



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS SOBRAL - CE
CURSO DE ENGENHARIA DA COMPUTAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO

RAUL DE VASCONCELOS BOTELHO

MEDIDOR DO CONSUMO DE ENERGIA

Sobral

2018

RAUL DE VASCONCELOS BOTÊLHO

MEDIDOR DO CONSUMO DE ENERGIA

Monografia apresentada ao Curso de Engenharia da Computação no Campus de Sobral da Universidade Federal do Ceará como exigência parcial para obtenção do Título de Engenheiro de Computação.

Orientador: Prof. Me. Reuber Regis de Melo
Coorientador: Eng. João Lucas de Oliveira Torres

Sobral
2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

B764m Botêlho, Raul de Vasconcelos.

Medidor do Consumo de Energia / Raul de Vasconcelos Botêlho. – 2018.
68 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral,
Curso de Engenharia da Computação, Sobral, 2018.

Orientação: Prof. Me. Reuber Regis de Melo.

Coorientação: Prof. João Lucas de Oliveira Torres.

1. Consumo de Energia. 2. Automação Residencial. 3. Internet of Things. 4. Sistemas Embarcados. I.
Título.

CDD 621.39

RAUL DE VASCONCELOS BOTÊLHO

MEDIDOR DO CONSUMO DE ENERGIA

Monografia apresentada ao Curso de Engenharia da Computação no Campus de Sobral da Universidade Federal do Ceará como exigência parcial para obtenção do Título de Engenheiro de Computação.

Orientador: Prof. Me. Reuber Regis de Melo
Coorientador: Eng. João Lucas de Oliveira Torres

Aprovada em: ___ / ___ / ____.

BANCA EXAMINADORA

Prof. Me. Reuber Regis de Melo (Orientador)
Universidade Federal do Ceará (UFC)

Eng. João Lucas de Oliveira Torres (Coorientador)

Prof. Me. David Nascimento Coelho
Universidade Federal do Ceará (UFC)

Aos meus pais, Everson e Romana.

AGRADECIMENTOS

A construção deste trabalho só foi possível por conta da colaboração, estímulo e compreensão de diversas pessoas. Agradeço primeiramente à minha família pela grande compreensão, companheirismo para comigo e sempre me apoiaram em seguir fazendo o que gosto. Em especial agradeço minha mãe Maria Romana de Vasconcelos que dedicou seu árduo trabalho e amor para investir no meu crescimento e não mediu esforços para me dar uma boa educação. Agradeço também ao meu primo Hygor Melo e minha tia Maria Socorro Vasconcelos pela boa convivência e apoio no dia a dia. À Thiago Araujo pelos conselhos e inspirações que me fizeram decidir este curso.

Gostaria de agradecer a minha namorada Naianna Melo pelo amor, amizade, compreensão e cuidado durante todo meu percurso ao longo destes 5 anos na universidade e muito além. Agradeço pela ajuda de sempre, com esforço máximo mesmo quando não ter o que oferecer além do apoio e carinho.

Aos meus amigos que ao longo destes anos me acompanharam e suportaram de diversas maneiras sempre na medida do possível e às vezes do impossível. Em especial destaco José Aragão da Silva pelo estímulo constante a respeito deste trabalho e ao quinteto fantástico (Alysson, Tamy, Reno e Danilo) pelo apoio ao longo de mais de 10 anos. À Matheus Gomes por me incentivar na área de tecnologia dès da minha infância.

Ao Colégio Sant'Ana e toda sua equipe por me propiciar uma formação de ensino fundamental e médio de qualidade.

Deixo minha gratidão à 9ª turma de Engenharia da Computação de Sobral pela parceria ao longo destes anos de muita luta e dedicação no curso diante de todas as conquistas e derrotas.

Gostaria de agradecer a Renan Carneiro pela imensa parceria durante os períodos mais difíceis desse curso, em conjunto com toda turma de estudo da Engenharia Elétrica que me acompanhou e apoiou em noites de café com leite e suco de umbu-cajá (Pedro Matheus, Felipe Davi, Lucas Aguiar, Thais, Ildenor e João Gomes).

Agradeço também a João Lucas Torres por, além de toda tutela e construção da idealização deste trabalho, me aconselhar e potencializar meus conhecimentos.

À Grendene S/A por dar disponibilidade e incentivo a realizar a conclusão deste projeto e aos meus colegas de trabalho, em especial a minha equipe da Célula de Pesquisa e Desenvolvimento.

Sou grato a Universidade Federal do Ceará, Campus Sobral, com um agradecimento especial aos professores Carlos Elmano, Iális Cavalcante e Reuber Regis por toda orientação ao longo da minha formação. Agradeço também ao Prof. Marco Antônio Toledo e todo grupo PesquisaMus pela imensa aprendizagem diante das experiências vivenciadas.

RESUMO

O mundo tem passado por grandes mudanças tecnológicas trazendo entre diversas outras melhorias, um maior conforto e segurança para nosso lar através da automação residencial. Porém é preciso estar consciente da quantidade de energia elétrica que é consumida. Utilizando conhecimentos de sistemas embarcados e conceitos como *Internet of Things*, o presente trabalho propõe um projeto de construção de um medidor de consumo de energia, onde é possível coletar informações de tensão e corrente de equipamentos e exibir seus valores através de uma página HTML em uma rede local.

Palavras-chave: Consumo de energia. Automação residencial. *Internet of Things*. Sistemas Embarcados.

ABSTRACT

The world underwent major technological changes bringing among many other improvements, greater comfort and safety for our home through residential automation. But you have to be aware of the amount of electricity that is consumed. Using knowledge of embedded systems and concepts such as Internet of Things, the present work proposes a project to build an energy consumption meter, where it is possible to collect voltage and current information from equipment and display its values through an HTML page in a local network..

Keywords: Energy consumption. Home automation. *Internet of Things*. Embedded systems.

LISTA DE FIGURAS

Figura 1 – Composição tarifária das distribuidoras.....	20
Figura 2 – Entradas e saídas do Arduino Nano.....	23
Figura 3 – Entradas e saídas do ESP8266MOD.....	25
Figura 4 – Esquema de comunicação SPI entre um mestre e dois escravos.....	28
Figura 5 – Diagrama básico de um circuito elétrico.....	31
Figura 6 – Representação dos valores pontuais de tensão para instantes de tempo.....	35
Figura 7 – Arquitetura geral do medidor de consumo.....	37
Figura 8 – Etapas de implementação das funcionalidades.....	39
Figura 9 – Circuito de alimentação.....	39
Figura 10 – Formas de onda da ponte retificadora.....	40
Figura 11 – Circuito de comunicação.....	42
Figura 12 – Fluxograma do algoritmo de funcionamento da coleta de dados.....	43
Figura 13 – Ferramenta Serial Plotter do Arduino.....	44
Figura 14 – Fluxograma de operações sobre sinal de A2.....	44
Figura 15 – Forma de onda do sinal de A2.....	45
Figura 16 – Diagrama de blocos do circuito de alimentação.....	46
Figura 17 – Representação da troca de informações ordenada por SPI.....	47
Figura 18 – Diagrama com as conexões para comunicação SPI entre os SoCs.....	48
Figura 19 – Fluxograma do Web Server.....	49
Figura 20 – Pagina HTML exibindo o arquivo data.txt.....	50
Figura 21 – Protótipo do projeto.....	51
Figura 22 – Página HTML exibindo as informações da carga.....	52
Figura 23 – Interface da função de leitura de tensão e corrente RMS do ET-5060C.....	53
Figura 24 – Montagem do primeiro ensaio em bancada.....	54
Figura 25 – Montagem do segundo ensaio em bancada.....	55
Figura 26 – Montagem do terceiro ensaio em bancada.....	56

LISTA DE TABELAS

Tabela 1 – Lista de componentes do protótipo.....	38
Tabela 2 – Relação de bytes com informação.....	48
Tabela 3 – Resultados do primeiro ensaio.....	54
Tabela 4 – Resultados do segundo ensaio.....	55
Tabela 5 – Resultados do terceiro ensaio.....	56

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog Digital Converter</i>
AJAX	<i>Asynchronous Javascript and XML</i>
ANEEL	<i>Agência Nacional de Energia Elétrica</i>
CA	<i>Corrente Alternada</i>
CC	<i>Corrente Continua</i>
CPU	<i>Central Process Unit</i>
DIY	<i>Do It Yourself</i>
DSP	<i>Digital Signal Processor</i>
FTDI	<i>Future Technology Devices International</i>
GPIO	<i>General Purpose Input/Output</i>
GSM	<i>Global System for Mobile Communications</i>
HTML	<i>Hyper Text Markup Language</i>
IC	<i>Inter-Integrated Circuit</i>
PS	<i>Inter-IC Sound</i>
IC	<i>IntegredCircuit</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IEEE-SA	<i>Institute of Electrical and Electronics Engineers Standards Association</i>
IETF	<i>Internet Engineering Task Force</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol address</i>
ISO	<i>International Organization for Standardization</i>
ITU-T	<i>International Telecommunications Union Telecommunication Standardization Sector</i>
LED	<i>Light Emitting Diode</i>
MISO	<i>Master IN Slave OUT</i>
MOSI	<i>Master OUT Slave IN</i>
NOR	<i>Not OR</i>
OTP	<i>One Time Programmable</i>
PC	<i>Personal Computer</i>
PSTN	<i>Public switched telephone network</i>
PWM	<i>Pulse Width Modulation</i>

RAM	<i>Random Access Memory</i>
RF	Radio Frequência
RMS	<i>Root Mean Square</i>
ROM	<i>Read Only Memory</i>
RX	<i>Receiver</i>
SoC	<i>System-on-a-Chip</i>
SPI	<i>Serial Peripheral Interface bus</i>
SS	<i>Slave Select</i>
SSI	<i>Synchronous Serial Interface</i>
TTL	<i>Time To Live</i>
TV	<i>Television</i>
TX	<i>Transmitter</i>
UART	<i>Universal Synchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1.INTRODUÇÃO	13
1.2 Motivação	14
1.3 Objetivos.....	14
2.1 Automação Residencial	15
2.1.1 <i>Smart Home</i>	16
2.1.2 <i>Internet das Coisas</i>	18
2.1.3 <i>Consumo de Energia</i>	18
2.2 Microprocessadores e Microcontroladores	20
2.2.1 <i>Arduino Nano</i>	22
2.2.2 <i>ESP8266</i>	24
2.3 Protocolo de Comunicação	26
2.3.1 <i>Serial Peripheral Interface</i>	27
2.3.2 <i>Padrão IEEE 802.1</i>	29
2.4 Circuitos Elétricos.....	30
2.4.1 <i>Fonte de alimentação</i>	32
2.4.2 <i>Efeito Hall</i>	33
2.4.3 <i>Tensão RMS</i>	34
2.4.4 <i>Amostragem</i>	35
3. MATERIAIS E MÉTODOS	37
3.1 Visão geral da arquitetura do medidor de consumo.....	37
3.2 <i>Alimentação</i>	39
3.2.1.1 <i>Circuito de Alimentação</i>	39
3.2.1.2 <i>Circuito de Comunicação</i>	41
3.2.2 <i>Coleta de dados</i>	42
3.2.2.1 <i>Captura de Corrente</i>	43
3.2.2.2 <i>Captura de Tensão</i>	45
3.2.3 <i>Comunicação SPI</i>	47
3.2.3.1 <i>Master</i>	48
3.2.3.2 <i>Slave</i>	48
3.2.4 <i>WebServer</i>	49
4. RESULTADOS E DISCUSSÕES	51
4.1 Montagem do Circuito e coleta de informações.	51
4.2 Monitoramento do consumo.	52

4.3 Assertividade das informações reproduzidas.....	52
4.3.1 <i>Ensaio 1</i>	53
4.3.2 <i>Ensaio 2</i>	54
4.3.3 <i>Ensaio 3</i>	55
6. REFERÊNCIAS	58
APÊNDICE A - INICIANDO HOTSPOT E WEB SERVER NO ESP8266	63
APÊNDICE B - EXECUÇÃO DO SERVIDOR, PAGINA HTML E SCRIPT AJAX	64
APÊNDICE C – TRABALHANDO COM SPI NO ESP8266 (MESTRE)	65
APÊNDICE D – TRABALHANDO COM SPI NO ARDUINO NANO (ESCRAVO)	66
APÊNDICE E – CAPTURA E TRATAMENTO DE CORRENTE	67
APÊNDICE F – CAPTURA E TRATAMENTO DE TENSÃO	68

1.INTRODUÇÃO

Sistema embarcado consiste basicamente de um sistema eletrônico que integra os circuitos de hardware com técnica de programação de software para fornecer determinada solução (LEE, 2011). Ainda de acordo com o autor, um sistema embarcado pode ser visto em um âmbito macro como pequenos computadores com múltiplos sensores que monitoram e controlam os processos físicos. Existem diversas áreas que integram os sistemas embarcados, tais como: sistemas automotivos, sistemas aviônicos, sistemas industriais, automação residencial.

Na automação residencial, os eletrodomésticos surgem com novas funções que são transcendentais às suas funções originais. Tecnologias como GSM, *Bluetooth*, Wi-Fi e Internet estão cada vez mais presentes nas residências devido esta crescente evolução dos dispositivos que estão sendo desenvolvidos com sistemas computacionais embarcados para processar informações e comandos. Este novo conceito de tecnologia está sendo incorporado neste ambiente onde a premissa é o acesso remoto, monitoramento de informações e maior comodidade ao cliente. (PALANIAPPAN, 2015)

Considerando o crescente número de tecnologias em residências e suas potencialidades, a eficiência energética é um ponto necessário de ser abordado. Simplificando, a eficiência energética é a tentativa de reduzir a quantidade de energia utilizada para funcionamento adequado de algum dispositivo visando o desempenho e otimização da tecnologia (VIANA,2012). Esta lógica pode ser estendida a dispositivos individuais, como lâmpadas, para permitir que elas usem menos energia para a mesma quantidade de iluminação como também para todo um conjunto de dispositivos como, por exemplo, uma sala manter uma temperatura agradável devido a redução de consumo energético de diversos dispositivos levando a menor quantidade de calor dissipado. A abordagem pode ir a diversos contextos, seja de impacto econômico, gastos financeiros e até melhoria de qualidade de vida (ASE, 2012).

Há muitas maneiras de melhorar a eficiência energética dentro do lar, que podem ser soluções simples de baixo custo, tais como a substituição das lâmpadas tradicionais fluorescentes pela iluminação LED. Porém, existem soluções que demandam uma mão-de-obra especializada no qual é necessário um investimento inicial alto para obter um retorno, como um sistema fotovoltaico para geração de energia como estratégia para minimizar a conta de luz. Visto este cenário, ações mais simples como controlar o consumo mensal do lar se tornam a primeira opção a se considerar.

1.2 Motivação

Os sistemas de computação trouxeram ao longo dos anos diversas aplicabilidades para as mais distintas áreas, seja na indústria, na agropecuária, na mídia e ganhando cada vez mais seu espaço para desempenhar uma ou diversas funções específicas para respectiva aplicação (NEVES, 2007). Neste trabalho estudou-se uma das crescentes áreas: a Automação residencial.

O cenário em relação a esse contexto é qualquer ambiente que consista em aparelhos como televisores, aparelhos de som, ventiladores, ares-condicionados, motores, aquecedores, sistemas de iluminação, etc. Existe uma demanda hoje que o ambiente residencial seja acessível remotamente, permitindo que cada aparelho possa ser acessado e controlado através de uma interface de software, seja ela em um *smartphone* ou em uma aplicação *web*. Esses sistemas acessíveis de forma remota já estão disponíveis no mercado, através de aparelhos eletrodomésticos que possuem sistemas embarcados cada vez mais robustos devido às diversas funcionalidades exigidas pelos usuários ao longo do tempo.

Considerando o crescente aumento do consumo de energia nas residências nos últimos anos, devido ao surgimento de novos conceitos, tais como Internet das Coisas (IoT, do inglês *Internet of Things*), que será tratado com mais atenção no decorrer deste trabalho, é pertinente desenvolver um medidor como uma solução viável aplicado ao gerenciamento de consumo de energia residencial.

1.3 Objetivos

Neste trabalho é proposto um projeto de construção de um medidor de consumo de energia utilizando sistemas embarcados. Este tem como objetivo geral realizar a medição das grandezas elétricas de uma carga conectada à rede residencial. Alguns objetivos específicos são:

- Implementar a comunicação de um *System-on-a-Chip* (SoC) com sensores, amperímetros e voltímetros;
- Gerar de um conjunto de informações para produzir um monitoramento do consumo de energia;
- Visualizar através de uma aplicação *web* os dados capturados.

2. REFERENCIAL TEÓRICO

Neste capítulo serão abordados alguns fundamentos que serviram como base para o desenvolvimento deste trabalho. Entre eles estão os conceitos de automação residencial, microprocessadores e microcontroladores, protocolos de comunicação e circuitos elétricos.

2.1 Automação Residencial

A automação residencial é a integração ou inserção de tecnologias no ambiente residencial a fim de obter maior controle sobre o mesmo. Um sistema de automação residencial controlará iluminação, clima, sistemas de entretenimento e eletrodomésticos. Também pode incluir segurança doméstica, como controle de acesso e sistemas de alarme. Quando conectados à Internet, os dispositivos domésticos são um importante componente da Internet das Coisas (BARROS, 2010).

Um sistema de automação residencial normalmente conecta dispositivos controlados a um *hub* central ou *gateway*. A interface do usuário para o controle do sistema usa terminais montados na parede, computadores *tablet* ou *desktop*, um aplicativo de telefone celular ou uma interface da *Web*, que também pode ser acessada externamente pela Internet (TERUEL, 2008).

Embora existam muitos fornecedores concorrentes, existem muito poucos padrões da indústria aceitos mundialmente e o espaço doméstico inteligente é bastante fragmentado. Os fabricantes geralmente impedem implementações independentes por meio da retenção de documentação e por litígios, o que dificulta o desenvolvimento deste campo. Apesar disso, o mercado de automação residencial movimentou US \$ 5,77 bilhões em 2013, com previsão de atingir um valor de mercado de US \$ 12,81 bilhões até o ano 2020 (KITCHIN; DODGE, 2011).

A automação residencial inicial começou com máquinas que economizam mão-de-obra. Eletrodomésticos autônomos movidos a eletricidade ou a gás tornaram-se viáveis em 1900 com a introdução da distribuição de energia elétrica e levaram à introdução de lavadoras (1904), aquecedores de água (1889), refrigeradores, máquinas de costura, lava-louças e secadores de roupa (KITCHIN; DODGE, 2011).

Em 1975, foi desenvolvida a primeira tecnologia de automação doméstica de uso geral, X10. Este é um protocolo de comunicação para dispositivos eletrônicos, o qual usa principalmente a fiação de transmissão de energia elétrica para sinalização e controle e continua sendo o mais amplamente disponível (LI et al., 2012).

Em 1978, os produtos X10 incluíam um console de comando de 16 canais, um módulo de lâmpada e um módulo de *appliance*. Logo depois veio o módulo de interruptor de parede e o primeiro temporizador X10 (KITCHIN; DODGE, 2011).

Até 2012, nos Estados Unidos, de acordo com a ABIResearch, 1,5 milhões de sistemas de automação residencial foram instalados. Segundo Li et al. (2012) existem três gerações de automação residencial:

- a) Primeira geração: tecnologia sem fio com servidor *proxy*, por exemplo Automação Zigbee;
- b) Segunda geração: a inteligência artificial controla dispositivos elétricos, por exemplo AmazonEcho;
- c) Terceira geração: amigo robô que interage com humanos, por exemplo Robô Rovio, Roomba.

A automação residencial sofre de fragmentação de plataforma e falta de padrões técnicos, onde a variedade de dispositivos de automação residencial, resultante de uma diversidade de hardware e software, dificultam a tarefa de desenvolver aplicativos que trabalhem consistentemente entre diferentes ecossistemas tecnológicos. Os clientes podem hesitar em investir em dispositivos *IoT* com software proprietário ou hardware que usam protocolos proprietários que podem desaparecer ou se tornar difíceis de personalizar e interconectar (BARROS, 2010).

A natureza dos dispositivos de automação residencial também pode ser um problema para a segurança, já que as correções para as falhas encontradas no sistema operacional central geralmente não alcançam os usuários de dispositivos mais antigos e de menor preço. Um grupo de pesquisadores afirma que a falha dos fornecedores em suportar dispositivos mais antigos com correções e atualizações deixa mais de 87% dos dispositivos ativos vulneráveis (BOLZANI, 2007).

2.1.1 Smart Home

Smart home é um termo usado para definir uma ampla gama de soluções para controle, monitoramento e automação de funções em casa. Um *smart home* requer um aplicativo de *smartphone* ou um portal *web* como uma interface de usuário (SARIJARI et al., 2014). Dispositivos que só podem ser controlados com interruptores, temporizadores, sensores

e controles remotos não são o escopo deste estudo. Os sistemas domésticos inteligentes podem ser agrupados em seis categorias primárias (ANDERSON, 2016):

- a) Sistemas de gerenciamento de energia e controle climático.
- b) Sistemas de segurança e controle de acesso.
- c) Iluminação, janela e aparelho sistemas de controle.
- d) Eletrodomésticos.
- e) Audiovisual e entretenimento de sistemas.
- f) Sistemas de assistência médica assistida.

De acordo com as projeções feitas pela CISCO / ERICSSON / ITU, pode haver cerca de 50 bilhões de dispositivos conectados globalmente até 2020. Como em outro estudo realizado pela CISCO em 2015, a participação de computadores (incluindo *PCs*, *tablets*, e *smartphones*) serão cerca de 17 por cento de todas as conexões de Internet; os outros 83 por cento resultarão de dispositivos a serem usados em vários setores verticais do domínio *IoT* (MENDES et al., 2015).

A Índia deve ter cerca de 2,6 bilhões dispositivos conectados em 2020. O mercado de automação residencial na Índia deverá atingir INR 8800 Crores (US \$ 1,3 B) até 2017 e está crescendo a uma taxa de 30% CAGR (SAMUEL, 2016).

A idéia do *smarthome* foi originada há muito tempo antes do advento da *IoT*. O mundo passou por uma grande mudança devido à invenção de vários dispositivos para automatizar as atividades domésticas durante 1901-1920. Entre as poucas invenções do início do século XX, estava o primeiro aspirador com motor em 1901 seguido pela primeira geração de refrigeradores, máquinas de lavar etc. Em 1966, ECHO IV foi o primeiro dispositivo inteligente inventado, mas não foi vendido no mercado comercial (WITHANAGE et al., 2014).

Houve um mito de que o *Smart Home* fosse um item de luxo. Embora isso tenha sido verdade anteriormente, as necessidades modernas transformaram as Casas Inteligentes em uma categoria exigida. Isso aconteceu principalmente por conta da penetração de *Smart Devices* a preços mais acessíveis e integração no ecossistema da *IoT*. As barreiras de entrada baixas, especialmente com kits *Do It Yourself* (DIY) também estão contribuindo para isso (MENDES et al., 2015).

Com o advento de tais tecnologias, a idéia de *smart home* está sendo estendida para automóveis também. Muitos fabricantes de automóveis, como Ford, Toyota estão trabalhando para sincronizar seu automóvel com aparelhos domésticos inteligentes. A maioria deles

investiram em sua pesquisa e desenvolvimento para estabelecer um sistema à prova de erros. (ANDERSON, 2016).

2.1.2 Internet das Coisas

O conceito de *IoT* ou em tradução livre para o português, Internet das Coisas, pode ser definido como uma rede de dispositivos físicos que permite que esses objetos se conectem e troquem dados. Dentro desta rede, cada dispositivo possui seu próprio sistema embarcado para desempenhar suas funções exclusivas, mas também é capaz de interagir dentro da infraestrutura existente na Internet (LI et al., 2012). Por exemplo, eletrodomésticos ou veículos podem ter conectividade com software, sensores e atuadores.

A aplicação de *IoT* necessita da combinação de três componentes: dispositivos, sistema de controle e redes de comunicação. Os eletrônicos possuem um sistema embarcado equipados com chips, sensores, antenas, entre outros dispositivos que irão proporcionar a comunicação e controle. As redes de comunicação são as tecnologias Wi-Fi, *Bluetooth* e NFC, de redes móveis como 3G e 4G não bastam para que o dispositivo se conecte à internet ou troque informações com outros objetos, esses dados precisam ser processados de forma dependente da aplicação. Há diversas aplicações potenciais de *IoT* e em diferentes áreas, podendo ser vistos em hospitais, agropecuária, fábricas, comércios, transportes públicos, logística, serviços públicos, dentre outros. Aplicações simples como na localização de determinado ônibus até monitoramento cardíaco de um paciente.

Os dispositivos com capacidade *on-line* aumentaram 31% de 2016 para 8,4 bilhões em 2017. A *IoT* é estimada a constituir em cerca de 30 bilhões de objetos até 2020. Estima-se também que o valor de mercado global da *IoT* chegará a US \$ 7,1 trilhões até 2020. (GREENBERG, 2017). As suas aplicabilidades resultam em impacto benéfico a economia visto o potencial ganho de eficiência, precisão além da redução da intervenção humana nas integrações com o mundo físico. Os estudiosos jurídicos sugerem que as "coisas" são uma "mistura inextricável de hardware, software, dados e serviço". Esses dispositivos coletam dados úteis com a ajuda de várias tecnologias existentes e, em seguida, transferem de forma autônoma os dados entre outros dispositivos (ATZORI; IEIRA; MORABITO, 2010).

2.1.3 Consumo de Energia

A automação residencial é apenas energeticamente eficiente quando a casa é programada para ser assim, caso contrário, a energia usada para alimentar os outros sistemas

automatizados supera a que é economizada. Instalar uma tecnologia com eficiência energética e programar sua casa inteligente para ajudar a economizar energia pode ajudá-lo a reduzir seu impacto no ambiente, além do impacto de sua conta de energia (BARROS, 2010).

Como há simplesmente mais tecnologia instalada em uma casa automatizada do que uma casa comum, as casas inteligentes têm a tendência de consumir mais energia do que economizam. Equipamentos como sensores, processadores e motores/controles consomem um pouco de energia, o que pode aumentar com o tempo. Uma casa inteligente com eficiência energética requer um planejamento cuidadoso e uma instalação adequada para garantir que você economize o máximo de energia possível (BARROS, 2010).

Talvez a maior economia possível seja a maneira como a casa regula o clima. Através do uso de sensores e temporizadores, unidades de resfriamento e aquecimento, ventiladores de teto, janelas, revestimentos de janelas e exaustores podem ser controlados para aproveitar ao máximo o calor natural e o frescor. Eles também podem ser operados em zonas específicas que o usuário solicita sobre como e quando for necessário (com base em sensores de ocupação) para garantir que o equipamento esteja apenas aquecendo e esfriando em áreas que precisam em qualquer ponto específico (BARROS, 2010).

Um relatório da Agência Internacional de Energia afirmou que uma mudança mundial para sistemas de iluminação com eficiência energética poderia economizar até 10% da conta de eletricidade do mundo (BARROS, 2010).

A maioria dos controladores de iluminação reguláveis é definida com um nível de saída de 90%, o que significa que você economiza 10% de energia toda vez que aciona o botão. Controles automáticos de iluminação que desligam e apagam as luzes somente quando (e onde) são necessários também podem ajudar a economizar uma pequena fortuna em iluminação. Faixas de energia que cortam todo o suprimento de energia para aparelhos em modo de espera, como televisores e microondas, podem ajudar a reduzir em até 20% sua conta de energia (BARROS, 2010).

Alguns dispositivos de gerenciamento de energia podem ler seu uso de energia em cada ponto de energia, e também detectar e ativar quando você precisar alterar seu consumo. Os medidores inteligentes também permitem que o usuário monitore seu próprio consumo de energia em tempo real e, portanto, lhe dá a capacidade de mudar seu estilo de vida, dependendo de quanto excesso de energia você está usando. Um painel de gerenciamento de energia coleta dados sobre o uso de energia quando os equipamentos atingem o pico de uso. Isso permite que se possa entender quais dispositivos estão operando quando o pico de uso é

atendido, além de mostrar quando pode usar energia em horários de maior uso (BARROS, 2010).

Como todas as casas inteligentes são diferentes, todas elas consomem e economizam de forma igualmente diferente. No entanto, deve-se entender a capacidade do sistema automatizado de economizar energia, para ter uma compreensão mais ampla sobre se a casa é ou não eficiente em energia. Dependendo dos sistemas de automação que são instalados e a forma como a casa é usada e construída, a economia de energia pode até mesmo superar o custo de instalação em apenas alguns anos. Projetos menores baseados em iluminação têm maior probabilidade de gerar economias de curto prazo, enquanto projetos maiores, incluindo controles climáticos e cegos, podem levar mais tempo (BARROS, 2010).

Todos pagamos o nosso consumo de energia com base de uma tarifa. No Brasil relaciona cada kWh (quilowatt por hora) como unidade de energia. O preço da energia elétrica é formado pelos custos da geração até a sua disponibilidade aos consumidores dentro de suas residências 24 horas por dia, 7 dias por semana, cujos valores são estabelecidos por resoluções ou despachos da Agência Nacional de Energia Elétrica, ANEEL (ABRADEE, 2012).

Figura 1 – Composição tarifária das distribuidoras



Fonte: ABRADEE, Tarifas de Energia.

Desta forma o preço da energia se faz suficiente para arcar com os custos de operação e expansão de todo o sistema elétrico, cobrindo os investimentos realizados na rede e a sua diária operação, reduzindo assim os índices de falhas e tempo de manutenções. Está compreendido desde a usina geradora até a distribuição de ligação dos consumidores de baixa tensão de energia.

2.2 Microprocessadores e Microcontroladores

Um microprocessador é um processador de computador que incorpora as funções de uma unidade central de processamento em um único circuito integrado (IC, do inglês

IntegredCircuit), ou no máximo alguns circuitos integrados. O microprocessador é um circuito integrado digitalmente multiuso, acionado por *clock* e baseado em registro que aceita dados binários como entrada, processa-os de acordo com instruções armazenadas em sua memória e fornece resultados como saída. Os microprocessadores contêm lógica combinacional e lógica digital sequencial. Os microprocessadores operam em números e símbolos representados no sistema numérico binário (FRIELDEWALD, 2011).

Um microcontrolador (MCU para a unidade microcontroladora ou UC para o controlador μ) é um pequeno computador em um único circuito integrado. Na terminologia moderna, é similar, mas menos sofisticada que, um sistema em um chip ou SoC; um SoC pode incluir um microcontrolador como um de seus componentes. Um microcontrolador contém um ou mais processadores (núcleos de processador), juntamente com periféricos de entrada e saída de memória e programáveis. A memória de programa na forma de RAM ferroelétrica, flash NOR ou ROM OTP também é frequentemente incluída no chip, bem como uma pequena quantidade de RAM (CARRO; WAGNER, 2003).

Os microcontroladores são usados em produtos e dispositivos controlados automaticamente, como sistemas de controle de motores de automóveis, dispositivos médicos implantáveis, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos e outros sistemas embarcados. Ao reduzir o tamanho e o custo em comparação com um projeto que usa um microprocessador separado, memória e dispositivos de entrada / saída, os microcontroladores tornam econômico o controle digital de ainda mais dispositivos e processos. Microcontroladores de sinais mistos são comuns, integrando componentes analógicos necessários para controlar sistemas eletrônicos não digitais (EAGLE, 2016).

Alguns microcontroladores podem usar palavras de quatro bits e operar em frequências tão baixas quanto 4 kHz, para baixo consumo de energia (miliwatts de um dígito ou microwatts). Eles geralmente terão a capacidade de manter a funcionalidade enquanto esperam por um evento, como um pressionamento de botão ou outra interrupção; consumo de energia durante o sono (relógio da CPU e a maioria dos periféricos desligados) pode ser apenas nanowatts, tornando muitos deles adequados para aplicações de bateria de longa duração. Outros microcontroladores podem servir a funções de desempenho crítico, onde eles podem precisar agir mais como um processador de sinal digital (DSP, do inglês *Digital Signal Processor*), com velocidades de *clock* e consumo de energia mais alta (FAIRCHILD, 2016).

Comparando microcontrolador e microprocessador em termos de custo não se justifica. Sem dúvida, um microcontrolador é muito mais barato que um microprocessador. No entanto microcontrolador não pode ser usado no lugar do microprocessador e usando um microprocessador não é recomendado no lugar de um microcontrolador, pois torna a aplicação bastante onerosa. O microprocessador não pode ser usado sozinho. Eles precisam de outros periféricos como RAM, ROM, buffer, portas de E / S, etc, e, portanto, um sistema projetado em torno de um microprocessador é bastante caro (LABCENTER, 2016).

2.2.1 Arduino Nano

O Arduino Nano é uma placa pequena, completa e fácil de usar, baseada no microcontrolador ATmega328 (Arduino Nano 3.0) ou ATmega168 (Arduino Nano 2.x). Esse Arduino tem funcionalidades semelhantes ao ArduinoDuemilanove, porém, com algumas diferenças, como a falta de um conector DC e o funcionamento através de um cabo USB Mini-B. O Nano foi projetado e está sendo produzido pela Gravitech (ARDUINO, 2017).

O Arduino Nano pode ser alimentado através da conexão USB Mini-B, fonte de alimentação externa não regulada de 6-20V (pino 30) ou fonte de alimentação externa regulada de 5V (pino 27). A fonte de energia é automaticamente selecionada para a fonte de tensão mais alta (JUNIOR, 2014).

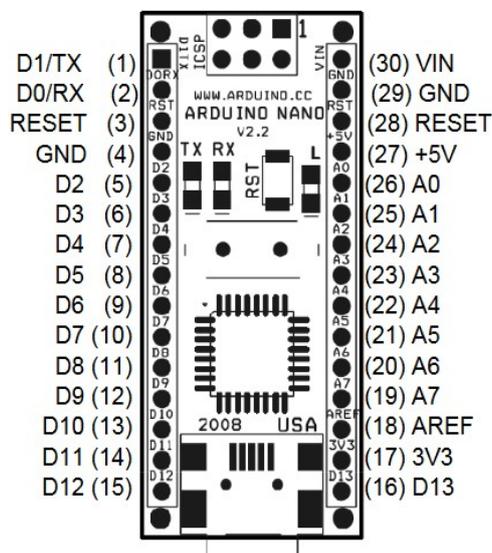
O chip FTDI FT232RL no Nano só é energizado se a placa estiver sendo alimentada via USB. Como resultado, ao funcionar com energia externa (não USB), a saída de 3,3V (que é fornecida pelo chip FTDI) não está disponível e os LEDs RX e TX piscarão se os pinos digitais 0 ou 1 estiverem altos (BANZI, SHILOH, 2015).

O Arduino Nano possui várias facilidades para comunicação com um computador, outro Arduino ou outros microcontroladores. O ATmega168 e o ATmega328 fornecem comunicação serial UART TTL (5V), que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um FTDI FT232RL na placa canaliza essa comunicação serial via USB e os drivers FTDI (incluídos no software Arduino) fornecem uma porta de comunicação virtual para o software no computador. O software Arduino inclui um monitor serial que permite que dados textuais simples sejam enviados de e para a placa Arduino. Os LEDs RX e TX na placa piscarão quando os dados estiverem sendo transmitidos via chip FTDI e conexão USB ao computador (mas não para comunicação serial nos pinos 0 e 1) (COSTA; GRUNDSTEIN, 2016).

Uma biblioteca *Software Serial* permite a comunicação serial em qualquer um dos pinos digitais do Nano. O ATmega168 e o ATmega328 também suportam comunicação I2C (TWI) e *SPI*. O software Arduino inclui uma biblioteca *Wire* para simplificar o uso do

barramento I2C; Veja a documentação para detalhes. Para usar a comunicação *SPI*, consulte a folha de dados ATmega168 ou ATmega328 (JUNIOR, 2014).

Figura 2 – Entradas e saídas do Arduino Nano



Fonte: Arduino Nano Documentation, 2008.

O Arduino Nano pode ser programado com o *software* Arduino.

O ATmega168 ou ATmega328 no Arduino Nano vem pré-gravado com um gerenciador de inicialização que permite que você faça o *upload* de um novo código para ele sem o uso de um programador de hardware externo. Ele se comunica usando o protocolo STK500 original (referência, arquivos de cabeçalho C) (COSTA; GRUNDSTEIN, 2016).

Em vez de exigir um pressionamento físico do botão de reset antes de um *upload*, o Arduino Nano é projetado de forma a permitir que ele seja redefinido pelo *software* executado em um computador conectado. Uma das linhas de controle de fluxo de hardware do FT232RL é conectada à linha de reset do ATmega168 ou ATmega328 através de um capacitor de 100 nanofarad (ARDUINO, 2017).

Quando esta linha é declarada (baixa), a linha de *reset* cai o suficiente para resetar o *chip*. O *software* Arduino usa esse recurso para permitir que você faça *upload* de código simplesmente pressionando o botão de *upload* no ambiente do Arduino. (JUNIOR, 2014).

Essa configuração tem outras implicações. Quando o Nano está conectado a um computador com Mac OS X ou Linux, ele é redefinido toda vez que uma conexão é feita a partir do software (via *USB*). Durante o seguinte meio segundo, o *bootloader* está sendo executado no Nano (ARDUINO, 2017).

Embora esteja programado para ignorar dados malformados (ou seja, qualquer coisa além do *upload* de um novo código), ele interceptará os primeiros *bytes* de dados enviados

para a placa após a abertura de uma conexão. Se um esboço executado na placa receber uma configuração única ou outros dados quando for iniciado pela primeira vez, certifique-se de que o *software* com o qual ele se comunica espere um segundo depois de abrir a conexão e antes de enviar esses dados (BANZI, SHILOH, 2015).

2.2.2 ESP8266

O ESP8266 é um microchip Wi-Fi de baixo custo com comunicação TCP / IP completa e capacidade de microcontrolador produzido pela fabricante chinesa Espressif Systems, sediada em Xangai (ALOI, 2013).

O chip chamou a atenção dos fabricantes ocidentais em agosto de 2014 com o módulo ESP-01, feito por um fabricante terceirizado, Ai-Thinker. Este pequeno módulo permite que microcontroladores se conectem a uma rede Wi-Fi e façam conexões TCP / IP com comandos simples (CAMPOS, 2015).

No entanto, na época não havia quase nenhuma documentação em inglês sobre o chip e os comandos aceitos. O preço muito baixo e o fato de que havia poucos componentes externos no módulo que sugeriam que poderia eventualmente ser muito barato em volume, atraiu muitos hackers a explorar o módulo, o *chip* e o *software* nele, bem como a traduzir a documentação chinesa (FLOP, 2016).

O ESP8285 é um ESP8266 com 1 MiB de *flash* embutido, permitindo dispositivos de chip único capazes de se conectar ao Wi-Fi. O sucessor destes chips microcontroladores é o ESP32. A série ESP8266, ou família, de chips Wi-Fi é produzida pela Espressif Systems, uma empresa de semicondutores *fabless* que opera em Xangai, na China. A série ESP8266 inclui atualmente os chips ESP8266EX e ESP8285 (FLOP, 2016).

ESP8266EX (simplesmente referido como ESP8266) é um system-on-chip (SoC) que integra um microcontrolador *Tensilica* de 32 bits, interfaces periféricas digitais padrão, comutadores de antena, *balun* de RF, amplificador de potência, amplificador de recepção de baixo ruído, filtros e gerenciamento de energia módulos em um pacote pequeno (PRASAD, 2009).

Ele fornece recursos para Wi-Fi de 2,4 GHz (802.11 b / g / n, WPA / WPA2), entrada / saída de uso geral (16 *GPIO*), Circuito Inter-Integrado (I^2C , do inglês *Inter-Integrated Circuit*), conversão analógico-digital (10 ADC), Interface Serial Periférica (*SPI*), interfaces I^2S (*Inter-ICSound*) com *DMA* (*Direct Memory Access*, para compartilhamento de pinos com *GPIO*), *UART* (em pinos dedicados, além de uma *UART* somente de transmissão pode ser habilitada no *GPIO2*) e modulação por largura de pulso (*PWM*, do inglês *Pulse*

da *Espressif* são atualmente ESP-WROOM-02 e ESP-WROOM-S2. Os módulos *Ai-Thinker* são rotulados sucintamente ¹de ESP-01 a ESP-14.

2.3 Protocolo de Comunicação

Na telecomunicação, um protocolo de comunicação é um sistema de regras que permite que duas ou mais entidades de um sistema de comunicações transmitam informações por meio de qualquer tipo de variação de uma grandeza física. O protocolo define a sintaxe das regras, semântica e sincronização de comunicação e possíveis métodos de recuperação de erros. Protocolos podem ser implementados por hardware, software ou uma combinação de ambos (TANENBAUM, 2003).

Os sistemas de comunicação usam formatos bem definidos (protocolo) para trocar várias mensagens. Cada mensagem tem um significado exato destinado a extrair uma resposta de uma gama de possíveis respostas pré-determinadas para aquela situação específica (TRONCO, 2006).

O comportamento especificado é normalmente independente da implementação. Os protocolos de comunicação devem ser acordados pelas partes envolvidas. Para chegar a um acordo, um protocolo pode ser desenvolvido em um padrão técnico. Uma linguagem de programação descreve o mesmo para cálculos, portanto há uma analogia próxima entre protocolos e linguagens de programação: os protocolos são para comunicar quais linguagens de programação são para cálculos (TANENBAUM, 2003).

Vários protocolos geralmente descrevem diferentes aspectos de uma única comunicação. Um grupo de protocolos projetado para trabalhar em conjunto é conhecido como um conjunto de protocolos; quando implementados em software, eles são uma pilha de protocolos (BATES, 2002).

Os protocolos de comunicação da Internet são publicados pela *Internet Engineering Task Force* (IETF). O IEEE lida com redes com e sem fio, e a Organização Internacional para Padronização (ISO) lida com outros tipos. O ITU-T lida com protocolos e formatos de telecomunicações para a rede telefônica pública comutada (PSTN). À medida que o PSTN e a Internet convergem, os padrões também estão sendo direcionados para a convergência (TANENBAUM, 2003).

A informação trocada entre dispositivos através de uma rede ou outra mídia é governada por regras e convenções que podem ser estabelecidas nas especificações do

¹Os módulos da Ai-Thinker são geralmente associados como os identificadores não oficiais de "ESP8266-01" através de "ESP8266-14" para facilitar a compreensão (MINATEL, 2015).

protocolo de comunicação. A natureza de uma comunicação, os dados reais trocados e quaisquer comportamentos dependentes do estado são definidos por essas especificações. Em sistemas de computação digital, as regras podem ser expressas por algoritmos e estruturas de dados. Os protocolos são para comunicação que algoritmos ou linguagens de programação são para cálculos (ROSEN, 2001).

Quando os algoritmos de protocolo são expressos em uma linguagem de programação portátil, o software de protocolo pode se tornar independente do sistema operacional. Os frameworks mais conhecidos são o modelo TCP / IP e o modelo OSI (ALOIA, 2003).

2.3.1 Serial Peripheral Interface

Serial Peripheral Interface (SPI) é um barramento usado comumente para enviar dados de curta distância entre microcontroladores e pequenos periféricos como sensores, cartões de memória, e entre outros. A interface foi desenvolvida pela Motorola em meados da década de 1980 e se tornou um padrão de fato. Aplicações típicas incluem cartões Secure Digital e displays de cristal líquido (PATTERSON, 2012).

Os dispositivos *SPI* se comunicam no modo *full duplex* usando uma arquitetura mestre-escravo com um único mestre. O dispositivo mestre origina o quadro para leitura e gravação. Há vários dispositivos escravos são suportados através de seleção com linhas individuais de *slaveselect (SS)*, que significa em tradução livre para o português seleção de escravos conforme enuncia (Pell, 2011).

Às vezes, o *SPI* é chamado de barramento serial de quatro fios, contrastando com os barramentos seriais de três, dois e um fio. O *SPI* pode ser descrito com precisão como uma interface *synchronous serial*, mas é diferente do protocolo SSI (Synchronous Serial Interface), que também é um protocolo de comunicação serial síncrono de quatro fios. O protocolo SSI emprega sinalização diferencial e fornece apenas um único canal de comunicação simplex. O barramento SPI pode operar com um único dispositivo mestre e com um ou mais dispositivos escravos(PATTERSON, 2012).

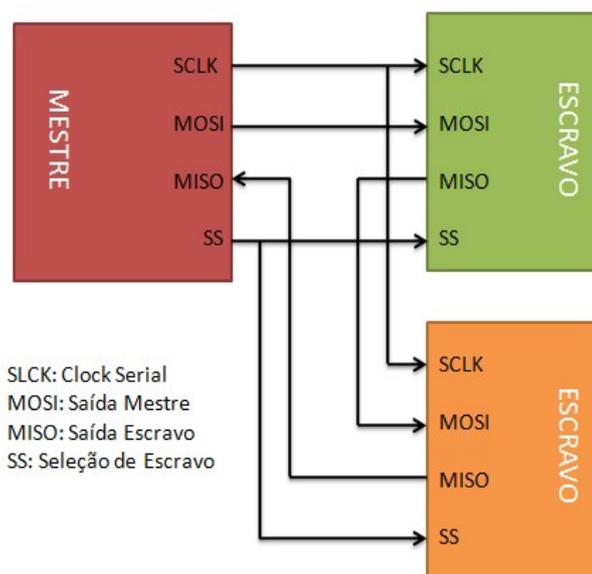
Se um único dispositivo escravo for usado, o pino SS pode ser fixado em lógica baixa se o escravo permitir. Alguns escravos exigem uma borda descendente do sinal de seleção de chip para iniciar uma ação. Um exemplo é o Maxim MAX1242 ADC, que inicia a conversão em uma transição alta → baixa. Com múltiplos dispositivos escravos, é necessário um sinal SS independente do mestre para cada dispositivo escravo(PATTERSON, 2012).

A maioria dos dispositivos escravos tem saídas *tri-state*, de modo que seu sinal MISO se torna de alta impedância (logicamente desconectado) quando o dispositivo não é selecionado. Dispositivos sem saídas *tri-state* não podem compartilhar segmentos de barramento SPI com outros dispositivos; apenas um desses escravos poderia conversar com o mestre (PATTERSON, 2012).

Para iniciar a comunicação, o mestre de barramento configura o relógio, usando uma frequência suportada pelo dispositivo escravo, normalmente até alguns MHz. O mestre então seleciona o dispositivo escravo com um nível lógico 0 na linha de seleção. Se for necessário um período de espera, como para uma conversão analógico-digital, o mestre deve aguardar pelo menos esse período de tempo antes de emitir os ciclos de *clock* (PELL, 2011).

As transmissões normalmente envolvem dois registradores de deslocamento de um determinado tamanho de palavra, como oito bits, um no mestre e outro no escravo; eles estão conectados em uma topologia de anel virtual. Os dados geralmente são deslocados primeiro com o bit mais significativo. Na borda do relógio, tanto o mestre quanto o escravo deslocam-se um pouco e emitem-no na linha de transmissão para a contraparte (PATTERSON, 2012).

Figura 4 – Esquema de comunicação *SPI* entre um mestre e dois escravos.



Fonte: Próprio autor.

As transmissões geralmente consistem em palavras de 8 bits. No entanto, outros tamanhos de palavras também são comuns, por exemplo, palavras de 16 bits para controladores de tela sensível ao toque ou *codecs* de áudio, como o TSC2101 da Texas Instruments, ou palavras de 12 bits para muitos conversores digital-analógicos ou analógico-

digital. Cada escravo no barramento que não tenha sido ativado usando sua linha de seleção de chip deve desconsiderar o relógio de entrada e os sinais MOSI, e não deve conduzir o MISO (PELL, 2011).

2.3.2 Padrão IEEE 802.1

Os padrões IEEE desempenham um papel de cidades inteligentes para dispositivos inteligentes. Os padrões impulsionam a funcionalidade, capacidades e interoperabilidade de uma ampla gama de produtos e serviços que transformam a forma como as pessoas vivem, trabalham e se comunicam (BARTH; SIEWERT, 2005).

O IEEE possui padrões que abrangem todas essas funções, mas é onde essas funções são distribuídas entre diferentes dispositivos que os padrões IEEE para redes de comunicação e seus protocolos se tornam críticos para interoperabilidade e transmissão de informações entre diferentes dispositivos e pessoas (RUBINSTEIN; REZENDE, 2002).

Dentro do grande conjunto de padrões de comunicação IEEE, os padrões IEEE 802 são fundamentais pois fornecem conectividade para infraestrutura de cidade conectada; edifícios comerciais conectados; casas conectadas; veículos conectados; e pessoas conectadas (RUBINSTEIN; REZENDE, 2002).

O Grupo de Trabalho IEEE 802.1 é um grupo da Associação de Padrões IEEE (IEEE-SA) estabelecido para garantir o gerenciamento de rede e os recursos de monitoramento em redes desenvolvidas de acordo com os padrões IEEE 802 (BARTH; SIEWERT, 2005).

O IEEE 802.1 lida com a arquitetura, segurança, gerenciamento e interconexão de redes de área local (LAN), redes de área metropolitana (MAN) e redes de área ampla (WAN) padronizadas pelo IEEE 802. A seguir estão as principais tarefas IEEE 802.1 (RUBINSTEIN; REZENDE, 2002):

- a) Projeta e implementa padrões que regulam as práticas de gerenciamento de rede
- b) Fornece serviços, incluindo gerenciamento LAN / MAN, *bridging* de controle de acesso de mídia (MAC), criptografia / codificação de dados e gerenciamento de tráfego de rede

O IEEE 802.1 é composto por quatro grupos que se concentram em diferentes padrões e políticas nas seguintes áreas (BARTH; SIEWERT, 2005):

- a) *Internetworking*
- b) Ponte de áudio / vídeo (A / V)
- c) *Data center bridging*
- d) Segurança

O grupo *Internetworking* lida com a arquitetura geral, agregação de links, endereçamento de protocolo, identificação / cálculo de caminho de rede e outras práticas e recomendações técnicas (RUBINSTEIN; REZENDE, 2002).

2.4 Circuitos Elétricos

Um circuito elétrico é um caminho no qual os elétrons de uma fonte de tensão ou corrente fluem. O ponto em que esses elétrons entram em um circuito elétrico é chamado de "fonte" de elétrons. O ponto em que os elétrons deixam um circuito elétrico é chamado de "retorno" ou "terra". O ponto de saída é chamado de "retorno", porque os elétrons sempre acabam na fonte quando completam o caminho de um circuito elétrico (FERRARO, 1991).

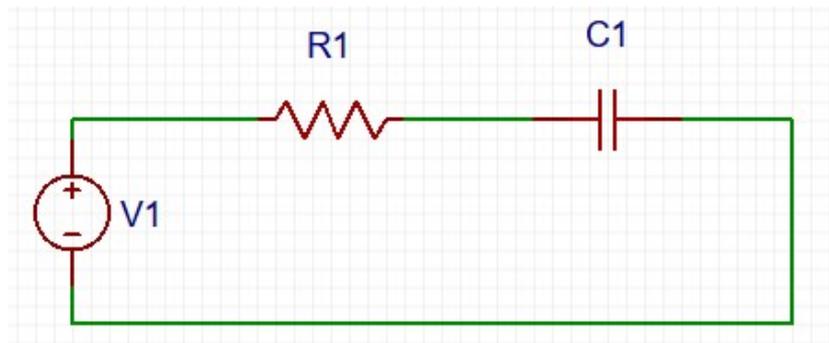
A parte de um circuito elétrico que está entre o ponto de partida dos elétrons e o ponto em que eles retornam à fonte é chamada de "carga" de um circuito elétrico. Uma carga de um circuito elétrico pode ser tão simples quanto aquelas que alimentam eletrodomésticos como geladeiras, televisores ou lâmpadas ou mais complicadas, como a carga na saída de uma estação geradora de energia hidrelétrica (GONÇALVES; BAROLLI, 1997).

Os circuitos usam duas formas de energia elétrica: corrente alternada (CA) e corrente contínua (CC). A CA alimenta frequentemente aparelhos e motores de grande porte e é gerada por estações de energia. A CC alimenta veículos operados a bateria e outras máquinas e componentes eletrônicos. Os conversores podem mudar de CA para CC e vice-versa. A transmissão de corrente contínua de alta tensão usa conversores grandes (CARRON; GUIMARÃES, 1997).

Circuitos eletrônicos geralmente usam fontes de corrente contínua de baixa tensão. A carga de um circuito eletrônico pode ser tão simples quanto alguns resistores, capacitores e uma lâmpada, todos conectados juntos para criar o flash em uma câmera. Ou um circuito eletrônico pode ser complicado, conectando milhares de resistores, capacitores e transistores. Pode ser um circuito integrado, como o microprocessador em um computador. Resistores e outros elementos de circuito podem ser conectados em série ou em paralelo. Resistência em circuitos em série é a soma da resistência (FERRARO, 1991)

Um diagrama de circuito ou diagrama de fiação usa símbolos para representar partes de um circuito.

Figura 5 – Diagrama básico de um circuito elétrico com uma fonte V1, resistor R1 e capacitor C1.



Fonte: Próprio autor.

Circuitos elétricos e eletrônicos podem ser complicados. Fazer um desenho das conexões para todas as partes componentes na carga do circuito torna mais fácil entender como os componentes do circuito estão conectados. Desenhos para circuitos eletrônicos são chamados de "diagramas de circuito". Desenhos para circuitos elétricos são chamados de "diagramas de fiação". Como outros diagramas, esses diagramas são geralmente desenhados por desenhistas e depois impressos. Os diagramas também podem ser criados digitalmente usando um software especializado (BORGES, 1998).

A fiação e diagramas de circuito utilizam símbolos especiais reconhecidos por todos que usam os desenhos. Os símbolos nos desenhos mostram como componentes como resistores, capacitores, isoladores, motores, caixas de tomadas, luzes, interruptores e outros componentes elétricos e eletrônicos são conectados juntos. Os diagramas são uma grande ajuda quando os trabalhadores tentam descobrir por que um circuito não funciona corretamente (GONÇALVES; BAROLLI, 1993).

A corrente que flui em um circuito elétrico pode aumentar subitamente quando uma peça componente falha. Isso pode causar sérios danos a outros componentes do circuito ou criar um risco de incêndio. Para proteger contra isso, um fusível ou um dispositivo chamado "disjuntor" pode ser conectado a um circuito. O disjuntor abrirá ou "quebrará" o circuito quando a corrente nesse circuito ficar muito alta ou o fusível "explodirá". Isso dá proteção (BORGES, 1998).

O retorno padrão para circuitos elétricos é o aterramento elétrico. Quando um dispositivo elétrico falha, ele pode abrir o circuito de retorno para o aterramento. O usuário do dispositivo pode se tornar parte do circuito elétrico do dispositivo, fornecendo um

caminho de retorno para os elétrons através do corpo do usuário, em vez do aterramento do circuito. Quando nosso corpo se torna parte de um circuito elétrico, o usuário pode ficar seriamente chocado, ou mesmo morto por eletrocussão (CARRON; GUIMARÃES, 1997).

2.4.1 Fonte de alimentação

Uma fonte de alimentação é um dispositivo elétrico que fornece energia elétrica a uma carga elétrica. A principal função de uma fonte de alimentação é converter a corrente elétrica de uma fonte para a tensão, corrente e frequência corretas para alimentar a carga. Como resultado, as fontes de alimentação são, às vezes, chamadas de conversores de energia elétrica (GHETTI, 2009).

Algumas fontes de alimentação são equipamentos autônomos separados, enquanto outras são incorporadas aos dispositivos de carga que alimentam. Exemplos deste último incluem fontes de alimentação encontradas em computadores desktop e dispositivos eletrônicos de consumo (HART, 2012).

Todas as fontes de alimentação têm uma conexão de entrada de energia, que recebe energia na forma de corrente elétrica de uma fonte e uma ou mais conexões de saída de energia que fornecem corrente para a carga. A fonte de energia pode vir da rede de energia elétrica, como uma tomada elétrica, dispositivos de armazenamento de energia, como baterias ou células de combustível, geradores ou alternadores, conversores de energia solar ou outra fonte de alimentação (SILVA, 2013).

Fontes de alimentação ajustáveis permitem que a tensão ou corrente de saída sejam programadas por controles mecânicos (por exemplo, botões no painel frontal da fonte de alimentação), ou por meio de uma entrada de controle, ou ambos. Uma fonte de alimentação regulada e regulável é aquela que é ajustável e regulada. Uma fonte de alimentação isolada tem uma saída de energia que é eletricamente independente de sua entrada de energia; isso está em contraste com outras fontes de alimentação que compartilham uma conexão comum entre entrada e saída de energia (SOUSA, 2013).

Uma fonte de alimentação integrada é aquela que compartilha uma placa de circuito impresso comum com sua carga. Uma fonte de alimentação externa, um adaptador CA ou uma fonte de energia, é uma fonte de alimentação localizada no cabo de energia CA da carga que é conectado a uma tomada de parede; uma verruga de parede é uma fonte externa integrada ao próprio plugue de saída. Estes são populares em eletrônicos de consumo por

causa de sua segurança; a corrente perigosa de 120 ou 240 volts da rede é transformada em uma voltagem mais segura antes de entrar no corpo do aparelho (GHETTI, 2009).

As fontes de alimentação podem ser amplamente divididas em tipos lineares e de comutação. Os conversores lineares de energia processam a energia de entrada diretamente, com todos os componentes ativos de conversão de energia operando em suas regiões operacionais lineares. Ao alternar conversores de energia, a energia de entrada é convertida em CA ou em pulsos CC antes do processamento, por componentes que operam predominantemente em modos não lineares (por exemplo, transistores que passam a maior parte do tempo em corte ou saturação) (SOUSA, 2013).

2.4.2 Efeito Hall

O efeito Hall é a produção de uma diferença de tensão (a tensão Hall) através de um condutor elétrico, transversal a uma corrente elétrica no condutor e a um campo magnético aplicado perpendicularmente à corrente. Foi descoberto por Edwin Hall em 1879. Para maior clareza, o efeito original é às vezes chamado de efeito Hall comum para distingui-lo de outros "efeitos Hall" que possuem diferentes mecanismos físicos (ABOBA; CALHOUN, 2003).

Quando uma partícula possuindo uma carga q e com velocidade v se move sobre um campo magnético B , passar a agir sobre esta partícula uma força F definida como:

$$F_B^{\rightarrow} = qv^{\rightarrow} \times B^{\rightarrow} \quad (1)$$

Se somarmos essa força magnética com o campo elétrico, temos:

$$F_L^{\rightarrow} = q(E^{\rightarrow} + v^{\rightarrow} \times B^{\rightarrow}) \quad (2)$$

A força é nomeada Força de Lorentz. O coeficiente de Hall é definido como a razão entre o campo elétrico induzido e o produto da densidade de corrente e o campo magnético aplicado. É uma característica do material do qual o condutor é feito, uma vez que seu valor depende do tipo, número e propriedades dos portadores de carga que constituem a corrente (DULANEY, 2011).

O efeito Hall é devido à natureza da corrente em um condutor. A corrente consiste no movimento de muitos pequenos portadores de carga, tipicamente elétrons, buracos, íons ou todos os três. Quando um campo magnético está presente, essas cargas experimentam uma força, chamada força de Lorentz (SARDINHA, 2005).

Quando tal campo magnético está ausente, as cargas seguem caminhos aproximadamente retos, "linha de visão" entre colisões com impurezas, fônons, etc. Entretanto, quando um campo magnético com um componente perpendicular é aplicado, seus

caminhos entre colisões são curvados, cargas em movimento se acumulam em uma face do material. Isso deixa cargas iguais e opostas expostas na outra face, onde há uma escassez de tarifas móveis (ABOBA; CALHOUN, 2003).

O resultado é uma distribuição assimétrica da densidade de carga através do elemento Hall, resultante de uma força que é perpendicular ao caminho da linha de visão e ao campo magnético aplicado. A separação de carga estabelece um campo elétrico que se opõe à migração de carga adicional, de modo que um potencial elétrico estável é estabelecido enquanto a carga estiver fluindo (ABOBA; CALHOUN, 2003).

Para um metal simples onde há apenas um tipo de portador de carga (elétrons), a tensão Hall V_H pode ser derivada usando a força de Lorentz e vendo que, na condição de estado estacionário, as cargas não estão se movendo na direção do eixo y . Assim, a força magnética em cada elétron na direção do eixo y é cancelada por uma força elétrica no eixo y devido ao acúmulo de cargas.

$$V_H = \frac{-IB}{nqd} \quad (3)$$

Temos que n representa a densidade de portadores e d a espessura do fio. (DULANEY, 2011).

2.4.3 Tensão RMS

Na estatística e suas aplicações, a raiz quadrada da média (RMS é definida como a raiz quadrada do quadrado médio (a média aritmética dos quadrados de um conjunto de números). O RMS é também conhecido como a média quadrática e é um caso particular da média generalizada com o expoente 2. O RMS também pode ser definido para uma função continuamente variável em termos de uma integral dos quadrados dos valores instantâneos durante um ciclo (CARTWRIGHT, 2007).

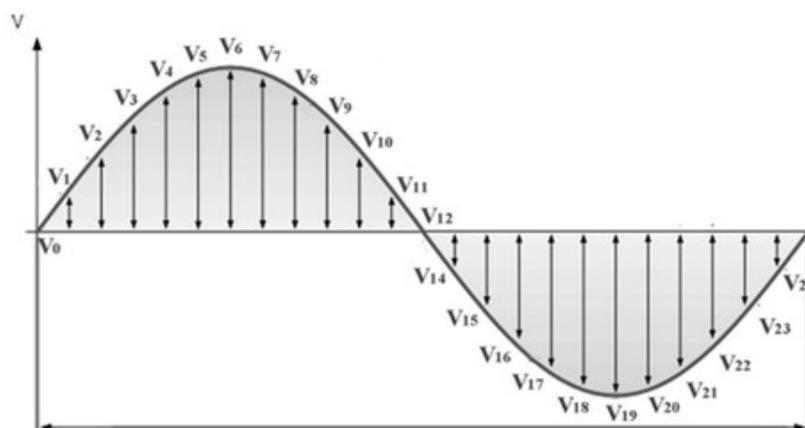
Para uma corrente elétrica ciclicamente alternada, o RMS é igual ao valor da corrente contínua que produziria a mesma dissipação de potência média em uma carga resistiva. Na teoria de estimação, o erro médio quadrático de um estimador é uma medida da imperfeição do ajuste do estimador aos dados (NASTASE, 2015).

$$V_{rms} = \sqrt{\frac{V_1^2 + V_2^2 + V_3^2 + V_4^2 + V_5^2 + \dots + V_n^2}{n}} \quad (4)$$

O valor RMS de um conjunto de valores (ou uma forma de onda de tempo contínuo) é a raiz quadrada da média aritmética dos quadrados dos valores, ou o quadrado da função

que define a forma de onda contínua. Na física, a corrente RMS é o "valor da corrente contínua que dissipa energia em um resistor" (BISSEL; CHAPMAN, 1992).

Figura 6 – Representação dos valores pontuais de tensão para determinados instantes de tempo. O *True* RMS se daria pela média aritmética da soma de todos os pontos.



Fonte: (ATHOS, 2015)

O RMS durante todo o tempo de uma função periódica é igual ao RMS de um período da função. O valor RMS de uma função contínua ou sinal pode ser aproximado tomando o RMS de uma sequência de amostras igualmente espaçadas. Além disso, o valor RMS de várias formas de onda também pode ser determinado sem cálculo, como mostrado por Cartwright. No caso da estatística RMS de um processo aleatório, o valor esperado é usado em vez da média (CARTWRIGHT, 2007).

2.4.4 Amostragem

O Teorema de Nyquist, também conhecido como o teorema da amostragem, é um princípio que deve ser seguido em processos de digitalização de sinais analógicos. Na conversão analógico-digital para resultar em uma reprodução fiel do sinal, amostras da forma de onda analógica devem ser obtidas com uma determinada frequência (LATHI, 2008). O número de amostras por segundo é chamado de taxa de amostragem ou frequência de amostragem.

Os sinais analógicos geralmente têm formas de onda complexas, com componentes em muitas frequências. O componente de frequência mais alta em um sinal analógico

determina a largura de banda desse sinal. Quanto maior a frequência, maior a largura de banda, se todos os outros fatores forem mantidos constantes.

Suponha que o componente de frequência mais alta, em hertz, para um dado sinal analógico seja f_{max} . De acordo com o Teorema de Nyquist, a taxa de amostragem deve ser de pelo menos $2f_{max}$, ou duas vezes o maior componente de frequência analógica. A amostragem em um conversor analógico-digital é acionada por um gerador de pulsos. Se a taxa de amostragem for menor que $2f_{max}$, alguns dos componentes de frequência mais alta no sinal de entrada analógica não serão representados corretamente na saída digitalizada (LATHI, 2008).

Quando um sinal digital desse tipo é convertido de volta para a forma analógica por um conversor de digital para analógico, aparecem componentes de frequência que não estavam no sinal analógico original. Essa condição indesejável é uma forma de distorção chamada *aliasing* (LATHI, 2008).

3. MATERIAIS E MÉTODOS

Neste capítulo serão descritos os materiais e métodos utilizados para o desenvolvimento deste trabalho. Inicialmente será mostrado uma visão geral da arquitetura do sistema embarcado proposto para o medidor de consumo. Em seguida os materiais utilizados e por fim uma descrição mais detalhada de como foi implementado as funcionalidades em *hardware* e *software*.

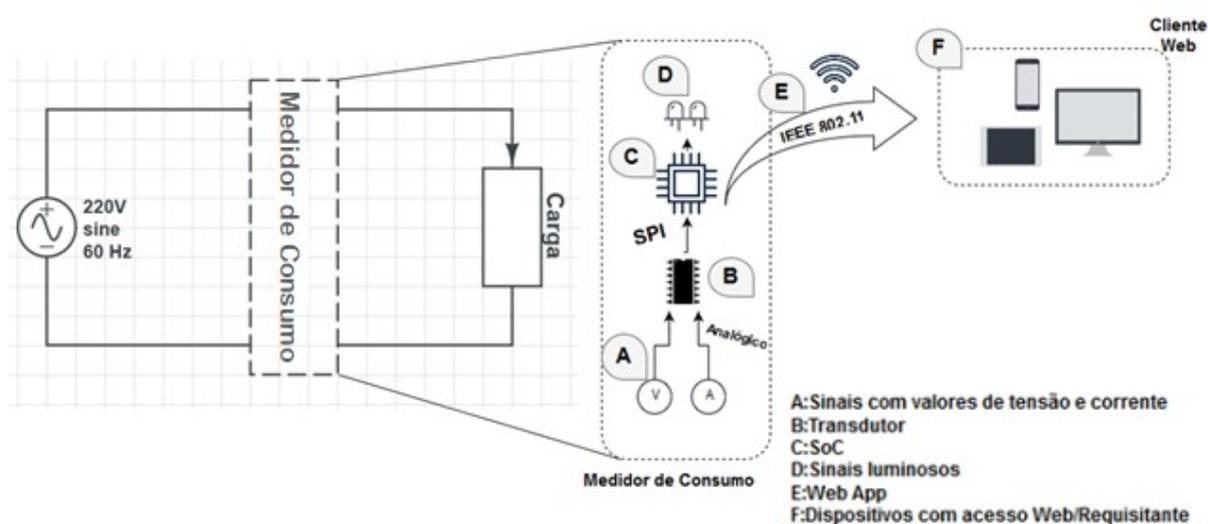
3.1 Visão geral da arquitetura do medidor de consumo

O medidor de consumo proposto possui basicamente os seguintes requisitos funcionais:

- Monitorar valores de corrente e tensão de uma carga ligada à uma rede elétrica monofásica com tensão nominal de 220V a 60Hz.
- Ser compatível com equipamentos resistivos que consumam no máximo 25A.
- Funcionar sem a necessidade de uma fonte externa.
- Exibir informações de potência em tempo real para uma página *web*.

Esses requisitos serviram como base para elaborar a arquitetura proposta do sistema embarcado do medidor de consumo. A visão geral dessa arquitetura é ilustrada na Figura 7, onde é possível observar a distribuição planejada dos componentes a serem utilizados para entregar o modelo com base nos requisitos propostos.

Figura 7 – Arquitetura geral do medidor de consumo.



Fonte: Próprio Autor(2018).

Primeiramente os sinais com valores de tensão e corrente da carga a ser monitorada são recebidos por um transdutor através de uma entrada analógica. Este transdutor é um Arduino Nano, que é uma plataforma de prototipagem eletrônica micro processada, a qual possui os requisitos necessários para leitura de sinais analógicos e tratamento destes para os próximos passos através de uma conversão de sinais analógicos para digital.

Após ter os sinais devidamente adequados, o transdutor irá realizar todos os cálculos necessários para obter valores significantes para o monitoramento. Obtendo valores de tensão e corrente, é possível calcular potência e energia consumida.

Por fim, estas informações serão passadas para uma página *web* embarcada em um SoC, o qual também é capaz de rotear uma rede Wi-Fi. O SoC em questão será o NodeMCU ESP8266MOD, o qual possui os requisitos necessários para processar as comunicações. Dispositivos com acesso *web* poderão conectar-se na rede do SoC e requisitar informações das grandezas elétricas para de fato realizar a análise de energia com os dados exibidos na tela do dispositivo de forma organizada e com acompanhamento em tempo real.

3.2 Implementação das funcionalidades

O desenvolvimento do projeto foi iniciado com a construção do protótipo do medidor de consumo. Ele se constitui na implementação do código de programação e montagem de dois circuitos elétricos: circuito de alimentação e circuito de comunicação. A construção do protótipo foi realizada em uma placa de fenolite perfurada de dimensões 12x18 centímetros utilizando os componentes mostrados na Tabela 1.

Tabela 1–Lista de componentes do protótipo.

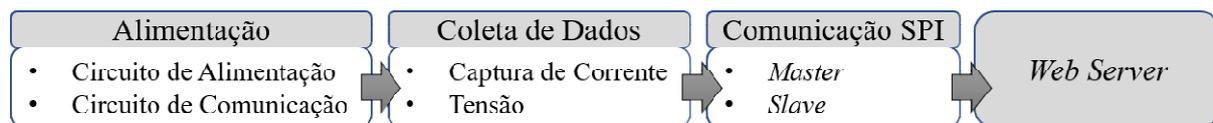
Descrição	Quantidade
Arduino Nano ATmega328P	1
Capacitor 1000 μ F	3
Diodo 1n4007	4
ESP8266MOD	1
LED	1
Regulador L7805CV	1
Resistor 1k Ω	4
Resistor 2k Ω	5
Resistor 3k3 Ω	1
Resistor 8k2 Ω	1
Sensor de Corrente ACS712ELCTR-30A-T	1

Transformador Bivolt 12V 1A	1
Zener 5.1V 250mW	1

Fonte: Próprio Autor (2018).

A implementação das funcionalidades em *software* e *hardware* foi dividida em quatro partes: alimentação, coleta de dados, comunicação *SPI*, *web Server*. As etapas da implementação podem ser resumidamente visualizadas na Figura 8.

Figura 8 – Etapas da implementação das funcionalidades.

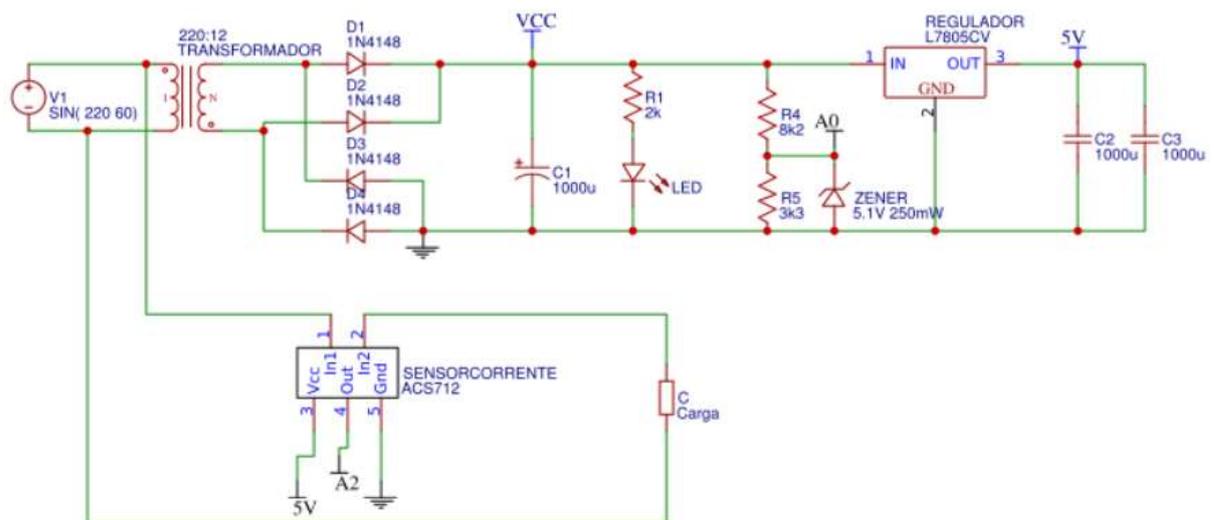


3.2.1 Alimentação

3.2.1.1 Circuito de Alimentação

Primeiramente iniciou-se o circuito de alimentação. Seguindo o circuito da Figura 9, podemos observar seu funcionamento. Temos a tensão de entrada da rede, que fornece corrente alternada em 60Hz com 220V e alimenta dois circuitos: a carga e o medidor. No medidor temos um transformador de 12V para reduzir a tensão na placa de modo a trabalhar de forma segura, sem sobreaquecer os componentes.

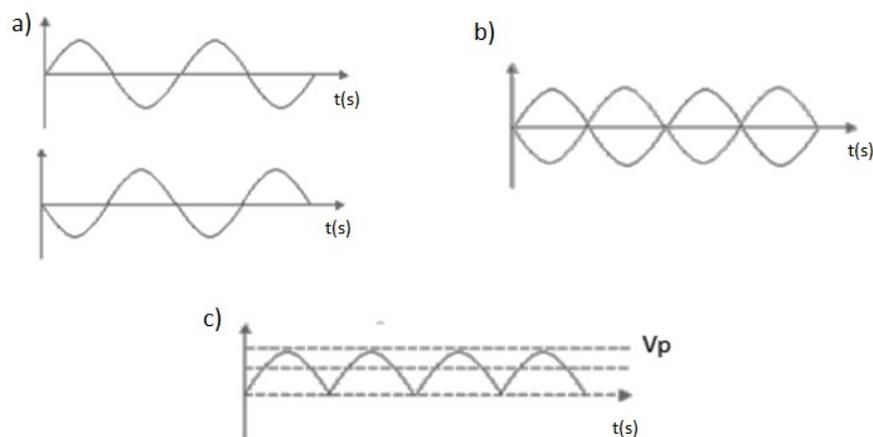
Figura 9– Circuito de Alimentação



Fonte: Reprodução do Autor no software EasyEDA(2018).

Logo em seguida a tensão passa por 4 diodos ligados em forma de ponte retificadora². Quando a tensão de saída do transformador é positiva, os diodos D1 e D3 são polarizados e a corrente flui através deles. Durante outro meio ciclo, quando a saída é negativa, D2 e D4 são polarizados diretamente e a corrente entra através de D2 e sai através de D4 para a fonte. A ponte fornece corrente para a carga durante os dois semicírculos (AHMED, 2008). Desta forma, conforme a Figura 10, é possível obter uma corrente com forma de onda mista entre os semicírculos defasados tornando com característica similar a uma onda contínua.

Figura 10 – a) Formas de onda dos dois pares de diodo respectivamente. b) Forma de onda mista. c) Forma de onda de saída da ponte com característica similar a onda contínua.



Fonte: Próprio Autor(2018).

Com a tensão retificada e reduzida, é colocado um LED para indicar o funcionamento do circuito e em seguida realizado um divisor de tensão para que o nó A0 possa servir como ponto de leitura de tensão do Arduino. A relação do divisor de tensão foi calculada para obter tensões menores que 5V no ponto A0, como calculado na equação 5. Os valores dos resistores foram selecionados com base dos disponíveis no mercado, portanto a relação de R_4 e R_5 foi determinada de forma a obter um V_{out} com valores menores que 5.

$$V_{out} = V_{in} \times \frac{R_5}{R_4 + R_5} \quad (5)$$

$$V_{out} = 15,57 \times \frac{3300}{8200 + 3300} \quad (6)$$

$$V_{out} = 4,47V \quad (7)$$

² Para melhor detalhamento do funcionamento da ponte retificadora de onda completa, consultar AHMED, Ashfaq. Eletrônica de potência.

Sendo V_{in} a tensão no ponto através da medição de um multímetro e V_{out} a tensão no ponto $A0$. Sabendo que a rede doméstica pode ter picos de tensão e não existe nenhum regulador até esse ponto, foi dimensionado um Zener de 5,1V e 250mW para proteger o ponto $A0$ de surtos de tensão maiores de 5V, visto que esse ponto é conectado à porta analógica do Arduino.

O circuito é concluído pela presença do regulador de tensão $L7805CV$ que disponibiliza uma tensão de 5V regulada em sua saída tornando esse nó o ideal para alimentar toda demanda de $5V_{CC}$ do projeto. Dois capacitores de $1000\mu F$ foram adicionados em paralelo para reduzir os possíveis ruídos.

3.2.1.2 Circuito de Comunicação

Para realizar a coleta de dados é necessário que o transdutor e o SoC estejam alimentados com $5V_{CC}$ assim como os sensores. Porém, existe a necessidade de realizar a ligação dos pinos que irão realizar a comunicação SPI entre eles. Todas as portas do Arduino Nano são 5V, porém o ESP8266 são de 3,3V e tolerantes até o máximo de 3.9V, levando a necessidade de se realizar um divisor de tensão para cada uma das ligações. A equação 8 mostra os valores de resistores utilizados nos divisores de tensão, sendo V_{in} os 5V do Arduino e o V_{out} a tensão desejada de 3,3V para o ESP8266.

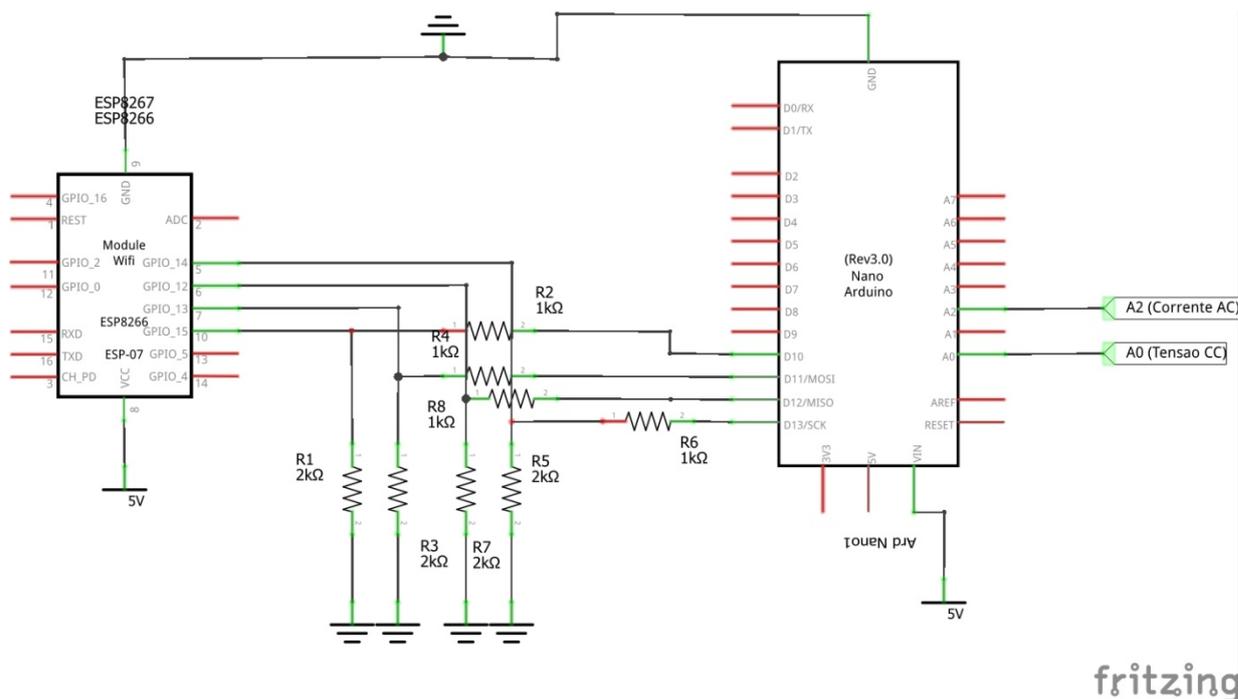
$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2} \quad (8)$$

$$V_{out} = 5 \times \frac{2000}{1000 + 2000} \quad (9)$$

$$V_{out} = 3,33 V \quad (10)$$

O circuito está representado na Figura 11. A forma detalhada de funcionamento desta comunicação será explicada na subseção 3.2.3.

Figura 11– Circuito de Comunicação

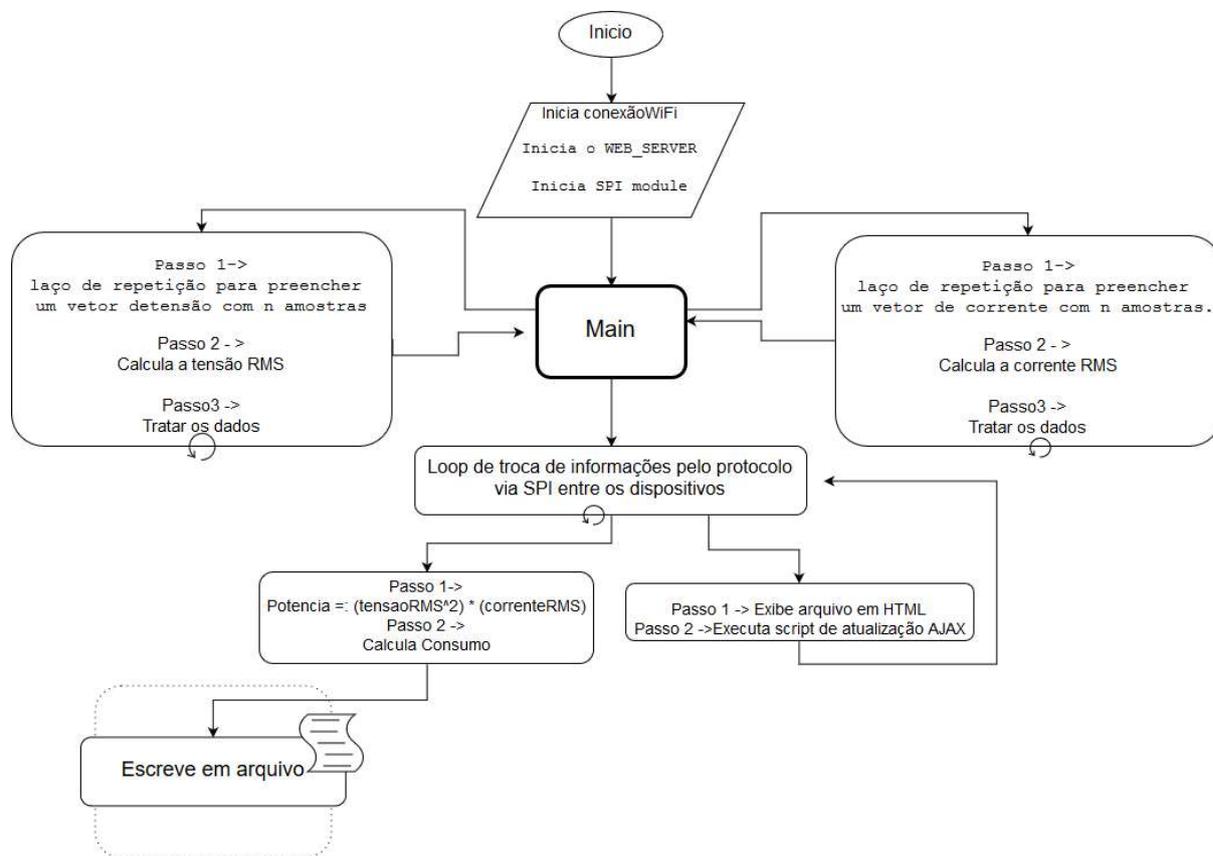


Fonte: Reprodução do Autor no software Fritzing(2018).

3.2.2 Coleta de dados

A coleta de dados vem como funcionalidade seguinte à alimentação do medidor de consumo. No apêndice podem ser encontrados os códigos de algumas funções que serão detalhadas posteriormente nesta subseção. Estes estão distribuídos de forma independente, onde posteriormente podem ser implementadas nas suas respectivas aplicações. O algoritmo base de funcionamento pode ser visualizado na Figura 12, onde é possível analisar de forma geral a lógica de funcionamento do sistema embarcado do medidor. É representado de forma unificada sem considerar as distinções entre as funções do Arduino e do ESP8266, sendo apenas uma representação lógica. Existem operações simultâneas de cálculo de tensão e corrente RMS, juntamente com execução do protocolo de comunicação e *web Server*. As operações de leitura e tratamento de dados é realizada completamente pelo Arduino Nano, já o *web Server* para monitoramento roda inteiramente no ESP8266. A integração entre estes dispositivos é feita em conjunto pelo *SPI*.

Figura 12 – Fluxograma do algoritmo de funcionamento da coleta de dados.

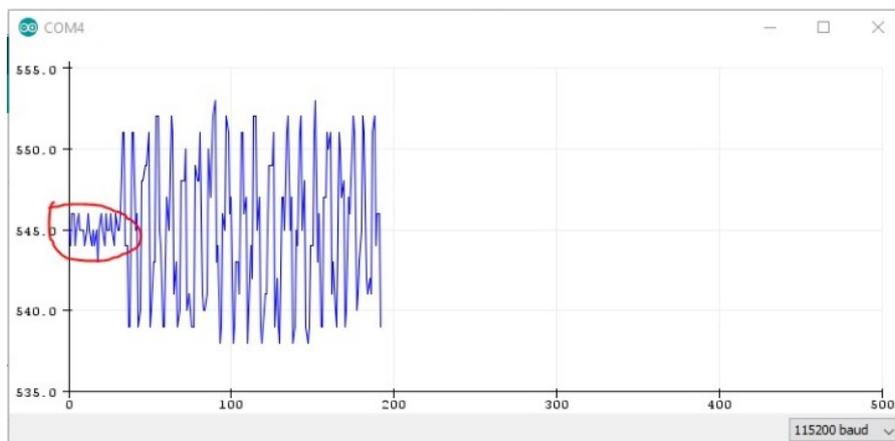


Fonte: Reprodução do Autor(2018).

3.2.2.1 Captura de Corrente

As amostras de corrente coletadas pelo sensor de corrente ACS712 no pino 2 do Arduino exibem valores sem calibração adequada. Na Figura 13 é possível observar os valores de retorno da função *analogRead(A2)* do Arduino quando o sensor lê a corrente de um ventilador. É importante notar que o valor destacado corresponde ao período em que o ventilador estava desligado, ou seja, deveria estar com valores próximos a 0.

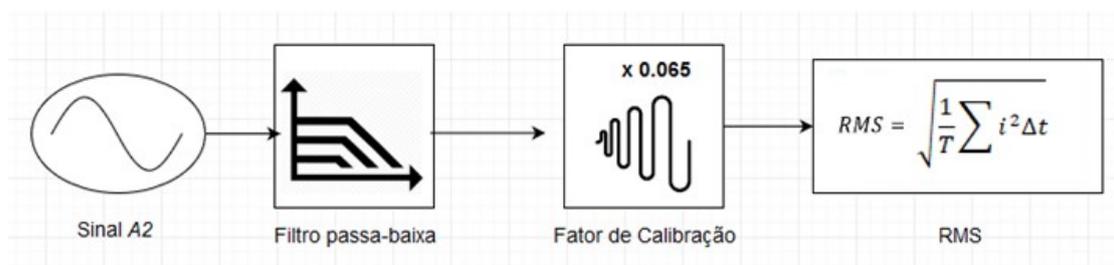
Figura 13 – Ferramenta *Serial Plotter* do Arduino exibindo os valores de corrente lido pela porta do *A2*. Está destacado o valor quando o ventilador estava desligado.



Fonte: Reprodução do Autor utilizando software Arduino(2018).

Para tratar este valor, foi necessário realizar 3 operações sobre o valor de entrada em *A2* para obter o valor de corrente devidamente tratado. O fluxograma das operações pode ser encontrado na Figura 14 e o código de programação utilizado pode ser visto no Apêndice E.

Figura 14 – Fluxograma de operações sobre sinal de *A2*.

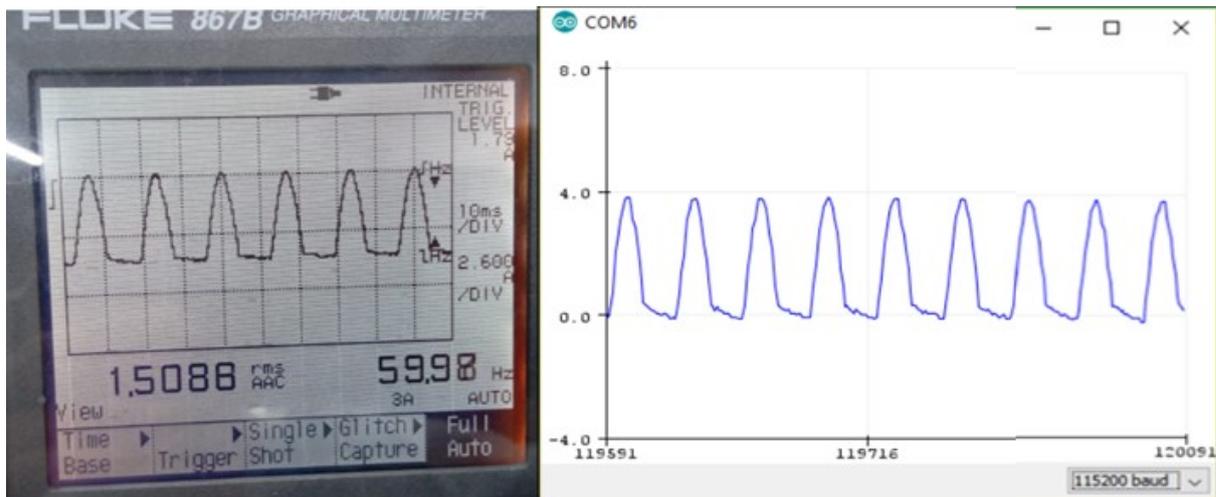


Fonte: Próprio Autor(2018).

Primeiramente foi aplicado um filtro passa baixa para remover todo sinal abaixo do valor referencial do sinal 545, que seria o valor equivalente ao valor 0 de corrente como visto na Figura 13.

Em seguida é feito um produto com o fator de calibração, valor este definido a partir da tentativa de replicar a forma de onda encontrada ao medir a corrente em *A2* com o multímetro gráfico 867b produzido pela Fluke Corporation. A Figura 15 exibe a comparação da forma de onda de *A2* exibida pelo 867b e o Arduino após aplicado o fator de calibração e filtro passa-baixa.

Figura 15 – Forma de onda sinal de A2. À esquerda é a reprodução do Fluke 867b e à direita a reprodução do Arduino com calibração.



Fonte: Reprodução do próprio Autor(2018).

Como último passo é aplicado o cálculo de *RMS* sobre o sinal. Como se trata de um conjunto de valores discretos, podemos utilizar estruturas de repetição para definir o período da função através da relação da equação 11, onde N representa o número de interações da estrutura de repetição e Δt o tempo entre cada interação.

$$T = N \times \Delta t \quad (11)$$

A integral da fórmula do cálculo de *RMS* para valores discretos pode ser representada como somatório, como visto na equação 12.

$$RMS = \sqrt{\frac{1}{T} \int_0^T i^2(t) \times dt} = \sqrt{\frac{1}{T} \sum i^2 \Delta t} \quad (12)$$

Desenvolvendo a equação, vamos obter operação para obter a variável *currentrms* que seria o próprio valor de *RMS*, representado na equação 13.

$$RMS = \sqrt{\frac{\sum i^2}{N}} \quad (13)$$

Desta forma é concluído o tratamento do valor de corrente obtido pelo sensor.

3.2.2.2 Captura de Tensão

Para obter o valor de tensão o vetor *sum_voltage* com 100 espaços recebe as amostras de tensão coletadas pelo pino 0 do Arduino diretamente do divisor tensão do circuito como pode ser visto no Apêndice F. Um diagrama de blocos do circuito representado na Figura

16 simplifica a identificação das etapas. Através de funções de repetição são armazenadas as amostras no vetor e coletado o maior valor para ser tratado de forma que possa ser utilizado.

Primeiramente é necessário converter o valor que se encontra entre 0 e 1023, para o valor de tensão no Arduino. Para isso, por meio de uma regra de 3 simples, é possível encontrar esta relação e desenvolver, como pode ser visto nas equações 14 - 16.

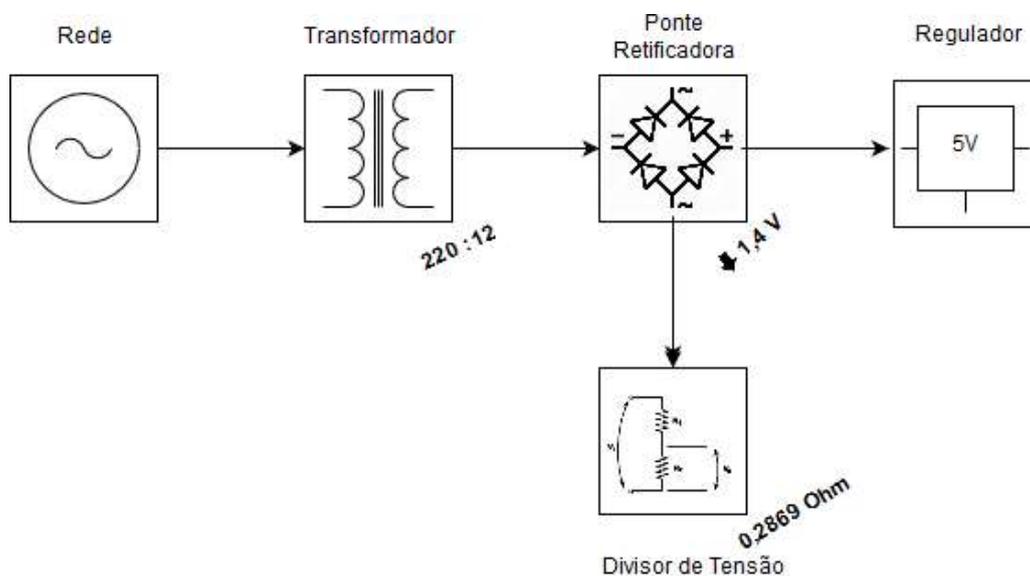
$$\begin{aligned} 1023 &\rightarrow 5 \\ A &\rightarrow X \end{aligned} \quad (14)$$

$$X = \frac{A \times 5}{1023} \quad (15)$$

$$X = A \times 0,0048828125 \quad (16)$$

Onde A é o maior valor do vetor de amostras e X é o valor que queremos encontrar.

Figura 16 – Diagrama de bloco do circuito destacando a relação de tensão do transformador, queda de tensão da ponte retificadora e a resistência equivalente do divisor de tensão.



Fonte: Próprio Autor.

O valor de tensão obtido em X não pode ser utilizado para calcular a potência da carga pois o mesmo é referente a tensão no divisor de tensão do circuito de alimentação do embarcado. Para obter o valor de tensão real, é necessário fazer a relação de tensão com base da Figura 16. Considerando a queda de tensão da ponte retificadora e adicionando as relações

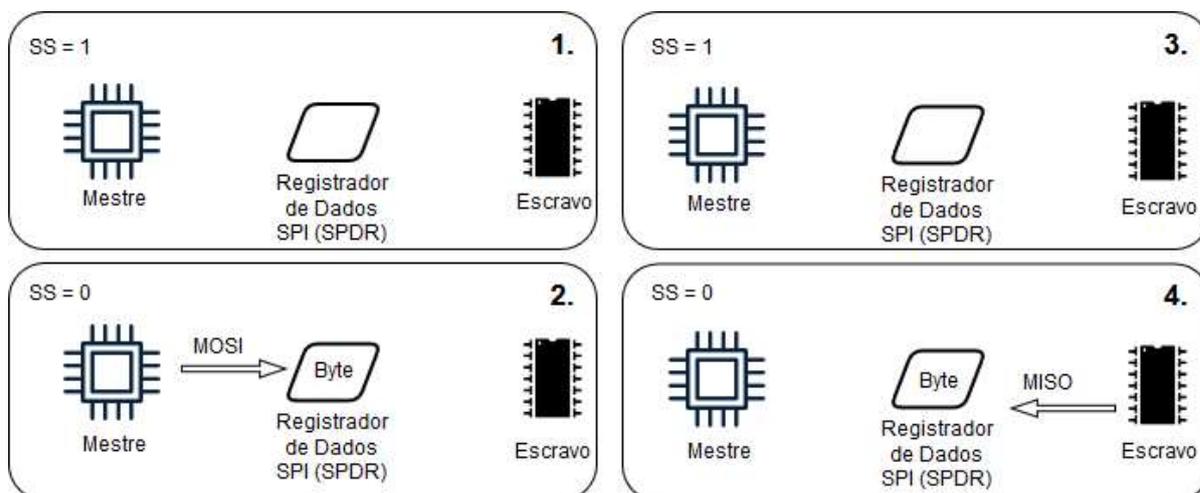
de tensão do divisor e transformador respectivamente obtemos a equação 17. Desta forma conseguimos calibrar o valor de tensão recebido na porta analógica.

$$V = \left[\left(\frac{V_{divisor}}{0,2869} \right) + 1,4 \right] \times \frac{220}{12} \quad (17)$$

3.2.3 Comunicação SPI

A comunicação entre Arduino Nano e ESP8266 utiliza um protocolo de envio de *bytes* pelo *Serial Peripheral Interface (SPI)*, onde a cada interação no *SlaveSelect (SS)* é feita uma troca de informações ordenada entre o mestre e escravo. Esta configuração foi escolhida por ser rápida se comparada com as demais disponíveis entre os dispositivos (BUILDBOT, 2018). A Figura 17 representa o protocolo de comunicação. O mestre envia um *byte* contendo qual informação ele deseja receber do escravo através de um registrador da própria interface *SPI*. Após receber o *byte* e identificar qual o dado requerido, o escravo envia um novo *byte* contendo a informação solicitada.

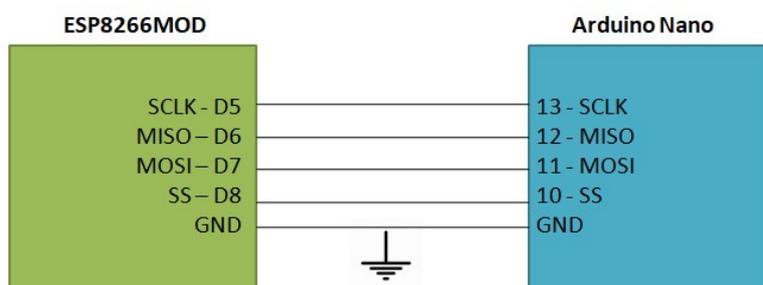
Figura 17 – Representação da troca de informações ordenada por *SPI*.



Fonte: Próprio Autor(2018).

Para realizar esta comunicação foi necessário implementar um circuito de comunicação representado na Figura 10, onde se encontra as ligações entre as portas do ESP8266 e do Arduino Nano, que são mestre e escravo respectivamente seguindo o diagrama da Figura 18.

Figura 18 - Diagrama com as conexões para comunicação *SPI* entre os SoCs.



Fonte: Próprio Autor(2018).

3.2.3.1 Master

Seguindo o código no Apêndice C o ESP8266 possui uma biblioteca com funções próprias para o *SPI*. Primeiramente ele é configurado como Mestre na comunicação, configura a velocidade de *clock* e inicializa a função principal. Dentro da função *voidloop()* ele executa a rotina de coleta de valores de tensão, corrente e potência. A coleta é realizada de forma ordenada, no qual o número de *byte* corresponde a informação desejada conforme a Tabela 2.

Tabela 2–Relação de *bytes* com informação

Ordem da Interação	Byte	Informação
1º	0x01	Tensão
2º	0x02	Corrente
3º	0x03	Potência

Fonte: Próprio Autor (2018).

A cada interação do *slaveslect* o *byte* é enviado e em seguida é devolvido com a informação, tendo um intervalo de 10 microssegundos para garantir que o Arduino processe corretamente e não retorne um valor vazio. As três informações são coletadas e a cada 1 segundo são novamente atualizadas.

3.2.3.2 Slave

O código no Apêndice D é executado no Arduino Nano com a biblioteca de funções dedicadas ao *SPI*. Inicialmente suas configurações são definidas para o modo Escravo na comunicação, os pinos e variáveis devidamente declarados. Através de estruturas

condicionais o valor do *byte* recebido na variável *SPIdata* é comparado de acordo com a Tabela 2 para definir qual informação será enviada.

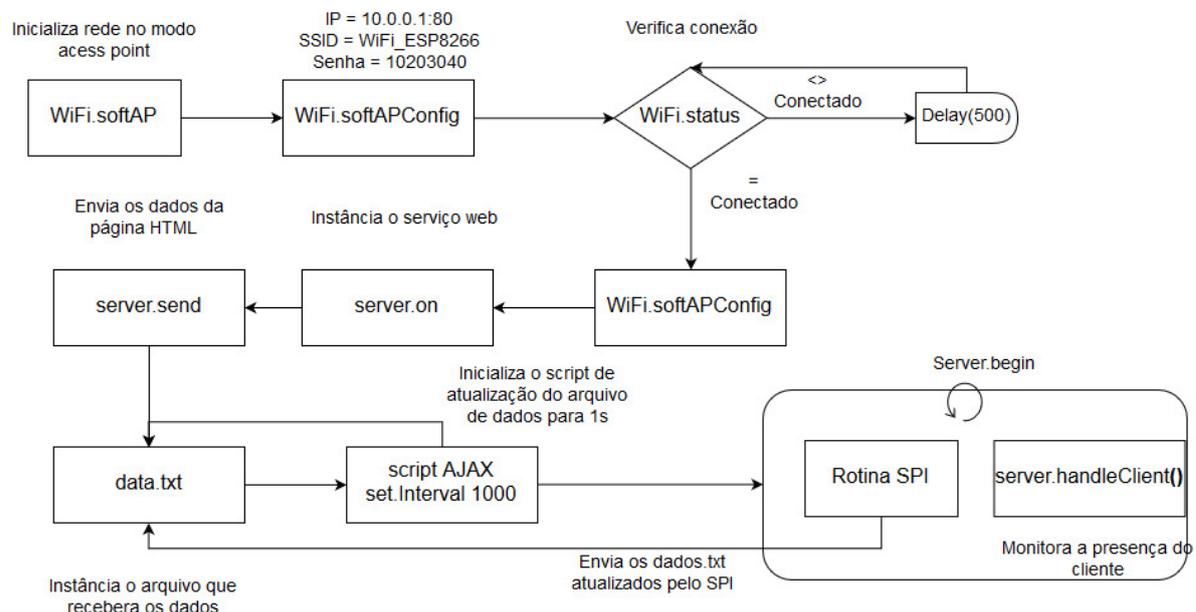
O dado recebido é encaminhado para uma variável temporária onde é feita a operação necessária. As operações são de acordo com a ordem de coleta seguindo a equação 18, onde *Byte_{out}* é o *byte* a ser enviado para o Mestre, *Byte_{in}* é o *byte* recebido, *Dado* é o valor que se deseja enviar e *n* é o número referente à ordem de interação.

$$Byte_{out} = \left(\frac{Byte_{in}}{n} \right) \times Dado \quad (18)$$

3.2.4 WebServer

O ESP8266 possui funções diversas para trabalhar com o protocolo *IEEE 802.1*, entre elas a função de criar uma rede Wi-Fi em modo *access point*. No Apêndice A é possível visualizar as chamadas das funções para configurar essa rede com o *SSID* “*WiFi_ESP8266*”, definido a senha para “10203040” e mantendo o IP fixo em 10.0.0.1 com máscara de rede padrão. O fluxograma da Figura 19 mostra o processo de *webServer* sequenciado.

Figura 19 – Fluxograma do *Web Server*



Fonte: Próprio Autor(2018).

O *web Server* cria um arquivo *data.txt* e exibe esses dados em uma página HTML. No *void loop* do ESP8266 o servidor já preparado fica executando a rotina *SPI* do Apêndice C para receber os valores do Arduino e atualizar no arquivo de texto. A Figura 20 contém o resultado do código.

Para que não seja necessário chamar novamente a execução da instância do *server.on* para que os dados do arquivo *data.txt* sejam exibidos novamente com dados atualizados, é executado o script conhecido como AJAX. Ele permite que as páginas da *web* sejam atualizadas de forma assíncrona, trocando dados XML com um servidor da *web* para transportar informações simples como um arquivo de texto (WOYCHOWSKY, 2007). Isso significa que é possível atualizar partes de uma página da *web*, sem recarregar a página inteira. O código pode ser verificado no Apêndice B.

Figura 20 – Página HTML exibindo o arquivo data.txt



Fonte: Reprodução do autor no software Mozilla Firefox(2018).

4. RESULTADOS E DISCUSSÕES

Neste capítulo é mostrado o protótipo do medidor de consumo desenvolvido e alguns ensaios realizados para validação da proposta.

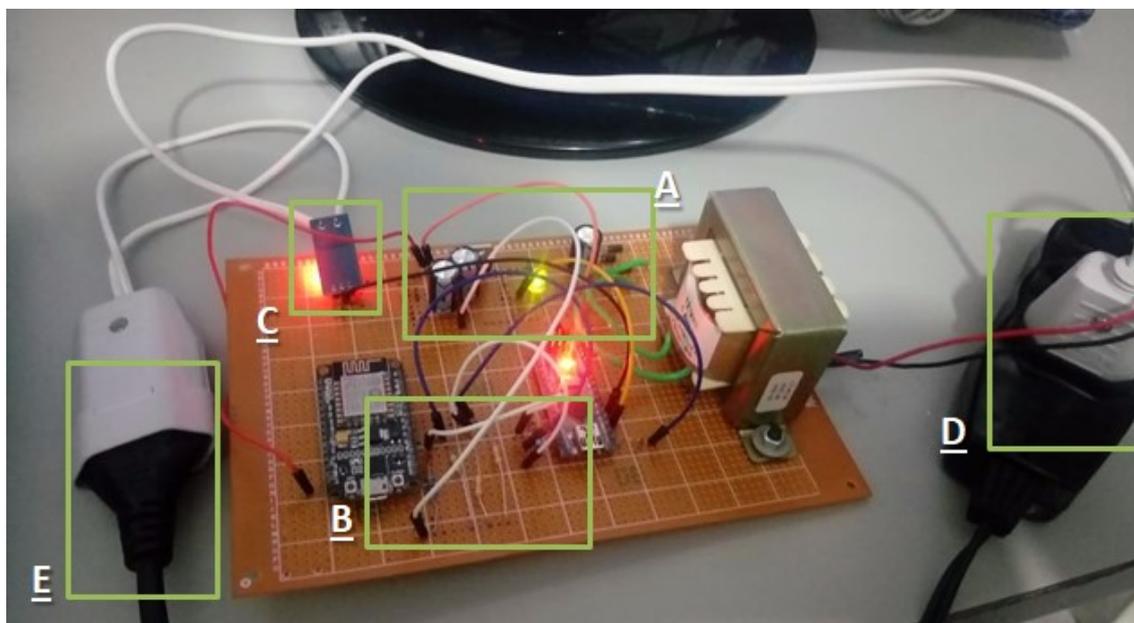
O protótipo construído no capítulo anterior tem finalidade de validar as simulações, diagramas de circuitos e lógica de algoritmos desenvolvidos para cumprir os requisitos de projeto. Após concluído todo o desenvolvimento, os seguintes ensaios foram realizados:

- a) Montagem do circuito e coleta de informações.
- b) Monitoramento do consumo.
- c) Assertividade das informações reproduzidas.

4.1 Montagem do Circuito e coleta de informações.

A montagem do circuito ficou de forma simples com inserção dos componentes em uma placa de fenolite perfurada utilizando solda para mantê-los fixos e realizar a trilha para condução de corrente. Sem muitos acabamentos, os circuitos na montagem final se encontram na Figura 21 onde *A* representa o circuito de alimentação, *B* circuito de comunicação, *C* sensor de corrente, *D* entrada da rede e *E* o carregador de um notebook como carga.

Figura 21 – Protótipo do projeto



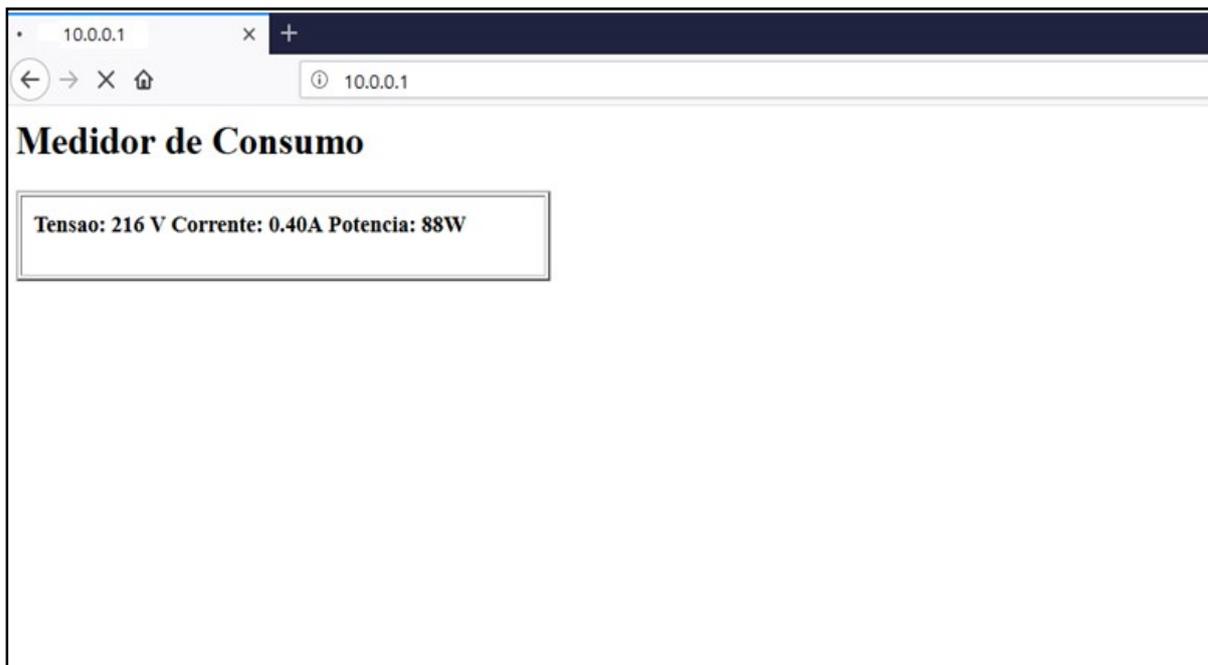
Fonte: Próprio Autor

Com auxílio de um multímetro, foi constatado valores de tensão chegando corretamente aos pontos do circuito e os sinais luminosos presentes nos dispositivos evidenciam o correto funcionamento do circuito.

4.2 Monitoramento do consumo.

Seguindo a validação anterior, o ponto E da Figura 21 indica um carregador de notebook ligado no medidor, representando a carga do circuito. Desta forma, o *web Server* deve retornar o consumo em uma página HTML.

Figura 22 – Página HTML exibindo o arquivo data.txt



Fonte: Reprodução do autor no software Mozilla Firefox(2018).

Na Figura 22 o medidor está indicando valor potência provenientes da leitura dos valores de tensão e corrente da carga evidenciando o seu correto funcionamento. Como se trata de uma carga não linear e sujeita a influência de ruídos na rede, os valores de corrente e tensão podem variar e a cada segundo são exibidos novos valores após o script AJAX requisitar os valores dos dados.

4.3 Assertividade das informações reproduzidas.

Para esta validação, no laboratório de eletromecânica da Universidade Federal do Ceará no *Campus* de Sobral foram realizados 3 ensaios utilizando como comparação o analisador de qualidade de energia ET-5060C da Minipa Indústria e Comércio Ltda. Este analisador possui funções de métrica de consumo de energia em cargas trifásicas ou monofásicas, como pode ser visto na Figura 23. A fim de melhor reproduzir uma aplicação em uma rede residencial, os ensaios foram realizados com as seguintes cargas monofásicas:

1. Banco de carga resistiva 2740Ω
2. Banco de carga resistiva 2740Ω e 1x Lâmpada Incandescente
3. Banco de carga resistiva 2740Ω e 2 x Lâmpadas Incandescentes

Figura 23 – Interface da função de leitura de tensão e corrente RMS. Deve ser considerado apenas os valores de L1 por tratar-se de uma carga monofásica.



Fonte: Próprio autor.

4.3.1 Ensaio 1

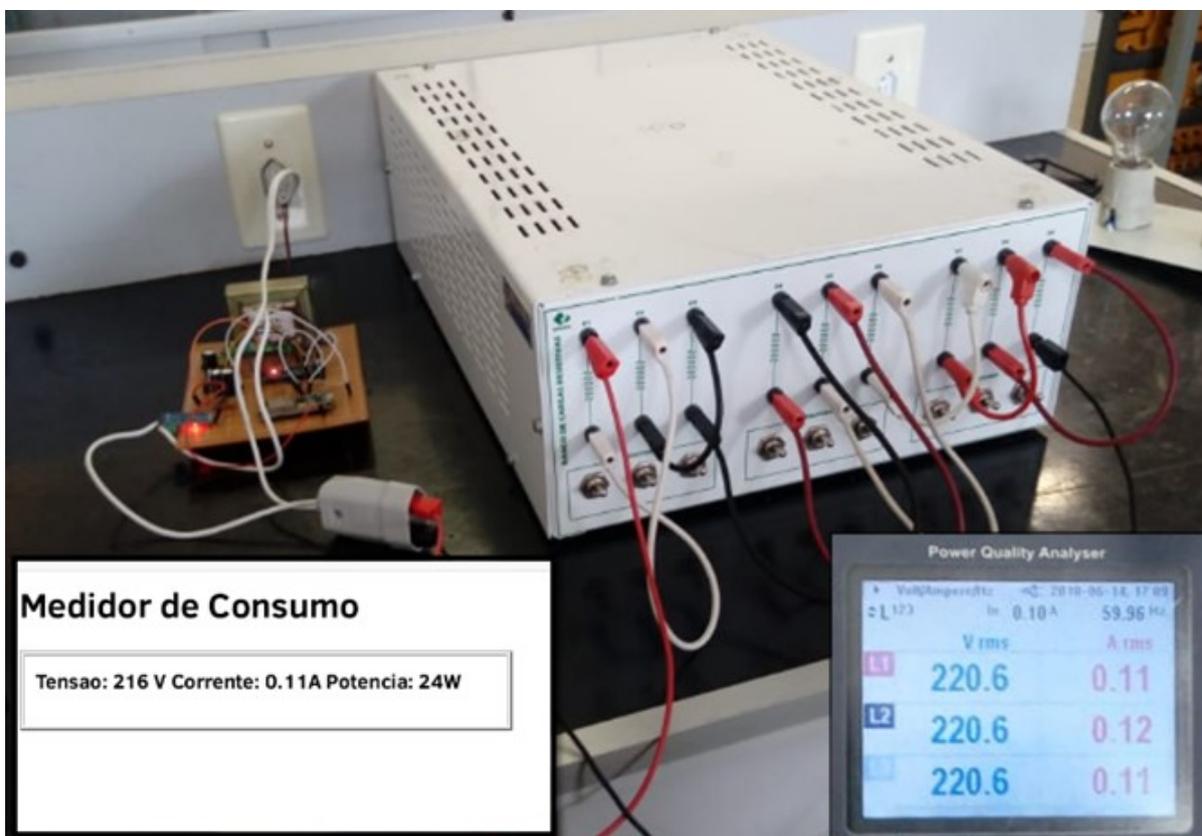
Para o ensaio foi utilizado um banco de cargas resistivas ligadas em série conforme a Figura 24. O banco de carga resistiva utilizado consiste de um conjunto de 9 resistores com valores nominais de 330 Ω . Para verificar o valor total de resistência do banco, foi calculado a soma de cada resistência utilizando um multímetro, obtendo desta forma a equação 19.

$$R_{eq} = R_1 + R_2 + R_3 + R_4 + R_5 + R_6 + R_7 + R_8 + R_9 \quad (19)$$

$$R_{eq} = 270 + 280 + 280 + 280 + 270 + 270 + 280 + 270 + 270 \quad (20)$$

$$R_{eq} = 2470 \Omega \quad (21)$$

Figura 24 – Montagem do primeiro ensaio em bancada. Na imagem exibe a ligação do banco de carga resistiva com o protótipo e os resultados.



Fonte: Próprio autor.

O primeiro ensaio teve como resultado a Tabela 3, onde é possível ver que o valor de potência do protótipo comparado com o ET-5060C foi muito satisfatório.

Tabela 3 – Resultados do primeiro ensaio.

Informação	Protótipo	ET-5060C	Erro Relativo
Tensão	218 V	220,7 V	-2,7
Corrente	0,11 A	0,11 A	0,0
Potência	24 W	24,2 W	0,2

Fonte: Próprio Autor (2018).

4.3.2 Ensaio 2

O segundo ensaio foi realizado de forma similar ao ensaio 1 porém adicionando ao circuito uma lâmpada incandescente. A ligação foi feita em paralelo devido a alta queda de tensão dos resistores no qual não iria alimentar a lâmpada o suficiente. Na Figura 25 é possível observar a montagem em bancada.

Figura 25 – Montagem do segundo ensaio em bancada e os resultados.



Fonte: Próprio autor.

O resultado está representado na Tabela 4, onde é possível ver o valor de potência divergindo entre os equipamentos. É importante destacar que os ensaios não foram feitos simultaneamente, o que pode justificar a variação de tensão devido ao consumo variável do prédio da universidade. Esta variação ocorre devido a influência de ruídos na rede que pode gerar um fenômeno de defasagem entre as ondas da tensão e da corrente.

Tabela 4 – Resultados do segundo ensaio.

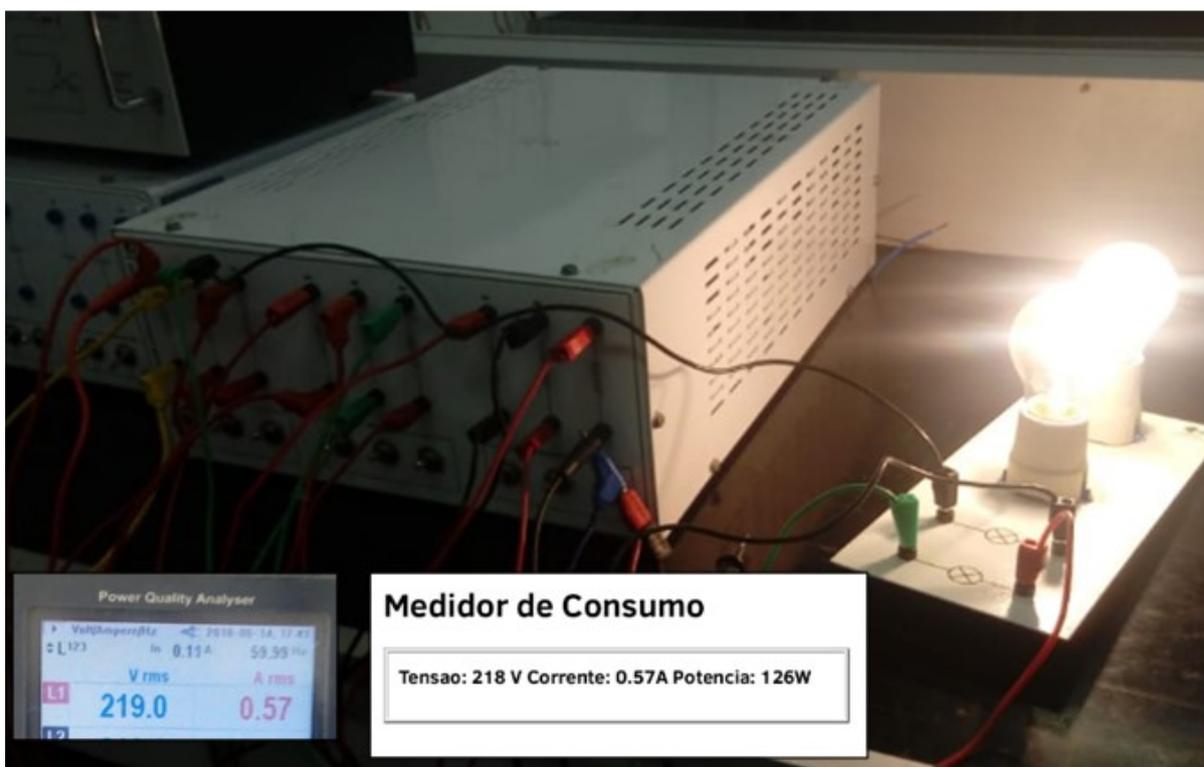
Informação	Protótipo	ET-5060C	Erro Relativo
Tensão	217 V	219,40 V	-2,40
Corrente	0,33 A	0,31 A	0,02
Potência	72 W	68,32 W	3,68

Fonte: Próprio Autor (2018).

4.3.3 Ensaio 3

Para o último ensaio, foi adicionado uma segunda lâmpada ao circuito também em paralelo, que pode ser visto na Figura 26.

Figura 26 – Montagem do terceiro experimento em bancada e os resultados.



Fonte: Próprio autor.

A Tabela 5 exibe os resultados. Existe divergência nos resultados de potência, porém novamente ocasionado pela variação de tensão.

Tabela 5 – Resultados do terceiro ensaio.

Informação	Protótipo	ET-5060C	Erro Relativo
Tensão	218 V	219 V	
Corrente	0,57 A	0,57 A	
Potência	126 W	124,83 W	

Fonte: Próprio Autor (2018).

É importante destacar o protocolo de comunicação utilizado na coleta de dados. Variações de tensão e corrente em intervalos menores que 1 segundo podem gerar valores de potência não correspondentes aos valores de tensão e corrente exibidos, e sim a valores que ocorreram após a interação. Apesar de ser um protocolo síncrono, a coleta dos valores de tensão, corrente e potência são assíncronos.

5. CONCLUSÕES

Neste trabalho foi realizada a construção de um medidor do consumo de energia utilizando conhecimentos de sistemas embarcados e plataformas abertas, como por exemplo o Arduino. A partir das simulações foi desenvolvido um protótipo funcional do projeto, o qual foi submetido a ensaios para validar seu funcionamento na medição do consumo de uma carga em rede residencial.

Foi realizado a comunicação de um SoC com amperímetros e voltímetros, onde o circuito de alimentação e comunicação são funcionais e troca-se dados de tensão e corrente entre dois SoCs de forma síncrona. Através das informações coletadas é possível gerar outras informações mais precisas, como por exemplo valor de corrente RMS e potência consumida pela carga. Tudo isso podendo ser monitorado diretamente de um *web Server* através de uma rede local. Desta forma é possível concluir que os objetivos do trabalho foram atingidos.

Mesmo se referindo a um simples protótipo, os resultados foram expressivos ao comparado com um analisador de qualidade já consolidado no mercado. As informações coletadas nos ensaios em laboratório foram muito próximas do real aumentando a credibilidade do dispositivo.

Para futuros trabalhos é interessante adicionar novas funções ao software, como um cadastro de alarmes de consumo, um *web Server* mais avançado com página HTML mais robusta, reconstruindo os circuitos em uma placa de cobre com maior qualidade e menores dimensões, etc.

6. REFERÊNCIAS

ABOBA, B.; CALHOUN, P. **RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)**. 2003

ACROBOTIC, **DevKit (ESP8266): Getting Started**, Disponível em <<https://learn.acrobotic.com/tutorials/post/esp8266-getting-started>>. Acesso em 19/06/2018

AHMED, Ashfaq. **Eletrônica de potência**. Pearson Education do Brasil, 2008.

Alliance to Save Energy, **Top 5 Reasons To Be Energy Efficient**, 2012.<<http://www.ase.org/resources/top-5-reasons-be-energy-efficient>> Acesso 05/10/2017

ALOI, R. **Códigos da Aula**. Serial Link, 2013.

ALOIA, E. J. **Sistematização crítica das tendências de padronização de arquitetura de protocolos em redes ópticas**. São Carlos: sn, 2003.

ANDERSEN, Ø. L. **Security of Internet of Things Protocol Stacks**. 2016. 105 p. Dissertação (Department of Telematics) — Norwegian University of Science and Technology.

Arduino Nano Documentation, Disponível em <<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>>. Acesso em 13/10/2017

ARDUINO. **Arduino Uno**. Disponível em: . Acesso em: 8 de Abril de 2017.

ASHTON, K. **That ‘Internet of Things’ thing**. Publicano no RFID Journal, 2009.

ATZORI, L; IERA, A; MORABITO, G. **The Internet of Things: A survey**. Computer Networks, [s.l.], v. 54, n. 15, p.2787-2805, out. 2010. Elsevier BV.

BANZI, M.; SHILOH, M. **Primeiros Passos com o Arduino–2.ed**. São Paulo: Novatec Editora Ltda, 2015.

BARROS, A. **Edifícios Inteligentes e a Domótica: Proposta de um Projeto de Automação Residencial utilizando o protocolo X-10**. 2010. 105 f. Monografia (Graduação) - Curso de Licenciatura em Engenharia de Construção Civil, Universidade Jean Piaget de Cabo Verde, Cidade da Praia, 2010.

BARTH, D.; SIEWERT, V. **Lightweight Directory Access Protocol**. Florianópolis, 2005. Artigo Científico. Serviço Nacional de Aprendizado Industrial – Santa Catarina.

- BATES, R. J. **Broadband Telecommunications Handbook**. 2 nd ed. Phoenix: McGraw - Hill telecommunications, 2002.
- BISSEL, C. C.; CHAPMAN, D. A. **Digital signal transmission (2nd ed.)**. Cambridge University Press. p. 64. 1992.
- BOLZANI, C. A. M. **Residências inteligentes**. São Paulo, Ed. Editora e Livraria da Física, 2007.
- BOYLESTAD, R. L; NASHELSKY, L. **Dispositivos eletronicos e teorias de circuitos**. 8ª ed. São Paulo: Pearson Prentice Hall, 2004 671 p.
- BREVE, M. M; BERNUY, M. A. C. **Introdução ao Desenvolvimento de Sistemas com Microcontroladores**. In: ANAIS III Seminário de Pesquisa Jr (SepesqJr). Universidade Tecnológica Federal do Paraná. (2012)
- BUILDBOT, **Interfaces de comunicação OneWire, SPI e I2C**. Disponível em:<<http://buildbot.com.br/blog/interfaces-de-comunicacao-onewire-spi-i2c/>>. Acesso em 13/01/2018
- CAMPOS, A. **Arduino e cartões SD, MicroSD e SDHC**. BR-Arduino.org, 2015.
- CARRO, L., WAGNER, F. R. **Sistemas Computacionais Embarcados**. Jornadas de Atualização em Informática, XXII JAI 2003, v. 1, p. 45-95. Campinas: UNICAMP, 2003.
- CARTWRIGHT, K. V. "**Determining the Effective or RMS Voltage of Various Waveforms without Calculus**" Technology Interface. 8 (1): 20 pages. 2007.
- COSTA, D.; GRUNDSTEIN, A. **An Analysis of Children: Left Unattended in Parked Motor Vehicles in Brazil**, 2016.
- DULANEY, E. **CompTIA Security+**: Study Guide, 5. ed., 186-187 USA: Sybex Wiley, 2011
- EAGLE.**Eagle PCD Design**. Disponível em: . Acesso em: 25 de set. 2016.
- FAIRCHILD. **High Conductance Fast Diode**. (2016).
- FEKI, M. A. et al. **LievenThe Internet of Things: The Next Technological Revolution**. IEEE Computer Society, 2013
- FLOP, F. E. **Kit modulo leitorRFId MFRC522**. Filipe e Flop, 2016.

FRIEDEWALD, M.; RAABE, O. **Ubiquitous computing**: An overview of technology impacts. TelematicsandInformatics, 2011

Gajski, D.D.; Abdi, S.; Gerstlauer, A.; Schirner, G. **Embedded System Design - Modeling, Synthesis and Verification**, XXV, 352p, Springer, ISBN 978-1-4419-0503-1, 2009.

GHETTI, F. T. **Análise de variações topológicas aplicadas a uma UPS tipo delta monofásica**. Dissertação (Mestrado), Universidade Federal de Juiz de Fora, MG, Brasil, 2009.

GREENBERG, A. **THE REAPER IOT BOTNET HAS ALREADY INFECTED A MILLION NETWORKS**. Wired. 2017.

GUBBI, J. et al. **Internet of Things (IoT): A vision, architectural elements, and future directions**. Future Generation Computer Systems, 29(7), 1645-1660, 2013.

HART, D. W. **Eletrônica de Potência**: Análise e projetos de circuitos. Porto Alegre: AMGH Editora, 2012.

JUNIOR, E. **Sistema de segurança para cadeira de bebê controlado por aplicativo computacional**. Taquara – RS. 2014

KITCHIN, R.; DODGE, M. **Code/Space**: Software and Everyday Life. Cambridge, MA, MIT Press, 2011.

LABCENTER. **Proteus Design Suite 8.5**. (2016).

Lalitnarayan, K.; Rao, K. M.; Sarcar, M.M.M. **Computer Aided Design and Manufacturing**, 2008.

Lathi, B. P. **Sinais e sistemas lineares**. Tradução Gustavo Guimarães Parma. – 2. ed. – Dados eletrônicos. – Porto Alegre: Bookman, 2008.

LEE, Edward Ashford; SESHIA, Sanjit A. Introduction to embedded systems: A cyber-physical systems approach. MIT Press, 2016. Disponível em <http://leeseshia.org/releases/LeeSeshia_Digital_V1_08.pdf> Acesso 05/10/2017

LI, Y. et al. **Towards a theoretical framework of strategic decision, supporting capability and information sharing under the context of Internet of Things**. Information Technology & Management. 2012.

MALVINO, A. P. **Eletrônica**. Vol II. São Paulo: McGraw-Hill, 1987.

MENDES, T. D. P. et al. **Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources**. *Energies*, v. 8, n. 7, 2015. ISSN 1996-1073.

MINATEL, P. **ESP8266: O guia básico de hardware**. Pedro Minatel, 2015.

NASTASE, A. S. "How to Derive the RMS Value of Pulse and Square Waveforms". 2015.

Neves, C.; Duarte, L.; Viana, N.; Lucena, V.F.J. **Os Dez Maiores Desafios da Automação Industrial: as Perspectivas Para o Futuro**, II CONNEPI, 2007

NodeMCUDocumentation, Disponível em <<https://nodemcu.readthedocs.io/en/master/>>. Acesso em 13/10/2017

PALANIAPPAN, Satish et al. Home automation systems-A study. **International Journal of Computer Applications**, v. 116, n. 11, 2015. Disponível em <<http://research.ijcaonline.org/volume116/number11/pxc3902601.pdf>>

PATTERSON, D. **Quad Serial Peripheral Interface (QuadSPI) Module Updates (PDF) (Application note)**. Freescale Semiconductor. 2012.

PELL, R. **Improving performance using SPI-DDR NOR flash memory**. EDN. 2011.

PERTENCE JUNIOR, A. **Amplificadores operacionais e filtros ativos**. 6ª edição. Porto Alegre: Bookman, 2003 p. 302.

PRASAD, N. **Aalborg University. ASPire**, 2009.

ROSEN, E. C. et al. **MPLS Label Stack Ecoding**. RFC 3032. 2001.

RUBINSTEIN M.; REZENDE J. **Qualidade de Serviço em Redes 802.11**. In XX Simposio Brasileiro de Redes de Computadores, 2002.

SAMUEL, S. S. I. **A review of connectivity challenges in IoT-smart home**. In: 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC) IS - SN - VO - VL -. [S.l.: s.n.], 2016. p. 1 – 4.

SARDINHA, G. **Virtualização de Servidores**. Voolivrelinux, 2009.

SARIJARI, M. A. et al. **Experimental studies of the ZigBee frequency agility mechanism in home area networks**. In: Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on IS - SN - VO - VL -. [S.l.: s.n.], 2014. p. 711 – 717.

SILVA, W. W. **Estudo e implementação de um conversor bidirecional como interface na regulação de tensão em barramento c.c. e carregamento de baterias em um sistema nano rede.** Belo Horizonte, MG, Brasil, 2013.

SOUSA, J. M. **Sistema bidirecional de carga de baterias para FEUP VEC.** Dissertação (Mestrado) – Faculdade de Engenharia do Porto, Porto, 2013.

TENENBAUM, A. S. **Redes de Computadores.** 4. ed. Rio de Janeiro: Campus Elsevier, 2003.

TERUEL, E. C. **Uma proposta de framework para sistemas de automação residencial com interface para WEB. 2008.** 158 f. Dissertação (Mestrado em Tecnologia: Gestão, Desenvolvimento e Formação) – Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2008.

TRONCO, T. R. **Redes de Nova Geração .1.** ed. São Paulo: Érica, 2006.

VIANA, Augusto Nelson Carvalho et al. **Eficiência energética: fundamentos e aplicações.** Elektro, Universidade Federal de Itajubá, Excen, Fupai, v. 1, 2012.

WITHANAGE, C. et al. **A comparison of the popular home automation technologies.** In: 2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA) IS -. [S.l.: s.n.], 2014. p. 600 – 605. ISSN 2378-8534 VO - VL -.

WOLF, Marilyn Claire - **Computers as Components: Principles of Embedded Computing System Design, Third Edition,** 2012

WOYCHOWSKY, Edmond; WOYCHOWSKY, Edmond. **AJAX: Creating web pages with asynchronous JavaScript and XML.** Prentice Hall, 2007.

APÊNDICE A - INICIANDO HOTSPOT E WEB SERVER NO ESP8266

```

1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3 #include <ESP8266WebServer.h>
4 // Substitua pelos dados desejados para o hotspot Wi-Fi
5 const char* ssid = "WiFi_ESP8266";
6 const char* password = "10203040";
7 ESP8266WebServer server(80); //instancia o servidor web na porta 80 (porta http)
8
9 void setup()
10 {
11     Serial.begin(115200); //baud rate
12     // Criando conexão Wi-Fi com Gateway e SubMask padrão.
13     Serial.print("Connecting to ");
14     Serial.println(ssid);
15     IPAddress apIP(10, 0, 0, 1);
16     WiFi.softAPConfig(apIP, apIP, IPAddress(255, 255, 255, 0));
17     WiFi.softAP(ssid, password);
18     IPAddress apip = WiFi.softAPIP();
19     //Validando sucesso da conexão
20     while (WiFi.status() != WL_CONNECTED) {
21         delay(500);
22         Serial.print(".");
23     }
24     Serial.println("");
25     Serial.print("Connected to ");
26     Serial.println(ssid);
27     Serial.print("IP address: ");
28     Serial.println(WiFi.localIP());
29     Serial.println("");
30     Serial.println("WiFi connected");
31 //
32 void loop(void)
33 {
34     server.handleClient();
35 }

```

APÊNDICE B - EXECUÇÃO DO SERVIDOR, PAGINA HTML E SCRIPT AJAX

```

1 String page = "";
2 String text = "";
3 String data ;
4
5 void setup() {
6   server.on("/data.txt", [](){
7     text = data;
8     server.send(200, "text/html", text);
9   });
10  server.on("/", [](){
11    page = ""; //Insira aqui o código HTML da página. Deve conter o espaço para chamar o arquivo de texto com id="data"
12    //Inserindo na página o Script do código AJAX para a página HTML inserida.
13    page += "<script>\r\n";
14    page += "var x = setInterval(function() {loadData(\"data.txt\",updateData)}, 1000);\r\n";
15    page += "function loadData(url, callback){\r\n";
16    page += "var xhttp = new XMLHttpRequest();\r\n";
17    page += "xhttp.onreadystatechange = function(){\r\n";
18    page += " if(this.readyState == 4 && this.status == 200){\r\n";
19    page += "   callback.apply(xhttp);\r\n";
20    page += " }\r\n";
21    page += "};\r\n";
22    page += "xhttp.open(\"GET\", url, true);\r\n";
23    page += "xhttp.send();\r\n";
24    page += "};\r\n";
25    page += "function updateData(){\r\n";
26    page += " document.getElementById(\"data\").innerHTML = this.responseText;\r\n";
27    page += "};\r\n";
28    page += "</script>\r\n";
29    server.send(200, "text/html", page);
30  });
31  // Iniciando o web server
32  server.begin();
33  Serial.println("Web server running. Waiting for the ESP IP...");
34  delay(100);
35  Serial.println("Web server started!");
36 }

```

APÊNDICE C – TRABALHANDO COM SPI NO ESP8266 (MESTRE)

```

1 #include <SPI.h>
2 #define CS 15 //Define o CS
3
4 void setup()
5 {
6     pinMode(CS, OUTPUT); //Configura o SS como saída, por ser mestre.
7     Serial.begin(115200); //baud rate
8     digitalWrite(CS, HIGH); //Inicializa o SS como HIGH
9     SPI.begin(); //Inicialização do módulo SPI --> configura MOSI, MISO e CLOCK
10    /*Configurações definidas 1. SPI bus speed = 1 MHz 2. Data Out = From MSB bit --> To LSB bit 3. Data Mode = SPI MODE0*/
11    SPI.beginTransaction(SPISettings(1000000, MSBFIRST, SPI_MODE0));
12 }
13 void loop(void)
14 {
15     byte tensao; //Variáveis de Transferencia em byte
16     byte corrente; //tensao, corrente e consumo para receber os respectivos valores. byte (0x01) = tensao ; byte (0x02) = corrente; byte (0x03) = consumo
17     byte potencia;
18     float corrente_fix; //variavel que ira exibir valor ajustado para duas casas decimais - > 0.00 A
19    //OBTER VALORES DE TENSAO E CORRENTE
20    digitalWrite(CS, LOW); //Seta a CS em LOW
21    SPI.transfer(0x01); //Manda o byte (0x01) para o escravo Arduino NANO.
22    digitalWrite(CS, HIGH); //Seta a CS em High
23
24    delayMicroseconds(10); //Tempo insignificante apenas para evitar que o Arduino NANO nao mande de volta valor vazio.
25
26    digitalWrite(CS, LOW); //Seta a CS em LOW
27    tensao = SPI.transfer(0x00); //Receber o valor de tensao do Arduino
28    digitalWrite(CS, HIGH); //Seta a CS em HIGH
29    Serial.println("Tensao: ");
30    Serial.print(tensao); //Escreve o valor de tensao
31    Serial.println("\r\n");
32
33    digitalWrite(CS, LOW); //Seta a CS em LOW
34    SPI.transfer(0x02); //Mandar o byte (0x02) parao escravo Arduino NANO
35    digitalWrite(CS, HIGH); //Seta a CS em HIGH
36
37    delayMicroseconds(10); //Tempo insignificante apenas para evitar que o Arduino NANO nao mande de volta valor vazio.
38
39    digitalWrite(CS, LOW); //Seta a CS em LOW
40    corrente = SPI.transfer(0x00); //Receber o valor de tensao do Arduino
41    digitalWrite(CS, HIGH); //Seta a CS em HIGH
42    Serial.println("Corrente: ");
43    Serial.print(corrente); //Escreve o valor de corrente
44    Serial.println("\r\n");
45    corrente_fix = float (corrente) / 100; //Corrige a exibição da corrente de mA para A.
46    Serial.println(corrente_fix);
47
48    digitalWrite(CS, LOW); //Seta a CS em LOW
49    SPI.transfer(0x03); //Mandar o byte (0x03) parao escravo Arduino NANO
50    digitalWrite(CS, HIGH); //Seta a CS em HIGH
51
52    delayMicroseconds(10); //Tempo insignificante apenas para evitar que o Arduino NANO nao mande de volta valor vazio.
53
54    digitalWrite(CS, LOW); //Seta a CS em LOW
55    potencia = SPI.transfer(0x00); //Receber o valor de tensao do Arduino
56    digitalWrite(CS, HIGH); //Seta a CS em HIGH
57    Serial.println("Potencia: ");
58    Serial.print(potencia); //Escreve o valor de potencia
59    Serial.println("\r\n");
60
61    delay(1000); //Delay de 1s para repetir o processo
62    data = "Tensao: " + (String)tensao + " V Corrente: " + (String)corrente_fix + "A \r\n Potencia: "+(String)potencia+"W"; //Código de atualização do arquivo data.txt
63    server.handleClient();
64
65 }

```

APÊNDICE D – TRABALHANDO COM SPI NO ARDUINO NANO (ESCRAVO)

```

1 #include <SPI.h>
2
3 float current = 0.0;           // variável que recebe o valor calculado de corrente
4 float voltage = 0.0;          // variável que recebe o valor calculado de tensao
5 float power = 0.0;            // variável que recebe o valor calculado de potencia
6 volatile boolean process_it;  //Flag para validar a chegada de dados no Mestre (ESP8266)
7 byte a;                       //Byte auxiliar para armazenar dado
8 void setup(void)
9 {
10     Serial.begin(115200);      //baud rate tem que ser o mesmo do Mestre (ESP8266)
11     SPCR |= bit(SPE);         //Configura ATMEGA328P/Arduino no modo escravo
12     pinMode(MISO, OUTPUT);    //Configura MISO como saída, SlaveOut
13     process_it = false;       //Inicializa a flag para FALSE
14     SPI.attachInterrupt();    //Função para possibilitar interrupção do SPI e realizar o protocolo de comunicação síncrono
15 }
16 // Rotina do Protocolo de Comunicação
17 ISR(SPI_STC_vect)
18 {
19     byte spidata = SPDR;      //Pega o byte vindo do SPI Data Register (SPDR)
20     if (spidata == 0x01)      //Condição para enviar Tensao
21     {
22         a = spidata; //Armazena numa variável auxiliar
23
24         SPDR = spidata * voltage; //processa a informação byte e devolve por SPDR para o Mestre ler
25         process_it = true; //Seta a Flag como TRUE
26     }
27     if (spidata == 0x02) //Condição para enviar corrente
28     {
29         a = spidata;
30         SPDR = (spidata / 2) * (currentrms)*100; //Divisao de acordo com ordem de envio // Corrente é multiplicado por 100 devido ao mestre querer exibir em 0.00A
31         process_it = true;
32     }
33     if (spidata == 0x03) //Condição para enviar Potencia
34     {
35         a = spidata;
36         SPDR = (spidata / 3) * power; //Divisao de acordo com ordem de envio
37         process_it = true;
38     }
39 }
40
41 void loop(void)
42 {
43     if (process_it)           //Condição de envio de dado
44     {
45         Serial.println("Recieved\r\n"); //Caso o mestre receba, é notificado.
46         Serial.println("Voltage:");
47         Serial.println(voltage);
48         Serial.println("Current:");
49         Serial.println(current);
50         Serial.println("Power:");
51         Serial.println(power);
52         process_it = false;    //Seta a Flag pra Falso para reiniciar rotina.
53     }
54 }

```

APÊNDICE E – CAPTURA E TRATAMENTO DE CORRENTE

```
1 void loop() {
2   max_current = 0; //variável que irá receber o valor do somatório.
3   for(int i = 0; i < 300; i++) //estrutura de repetição
4   {
5     //obtenção do sinal de corrente do sensor ACS712
6     current = (analogRead(A2) - 545)*0.065; //aplicando o filtro passa-baixa e o fator de calibração
7     current = current*current; //calcula o quadrado da corrente
8     max_current += current; //recebe o somatório
9     delayMicroseconds(600); //tempo de execução de cada interação
10  }
11  currentrms = sqrt(max_current/300); //Calculando a raiz quadrada do (somatório/número de interações)
12 }
```

APÊNDICE F – CAPTURA E TRATAMENTO DE TENSÃO

```
1 void loop() {
2   for(int j = 0; j < 100; j++) //Estrutura de repetição para armazenar as amostras de tensão da porta analógica A0
3   {
4     sum_voltage[j] = analogRead(A0);
5     delayMicroseconds(600);
6   }
7   for(int j = 0; j < 100; j++) //Estrutura de repetição para encontrar o valor de pico
8   {
9     if(max_voltage_divisor < sum_voltage[j])
10    {
11      max_voltage_divisor = sum_voltage[j];
12    }
13  }
14
15  max_voltage_divisor = max_voltage_divisor * 0.004882812; //Relação proporcional da tensão 5V entre os valores de 0 e 1023 (10 bit).
16  voltage_divisor = max_voltage_divisor;
17  voltage_divisor = voltage_divisor / 1.41421356; //Calcula o RMS (MáxAmostragem dividido por raiz de dois)
18  //calcula para relacionar a tensão de 220V, usando os valores de tensão referentes
19  //ao divisor de tensão (0.2869) somados com a tensão de 1.4 V da ponte de diodo.
20  voltage = (((voltage_divisor / 0.2869) + 0.5) * (220/12));
21 }
```