria

### 2.2.2 O Hardware do Arduino

As características de uma placa Arduino dependem do modelo selecionado, porém a maioria possui um microcontrolador 8-bits Atmel AVR, entradas e saídas digitais e analógicas, com a possibilidade de gerar sinais modulados por largura de pulso, plug P4 fêmea para alimentação e chave de toque para reiniciar o microcontrolador. Ademais, possui interface serial e conexão USB, que é utilizada para transferir a programação desenvolvida na IDE para a placa, interagir com o computador em tempo real ou para alimentar o dispositivo. As placas também contam com memória de programação (*FLASH Memory*), memória RAM (*Random Access Memory* – Memória de Acesso Aleatório, em português) e memória EEPROM (*Electrically-Erasable Programmable Read Only Memory* – Memória Somente de Leitura, Programável e Apagável Eletricamente, em português). Na figura 2.2, estão ilustrados alguns modelos disponíveis no mercado.

**Figura 2.2** Arduino (a) MEGA, (b) UNO, (c) YÚN e (d) GEMMA



Fonte: Robocore [6].

Há no mercado extensões para as placas Arduino com o objetivo de implementar novas funções ou aumentar as características já existentes. Estas extensões são chamadas de *Shields* e como exemplo existe o *Keypad Shield LCD*, figura 2.3, que permite aos Arduínos a ele acoplado imprimir dados em uma tela e também adiciona botões às placas, de forma que o usuário possa programá-los na IDE para realizar as funções que desejar.

**Figura 2.3** Keypad Shield LCD acoplado a um Arduino Uno



Fonte: Robocore [6].

### 2.2.3 Arduino como plataforma de desenvolvimento do Protótipo

A plataforma Arduino foi selecionada como base para a confecção do presente trabalho pensando nas necessidades do público alvo, os consumidores residenciais. Por ser um público diverso, composto de diferentes classes econômicas e níveis de escolaridade, buscou-se uma plataforma que possuísse um preço de aquisição relativamente baixo e facilidade no manuseio. A plataforma Arduino, das atualmente presentes no mercado, é a que melhor se adequa a estes requisitos. Primeiro, por ter um preço acessível, na maioria dos modelos disponíveis no mercado, tanto das placas Arduino quanto dos *Shields*. O segundo, por ser uma plataforma construída com o objetivo de facilitar a confecção de projetos e possuir suporte online [3], que inclui informações, exemplos, fóruns de discussões e outros. Além do mais, a própria

comunidade que utiliza esta plataforma, disponibiliza, de forma gratuita, materiais complementares aos encontrados no suporte online, citando exemplos, tutorias de projetos pessoais, tradução de trabalhos em língua estrangeira para a língua nativa ou explicações de termos técnicos em uma linguagem comprehensível para os leigos. Todo este conteúdo, pode ser encontrado facilmente na internet em barras de pesquisa, youtube, *blogs*, fóruns e outros. Por fim, ainda pode-se incluir eventos organizados pela própria comunidade para a divulgar a plataforma, como os *ArduinoDay* ou *GenuinoDay*.

# CAPÍTULO III

---

## MATERIAIS E MÉTODOS

### 3.1 Considerações iniciais

Este capítulo destina-se a apresentar os materiais e métodos empregados no desenvolvimento deste trabalho. Os estudos foram conduzidos, inicialmente, com o estudo das partes componentes que iriam compor o protótipo, em seguida, foi desenvolvido o software que iria controla-lo. Na parte final, procedeu-se com a calibração e validação do medidor com dados colhidos experimentalmente.

### 3.2 Materiais

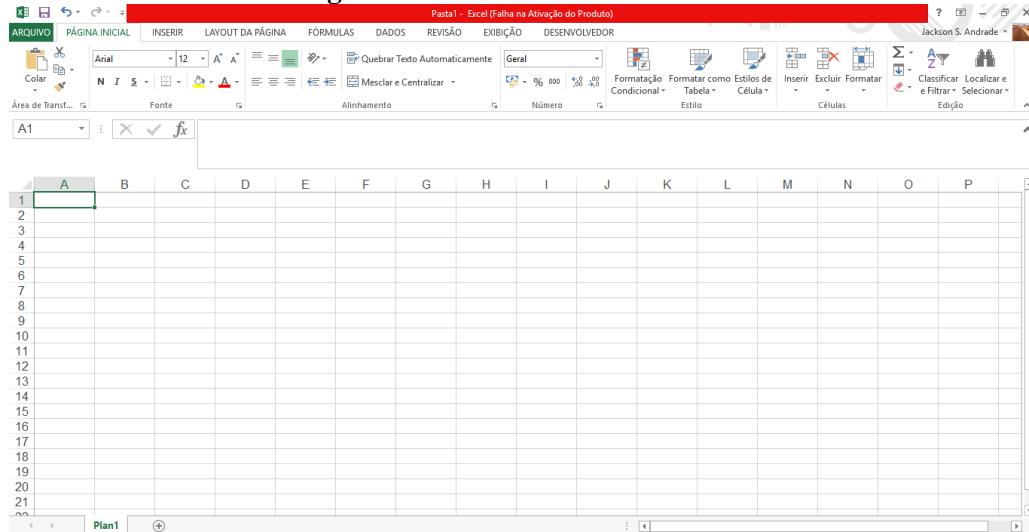
Neste item são apresentados os materiais utilizados neste trabalho. Os materiais estão divididos em dois grupos, um relacionado com os recursos computacionais e outro com os recursos materiais empregados na implementação do protótipo.

#### 3.2.1 Recursos Computacionais

Para o desenvolvimento do software do protótipo foi utilizado o Arduino IDE 1.6.12. Para a coleta de dados do protótipo e do SAGA3000 foram utilizados os programas EXCEL 2013 e o ESB Electronic Services.

##### 3.2.1.1 Excel® 2013

O Excel® é um editor de planilhas produzido pela Microsoft para computadores que utilizam o sistema operacional Windows, além de computadores Macintosh e dispositivos móveis como o Windows Phone, Android ou o iOS. A tela inicial do programa é ilustrada na figura 3.1.

**Figura 3.1 Tela inicial do Excel 2013**

Fonte: Autoria própria

### 3.2.1.2 ESB Electronic Services

Este programa computacional é a interface de comunicação entre o computador e os medidores SAGA3000, a qual possibilita descarregar os dados registrados. A tela inicial do programa é ilustra na Figura 3.2.

**Figura 3.2 Tela inicial do programa ANAWIN**

Fonte: Autoria própria

### 3.2.2 Recursos Materiais

#### 3.2.2.1 Arduino Mega 2560

O Arduino Mega 2560, neste trabalho denominado como Mega, foi selecionado como unidade principal de controle do protótipo, pois é um dos modelos do fabricante que apresenta a maior capacidade de memória e a maior quantidade de portas. A sua especificação técnica pode ser encontrada na figura 3.3 e o layout da placa, pinos de alimentação, pinos de entrada/saída e demais periféricos no Anexo A.1.

**Figura 3.3** Mega 2560



Fonte: Autoria própria

- Microcontrolador: ATmega2560;
- Tensão de alimentação recomendada: 7-12 V;
- Tensão de alimentação limite: 6-20 V;
- Tensão de operação: 5 V;
- Corrente por pino de E/S: 20 mA;
- Fonte interna: 3,3V (máx. 50 mA);
- E/S digitais: 54 pinos (15 podem ser saídas PWM);
- Entradas analógicas: 16 pinos;
- Memória FLASH: 256 kb;
- SRAM: 8 kb;
- EEPROM: 4 kb;
- Cristal oscilador: 16 MHz;
- Plug P4 para alimentação da placa;
- Entrada USB;
- Programável na própria plataforma via USB.

#### 3.2.2.2 Arduino Uno

O Arduino Uno, neste trabalho denominado como Uno, é o modelo utilizado para realizar as medições de tensão e corrente do protótipo, pois realiza esta função de maneira satisfatória e apresenta um custo de aquisição menor se comparado ao Mega 2560. A sua especificação técnica pode ser encontrada na figura 3.4 e o layout da placa, pinos de alimentação, pinos de entrada/saída e demais periféricos no Anexo A.2.

**Figura 3.4** Arduino Uno



Fonte: Autoria própria

- Microcontrolador: ATMega328P;
- Tensão de alimentação recomendada: 7-12 V;
- Tensão de alimentação limite: 6-20 V;
- Tensão de operação: 5 V;
- Corrente por pino de E/S: 20 mA;
- Fonte interna: 3,3V (máx. 50 mA);
- E/S digitais: 14 pinos (6 podem ser para saídas PWM);
- Entradas analógicas: 6 pinos;
- Memória FLASH: 32 kb;
- SRAM: 2 kb;
- EEPROM: 1 kb;
- Cristal oscilador: 16 MHz;
- Plug P4 para alimentação alternativa da placa;
- Entrada USB;
- Programável na própria plataforma via USB.

### 3.2.2.3 Ethernet Shield V1

O Arduino *Ethernet Shield V1* possibilita que uma placa Arduino possa conectar à internet via cabo RJ45, ademais, possui uma entrada para cartão MicroSD, assim, possibilitando o armazenamento de dados. É acoplada sobre as placas Arduino, figura 3.5, porém, não ocupa E/S deste, pois os seus sockets fêmeas estão interligados com os sockets da placa Arduino abaixo.

**Figura 3.5** Ethernet Shield V1 acoplado ao Arduino Mega



Fonte: Autoria própria

A sua especificação técnica pode ser encontrada na figura 3.6 e o layout da placa, pinos de alimentação, pinos de entrada/saída e demais periféricos no Anexo A.3.

**Figura 3.6 Ethernet Shield VI**



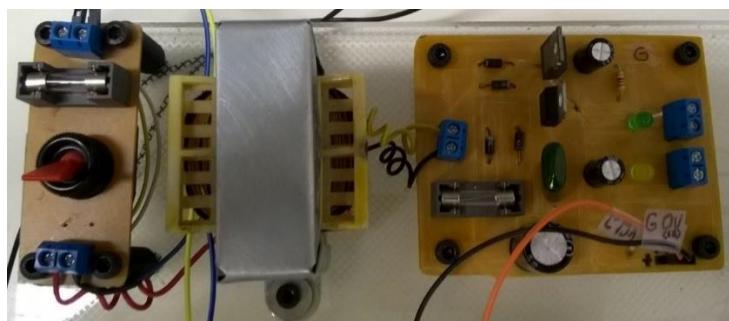
- *Ethernet Controller – W5100;*
- Tensão de operação: 5 V;
- Velocidade de conexão: 10/100 Mb;
- Conexão com o Arduino via porta SPI;
- Alimentação fornecida pela própria placa do Arduino.

Fonte: Autoria própria

### 3.2.2.4 Fonte de Alimentação

A fonte de alimentação, figura 3.7, é responsável por alimentar o protótipo com tensão contínua de 5V e 12V, sendo a corrente máxima comum das duas saídas de 1A. Ela é composta por dois módulos: o de transformação e o de conversão.

**Figura 3.7** Fonte de Alimentação



Fonte: Autoria própria.

#### a) Módulo de Transformação

O módulo de transformação, figura 3.8, tem a função de alimentar o módulo de conversão com tensão alternada de 12Vrms. Além disto, também permite a abertura do circuito de alimentação via chave interruptora e protege a fonte de alimentação de curto-circuito por meio de fusível de vidro.

**Figura 3.8** Módulo de Transformação



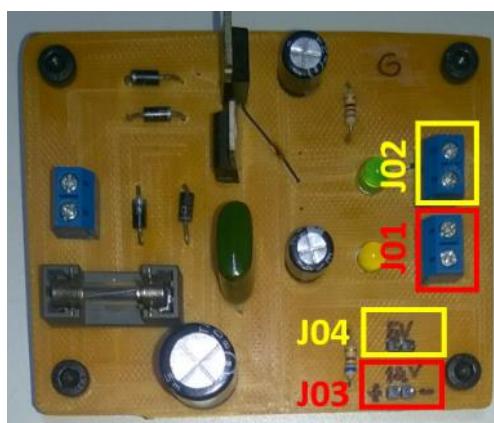
- Tensão de Entrada: 127V AC;
- Tensão de Saída: 12 V AC;
- Corrente de Saída: 1A AC.

Fonte: Autoria própria.

### b) Módulo de Conversão

O módulo de conversão, figura 3.9, tem a função de converter a tensão alternada proveniente do módulo de transformação em dois níveis de tensão CC: 5V e 12V. A primeira pode ser obtida no par de pinos J03 ou no borne J01 e a segunda no par de pinos J04 ou no borne J02.

**Figura 3.9** Módulo de Conversão



- Tensão de Entrada: 12Vac;
- Tensão de Saída 1: 5 Vcc;
- Tensão de Saída 2: 12Vcc;
- Corrente de Saída: 1A CC.

Fonte: Autoria própria.

#### 3.2.2.5 RTC DS 1307

O RTC DS 1307, figura 3.10, tem a função de fornecer as informações de horário (formatos 24 ou 12 horas) e calendário ao protótipo, sendo capaz de armazenar e transmitir informações de dia, mês e ano, com correções automáticas de data.

A comunicação com outros dispositivos é realizada por meio de protocolo serial I2C. O RTS ainda contém uma bateria de lítio que garante a preservação dos dados internos em caso de falha na alimentação.

**Figura 3.10** RTC DS1307



- Chip: DS1307;
- SRAM: 56 bytes;
- Comunicação: protocolo serial I2C;
- Circuito de detecção de falha de energia;
- Consome menos de 500nA no modo bateria com oscilador em funcionamento.

Fonte: Felipeflop [7].

### 3.2.2.6 Módulo SD

O módulo SD tem a função de armazenar determinadas informações do protótipo em cartão de memória SD. Ele é composto por dois itens: um *Shield Sd Card* e um *microSD* de 2Gb acoplado a um adaptador. O primeiro, imagem 3.11, realiza a leitura e a escrita das informações no segundo, imagem 3.12.

**Figura 3.11** *Shield Sd Card*



Fonte: Flipflop [7].

**Figura 3.12** *MicroSD* e adaptador de cartão

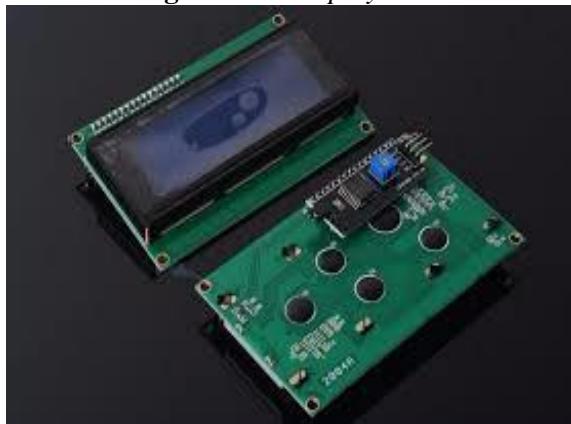


Fonte: Flipflop [07].

### 3.2.2.7 Display LCD

Este dispositivo, figura 3.13, tem a função de permitir ao usuário visualizar os menus contidos no protótipo. Ele é composto por dois itens interligados por meio de solda: o display e o módulo adaptador I2C. Sendo o primeiro o responsável por imprimir os caracteres na tela e o último por realizar a comunicação com outros dispositivos, utilizando para isto o protocolo I2C.

**Figura 3.13** Display LCD



- Tensão de Alimentação: 5V;
- Tela: Display 20x4 azul;
- Comunicação: protocolo I2C.

Fonte: Flipflop [7].

### 3.2.2.8 Placa de Controle

A placa de controle tem a função de permitir ao usuário navegar pelos menus do protótipo e reinicia-lo por meio de botões. Há cinco deles para uso livre na programação e um específico para a função de *reset* do dispositivo desejado. A placa em si pode ser visualizada na figura 3.14 e o circuito eletrônico no anexo A.4.

**Figura 3.14** Placa de controle



Fonte: Autoria própria.

### 3.2.2.9 *Protoboard*

A *Protoboard*, figura 3.15, tem a função de meio condutor de determinados circuitos do protótipo.

**Figura 3.15** *Protoboard*



- Furos: 1660;
- Furo de 0,8 mm de diâmetro;
- Resistência de Isolamento:  $100\text{M}\Omega$  minuto;
- Tensão Máxima: 500v AC por minuto;

Fonte: Autoria própria.

### 3.2.2.10 Sensor de Corrente

O sensor de corrente tem a função de medir a corrente no ponto de medição. Sendo esta ação realizada por efeito Hall, desta forma, de maneira não invasiva. A sua especificação técnica pode ser encontrada na figura 3.16 e o layout da placa, pinos de alimentação, pinos de entrada/saída e demais periféricos no Anexo A.5.

**Figura 3.16** Sensor de Corrente



- Modelo: SCT-013-060;
- Corrente de entrada: 0-50A;
- Sinal de saída: 0 - 1V;
- Material do Core: Ferrite;
- Plug de saída: Conector Jack BNC;
- Temperatura de trabalho: -25 a +70°C;
- Comprimento do cabo: 150 mm;

Fonte: Autoria própria.

### 3.2.2.11 Placa de Medição

A placa de medição, figura 3.17, realiza duas funções: Meio de conexão e tratamento de sinal. A primeira se refere ao fato de que ela é o meio por onde os sinais de corrente e tensão chegam ao protótipo, através dos cabos V e o sensor de corrente conectados a ela. Já a segunda função, trata-se da realização da adequação do sinal recebido, para que este possa ser lido pelas unidades Arduino. O circuito eletrônico pode ser encontrado no apêndice A.1.

**Figura 3.17** Placa de Conexão



- 2 entradas para cabo V;
- 1 entrada para sensor de corrente;
- 1 saída de sinal de tensão;
- 1 saída de sinal de corrente;

Fonte: Autoria própria.

### 3.2.2.12 Cabo V

O cabo V, figura 3.18, tem como função interligar o ponto de medição de tensão ao protótipo. Ele foi construído de forma a ter em uma das extremidades um conector Jack BNC macho para se conectar ao conector fêmea localizado no protótipo, especificamente na placa de conexão, e na outra ponta um conector jacaré para ser interligado ao ponto de medição de tensão.

**Figura 3.18** Cabo V



Fonte: Autoria própria.

### 3.2.2.13 Medidor Eletrônico SAGA3000

O SAGA3000 modelo 1360, figura 3.19, pertence a uma família de medidores eletrônicos de multigrandeza e de multifunção desenvolvidos pela ESB Electronic Services. Utilizam um DPS – *Digital Signal Processor* para realizar a aquisição de sinais e um microprocessador para o processamento das grandezas medidas.

**Figura 3.19** Saga 3000



- Fornece as grandezas primárias de tensão, corrente e fator de potência;
- Grandezas elétricas medidas: Tensão máxima, média e mínima, corrente, Potência ativa, aparente e reativa tanto as recebidas quanto as fornecidas;
- Medição indireta 2,5 / 20 A;
- Memória de massa 1Mbyte;
- Classe de exatidão de 1%;

Fonte: CCk [8].

### 3.2.2.14 Analisador Fluke 434

Este aparelho, figura 3.20 neste trabalho denominado de Fluke, é um analisador de energia elétrica que permite avaliar o sistema de distribuição de força a partir de um conjunto de possibilidades de medições. É uma ferramenta trifásica que mede cada um dos parâmetros do sistema de energia: tensão, corrente, frequência, energia, consumo de energia,  $\cos \phi$ , desequilíbrio, harmônicos e inter-harmônicos [9].

**Figura 3.20** Analisador Fluke 434



Fonte: Fluke [9].

### 3.2.2.15 Demais Materiais

Também foram utilizados os demais aparelhos descritos abaixo, porém devido a sua simplicidade serão apenas citados, porém é possível obter maiores detalhes em seus respectivos anexos quando disponível.

- Três Indutores [1240mH / 0,5A] – Veja Anexo A.6;
- Três capacitores [20 uF/230VAC/ tolerância:  $\pm 10\%$ ];
- Uma Lâmpada incandescente 220V/500W.

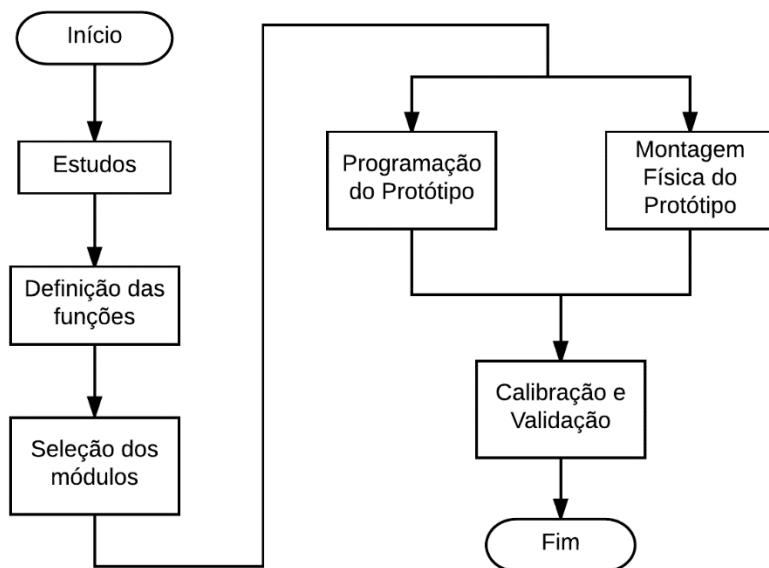
## 3.3 Métodos

Neste item são abordados os métodos adotados para o desenvolvimento deste trabalho. Na primeira etapa são selecionados as funções e os módulos que irão compor o protótipo, em seguida, na segunda etapa o protótipo é construído e testado no laboratório e na etapa final ele é calibrado e validado com os dados coletados no laboratório de forma comparativa com outros medidores.

### 3.3.1 Etapas seguidas para o desenvolvimento do protótipo

Este item evidencia as diversas etapas que foram seguidas, computacionais e experimentais, até a montagem e validação do protótipo. A sequência de trabalho é ilustrada na figura 3.21.

**Figura 3.21** Fluxograma do desenvolvimento do protótipo



Fonte: Autoria própria

### 3.3.2 Desenvolvimento e construção do protótipo

O desenvolvimento e construção do protótipo iniciou-se com a definição das funções que o protótipo deveria ser capaz de executar, sendo elas descritas a seguir:

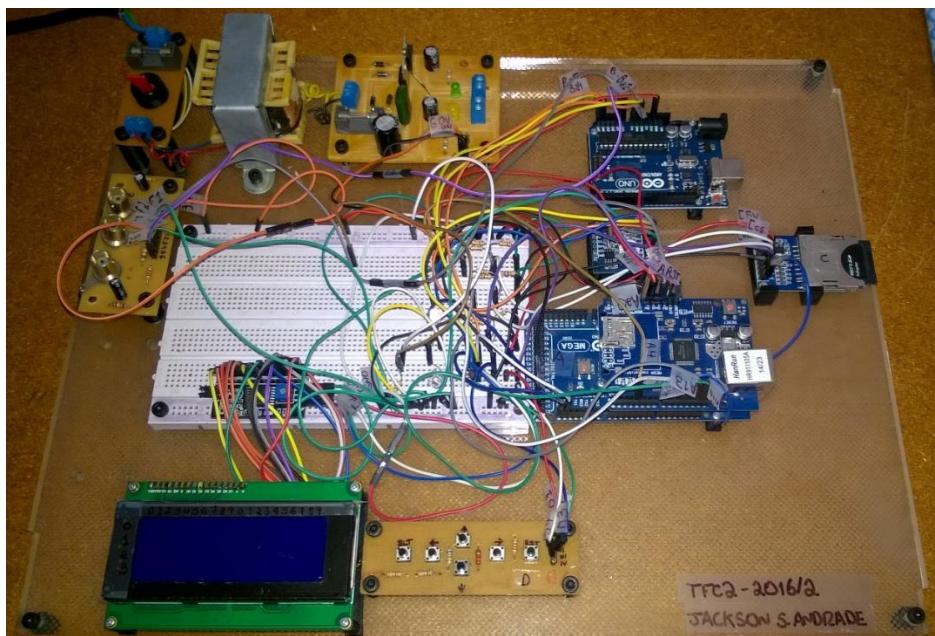
- Realizar medições das grandezas elétricas;
- Calcular o consumo de energia elétrica e calcular o custo estimado com base no valor da tarifa e dos impostos PIS, COFINS e ICMS;
- Ser capaz de armazenar os dados obtidos em cartão de memória SD e servidor WEB;
- Possuir tela e botões que permita ao usuário acessar dados do protótipo e configura-lo.

Em seguida, foram selecionados os itens que iriam compor o protótipo e, posteriormente, foi iniciada a construção, tanto a física quanto a de programação, sendo que elas ocorreram de forma paralela, porém neste trabalho elas serão demonstradas de forma separada.

### 3.3.2.1 Construção Física

Por ser a primeira unidade, o protótipo, figura 3.22, foi construído com foco em permitir ao usuário a manipulação facilitada dos seus itens componentes, desta forma, foi arquitetado de forma a admitir a rápida montagem e/ou desmontagem de circuitos, ademais, as partes componentes estão posicionadas estratégicamente para tornar as conexões por meio de cabos internos ao protótipo curtas, além do mais, as conexões a itens externos são realizadas em suas extremidades, por fim, pode-se adicionar que ele foi construído de forma a permitir futuras expansões, como novas unidade de medição e módulo de conexão wi-fi.

**Figura 3.22** Visão geral do protótipo



Fonte: Autoria própria

Os módulos que compõe o protótipo estão conectados entre si de maneira elétrica, por meio de fios. Estas conexões serão explicadas demonstrando as conexões de módulo a módulo, de maneira a tornar a compreensão mais fácil, além do mais, as ligações comuns entre dois dispositivos serão somente explicadas em um e no outro somente indicadas. Ainda em relação a este assunto, as conexões entre dispositivos serão indicadas por meio de fios denominados na seguinte ordem: origem, fim e número da ligação. Ademais, serão utilizadas as letras referentes a cada dispositivo, tabela 3.1, em vez dos seus respectivos nomes indicados no item materiais, isto se faz necessário para tornar as descrições menores e, assim, otimizando o tamanho das imagens e diminuindo a poluição visual. Exemplificando, uma ligação que tem origem da fonte

de alimentação, o fim na placa de conexão e sendo o número do fio igual a um, de acordo com a tabela 3.1, ficaria G,F-1.

**Tabela 3.1** Denominação dos módulos do protótipo

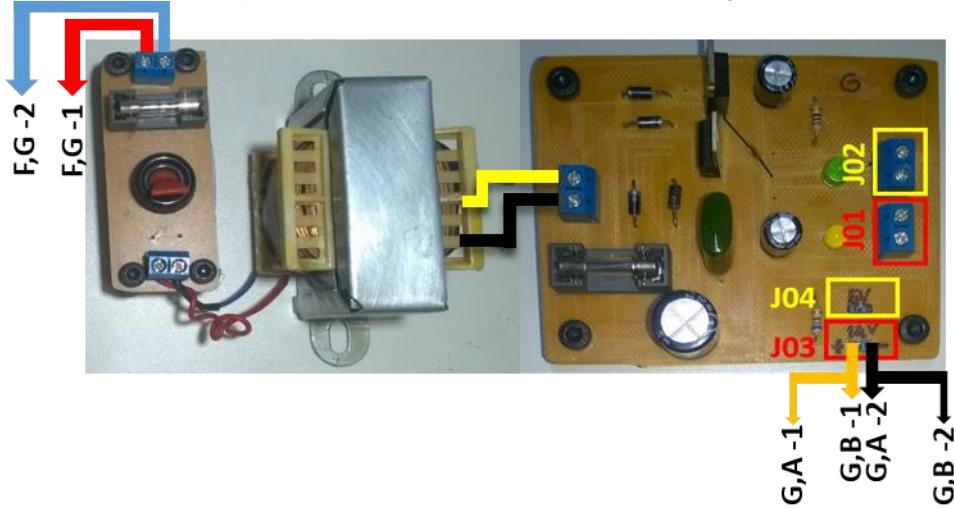
Letra	Módulo
A	Arduino Mega 2560
B	Arduino Uno
C	Módulo SD
D	Placa de Controle
F	Placa de Conexão
G	Fonte de Alimentação
H	<i>Display LCD</i>
I	RTC DS 1307
Sem letra	<i>Ethernet Shield V1</i>
Sem letra	<i>Protoboard</i>

Fonte: Autoria própria

- a) Fonte de Alimentação

As conexões estão conforme a figura 3.23.

**Figura 3.23** Conexões da fonte de alimentação



Fonte: Autoria própria

Observa-se que a alimentação da fonte (conexão F,G-1 e F,G-2), provém da placa de conexão. Esta alimentação tem a mesma diferença de potencial que a localizada entre os cabos

V conectados a esta placa, portanto, é necessário que esta tensão esteja dentro dos limites tolerados pela fonte, ou seja, entre 118V e 138V.

Outro ponto a se notar, é que a tensão de 5V (JO4 E JO2) não foi utilizada, mesmo que em vários pontos do circuito do protótipo apresentem a disponibilidade de uso deste nível de tensão, isto se deve ao fato de que os Arduínos já a disponibilizam e é preferível utiliza-la, pois apresentam uma regulagem de tensão melhor do que a da fonte de alimentação.

### b) MEGA 2560 e ETHERNET SHIELD

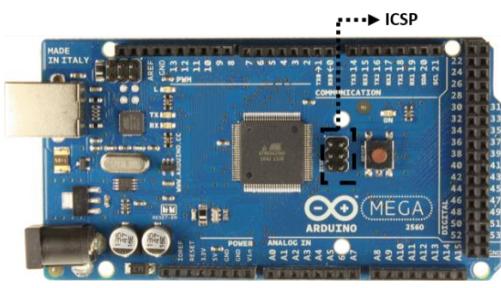
Estes dispositivos foram montados um sobre o outro, conforme a figura 3.24.

**Figura 3.24** Ethernet Shield V1 acoplado ao Mega 2560



Fonte: Flipflop [7].

**Figura 3.25** Conector ICSP do MEGA



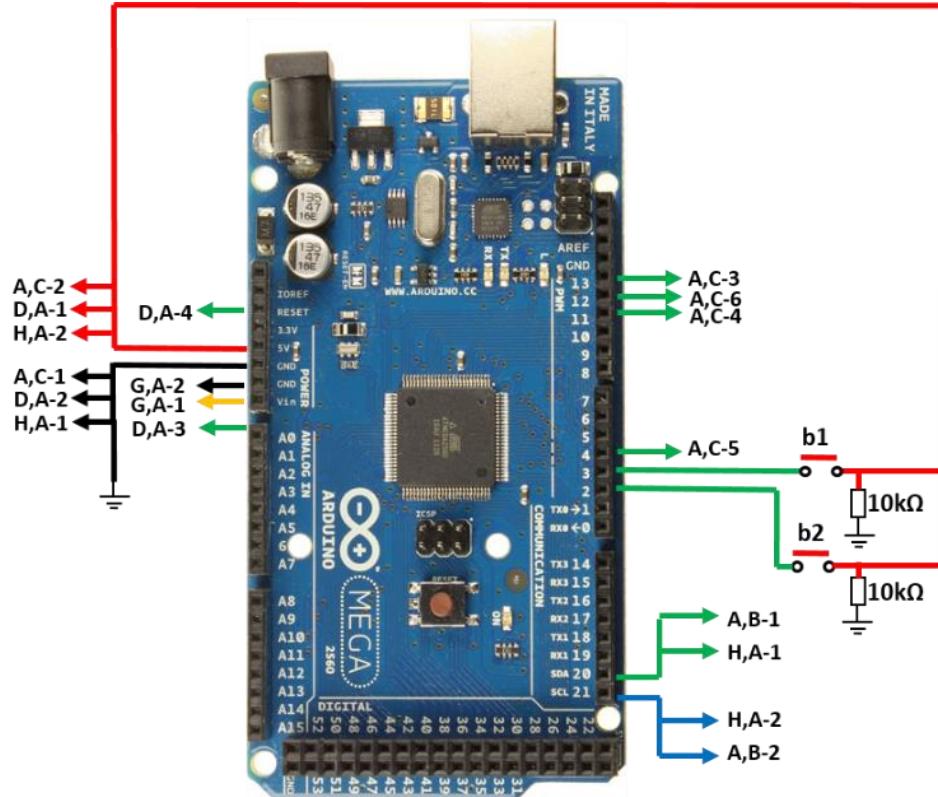
Fonte: Autoria própria.

A comunicação entre as placas é realizada via *SPI bus*, através da conexão dos conectores ICSP de cada placa, sendo um deles indicado na figura 3.25. Este conector no Mega tem seus pinos interligados aos pinos 50, 51, 52, assim, não é possível utilizar estes pinos para outros fins, ademais, o pino 53 também fica inviabilizado de uso devido a limitação do microcontrolador do Mega que não permite o uso desta porta quando se utiliza este tipo de comunicação [18].

Em ambas as placas, o pino 10 é usado para selecionar a opção Ethernet e o pino 4 para o MicroSD do *Ethernet Shield*, opção esta não utilizada no protótipo. O *Ethernet Shield* obtém a sua alimentação por meio dos pinos 5V e GND do Mega, assim, não sendo necessário fornecimento de energia externa [18].

As demais ligações elétricas do Mega são demonstradas na figura 3.26.

**Figura 3.26 Conexões do Mega 2560**



Fonte: Autoria própria

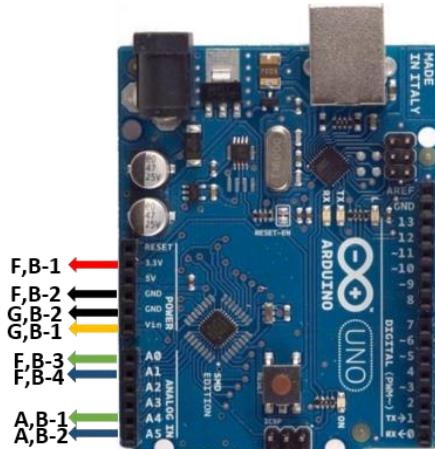
Os pinos VIN (G,A-1) e GND (G,A-2) são os pontos de alimentação do Mega. Já os demais pinos têm as seguintes funções:

- Pino 2: Quando o botão b2 é pressionado, o protótipo apaga o consumo do dia, consumo do mês e consumo total.
- Pino 3: Quando o botão b1 é pressionado, o protótipo apaga todos os dados da memória, assim, retornando as configurações originais.
- Pinos 20 e 21: São responsáveis por comunicar-se com os demais dispositivos que utilizam o protocolo I2C, sendo os pinos também denominados de, respectivamente, SDA e SCL do Mega.

c) Arduino UNO

As conexões deste dispositivo foram realizadas conforme a figura 3.27.

**Figura 3.27** Conexões do Uno.



Fonte: Autoria própria

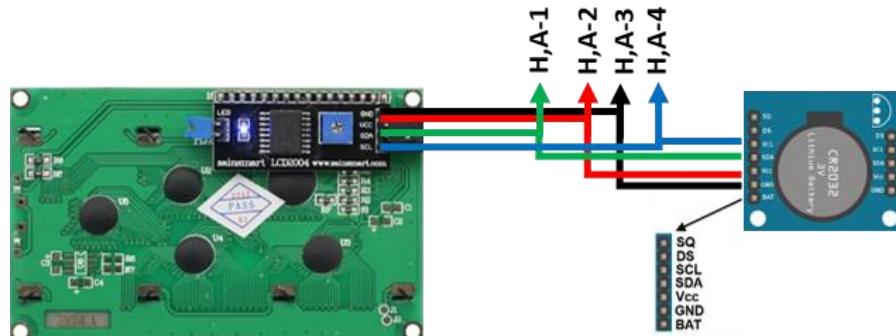
Os pinos VIN (G,B-1) e GND (G,B-2) são os pontos de alimentação do módulo. Já os demais pinos têm as seguintes funções:

- Pino A0: É utilizada como entrada analógica para o sinal de tensão.
- Pino A1: É utilizada como entrada analógica para o sinal de corrente.
- Pinos A4 e A5: São responsáveis por comunicar-se com MEGA por meio do protocolo I2C, sendo os pinos também denominados de, respectivamente, SDA e SCL do Uno.

d) RTC DS 1307 e *display LCD*

As conexões destes dispositivos foram realizadas conforme a figura 3.7.

**Figura 3.28** Conexões do *LCD* e do RTC DS 1307



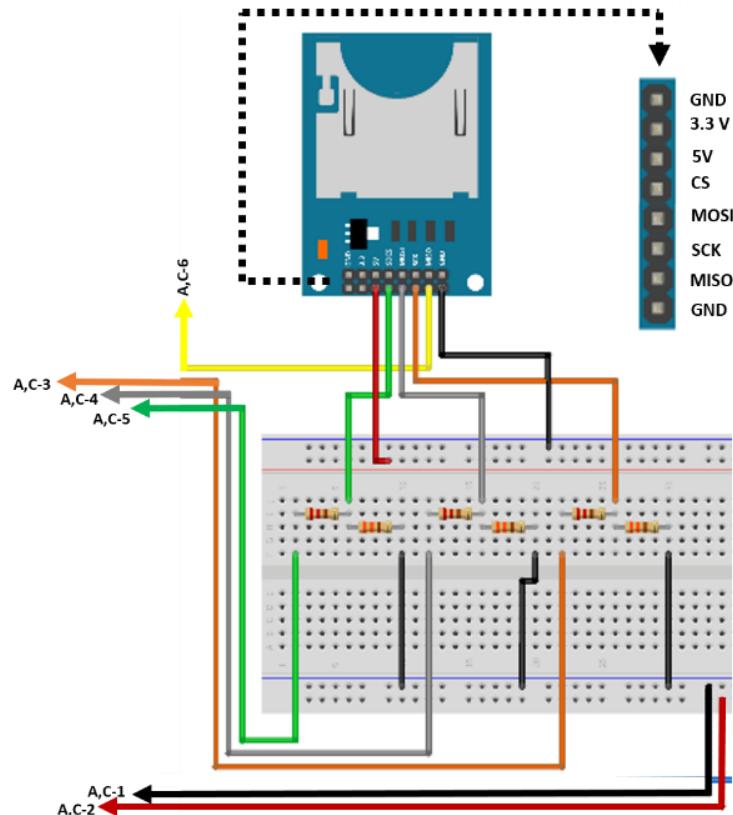
Fonte: Autoria própria

O par de pino SDA e SCL realiza a comunicação I2C dos dispositivos com o MEGA, já os pinos VCC e GND, estabelecem a alimentação de 5V nestes módulos.

e) Módulo SD

As conexões deste módulo foram realizadas conforme a figura 3.29.

**Figura 3.29** Conexões do módulo SD



Fonte: Autoria própria

Os pinos CS, MOSI, SCK, MISO são os responsáveis por estabelecer a interface ISP, comunicação de dados, no entanto o nível máximo de tensão no módulo é de 3,3V, no entanto o MEGA 2560 envia sinais em nível de 5V, assim sendo, foi necessário instalar os divisores de tensão por resistor visualizados na figura 3.29. Em relação aos pinos de 5V e GND, eles se fazem necessários para estabelecer a alimentação do dispositivo com 5V.

f) Placa de Controle

A placa foi conectada ao MEGA, conforme a figura 3.30.

**Figura 3.30** Conexões da placa de controle



Fonte: Autoria própria

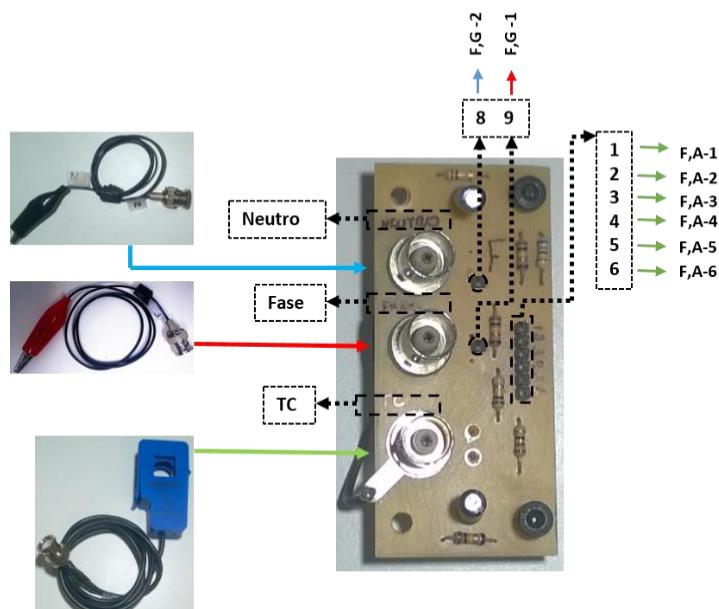
Os pinos têm as seguintes funções:

- Pinos I e II: Alimentar a placa com 5 Vcc;
- Pino III: Envia um sinal contínuo a porta analógica do MEGA quando um dos botões for pressionado, sendo que o módulo deste sinal (0 a 5Vcc) irá depender de qual botão foi acionado;
- Pino IV: Quando o botão RST é pressionado, o Arduino Mega reinicia.

g) Placa de Conexão

As ligações elétricas desta placa estão indicadas na figura 3.31.

**Figura 3.31** Ligações da placa de conexão.



Fonte: Autoria própria

Os pinos desta placa realizam as seguintes funções:

- Pinos 1 e 2: Alimentam o circuito de tratamento do sinal de tensão com 3.3V CC;
- Pino 3: Enviam ao Arduino Uno o sinal tratado de tensão.
- Pinos 4 e 5: Alimentam o circuito de tratamento do sinal de corrente com 3.3V CC;
- Pino 6: Enviam ao Arduino Uno o sinal tratado de corrente.
- Pinos 8 e 9: Fornecem à fonte de alimentação a tensão proveniente da diferença de potencial entre os conectores Fase e Neutro, ou seja, a tensão entre os cabos V conectados a esta placa.

Já os conectores BNC fêmea têm como tarefa:

- FASE: receber a onda de tensão de uma fase por meio de um cabo V;
- NEUTRO: receber a referência de tensão, o neutro, por meio de um cabo V.
- TC: receber o sinal de um sensor de corrente.

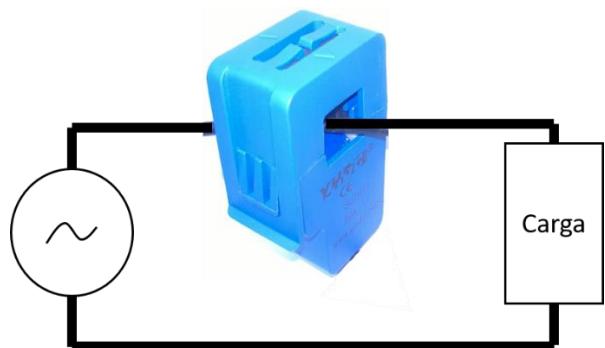
#### h) Cabo V e Sensor de Corrente

Para que o protótipo possa realizar a medição da tensão, é necessário que dois cabos V estejam conectados a dois pontos distintos de tensão, sendo um deles o neutro e o outro em uma fase. A diferença de potencial deve estar entre 118V e 138V, portanto valores superiores ou inferiores de tensão afetarão o funcionamento da fonte e, por conseguinte, o protótipo.

A conexão dos cabos V aos pontos de medição deve ser realizada conforme ilustrado na figura 3.32. Em relação a medição de corrente, basta que o sensor de corrente seja conectado ao condutor que se deseja realizar a medição, sendo que o limite de leitura é limitado pela capacidade do sensor, neste caso 50 A. Esta conexão deve ser realizada de acordo com a orientação localizada em cima do sensor, onde a seta indica o sentido da carga, conforme figura 3.33.

**Figura 3.32** Conexão de um cabo V

Fonte: Autoria própria

**Figura 3.33** Conexão de um sensor de corrente

Fonte: Autoria própria

### 3.3.3 Desenvolvimento do Software

Esta etapa consistiu no desenvolvimento de dois programas, especializados cada um com sua devida função, o primeiro em realizar a medição de tensão, corrente e fator de potência, enquanto que o segundo, no gerenciamento destas medições, no controle dos módulos, armazenamento de informações e meio de acesso do usuário aos dados por meio da placa de controle e do *display LCD*.

O projeto foi desenvolvido de forma a permitir que os programas trabalhem de forma independentemente, assim, possibilitando que haja alterações na programação somente naquele em que é desejado, permitindo assim, que, por exemplo, possa se utilizar diferentes métodos de medição sem que seja necessário alterar o segundo programa. Além do mais, esta independência também permite aos programas trabalharem de maneira mais rápida, já que independem da ação do outro programa para agirem.

O primeiro programa foi instalado no Arduino Uno, enquanto que o segundo foi inserido no Mega. A comunicação entre eles se dá por meio uma relação de *master* e *slave*, sendo este último o Uno. A comunicação ocorre da seguinte maneira, o *master* envia as ordens ao *slave*, que após realizar o que foi solicitado, retorna a suas tarefas originais, assim, por exemplo, o Mega solicita ao Uno os valores das medições e este por sua vez os envia e depois retorna a sua rotina principal, neste caso, rodar o primeiro programa, enquanto que o Mega retorna ao segundo. Desta forma, permite que o processo de medição seja praticamente contínuo, já que a comunicação é realizada de maneira rápida.

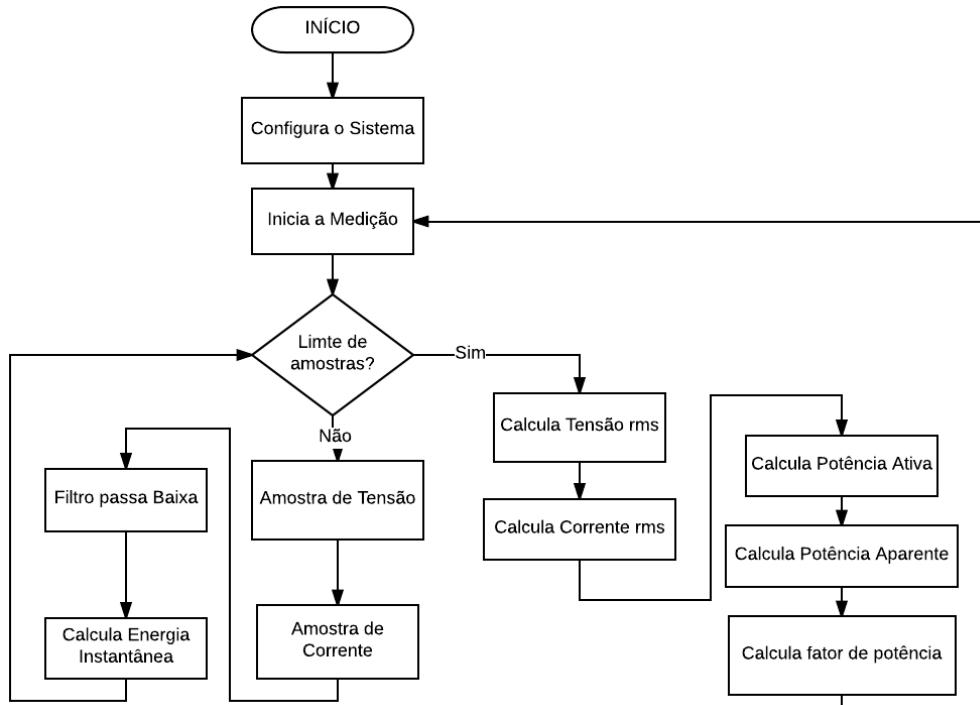
Nos itens seguintes, será dado uma visão macro do primeiro programa e algumas de suas peculiaridades, em seguida, como é realizada a comunicação entre o Uno e o Mega e, por fim, também uma visão macro do segundo programa, com algum detalhamento de pontos

cruciais de sua rotina. Para maiores detalhes, o primeiro programa encontra-se no apêndice A.2 e o segundo no apêndice A.3.

### 3.3.3.1 Primeiro Programa – Presente no Arduino Uno

A rotina deste programa obedece ao fluxograma indicado na figura 3.34.

**Figura 3.34** Rotina de funcionamento do primeiro programa



Fonte: Autoria própria

O início do programa é realizado juntamente com a energização do Uno, em seguida é efetivado a configuração das variáveis e bibliotecas utilizadas no programa. Logo em seguida, o Uno inicia o processo de obtenção das medições, sendo realizada em duas etapas, a primeira consiste em obter amostras de tensão, corrente e potência instantânea, já a segunda, a partir destas amostras, obter os valores de tensão e corrente rms, potência ativa, potência aparente e fator de potência.

Estas duas etapas são realizadas por meio da biblioteca VIP, que foi construída a partir da EmonLib, desenvolvida por Trystan Lea [10]. Essas etapas serão explicadas de maneira simplificada, para que se possa interpretar o processo de obtenção de valores pelo Uno, caso deseje-se maior aprofundamento, recomenda-se a leitura do trabalho do Trystan Lea [10]. A VIP encontra-se em GitHub[19] e a EmonLib pode ser encontrada em EmonLib[11].

a) 1º Etapa

Uma vez iniciada a primeira etapa, o Uno ficará obtendo amostras até que um limite pré-estabelecido seja atingido. Este limite é o número de meia-ondas do sinal que se deseja medir, sendo ele modificado na linha 46 do primeiro programa, figura 3.35, especificamente no primeiro parâmetro, sendo que o segundo estabelece um limite de tempo de segurança para que esta rotina seja realizada. Neste caso, foi estabelecido para que sejam medidas 20 meia-ondas e um limite de tempo de 2000 milissegundos.

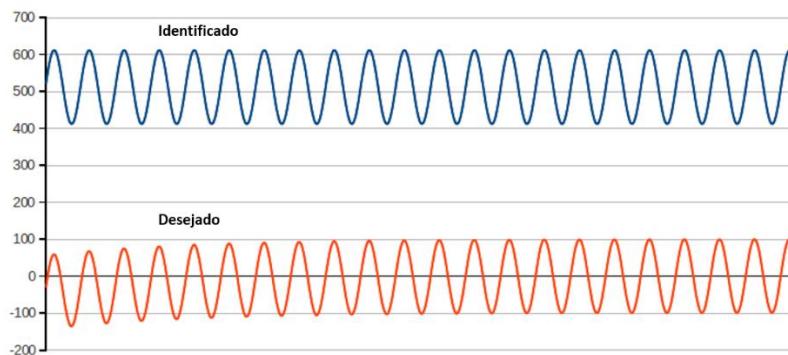
**Figura 3.35** Linha 46 do primeiro programa

```
46 | emon1.calcVI(20, 2000);
```

Fonte: Autoria própria

O primeiro item a ser obtido é uma amostra de tensão por meio da porta digital A0 do Uno, sendo que ela é obtida somente após a onda de tensão passar pelo zero. Em seguida, é obtida uma amostra de corrente, por meio da porta digital A1. Para obter tais amostras, o processo de captação, na placa de conexão, reduz o módulo do sinal e em seguida soma 1,65V<sub>CC</sub> ao sinal, para que possam ser lidos pelo Uno. Desta forma o sinal identificado pelo programa diferencia-se do sinal desejado, imagem 3.36.

**Figura 3.36** Exemplo de sinal de entrada do Uno



Fonte: Autoria própria.

Devido a isto, as medições de tensão e corrente passam por um filtro passa-baixa digital na rotina do programa que basicamente retira este sinal CC inserido anteriormente, assim, tornado o sinal identificado próximo do desejado.

Após o filtro, a medição de tensão é multiplicada pela de corrente, assim, obtendo uma medição de potência instantânea. Em seguida, a medição de tensão e de corrente são elevados

ao quadrado. Por fim, estes valores são armazenados juntamente com o valor da potência instantânea e somadas com uma nova medição, até que o limite de meia-ondas estabelecido no início seja atingido. Segundo Trystan Lea [12], está lógica de programação permite obter a cada 20 milissegundos até 50 amostras de tensão e outras 50 de corrente.

b) 2º Etapa

Após o limite de medições ser atingido, a segunda etapa inicia-se. Primeiramente, calcula-se a tensão RMS, por meio da equação 3.1, onde o somatório de  $u^2(n)$  é o valor armazenado de tensão na primeira etapa e N é o número de amostras obtidas. Em seguida, é realizada a obtenção da corrente RMS, utilizando a mesma equação, porém com o valor armazenado da corrente e seu respectivo número de amostras.

**Equação 3.1** Cálculo de tensão RMS.

$$U_{rms} = \sqrt{\frac{\sum_{n=0}^{N-1} u^2(n)}{N}}$$

Fonte: open.energy[13].

Posteriormente, a potência ativa é calculada a partir da equação 3.2. Sendo que o resultado do somatório, já foi obtido na primeira etapa, pois trata-se do valor da potência instantânea, portanto, neste momento só é realizado a divisão da potência instantânea pelo número de amostras.

**Equação 3.2** Cálculo da potência ativa

$$P \equiv \frac{1}{N} \sum_{n=0}^{N-1} u(n) \times i(n)$$

Fonte: open.energy[13].

Por fim, é obtido a potência aparente e o fator de potência pelas equações 3.3 e 3.4.

**Equação 3.3** Cálculo da potência aparente

$$S = U_{rms} \times I_{rms}$$

S - Potência Aparente

Fonte: Autoria própria

**Equação 3.4** Cálculo do fator de potência

$$Fp = P/S$$

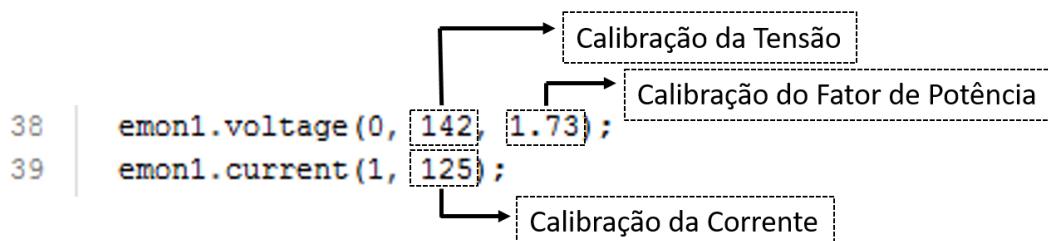
Fp – fator de potência

Fonte: Autoria própria

c) Calibração da Medição

O protótipo permite realizar três calibrações nas medições de forma independente, a primeira na tensão, a segunda no fator de potência e a última na corrente. Elas são realizadas no primeiro programa, sendo as duas primeiras na linha 38 e a última na linha 39, conforme os locais indicados na figura 3.37.

**Figura 3.37** Local de calibração de medição



Fonte: Autoria própria

O processo de calibração consiste em obter um número que inserido no local indicado permita obter uma medição exata ou aproximada a de um medidor de referência, conforme o método indicado no item a seguir 3.3.4.1 Calibração do protótipo. O procedimento a ser aplicado consiste em utilizar a equação 3.5 para adquirir tal número, por exemplo, para calibração da tensão, o protótipo mediu 127V enquanto que o medidor de referência mediu 123V, ademais, a calibração anterior do protótipo tinha como valor 142, assim, a calibração nova será 137,53, conforme pode ser visto na equação 3.6.

Equação 3.5 cálculo de calibração nova

$$\text{calibração nova} = \frac{\text{medição de referência}}{\text{medição protótipo}} * \text{calibração velha}$$

Fonte: Autoria própria

Equação 3.6 exemplo de calibração nova

$$\text{calibração nova} = \frac{123}{127} * 142 = 137,53$$

Fonte: Autoria própria

Por fim, basta inserir o novo valor de calibração no local de calibração da tensão, figura 3.38. Para as demais calibrações basta repetir o mesmo método utilizado na tensão, porém cada calibração com suas respectivas variáveis.

**Figura 3.38** Exemplo de inserção de calibração nova

```
38 | emon1.voltage(0, 137.53, 1.73);
```

Fonte: Autoria própria

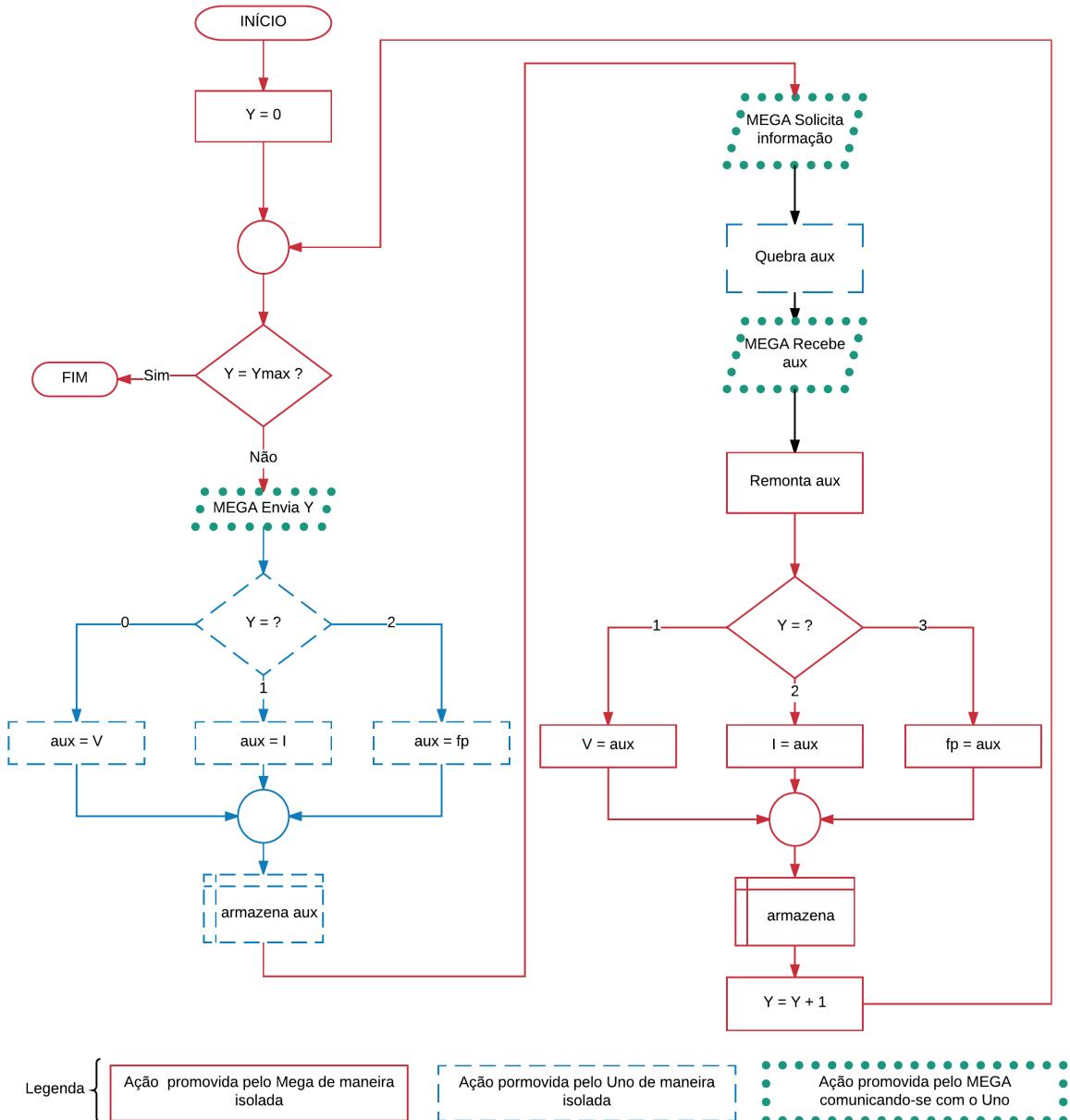
### 3.3.3.2 Comunicação entre o *Master* e o *Slave*

A comunicação entre o Uno e o Mega ocorre em ciclos de duas etapas, onde em cada ciclo ocorre a transmissão de uma informação referente as medições realizas pelo Uno. Irão ocorrer quatro ciclos, onde as informações transmitidas seguem a seguinte ordem: Tensão, Corrente, fator de potência e o sinal do fator de potência. Esta comunicação ocorre dentro da rotina interna denominada de “rotina principal” dentro do segundo programa, explicada no item 3.3.3.3 Segundo Programa.

Na primeira etapa, o Mega envia uma variável que contém um indicador da informação do qual necessita ao Uno, este por sua vez interpreta este sinal e, em seguida, busca a informação, desmonta-a e aguarda. Na segunda etapa, o Mega solicita ao Uno que envie a informação, o Uno obedece e a envia. O Mega recebe o que pediu, remonta a informação e a armazena de acordo com o indicador utilizado na primeira etapa. Após isto, inicia-se um novo ciclo até que o Mega receba todos os dados de medição.

Na figura 3.39 está indicado este processo de comunicação, onde a variável “Y” é o sinal utilizado como indicador da informação desejada e a tabela 3.2 demonstra o que esta variável armazena e significa a cada ciclo. Por motivo de simplificação, foi ocultado da figura 3.39 o caso em que Y é igual a três. Há também a variável “Y<sub>max</sub>”, que é igual a quatro, quando Y for igual a esta variável, o processo de comunicação encerra-se e o Uno e o Mega retornam a suas atividades originais, neste caso, ao primeiro e segundo programa.

**Figura 3.39** Fluxograma de comunicação entre o UNO e o MEGA



Fonte: Autoria própria

**Tabela 3.2** Valor e significado da variável Y a cada ciclo

Ciclo	Valor	Significado
1º	0	Tensão
2º	1	Corrente
3º	2	Fator de potência
4º	3	Sinal do fator de potência

Fonte: Autoria própria

O processo de desmontagem da informação consiste em quebra-la em três unidades de dezenas. Isto é realizado pois o maior valor que pode ser enviado em uma unidade de informação é de 255, um byte, sendo que é possível enviar até 6 destas unidades por comunicação [14]. Já o processo de remontagem utiliza o mesmo princípio, porém de maneira contrária.

A comunicação conforme descrita acima só é possível se for realizada duas configurações, a primeira deve ser inserida no Uno, no qual consiste em escrever um endereço no qual o *master* possa enviá-lo dados, sendo que as informações enviadas a um endereço somente podem ser lidas pelo *slave* que possui este endereçamento. O endereço é estabelecido pelo fabricante do dispositivo ou pelo usuário quando está opção está disponível. Sendo esta última a opção utilizada, no qual foi optado pelo endereçamento “0x08”, mas poderia ser qualquer outro que não coincidisse com o do display LCD ou do RTC 1307. O endereçamento foi inserido no primeiro programa na linha de programação descrita na figura 3.40.

**Figura 3.40** Endereço Slave do Uno

```
6 // endereço do módulo slave que pode ser um valor de 0 a 255
7 #define myAdress 0x08
```

Fonte: Autoria própria

A outra configuração deve ser realizada no Mega, para que o mesmo saiba o endereço do Uno, conforme figura 3.41.

**Figura 3.41** Inserção do endereço do Uno no Mega

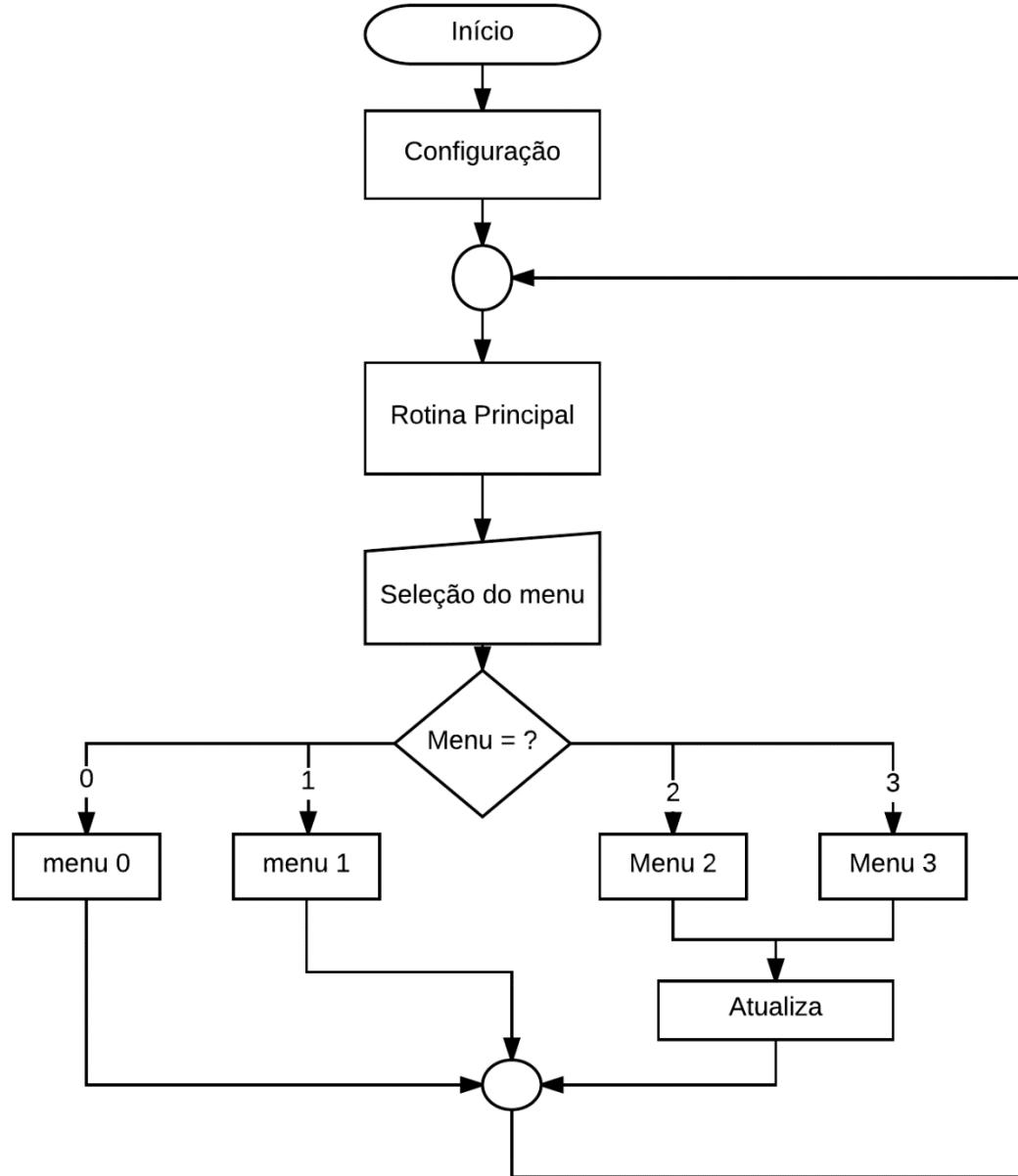
```
42 #define slaveAdress 0x08 // endereço do Arduino Uno
```

Fonte: Autoria própria

### 3.3.3.3 Segundo Programa

A rotina deste programa obedece ao fluxograma indicado na imagem 3.42.

**Figura 3.42** Fluxograma do segundo programa



Fonte: Autoria própria

O programa inicia-se juntamente com a energização do Mega, em seguida, o mesmo configura as variáveis voláteis, aquelas que apagam com desenergização do dispositivo, com os valores pré-estabelecidos no programa e também aquelas que possuem seus dados com base nos valores salvos na memória EEPROM, aquelas que não apagam quando ocorre a

desernegização do dispositivo. Isto garante que sempre que o protótipo for energizado continue com os valores anteriores ao momento de seu desligamento.

Em seguida, o programa entra na rotina chamada de principal, nela o dispositivo irá comunicar-se com o Uno para buscar os dados das medições, efetuar os cálculos de demanda, salvar, quando desejado pelo usuário, os dados no cartão SD e envia-los para o servidor WEB, por fim, armazenar os dados de consumo na EEPROM.

Posteriormente, por meio da placa de controle e da visualização dos caracteres impressos na tela do protótipo, o usuário seleciona em qual menu deseja entrar, sendo eles numerados de zero a três, no qual os dois primeiros são de visualização de dados de medição e os dois últimos de configuração. Caso seja realizada alteração de algum dado nestes menus, o programa atualiza novamente as variáveis voláteis, com a finalidade de mantê-las atualizadas. Logo após esta etapa, o programa reinicia o ciclo entrando novamente na rotina principal e continua assim até que o protótipo seja desligado.

Um ponto a se notar, é que se o programa não identificar o acionamento de qualquer tecla na placa de controle durante a seleção de menu, o mesmo irá retornar para a rotina principal. Porém há uma exceção para esta regra, caso seja selecionado o menu dois ou o três, neste caso, o programa aguarda o usuário fazer as alterações que desejar, no entanto, a rotina principal para e, por conseguinte, interrompe o processo de obtenção de medições.

#### a) Menus

É por meio dos menus que o usuário tem acesso aos dados do protótipo e a possibilidade de alterar determinadas configurações inerentes ao protótipo. A maneira de navegar pelos menus é indicado no apêndice A.2 e um resumo das suas funções é descrito abaixo.

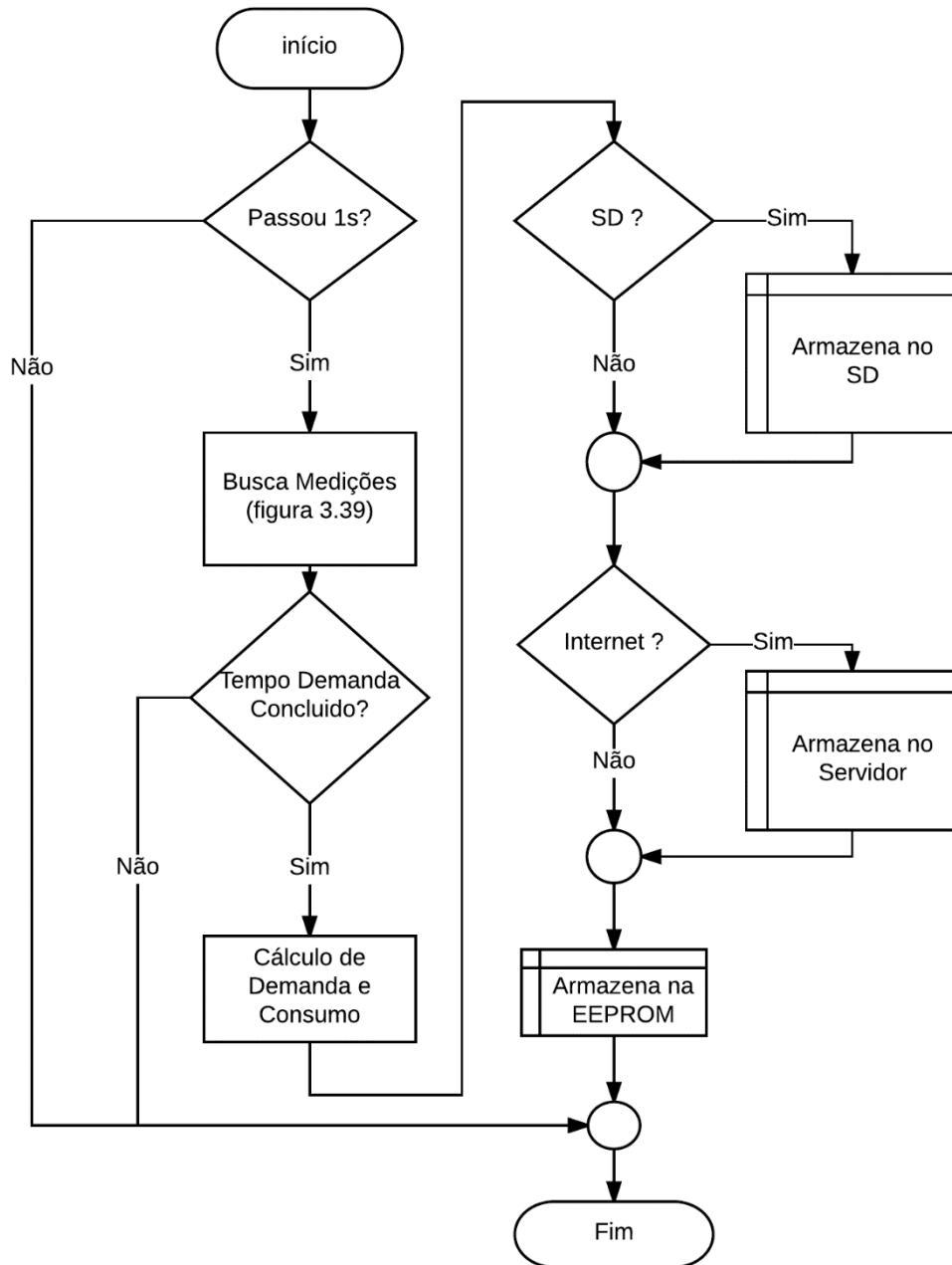
- Menu zero: Disponibiliza a visualização do relógio, data, tensão, corrente, fator de potência, custo da eletricidade utilizada no dia e no mês, sendo ambas contadas a partir da meia noite;
- Menu um: Disponibiliza a visualização da potência ativa, consumo de energia realizado no dia e no mês, ademais, o consumo de energia total registrado pelo protótipo;
- Menu dois: permite ao usuário configurar se o protótipo irá ou não salvar informações no cartão e/ou em servidor WEB.
- Menu três: permite ao usuário inserir o valor da tarifa de energia elétrica, dos impostos (ICMS, COFINS e PIS), ademais, ajustar o relógio e o calendário,

apagar o consumo acumulado do dia e do mês e, por fim, selecionar o dia em que o consumo do mês de energia elétrica tem início.

### b) Rotina Principal

Esta rotina obedece ao fluxograma indicado na figura 3.43.

**Figura 3.43** Fluxograma da rotina principal



Fonte: Autoria própria

Após ter início, a rotina irá verificar se passou mais de um segundo da última vez que o Mega buscou informações com o Uno, caso negativo, o programa sai da rotina, caso contrário, irá ocorrer a comunicação, conforme descrita no item 3.3.3.2 Comunicação entre o *Master* e o *Slave*. Em seguida, a rotina entra no processo decisório para verificar se o tempo de demanda foi atingido, caso negativo, o programa armazena os dados como amostras e sai da rotina. Este tempo de demanda é configurado no segundo programa, conforme a figura 3.44. Neste trabalho foi estabelecido o tempo de um minuto para esta tarefa, pois para tempos maiores seria necessário desenvolver novas rotinas para o cálculo e armazenamento de números com mais de 10 casas decimais, sendo que quanto maior o tempo, maior seria o número de casas.

**Figura 3.44** Definição do tempo de demanda

```
184 | int medminutos = 1;
```

Fonte: Autoria própria

Porém se o tempo for atingido, o programa irá calcular a demanda e o consumo a partir dos dados armazenados anteriormente, conforme, respectivamente, a equação 3.7 e a equação 3.8.

**Equação 3.7** Cálculo da Demanda

$$D = \frac{\sum_{n=1}^T P(n)}{N}$$

D = demanda (W);

P(n) = amostra de potência ativa (W);

N = Número de amostras (N=T);

T = tempo de demanda (s).

**Equação 3.8** Cálculo do Consumo

$$C = \sum_{n=1}^T P(n)$$

C = consumo (W);

P(n) = amostra de potência ativa (W);

T = tempo de demanda (s).

Fonte: Autoria própria

Fonte: Autoria própria

Posteriormente, se a opção de gravação dos dados no SD estiver ativa, ação esta ativada no menu dois, o programa irá salvar no cartão SD os dados de tensão e corrente rms, fator de potência, demanda e consumo, ademais, outras informações complementares como data e horários das medições. A quantidade de informação que pode ser salva está limitada pela

quantidade de memória do cartão SD, neste caso de dois Gb. Aproximadamente, cinquenta horas de medição consumiu apenas 165 Kb, tornando possível, assim, registrar mais de 60 mil horas de dados. Ademais, os dados são salvos no formato de texto (.txt), assim, podem ser abertos no Windows pelos programas Excel e Bloco de notas.

Em seguida, o programa verifica se a opção de envio de dados para a internet está ativa, opção está que também é ativada no menu dois. Caso positivo, o Mega irá enviar informações a um servidor na WEB via cabo de rede acoplado ao *Ethernet Shield*. O servidor escolhido para o protótipo é o fornecido pelo site ThingSpeak [15], que permite enviar até oito diferentes tipos de informações ao servidor, sendo que na conta gratuita, modelo de conta utilizada pelo protótipo, é possível enviar dados a cada 15s. O site disponibiliza o passo-a-passo em [16] de como realizar o envio de dados. Para cada conta é necessário realizar três configurações principais, a primeira é o número do canal do usuário e a sua respectiva senha, sendo que ambos são fornecidos na conta do usuário dentro do site. Estes dados foram inseridos no segundo programa, conforme imagem 3.45.

**Figura 3.45** primeira configuração do ThingSpeak

```
31 | unsigned long myChannelNumber = 178355; //obtido no site
32 | const char * myWriteAPIKey = "IYZK9CJEHVJ7P4GK"; //obtido no site
```

Fonte: Autoria Própria

A segunda configuração trata-se do que e por onde deseja-se enviar os dados ao servidor. É necessário configurar dois itens, o primeiro é o campo no qual será enviando a informação, é possível escolher até oito, por isso, escolhe-se um número de um a oito. O segundo item, trata-se da informação que será enviada no campo selecionado no primeiro item. Por exemplo, optou-se que o protótipo enviasse somente o valor da potência ativa, desta forma, o primeiro item é o campo um e o segundo item é a variável P, que armazena o valor daquela potência, assim, a configuração ficou conforme a imagem 3.46.

**Figura 3.46** segunda configuração do ThingSpeak



```
877 | ThingSpeak.writeField(myChannelNumber, 1, P, myWriteAPIKey);
```

Fonte: Autoria Própria

Caso deseje-se enviar outra informação, por exemplo, a tensão, a configuração ficaria conforme a imagem 3.47.

**Figura 3.47** Exemplo de segunda configuração

```
ThingSpeak.writeField(myChannelNumber, 2, V, myWriteAPIKey);
```

Fonte: Autoria Própria

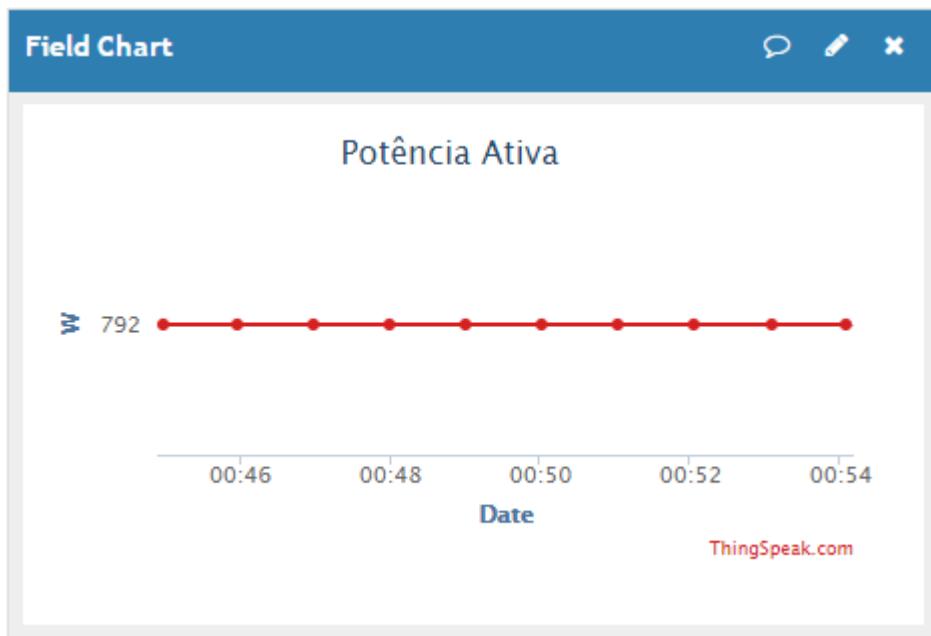
A terceira e última configuração é realizada dentro do próprio site da ThingSpeak [15], nesta etapa configura-se os canais que apresentaram os dados recebidos pelo protótipo. Uma vez dentro do site, entra-se em canais, meus canais, *New Channel*. Neste momento, figura 3.48, preenche-se as lacunas conforme as orientações do site, sendo que as lacunas descritas como “Campo 1”, “Campo 2” e assim por diante são os mesmos campos selecionados na segunda configuração, assim, é possível montar canais de diferentes tipos, por exemplo, canais de corrente e tensão ou até mesmo canais de diferentes tipos de potência.

**Figura 3.48** Terceira Configuração do ThingSpeak

Fonte: Autoria Própria

Uma vez finalizada esta configuração, obtém-se um canal com uma aparência semelhante ao demonstrado na figura 3.49.

**Figura 3.49** Exemplo de canal do ThingSpeak



Fonte: Autoria Própria

Por fim, o Mega soma os dados de consumo aos obtidos em rotinas anteriores e salva o resultado, valor este que é utilizado como registro de consumo total realizado pelo medidor. Em seguida, caso seja um novo dia e o primeiro registro, este resultado substitui o valor armazenado na variável “consumo\_dia”, variável esta que é utilizada como base para determinar o consumo de energia do dia e o seu respectivo custo. Caso também seja o dia em que o usuário registrou como dia do início de registro de consumo de energia do mês, registro realizado no menu três, o resultado também substitui o valor armazenado na variável “consumo\_mes”, variável esta que é utilizada como base para determinar o consumo de energia realizado no mês e o seu respectivo custo. De forma resumida, o consumo do dia é zerado todos os dias as 0h:00m e o consumo do mês é zerado as 0h:00m do dia de início programado no menu. O consumo total só será zerado caso o usuário pressione o botão b2, conforme descrito no item 3.3.2.b e identificado na figura 3.26.

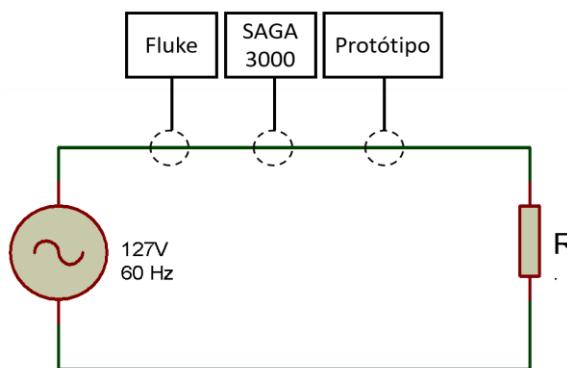
Estas duas substituições foram utilizadas para ser possível sempre calcular dados que expiram no fim do dia e/ou no final do mês a partir da primeira medição do intervalo de tempo. Além do mais, estas primeiras medições sempre são com base no registro geral e sempre atualizadas quando as condições ditas anteriormente são verdadeiras, assim, evitando que erros por arredondamentos inerentes ao programa acumulem-se. Por medida de segurança, o registro de “consumo total”, “consumo\_dia” e “consumo\_mes” são registrados na EEPROM.

### 3.3.4 Medições laboratoriais

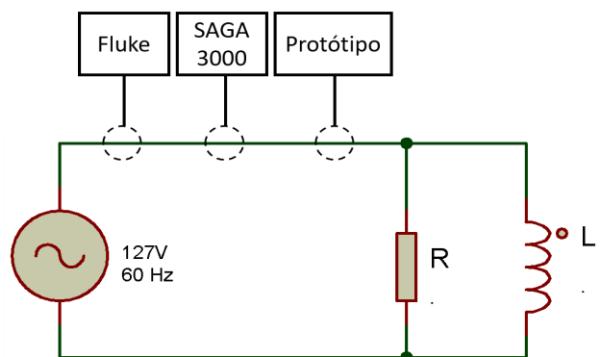
Foram realizados dois experimentos, o primeiro consistiu na calibração do protótipo e o segundo na sua validação como medidor de energia elétrica. Em ambos experimentos, foram utilizados os circuitos descritos na figura 3.50, onde R representa a lâmpada incandescente e L e C são uma associação em paralelo, respectivamente, dos três indutores e dos três capacitores descritos no item 3.2.2.14 Demais Materiais. Em relação aos equipamentos de medição, eles medem a tensão e a corrente presentes na carga, sendo que a forma de ligação ao circuito segue os procedimentos de cada dispositivo e não serão aqui indicados.

**Figura 3.50** Circuitos de medição laboratorial

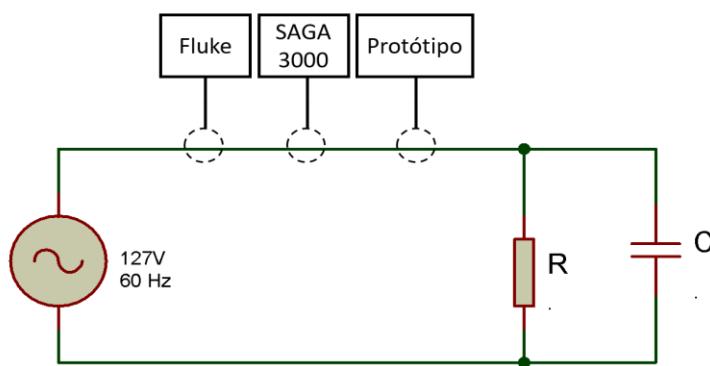
**1º Circuito**



**2º Circuito**



**3º Circuito**



Fonte: Autoria própria.

Cabe ressaltar que, sempre que o protótipo for utilizado para realizar medições, as suas portas USB devem estar desacopladas de qualquer cabo, pois caso contrário, há a possibilidade que isto altere os resultados das medições.

### 3.3.4.1 Calibração do Protótipo

Este experimento consistiu em obter e inserir três valores de calibração no protótipo, um para a tensão, outro para corrente e o último para o fator de potência, aplicando o método descrito no item 1.2.1.3 Calibração da Medição e utilizando o SAGA como medidor de referência.

### 3.3.4.2 Validação do Protótipo

Este experimento consistiu em comparar os dados de medição obtidos pelo protótipo com aqueles adquiridos pelo SAGA3000 e pelo Fluke. Os dados comparados são a energia ativa absorvida pela carga ao longo do tempo e o correspondente consumo de energia ativa total registrado.

# CAPÍTULO IV

---

## RESULTADOS LABORATORIAS E DISCUSSÕES

### 4.1 Calibração do Protótipo

Para a calibração do circuito foram utilizadas três características de cargas, sendo elas:

- Carga 1: Carga puramente resistiva formada pela lâmpada incandescente de 500W/220V;
- Carga 2: Carga resistiva/indutiva formada pela lâmpada incandescente de 500W/220V e pelos três indutores 1240mH / 0,5A.
- Carga 3: Carga resistiva/capacitiva formada pela lâmpada incandescente de 500W/220V e pelos três capacitores de 20 uF/230VAC.

#### 4.1.1 Primeiro Circuito – Carga 1

Na tabela 4.1, estão descritos os dados obtidos na primeira medição para uma carga puramente resistiva.

**Tabela 4.1** Dados obtidos no primeiro circuito antes da calibração – Carga 1

Equipamento	Tensão (V)	Corrente (A)	Fator de potência
SAGA 3000	128,75	1,82	0,99
Protótipo	140,00	2,00	1
Diferença (%)	8,73	9.89	1

Fonte: Autoria própria.

Com base nos dados de medição do SAGA 3000, tabela 4.1, e com os valores de 154,40 e 161,4 de calibração, respectivamente, de tensão e corrente do protótipo, foram obtidas as novas calibrações de 142 e 142,04, com base na equação 3.5. Em seguida, foi realizada nova medição, obtendo os dados descritos na tabela 4.2.

**Tabela 4.2** Dados obtidos no primeiro circuito após calibração – Carga 1

Equipamento	Tensão (V)	Corrente (A)	Fator de potência
SAGA 3000	128,75	1,82	0,99
Protótipo	128,84	1,76	1
Diferença (%)	0,07	-3,4	-1

Fonte: Autoria própria.

#### 4.1.2 Segundo Circuito – Carga 2

Na tabela 4.3, estão descritos os dados obtidos na primeira medição da carga com características resistiva/indutiva.

**Tabela 4.3** Dados obtidos no segundo circuito antes da calibração – Carga 2

Equipamento	Tensão (V)	Corrente (A)	Fator de potência
SAGA 3000	132,87	1,98	0,93
Protótipo	132,17	1,98	0,95
Diferença (%)	-0,53	0	2

Fonte: Autoria própria.

Com base nos dados de medição do SAGA 3000, tabela 4.3, e com o valor de 1,78 de calibração do fator de potência do protótipo, foi obtido a nova de calibração de 1.73, com base na equação 3.5. Em seguida, foi realizada nova medição, obtendo os dados descritos na tabela 4.4.

**Tabela 4.4** Dados obtidos no segundo circuito após calibração – Carga 2

Equipamento	Tensão (V)	Corrente (A)	Fator de potência
SAGA 3000	132,87	1,98	0,93
Protótipo	132,17	1,98	0,92
Diferença (%)	-0,53	0	1,08

Fonte: Autoria própria.

#### 4.1.3 Terceiro Circuito – Carga 3

Na tabela 4.5, estão descritos os dados obtidos na primeira medição da carga com característica resistiva/capacitiva.

**Tabela 4.5** Dados obtidos no terceiro circuito antes da calibração – Carga 3

Equipamento	Tensão (V)	Corrente (A)	Fator de potência
SAGA 3000	124,89	2,71	-0,64
Protótipo	125,86	2,58	0,66
Diferença (%)	0,78	4,79	2

Fonte: Autoria própria.

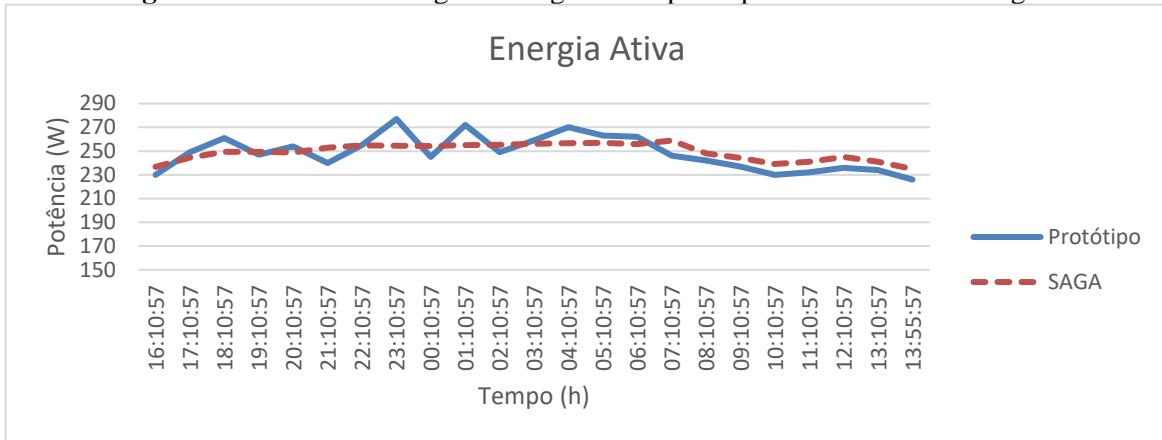
Com base nos dados de medição do SAGA 3000, tabela 4.5, e com o valor de 1,73 de calibração do fator de potência do protótipo, foi obtido a nova calibração de 1,67, com base na equação 3.5, porém este valor não foi inserido no protótipo, pois optou-se em manter a calibração anterior, a obtida no item 4.2.2 segundo circuito, já que as cargas predominantes no público alvo têm predomínio resistivo ou indutivo. Outro ponto a se notar, é a diferença de sinal do fator de potência, enquanto que no SAGA é negativo, no protótipo é positivo, demonstrando assim que na sua versão atual o protótipo não consegue diferenciar carga indutiva de capacitiva.

## 4.2 Validação do Protótipo

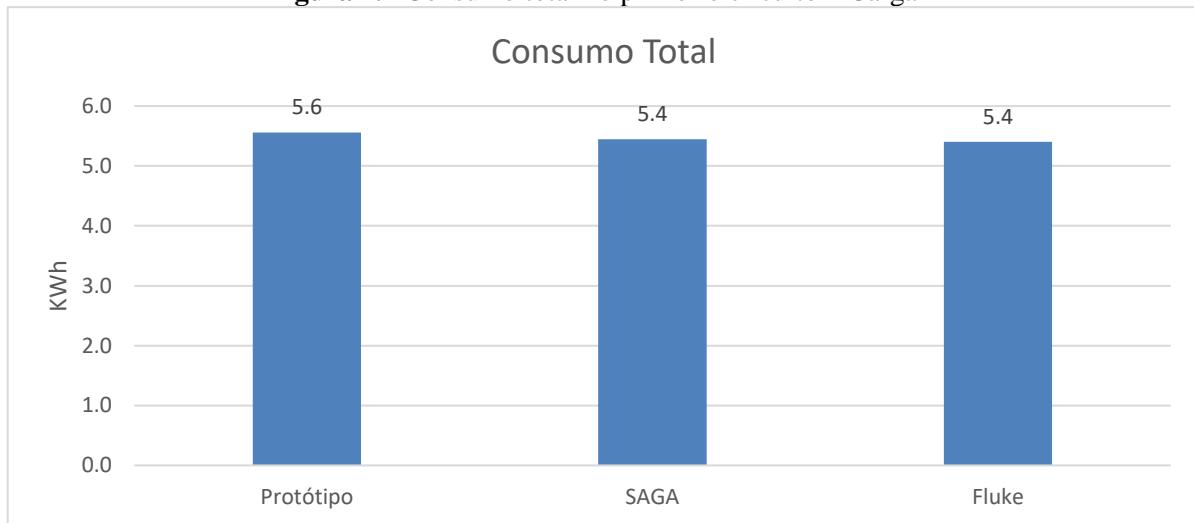
A validação foi realizada por meio da comparação da memória da massa do protótipo com a memória de massa do SAGA3000. Considerando que o equipamento Fluke 434 utilizado nesta validação não apresenta a funcionalidade de memória de massa, foi utilizado apenas o valor de consumo para validação com o protótipo. Todos os equipamentos foram zerados no início da medição.

### 4.2.1 Primeiro Circuito – Carga 1

A figura 4.1 mostra o perfil do consumo de energia ativa do circuito em estudo e a tabela a 4.1 fornece o correspondente consumo total de ativo medido pelos três medidores durante um período de 22 horas e 46 minutos.

**Figura 4.1** Gráfico de energia ao longo do tempo no primeiro circuito – Carga 1

Fonte: Autoria própria.

**Figura 4.2** Consumo total no primeiro circuito – Carga 1

Fonte: Autoria própria.

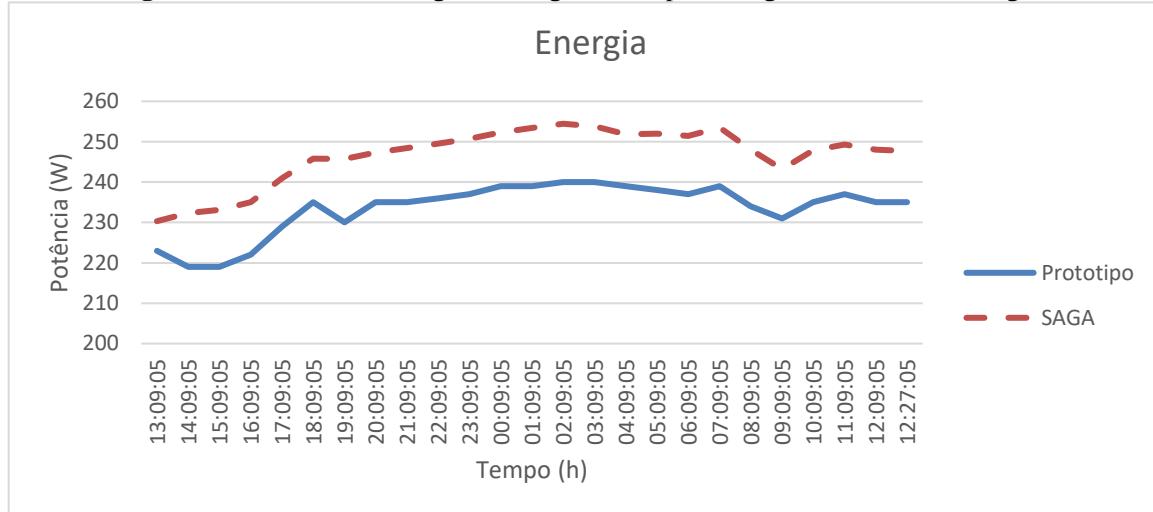
Observa-se na figura 4.1 que as medições registradas pelo protótipo apresentaram, em relação as medições do SAGA, períodos de perfil aproximado e em outros momentos um perfil oscilatório. Apesar disto, demonstra-se na figura 4.2 que o consumo total medido entre os três aparelhos foi próximo, com diferença de 0,2 kwh entre o protótipo e os demais aparelhos, correspondendo a uma diferença percentual de 3,7%.

Realizando uma estimativa para 30 dias, o consumo do protótipo e os demais aparelhos seria, de maneira aproximada, 177,1 e 170,7 KWh, portanto, uma diferença de 6,4 kwh a mais para o protótipo.

#### 4.2.2 Segundo Circuito – Carga 2

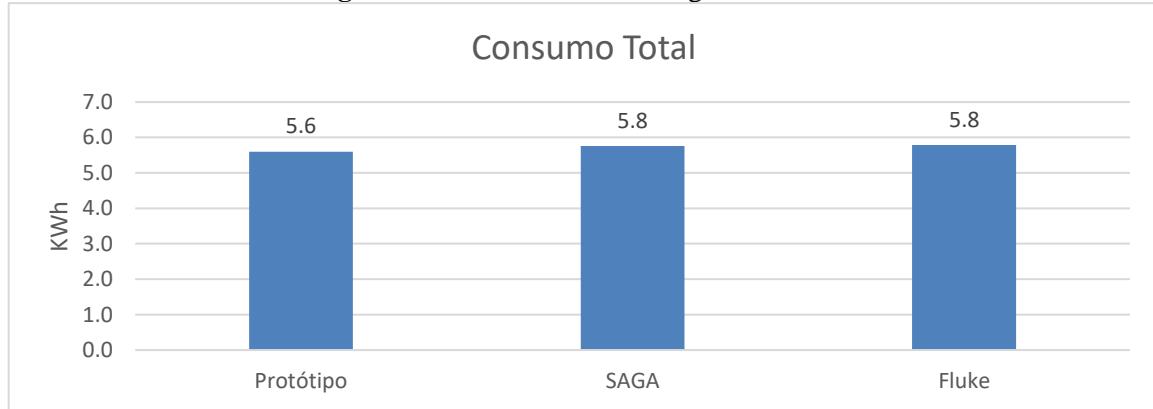
A figura 4.3 mostra o perfil do consumo de energia ativa do circuito em estudo e a figura 4.4 fornece o correspondente consumo total de ativo medido pelos três medidores durante um período de 23 horas e 19 minutos.

**Figura 4.3** Gráfico de energia ao longo do tempo no segundo circuito – Carga 2



Fonte: Autoria própria.

**Figura 4.4** Consumo total no segundo circuito



Fonte: Autoria própria.

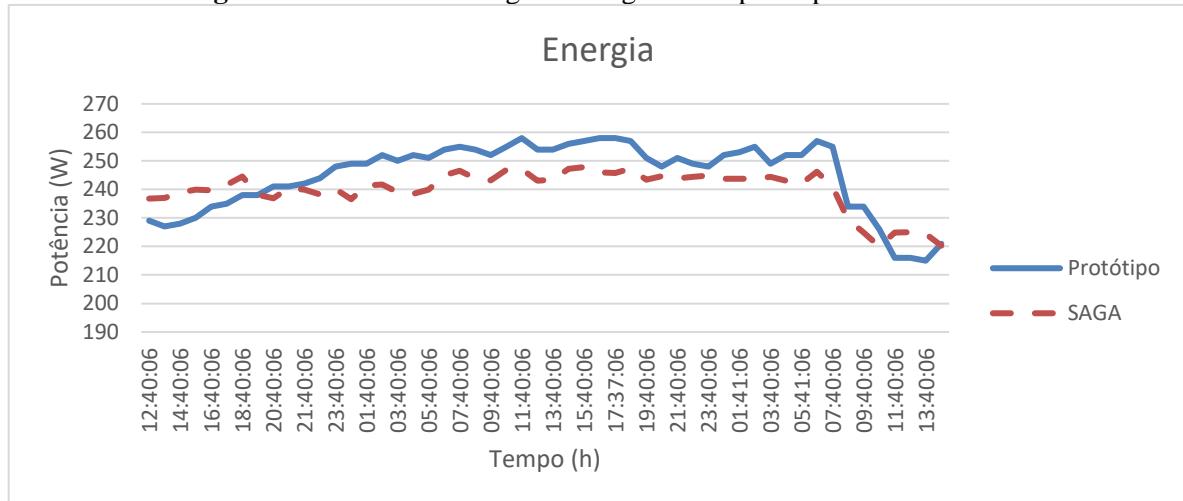
Nota-se na figura 4.3 que as medições registradas pelo protótipo apresentaram, em relação às medições do SAGA, um perfil com um formato próximo, contudo com valores inferiores aos do SAGA. Apesar disto, demonstra-se na figura 4.3 que o consumo total medido entre os três aparelhos foi próximo, com diferença de 0,2 kwh entre o protótipo e os demais aparelhos, correspondendo a uma diferença percentual de 3,7%.

Realizando uma estimativa para 30 dias, o consumo do protótipo e os demais aparelhos seria, de maneira aproximada, 172,9 e 179,1 KWh, portanto, uma diferença de 6,2 kwh a menos para o protótipo.

#### 4.2.3 Terceiro Circuito – Carga 3

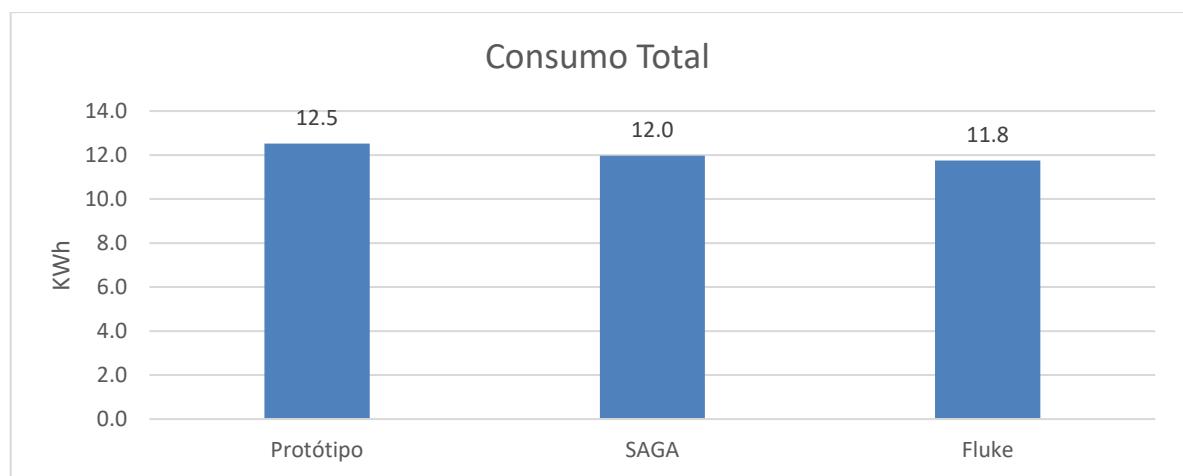
A figura 4.5 mostra o perfil do consumo de energia ativa do circuito em estudo e a figura 4.6 fornece o correspondente consumo total de ativo medido pelos três medidores durante um período de 49 horas e 37 minutos.

**Figura 4.5** Gráfico de energia ao longo do tempo no primeiro circuito



Fonte: Autoria própria.

**Figura 4.6** Consumo total do primeiro circuito



Fonte: Autoria própria.

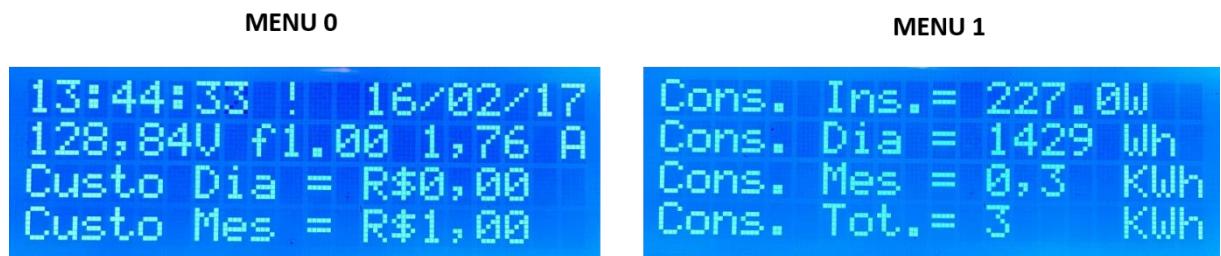
Observa-se na figura 4.6 que as medições registradas pelo protótipo apresentaram, em relação as medições do SAGA, um perfil com um formato próximo, contudo com valores ora inferiores ora superiores aos do SAGA. Apesar disto, demonstra-se na figura 4.7 que o consumo total medido pelo protótipo apresentou diferenças de 0,5 kwh e 0,7 kwh em relação ao SAGA e ao Fluke, correspondendo a uma diferença percentual de, respectivamente, 4,16 % e 5,9%.

Realizando uma estimativa para 30 dias, o consumo do protótipo, do SAGA e do Fluke seria, de maneira aproximada, respectivamente, 181,39 KWh, 174,13 KWh e 171,23 KWh, portanto, uma diferença, em relação ao SAGA e ao Fluke, de 7,26 e 10,16 kwh, ambas para mais.

#### 4.3 Demais Resultados

Este item trata-se inicialmente dos resultados referente as outras funções do protótipo obtidos durante o primeiro e segundo experimento. A primeira a se notar é o acesso aos menus do protótipo que ocorreram sem problemas. Por meio da figura 4.7 é possível visualizar os principais dados que a tela do dispositivo disponibiliza para o usuário.

**Figura 4.7** Telas de informação do protótipo



Fonte: Autoria Própria

Para um entendimento mais detalhado do display do protótipo, pode-se descrever as seguintes variáveis como:

- ! = Indicação de memória de massa ativada;
- # = Indicação de envio de dados para servidor WEB. O ícone é localizado ao lado direito da indicação “!”, não demonstrado a figura 4.7;
- f/+- = Indicação do fator de potência. Para “f” alternando com “+” o fator de potência é correspondente a uma carga indutiva, e para “f” alternando com “-“, a carga apresenta característica capacitiva;
- V = Indicação de Vrms;

- I = Indicação de Irms;
- Custo Dia = Indicação do custo do consumo de energia elétrica realizado no dia a partir das 0h:00m;
- Cons. Mês = Indicação do custo do consumo de energia elétrica realizado no mês a partir das 0h:00m do dia selecionado no menu “Alterar dia do ciclo de medição mensal”, indicado no apêndice A.2;
- Cons. Ins.= Indicação da potência ativa instantânea absorvida pela carga;
- Cons. Dia = Indicação do consumo de energia elétrica realizado no dia a partir das 0h:00m;
- Cons. Mês = Indicação do consumo de energia elétrica realizado no mês a partir das 0h:00m do dia selecionado no menu “Alterar dia do ciclo de medição mensal”, indicado no apêndice A.2;
- Cons. Total = Indicação do consumo de energia elétrica total registrado.

O “custo mês” e “custo dia” apesar de possuir indicação de centavos, eles estão indicados na presente versão como 00, pois apesar do programa calcular os valores corretos, a inserção destes valores precisa ser programada.

Quanto a função de armazenamento de dados no cartão SD, o mesmo ocorreu como planejado, por meio da figura 4.8 é possível visualizar um segmento extraído do cartão.

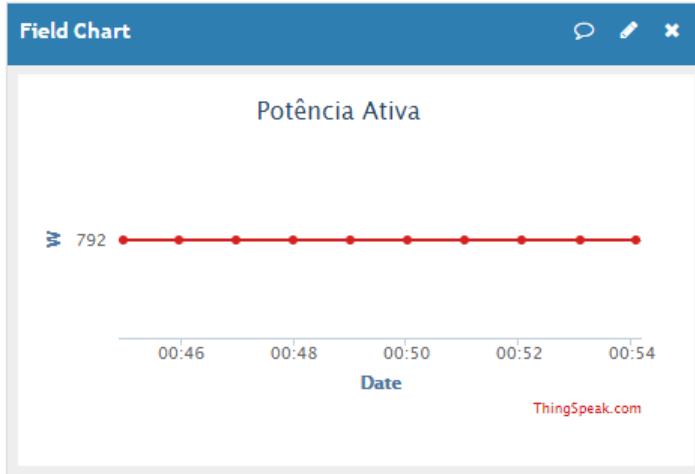
**Figura 4.8** Segmento de dados armazenado no cartão SD

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1														
2														
3	///////////													
4	Medidor_de_Energia	V.xx												
5														
6	Data_da_Letura:	15/02/2017												
7	Hora_da_Letura:	16:10:57												
8	Intervalo_de_Integração:	60 s												
9														
10	Data	Horario	Demandra(W)	Tensão(V)	Corrente(A)	FP	SFP(1-cap,2-ind)	Consumo(W)						
11	15/02/2017	16:10:57	230	130.1	1.76	1	2	13800						
12	15/02/2017	16:11:57	231	130.22	1.8	1	2	27660						
13	15/02/2017	16:12:57	231	130.04	1.77	1	2	41520						
14	15/02/2017	16:13:57	230	130.23	1.78	1	2	55320						
15	15/02/2017	16:14:57	230	130.31	1.77	1	2	69120						
16	15/02/2017	16:15:57	229	130.08	1.76	1	2	82860						
17	15/02/2017	16:16:57	228	130.14	1.77	1	2	96540						
18	15/02/2017	16:17:57	228	130.43	1.76	1	2	110220						
19	15/02/2017	16:18:57	229	130.44	1.78	1	2	123960						
20	15/02/2017	16:19:57	229	130.58	1.77	1	2	137700						
21	15/02/2017	16:20:57	229	130.4	1.78	1	2	151440						
22	15/02/2017	16:21:57	229	130.36	1.75	1	2	165180						
23	15/02/2017	16:22:57	229	130.44	1.77	1	2	178920						
24	15/02/2017	16:23:57	229	130.4	1.77	1	2	192660						
25	15/02/2017	16:24:57	229	130.42	1.78	1	2	206400						
26	15/02/2017	16:25:57	230	130.61	1.77	1	2	220200						

Fonte: Autoria Própria

Em relação ao envio de dados para um servidor WEB, não foi possível realizarlo durante os dois experimentos, pois não havia local para conectar à internet, porém na figura 4.9 é possível visualizar a aparência de um gráfico obtido por esta função dias antes.

**Figura 4.9** gráfico no canal do ThingSpeak



Fonte: Autoria Própria

O custo aproximado de uma unidade do protótipo levando-se em conta somente o custo material está descrito na tabela 4.6, onde é possível visualizar os itens orçados, sendo que a fonte de alimentação deste trabalho foi substituída por uma de modelo comercial e a placa de conexão, placa de controle, cabo V foram orçados os itens componentes que os compõe.

**Tabela 4.6** Custo material de uma unidade do protótipo

Material	Quantidade	Preço
Arduino Uno [20]	1	R\$ 29.50
Arduino Mega [21]	1	R\$ 44.00
Display LCD [22]	1	R\$ 42.00
Módulo SD [23]	1	R\$ 15.00
Base de Acrílico	1	R\$ 5.00
Placa de Controle [24]	1	R\$ 10.00
Placa de Conexão [24]	1	R\$ 10.00
Sensor de Corrente [25]	1	R\$ 48.00
Cabo V [24]	2	R\$ 6.00
Ethernet Shield [26]	1	R\$ 39.00
Cabos em geral [24]	1	R\$ 10.00
Fonte de Alimentação [27]	1	R\$ 5.00
<b>Total</b>		R\$ 234.00

Fonte: Autoria Própria

# CAPÍTULO V

---

## CONCLUSÕES

Ao analisar os resultados, nota-se que para cargas com características puramente resistiva e cargas com características resistiva/indutiva, o protótipo registrou o consumo de ativo com a mesma diferença percentual em relação aos outros equipamentos, demonstrado assim, certa regularidade nas medições. Porém, para cargas com característica resistiva/capacitativa, o protótipo não seguiu este padrão e elevou esta diferença. Ademais, para cada tipo de carga, as medições do protótipo possuíam um perfil próprio, seja ela acompanhando o perfil das medições do SAGA, ou apresentando medições com intensidade maior e em outros momentos de intensidades menores. Demonstrando assim, que o processo de medição das grandezas elétricas deve ser melhorado, porém, por ser a primeira versão, o protótipo apresentou resultados plausíveis.

Com relação a validação do protótipo para o período de um mês de medição, não foi possível tal realização devido ao curto período de tempo disponível para a finalização deste trabalho. Levando em consideração as características dos três tipos de cargas, seria necessários 90 dias para tal amostragem. Portanto, levou-se em consideração uma estimativa do erro de medição para um período de 30 dias.

Considerando ainda a falta de tempo, cada carga foi mensurada apenas uma vez, não sendo possível uma recalibração e nova aferição. Contudo, os resultados obtidos até então foram satisfatórios considerando a proposta deste trabalho que era desenvolver um medidor de baixo custo.

Pôde-se constatar que todas as funções do protótipo funcionaram conforme o planejado. Em relação aos programas desenvolvidos, eles encontram-se organizados e comentados, assim, fornecendo suporte para outros pesquisadores que queiram ingressar nesta mesma linha de pesquisa ou reproduzi-lo.

Como última observação, ressalte-se que o reconhecimento do sucesso da proposta deva-se atentar para o fato que, até a plena consolidação do protótipo, certamente outros aspectos ainda se fazem necessários e merecedores de maiores investigações. Destacam-se os seguintes pontos focais como sugestões para trabalhos futuros:

- Melhorar o processo de medição das grandezas elétricas, que pode ser obtido aperfeiçoando a placa de conexão com componentes melhores, melhorando a resolução de entrada de sinal no Uno, experimentar outros microcontroladores no lugar do Uno que possuam comunicação I2c.
- Desenvolver um involucro que armazene todos os módulos que compõe o protótipo, assim, tornando-o mais prático de uso.
- Realizar novas medições em períodos maiores e com cargas com fator de potência variável, para que assim, se possa avaliar melhor as medições do protótipo.
- Implementar e programar um módulo Wi-Fi ao protótipo.
- Criar um aplicativo para *Smartphone* que possibilite ao usuário monitorar o seu consumo de energia elétrica, sendo que os dados de consumo podem ser obtidos do servidor da Thingspeak.

## REFERÊNCIAS

- [1] EPE. **Projeção da demanda de energia elétrica:** para os próximos 10 anos (2015-2024). Rio de Janeiro, 2015. 78 p. (Estudos da demanda). Disponível em: <<http://www.epe.gov.br>>. Acesso em: 19 mar. 2017.
- [2] ARDUINO. **Introduction.** Disponível em:  
<<https://www.arduino.cc/en/Guide/Introduction#>>. Acesso em: 19 mar. 2017.
- [3] ARDUINO. **HOME.** Disponível em: <<https://www.arduino.cc/>>. Acesso em: 19 mar. 2017.
- [4] ARDUINO. **“What is Arduino?”.** Arduino. Disponível em:  
<<https://www.arduino.cc/en/Guide/Introduction>> Acesso em: 17 mar. 2017.
- [5] OTOMURA, Sólon Sadamitsu. **Uma proposta de eficiência energética através do controle e Supervisão de condicionadores de ar, utilizando a plataforma de hardware livre Arduino.** 2015. 129 f. TCC (Graduação) - Curso de Engenharia Elétrica, Engenharia Elétrica, Universidade Federal de Mato Grosso, Cuiabá, 2015.
- [6] ROBOCORE. **Loja virtual.** Disponível em: <<https://www.robocore.net>>. Acesso em: 19 mar 2017.
- [7] FILIPEFLOP. **Loja virtual.** Disponível em: <<http://www.filipeflop.com/>>. Acesso em: 19 mar 2017.
- [8] CCK. **Loja virtual.** Disponível em: <<http://www.cck.com.br/sistema/drivers.php>>. Acesso em: 19 mar 2017.
- [9] FLUKE. **Loja virtual.** Disponível em: <<http://www.fluke.com/>>. Acesso em: 19 mar 2017.
- [10] LEA, Trystan. **Learn.** Disponível em:

<<https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/introduction>>. Acesso em: 19 mar. 2017.

[11] LEA, Trystan. **EmonLib**. Disponível em:  
<<https://github.com/openenergymonitor/EmonLib>>. Acesso em: 19 mar. 2017.

[12] LEA, Trystan. AC Power Theory: **Arduino Maths**. Disponível em:  
<<https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/arduino-maths>>. Acesso em: 19 mar. 2017.

[13] LEA, Trystan. **AC Power Theory: Advanced Maths**. Disponível em:  
<<https://learn.openenergymonitor.org/electricity-monitoring/ac-power-theory/advanced-maths>>. Acesso em: 19 mar. 2017.

[14] ARDUINO. **Master Reader/Slave Sender**. Disponível em:  
<<https://www.arduino.cc/en/Tutorial/MasterReader>>. Acesso em: 19 mar. 2017.

[15] THINGSPEAK. **Understand Your Things**. Disponível em: <<https://thingspeak.com/>>. Acesso em: 19 mar. 2017.

[16] THINGSPEAK. **Send Data to ThingSpeak with Arduino**. Disponível em:  
<<http://community.thingspeak.com/tutorials/arduino/send-data-to-thingspeak-with-arduino>>. Acesso em: 19 mar. 2017.

[17] Mecanismo online para referências, versão 2.0. Florianópolis: UFSC Rexlab, 2013.  
Disponível em: <<http://www.more.ufsc.br/>>. Acesso em: 19 mar. 2017.

[18] ARDUINO. **Arduino Ethernet Shiled V1**. Disponível em:  
<<https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>>. Acesso em: 19 mar. 2017.

[19] GITHUB. **VIP**. Disponível em: <<https://github.com/jacksonsandrade/VIP>>. Acesso em: 23 mar. 2017.

[20] MERCADOLIVRE. **Loja Virtual**.

Disponível em: <<http://lista.mercadolivre.com.br/arduino-uno>>. Acesso em: 23 mar. 2017.

**[21] MERCADOLIVRE. Loja Virtual.**

Disponível em: <<http://lista.mercadolivre.com.br/arduino-uno>>. Acesso em: 23 mar. 2017.

**[22] MERCADOLIVRE. Loja Virtual.**

Disponível em: <[http://lista.mercadolivre.com.br/display-lcd-20x4-i2c#D\[A:display-lcd-20x4-i2c\]](http://lista.mercadolivre.com.br/display-lcd-20x4-i2c#D[A:display-lcd-20x4-i2c])>. Acesso em: 23 mar. 2017.

**[23] MERCADOLIVRE. Loja Virtual.**

Disponível em: <[http://lista.mercadolivre.com.br/cart%C3%A3o-memoria-2g#D\[A:cartão-memoria-2g\]](http://lista.mercadolivre.com.br/cart%C3%A3o-memoria-2g#D[A:cartão-memoria-2g])>. Acesso em: 23 mar. 2017.

**[24] PROESI. Loja Virtual.** Disponível em: <<http://www.proesi.com.br/>>. Acesso em: 23 mar. 2017.

**[25] MERCADOLIVRE. Loja Virtual.**

Disponível em: <[http://lista.mercadolivre.com.br/sensor-de-corrente-sct-013-30a#D\[A:sensor-de-corrente-sct-013-30a\]](http://lista.mercadolivre.com.br/sensor-de-corrente-sct-013-30a#D[A:sensor-de-corrente-sct-013-30a])>. Acesso em: 23 mar. 2017.

**[26] MERCADOLIVRE. Loja Virtual.**

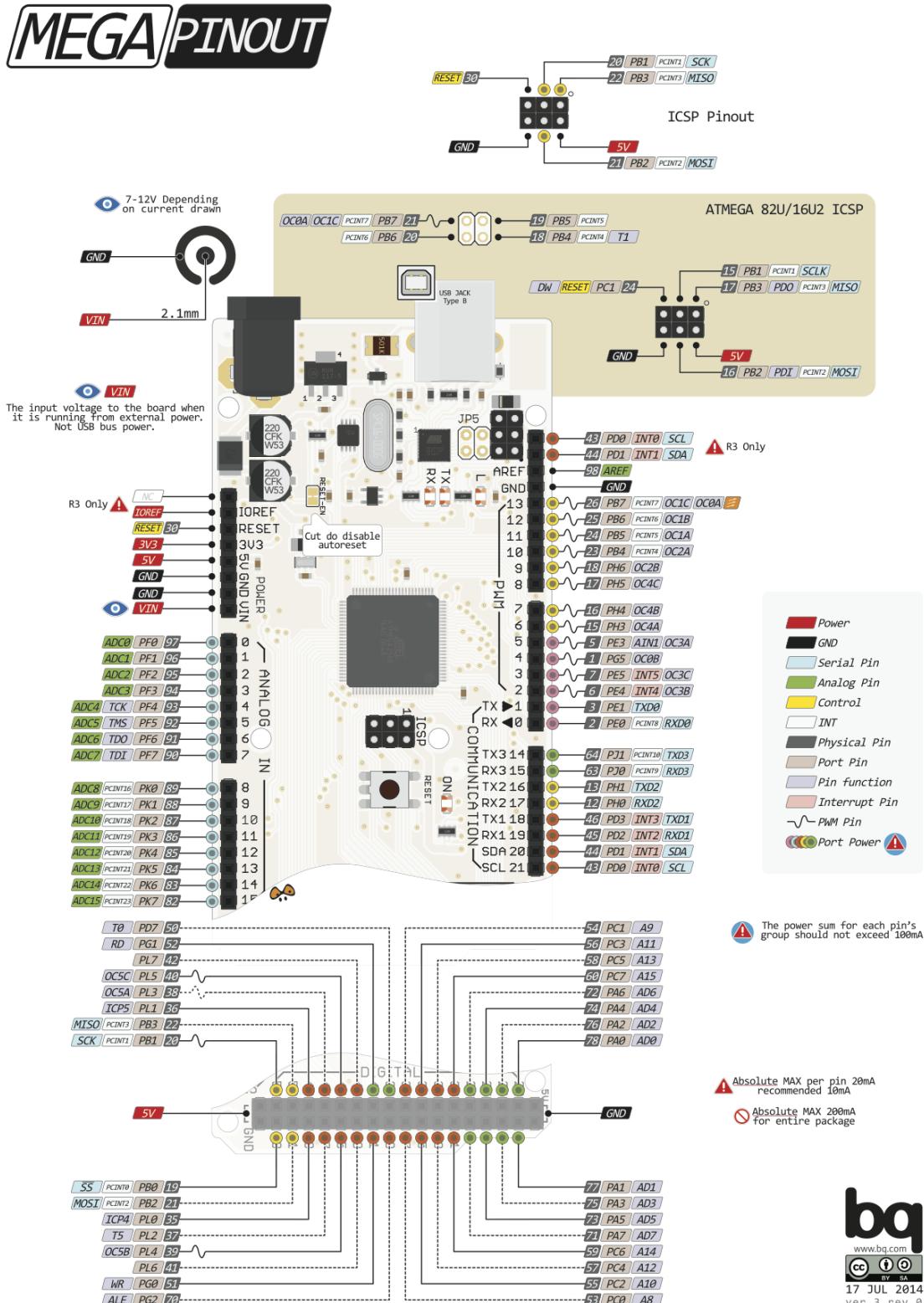
Disponível em: <[http://lista.mercadolivre.com.br/ethernet-shield#D\[A:ethernet-shield\]](http://lista.mercadolivre.com.br/ethernet-shield#D[A:ethernet-shield])>. Acesso em: 23 mar. 2017.

**[27] MERCADOLIVRE. Loja Virtual.**

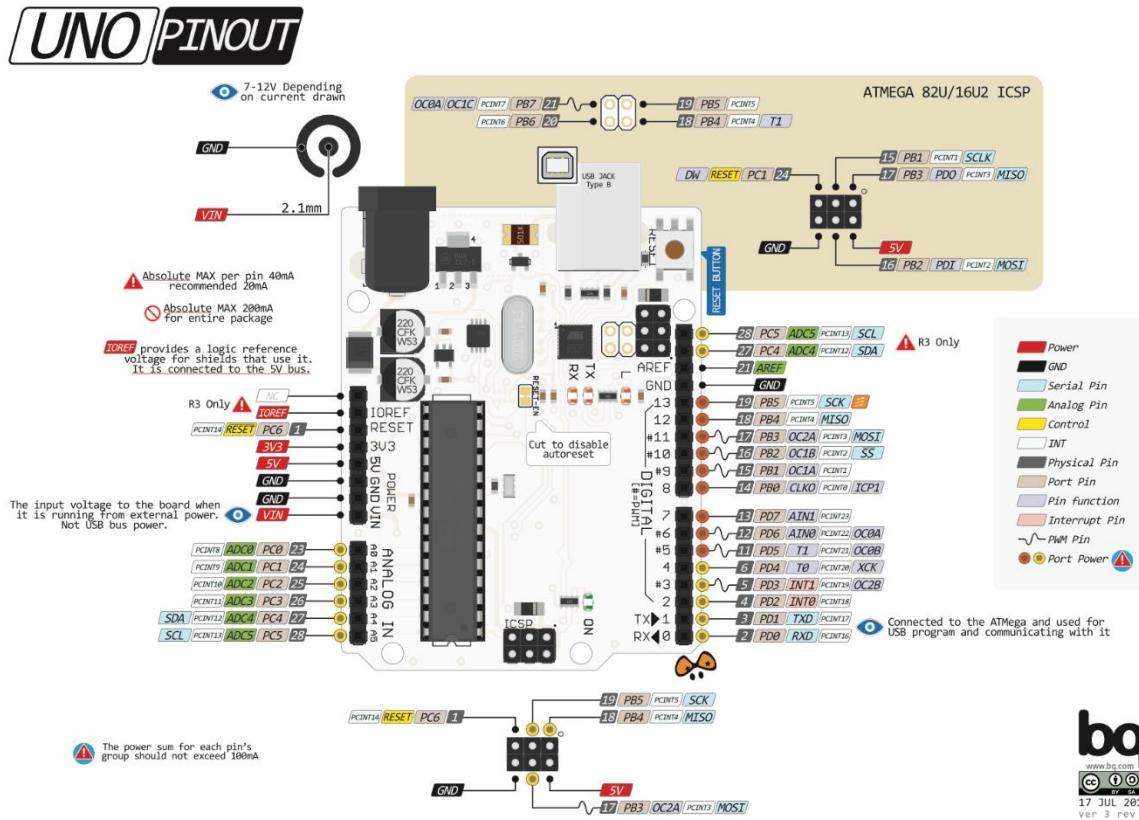
Disponível em: <[http://lista.mercadolivre.com.br/fonte-12v-1a#D\[A:fonte-12v-1a\]](http://lista.mercadolivre.com.br/fonte-12v-1a#D[A:fonte-12v-1a])>. Acesso em: 23 mar. 2017.

# ANEXOS

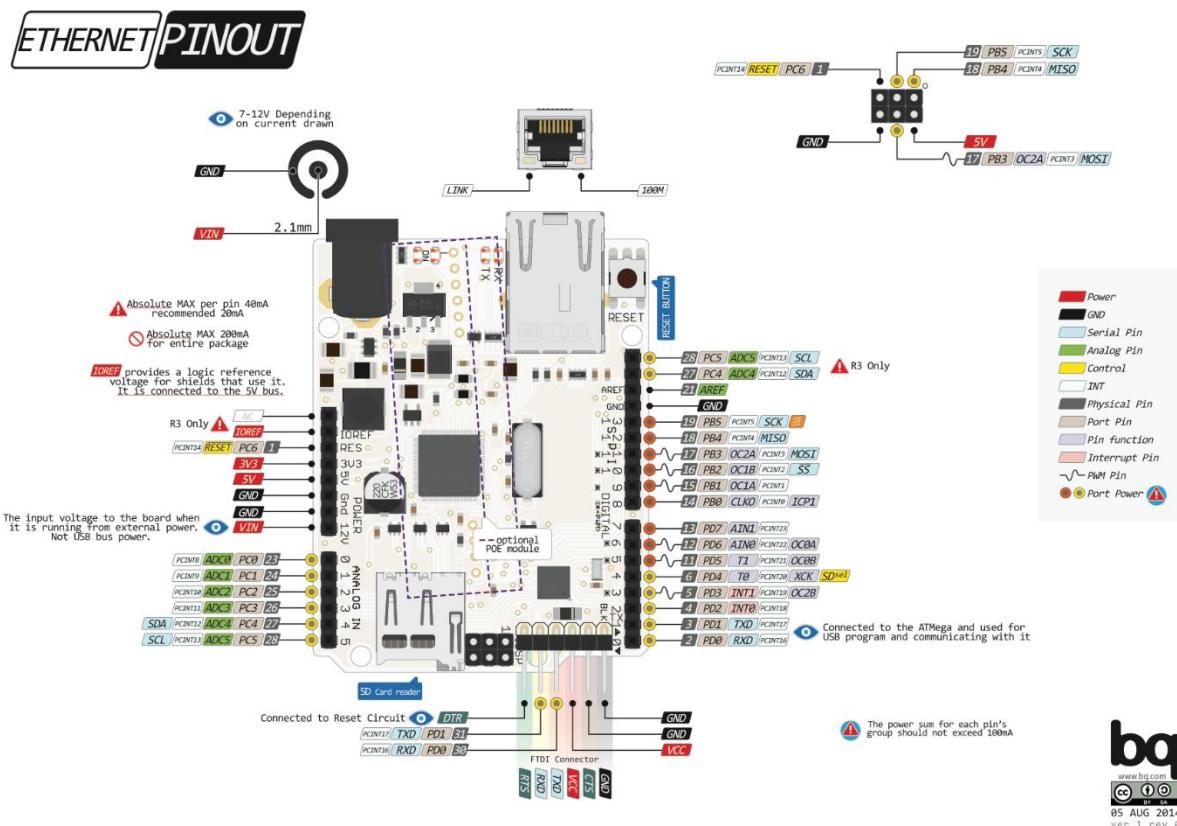
## ANEXO A.1



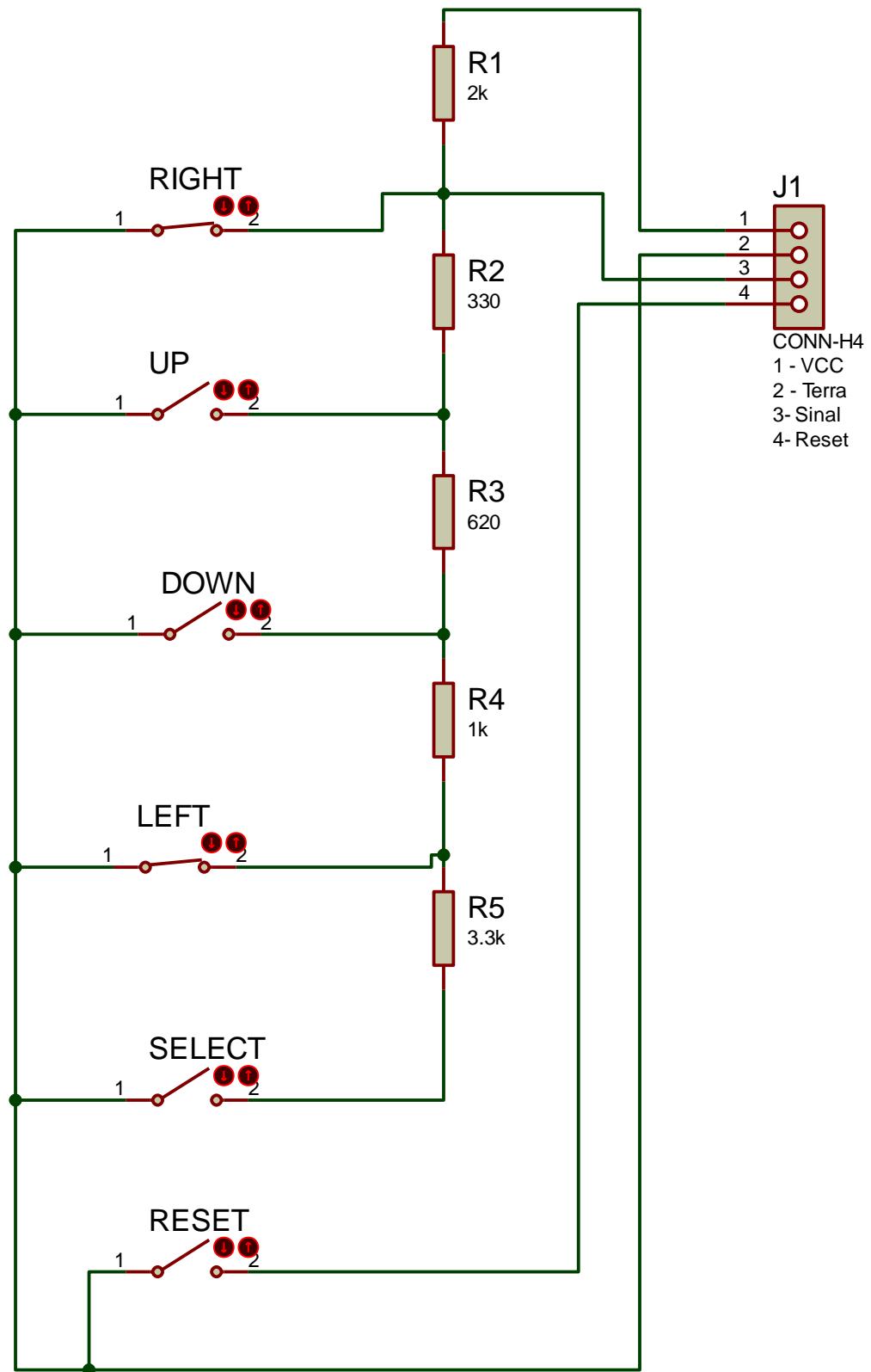
## ANEXO A.2



## ANEXO A.3



## ANEXO A.4



## ANEXO A.5



## Split-Core Current Transformer

## Model: SCT-013 Series

**Characteristics:** Opening size: 13mm×13mm

Non-linearity $\pm 3\%$  ( 10%—120% rated input current )

1.5m leading wire. Diagram for standard three-output

Current output type or Voltage output type

( Voltage output type built-in sampling resistor )

**Purpose:** Suitable for the current measuring

monitoring and protection of AC motor

lighting equipment air compressor and so on

**Core material:** Ferrite

**Mechanical strength:** The number of switching is not less than 1000 times

(Test under 20°C).

**Safety index:** Dielectric strength 6000V AC/1min

**Fire resistance property:** in accordance with UL94-VO

**Working temperature:** -25°C—+70°C

**Outline size diagram:** (in mm)

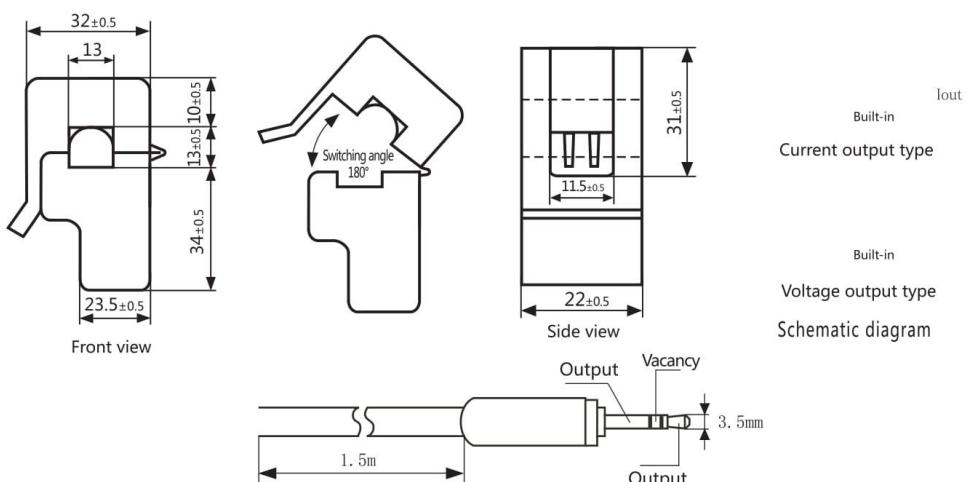


Table of technical parameters:

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0~100A	0~5A	0~10A	0~15A	0~20A
Output mode	Current/33mA	Voltage/1V	Voltage/1V	Voltage/1V	Voltage/1V
Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-070
Input current	0~25A	0~30A	0~50A	0~60A	
Output mode	Voltage/1V	Voltage/1V	Voltage/1V	Voltage/1V	

**Output mode:** Voltage output type built-in sampling resistor; Current output type built-in protective diode;  
Forbidden to be opening operating for current type.

## ANEXO A.6

### Indutor

Dados de Placa

Marca: Ultra Sinus

Entrada – Indutância: 1240mH

Entrada – Corrente: 0,5A

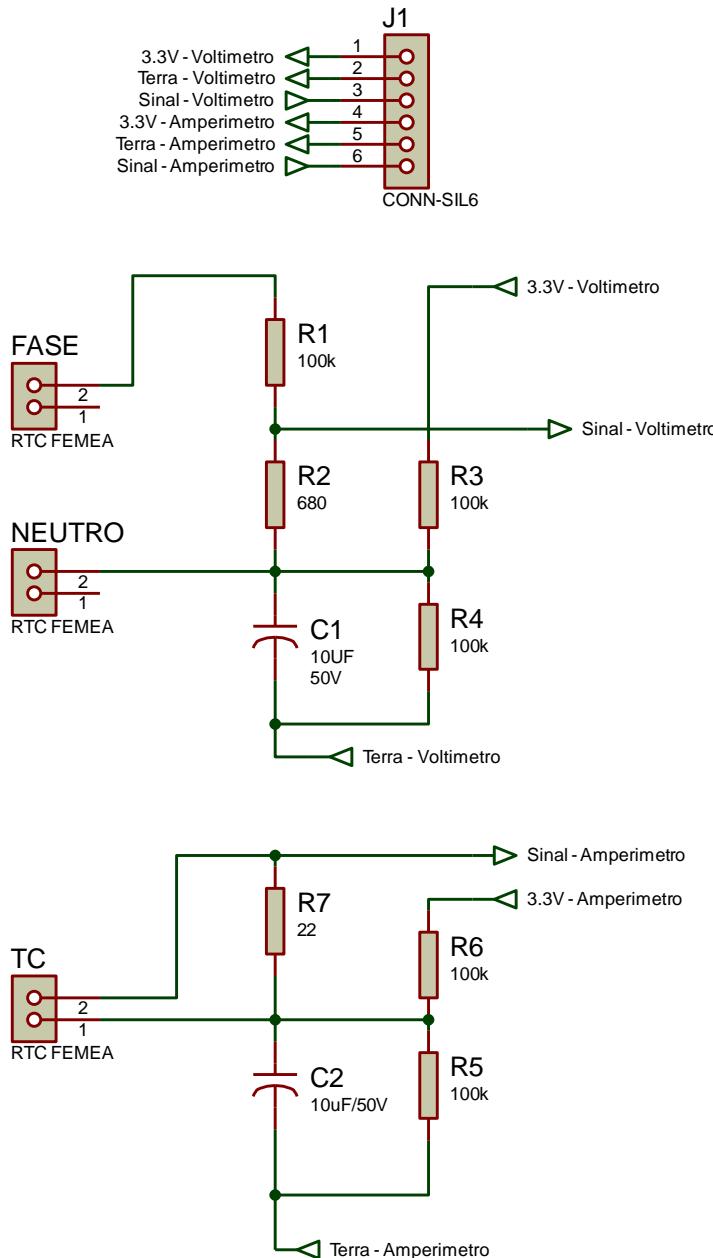
Nº A

CGC 60.747.870/0001-50 Industria Brasileira.



## APÊNDICES

### APÊNDICE A.1



## APÊNDICE A.2

