



FUNDAÇÃO DE ASSISTÊNCIA E EDUCAÇÃO - FAESA

CENTRO UNIVERSITÁRIO ESPÍRITO-SANTENSE / FAESA

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

HELICIO PAULO DA SILVA JUNIOR

JOÃO GABRIEL DA SILVA PEREIRA

**SISTEMA DE MONITORAMENTO DE CONSUMO DE ENERGIA ELÉTRICA NÃO
INVASIVO COM CONEXÃO À WEB**

Artigo do Trabalho de Conclusão de Curso de Graduação em Engenharia Elétrica, apresentado ao Centro Universitário - FAESA, sob a orientação do professor Felipe Machado Lobo.

VITÓRIA

2019

WEB CONNECTED NON-INVASIVE ELECTRIC POWER CONSUMPTION MONITORING SYSTEM

FELIPE MACHADO LOBO**

HELICIO PAULO DA SILVA JUNIOR*

JOÃO GABRIEL DA SILVA PEREIRA*

RESUMO

No presente trabalho de conclusão de curso foi desenvolvido um medidor de potência de baixo custo, conectado à Internet. O projeto do sistema como um todo foi feito utilizando-se conceitos e requisitos definidos por estudos na área da Internet das Coisas. Desenvolveu-se uma arquitetura baseada em serviços hospedados em uma plataforma *open source* chamada Emoncms, a qual processa e envia informações para um aplicativo de smartphones *Android* também *open source* chamado de *Open Energy Monitor*. Em conjunto, foi fabricado um protótipo utilizando principalmente um sensor de corrente não invasivo e um microcontrolador os quais monitoram o consumo de uma lâmpada de filamento de carbono, para fins de testes. Os testes mostraram-se satisfatórios, confirmando a prova de conceito. Aprimoramentos a nível de hardware e software ainda são necessários para obter-se um produto viável comercialmente.

Palavras-chave: *open source*, Emoncms, *Open Energy Monitor*, Medidor de potência.

ABSTRACT

In this course completion article, a low-cost power meter has been developed, connected to the Internet. The system design as it was done using concepts and requirements defined by studies in the area of the Internet of Things. An architecture based on services hosted on an open source platform called EMONCMS was developed, which processes and sends information to an open source Android smartphone app called Open Energy Monitor. Together, a prototype was manufactured using mainly a noninvasive current sensor and a microcontroller which monitor the consumption of a carbon filament lamp for testing purposes. The tests were satisfactory, confirming the proof of concept. Hardware and software enhancements are still needed to achieve a commercially viable product.

Keywords: *open source*, Emoncms, *Open Energy Monitor*, Power Meter

* Graduando de Engenharia de Produção da Instituição FAESA

** Professor MSc. da Instituição FAESA

1 INTRODUÇÃO

A informatização e o alto nível de conectividade dos dias atuais fizeram a necessidade de surgir o conceito de “IOT” (Internet das Coisas) e com isso novos modelos de negócios podem ser criados, contribuindo efetivamente para alavancar diversos setores da economia.

A Indústria 4.0 já é uma realidade em muitos setores da indústria brasileira, os princípios que movem esta nova indústria são o aumento massivo do monitoramento de processos industriais e a inserção do conceito de Internet das Coisas, que integra e interage por meio da computação sem fio entre diversas máquinas e sistemas computacionais, otimizando processos e reduzindo distâncias. No âmbito da energia, essa revolução tem o papel de coletar e analisar grandes volumes de dados, o chamado Big Data, e de proporcionar maior eficiência operacional e energética. (TRAVI,2019)

Uma clara evidência para tal tendência foi a iniciativa do ex-presidente dos Estados Unidos, Barack Obama, em 2009, de investir 3,4 bilhões de dólares em redes elétricas inteligentes e impulsionar outras dezenas de bilhões de dólares em investimentos privados. Segunda análise feita na época pelo *Electric Power Research institute* (EPRI), estima-se que a implementação de redes inteligentes nos Estados Unidos reduza em mais de 4% o consumo de energia elétrica no país até 2030, o que poderia implicar economias para os clientes da ordem de 20,4 bilhões de dólares (TOLEDO,2012).

Outro agravante que requer novas tecnologias é a alta demanda provocada por novos tipos de geração de energia elétrica, que está criando novos desafios exigentes, como por exemplo: as redes devem saber monitorar o fluxo de potência bidirecional e intermitência, e todo o sistema e todas as suas operações devem ser mantidas absolutamente num cenário seguro e confiável ao mesmo tempo para que não diminua a qualidade da energia fornecida aos consumidores finais. Além disso, novas capacidades precisam ser adicionadas, equipamentos existentes atualizados, e operação de rede otimizada para tornar toda a infraestrutura apta para o futuro.

1.2 Smart Grid

A rede elétrica inteligente ou *smart grid* já é uma realidade internacional e sua massificação é apenas uma questão de tempo. Apesar de serem muitos os desafios associados a sua implementação, condicionados pelo rompimento de paradigmas, maiores são as oportunidades de benefícios a serem gerados para a sociedade e para as diversas partes relacionadas (TOLEDO,2012).

A pesquisa e o desenvolvimento de novos produtos e serviços relacionados ao conceito de *smart grid* proporcionam uma participação mais ativa do consumidor na gestão de sua energia elétrica, que tomará conhecimento do consumo dos aparelhos elétricos e eletrônicos que ele possui em sua residência, empresa entre outros, e quanto isso significa em custo monetário em sua conta do final do mês e isso permitirá a oportunidade de estreitar ainda mais o relacionamento das concessionárias com os clientes, cuja participação será fundamental para o sucesso de tais implementações (TOLEDO, 2012).

Outro grande benefício para a sociedade será a maior eficácia na realização de programas de uso racional e eficiente de energia, atrelados à potencial redução dos gastos com energia elétrica, melhorias na qualidade do fornecimento de energia e redução de furtos de energia e emissões de CO₂ (TOLEDO,2012).

As *smart Grids* possuem sensores que facilitam a interação do cliente com o meio energético e a concessionária em tempo real, e isso contribui para um uso de energia elétrica mais

eficiente e racional por meio dos clientes. Algumas das características geralmente atribuídas às *Smart Grid* são, por exemplo: capacidade de automaticamente detectar, analisar, responder e restaurar falhas na rede. Reduzir o impacto ambiental do sistema produtor de eletricidade reduzindo perdas e utilizando fontes de baixo impacto ambiental entre outras. (FALCÃO,2009)

1.3 Smart Meter

Diferentemente dos antigos medidores mecânicos os *Smart Meters*, ou medidores inteligentes, possuem uma variedade de benefícios, como fazer com que os consumidores possam monitorar em tempo real os seus gastos com energia elétrica assim como a potência total consumida naquele exato momento (GOVUK.2013)

Com esses medidores podemos ter um maior conforto em relação aos custos de energia elétrica e assim racionalizar melhor o consumo de energia elétrica.

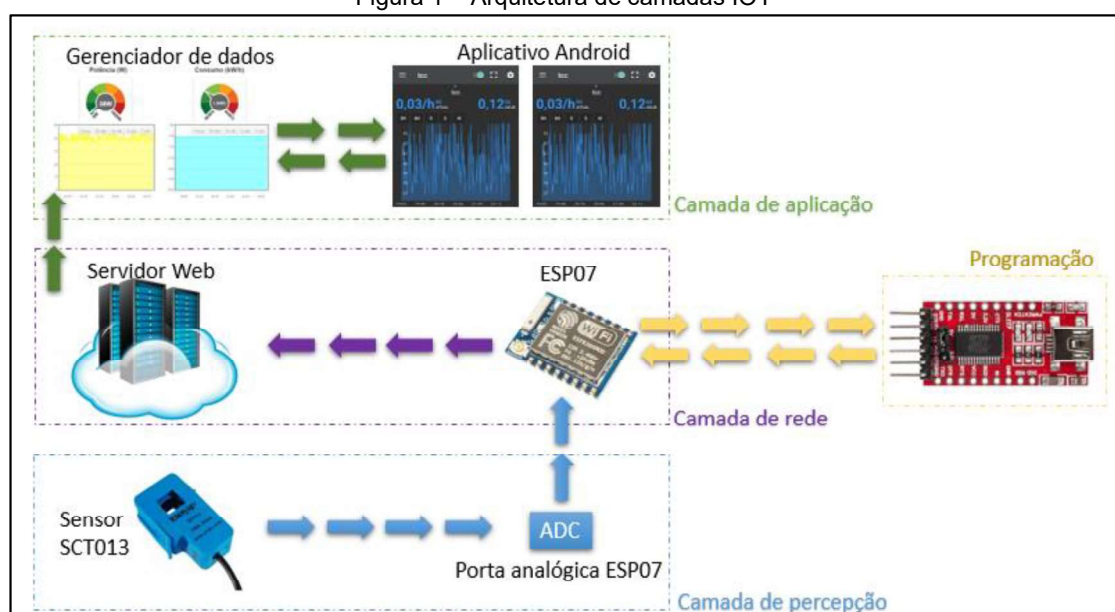
Com o objetivo de fundamentar os aspectos citados, é apresentado nesse projeto um estudo na área de energia elétrica, onde se quer saber qual o consumo de um aparelho ligado durante vinte e quatro horas. E em função desta variável, calcular seu custo por hora instantaneamente, prevenindo o usuário de eventuais surpresas na conta de energia elétrica no final do mês.

2 METODOLOGIA

A arquitetura da solução desenvolvida está ilustrada na Figura 1. Na solução desenvolvida, o sensor lê a corrente do condutor, envia as informações à porta analógica do ESP07 e conecta-se à rede local através de *WiFi*, que por sua vez conecta-se ao servidor através de TCP/IP e envia dados a ele utilizando o protocolo *HTTP*, podendo assim ser visualizado a potência utilizada e o consumo em tempo real na plataforma escolhida.

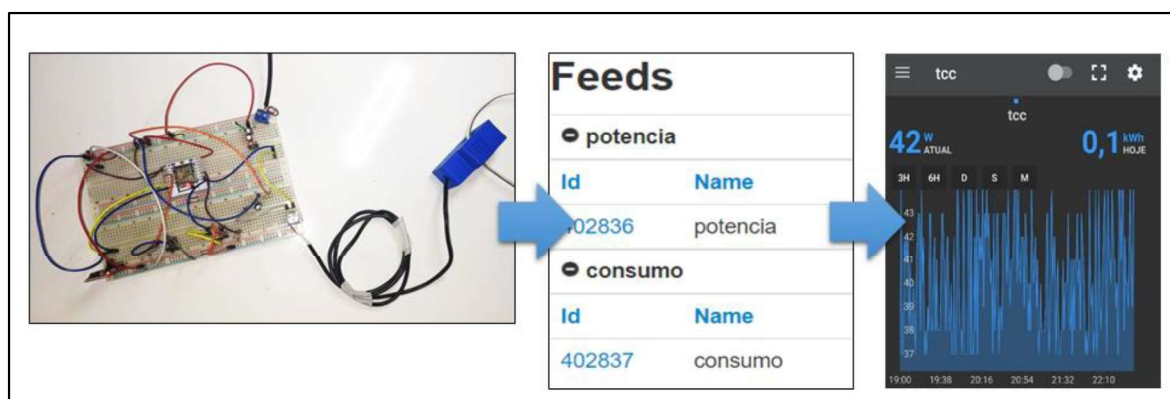
Arquitetura da solução:

Figura 1 – Arquitetura de camadas IOT



Fonte: elaborado pelos autores

Figura 2 – Sistema em funcionamento



Fonte: elaborado pelos autores

Para melhor entendimento da arquitetura do projeto, seus módulos foram separados em três camadas diferentes propostas pelo modelo *IoT*, sendo:

Camada de percepção: Tem a função de identificar os objetos e coletar informações (RODRIGUES,2014). No caso do presente trabalho, essa camada é preenchida pelo Sensor SCT013, o qual lê a corrente que passa pelo condutor e a entrega a porta analógica (a qual representa o microcontrolador embarcado na placa) do ESP07.

Camada de rede: Tem a função de transmitir as informações obtidas na camada de percepção (RODRIGUES,2014). Para essa função, foi designado o próprio ESP07 seu módulo *Wi-Fi*, o qual envia as informações do sensor de corrente recebidos pela sua porta analógica para o endereço *IP* do servidor *WEB* designado pelo EMONCMS.

Camada de aplicação: tem como função a de fornecer uma plataforma de suporte de operação que pode ser acessada por estações de monitoramento e aplicações, permitindo que qualquer pessoa possa usar os serviços de *IoT* (RODRIGUES,2014). A plataforma escolhida é um serviço gratuito de código aberto fornecido pela Emoncms.org, chamado de *Open Energy Monitor*. Além do serviço de armazenamento de dados, o *Open Energy Monitor* fornece API (*Application Programming Interface*) de comunicação, plataformas de visualização de dados e um aplicativo para *smartphones Android*.

Programação: Não faz parte das camadas do *IOT*, porém nesse presente projeto é embarcada através do conversor USB-UART em conjunto com a IDE Arduino

Figura 3 – Exemplo de aplicação do Open Energy Monitor na microprodução de energia solar



Fonte: disponível em: < <https://emoncms.org/> >

2.1 Projeto do hardware

O aspecto fundamental do sistema de monitoramento de consumo de energia elétrica proposto é a conexão com a Internet. Levando-se em conta a mobilidade necessária e o custo da conexão, excluem-se as opções de conexão cabeada, utilizando redes WiFi para a comunicação com a nuvem.

Figura (4) – Módulos eletrônicos selecionados, a) Conversor USB-UART FT232RL b) ESP8266 c) Sensor de corrente SCT013-000.



Fonte: Elaborado pelos autores

2.1.1 Conversor USB-UART FT232R

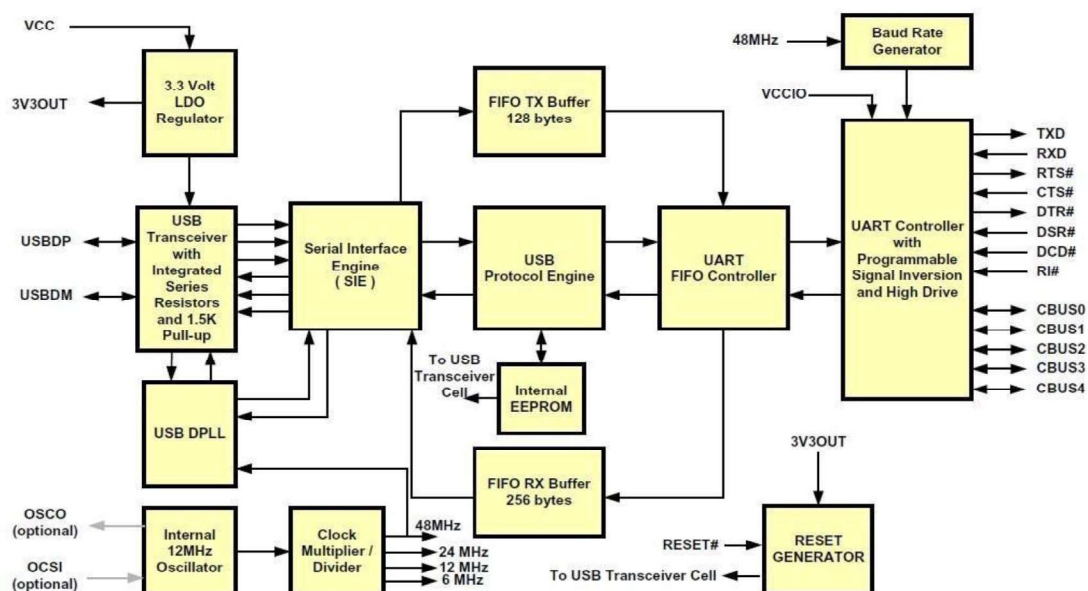
Criado pela empresa FTDI (*Future Technology Devices International*) uma empresa especialista em conversores para periféricos USB, fundada no ano de 1990 por Fred Dart, o FT232RL é um chip conversor USB – UART que usa seus pinos TX e RX para transmitir dados recebidos pela USB ou para transmitir dados para ela (FTDI,2019).

No presente trabalho, o conversor é a ponte de comunicação entre o ESP07, o qual se comunica em UART, e a plataforma de programação Arduino IDE, a qual foi escolhida pela sua compatibilidade a milhares de bibliotecas e facilidade de programação.

Como o ESP07 trabalha com 3.3V, é preciso compatibilizar o nível de tensão nos pinos TX e RX com o microcontrolador e para fazer a comunicação nesse nível de tensão, o FT232R

conta com um regulador interno do tipo LDO que disponibiliza essa tensão em um pino, o 3V3OUT, que pode facilmente ser acessado através de um jumper disposto na própria placa.

Figura (5) – Diagrama de blocos FT323R



Fonte: disponível em <<https://bit.ly/2QW8bjd>>

O FT232R conta com um oscilador interno de 12 MHz, que para funcionar precisa que o chip seja alimentado com no mínimo 4V, sendo que abaixo dessa tensão é necessário ligar um oscilador externo. A partir de um multiplicador/divisor interno, além dos 12 MHz, podem ser geradas as frequências de 6 MHz, 24 MHz e 48 MHz. Além disso, pode-se configurar um pino para oscilar em qualquer uma dessas frequências, e ainda esse pino pode ser usado para gerar o *clock* para um microcontrolador, por exemplo, eliminando a necessidade de um oscilador adicional (EMBARCADOS,2015).

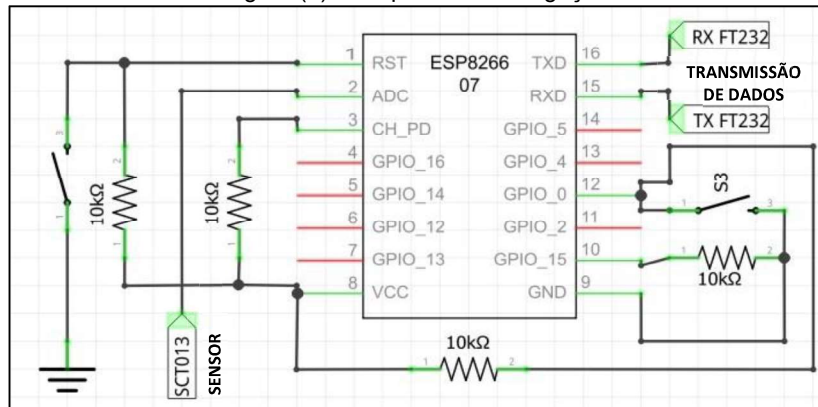
2.1.2 Microcontrolador ESP07

Desenvolvido pela *Ai-Thinker Team*, é um chip compacto feito para atender as necessidades de conectividade sem fio (padrão 802.11) podendo tanto ser utilizado independentemente, como é o caso do presente trabalho, ou em conjunto com um microcontrolador externo (*Ai-Thinker*, 2015).

Esse módulo possui um processador capaz de integração com sensores utilizando seus pinos de entrada/saída (GPIOs). Por se tratar de um módulo baseado em um Microcontrolador, requer poucos componentes, o que toma menos espaço tornando-se conveniente para o presente protótipo.

Como dito anteriormente, sua programação pode ser feita através de porta serial TX/RX, usando o conversor FT232R como ponte para a comunicação com a plataforma de código aberto Arduino.

Figura (6) – Esquemático de ligação ESP07

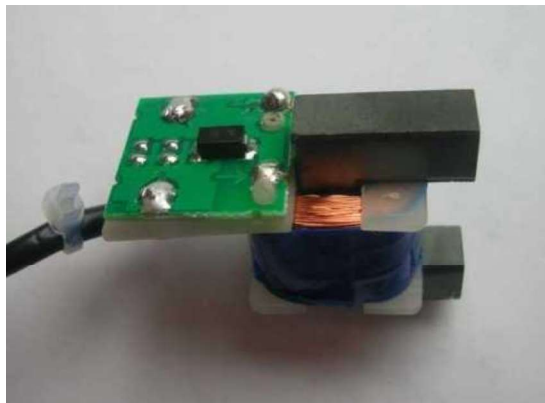


Fonte: Elaborado pelos autores

2.1.3 Sensor de corrente SCT013-000

SCT é a sigla para *Split-core Current Transformer*, ou seja, transformador de corrente de núcleo dividido. Esse sensor tem como principal vantagem a necessidade de estar apenas com uma fase do circuito passando pela bobina do seu transformador de corrente, não necessitando de interferir no curso do mesmo (Demetras,2017).

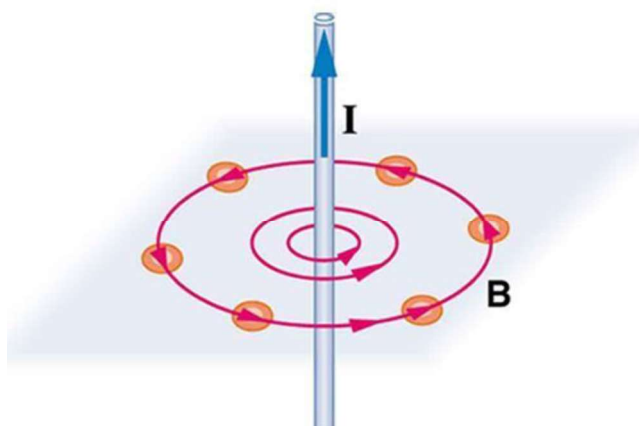
Figura (7) – Sensor SCT013 por dentro



Fonte: disponível em <<https://bit.ly/2QW8bjd>>

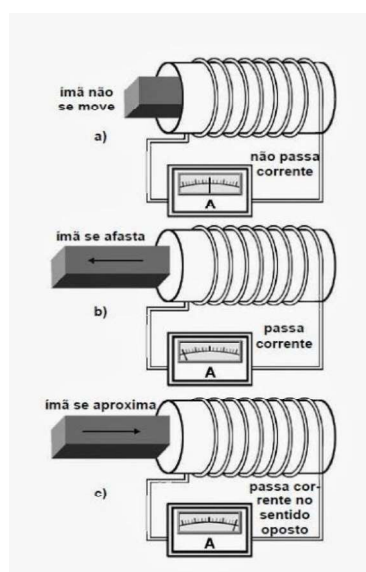
Para fazer essa leitura, o sensor usa da lei de Ampère, a qual diz que todo condutor por onde flui uma corrente elétrica induz ao seu redor um campo magnético proporcional à corrente e a lei de Faraday que diz um campo magnético variante no tempo induz numa espira uma tensão, e por consequência uma corrente elétrica, proporcional à intensidade do campo magnético (Demetras,2017).

Figura (8) – Ilustração da Lei de Ampère



Fonte: disponível em < <https://bit.ly/2XQeO84>>

Figura (9) – Ilustração da Lei de Faraday



Fonte: disponível em <<https://bit.ly/2OFSIRh>>

Para aplicar as leis de Ampère e Faraday, deve-se ter um conjunto de espiras ao redor do condutor, e uma vez que a corrente é induzida pro elas, deve-se realizar a leitura do sensor pela porta analógica ADC do ESP07 e para que isso ocorra, devemos adicionar um resistor de carga entre os polos do sensor de corrente.

Segundo o *Open Energy Monitor*, o valor desse resistor de carga é dado pela seguinte equação:

$$R = \frac{\left(\frac{V_{\text{sistema}}}{2}\right)}{\frac{(I_{\text{max}} * \sqrt{2})}{n^{\circ} \text{ espiras}}}$$

Onde:

$V_{\text{sistema}} = 3.3 \text{ V}$;

$I_{\text{max}} = \text{Corrente Máxima permitida pelo sensor (100A)}$;

Número de espiras = Quantidade de espiras que o Sensor possui (2000);

Então

$$R = \frac{\left(\frac{3,3}{2}\right)}{\frac{(100 * \sqrt{2})}{2000}}$$

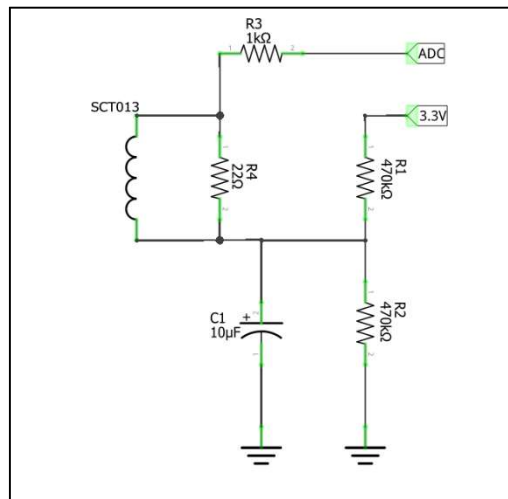
$$R = 23 \Omega$$

A aproximação que se encontra no mercado atual de um resistor nessa faixa é de 22Ω .

Somente com essas informações, o circuito está alternando entre 1,65V positivo e 1,65V negativo, como o ESP07 não pode medir tensão negativa, é necessário somar 1,65V ao sinal para que ele varie entre 0V a 3.3V. Para que isso ocorra, é necessário acrescentar no circuito um divisor de tensão de dois resistores de $470\text{k}\Omega$, para que a tensão fornecida possa estar igualmente dividida entre eles.

Foi acrescentado um resistor de $1\text{k}\Omega$ na saída para a porta analógica do ESP07 por segurança, pois o máximo de tensão permitida por ela é de 1V.

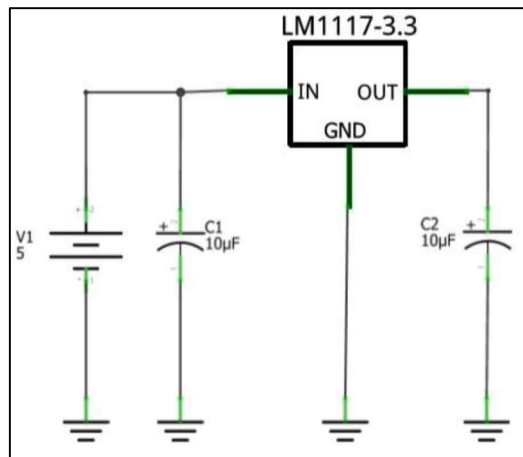
Figura 10 – Circuito utilizado para a leitura de corrente.



Fonte: Elaborado pelos autores

Para a alimentação de todo o circuito, é necessário 3.3V e aproximadamente 500mA (consumo ESP07 segundo *datasheet*). Como no mercado é mais prático encontrar fontes de 5V, foi utilizado um regulador de tensão LM1117, que recebe 5V de um cabo usb em conjunto com um carregador comum de celular e entrega 3,3V (máximo de 800mA).

Figura 11 – Circuito utilizado para o regulador de tensão



Fonte: Elaborado pelos autores

O circuito completo do *hardware* aqui abordado pode ser encontrado no anexo B deste trabalho.

2.2 PROJETO DO SOFTWARE

Para a implementação da arquitetura apresentada, além das configurações de funcionamento da plataforma, desenvolveu-se um software embarcado no módulo escolhido e foi configurado um aplicativo para *Android* disponibilizado pela plataforma.

2.2.1 Software Embarcado

O *software* embarcado lê a corrente instantânea fornecida pelo sensor, e calcula a potência ativa através da seguinte equação:

$$P = V * I_{rms}$$

Sendo que: “V” é uma constante que representa a tensão da fase medida (no caso do presente trabalho 127 V) e “I_{rms}” é a variável de corrente que é lida pelo sensor de corrente e transformada em corrente eficaz pela biblioteca “*EmonLib*” incluída no código embarcado no ESP07.

No caso do presente projeto foi utilizado uma carga totalmente resistiva, ou seja, tem seu fator de potência igual a um, não interferindo no cálculo da potência ativa.

Para a calibração do sensor no código embarcado, é necessária uma constante que é obtida pela seguinte equação fornecida pela *Open Energy Monitor*:

$$Cal = \frac{n^{\circ} \text{ espiras}}{R_{carga}}$$

Como calculado no projeto de hardware, o valor do resistor de carga é de 22 Ω, e o número de espiras do sensor é de 2000.

Então calculando:

$$Cal = \frac{2000}{22}$$

$$Cal = 90,9$$

Este software gerencia a comunicação do ESP07, montando as mensagens (*payload*) a serem enviadas à rede, solicitando conexão com a rede *Wi-Fi* local, solicitando conexão com o servidor, checando se a comunicação obteve sucesso e identificando possíveis problemas de comunicação com a rede local ou com o servidor.

2.2.2 Envio de dados

Antes do envio dos dados, é necessária uma *API KEY* de escrita que é embarcada no código e que serve como autenticação para a conta criada previamente no site do Emoncms, podendo após isso, acontecer o envio e edição dos dados recebidos pela plataforma.

Figura 12 – Conta criada no Emoncms e API KEY de leitura e escrita fornecidas



Fonte: elaborado pelos autores

A recepção e análise dos dados pela plataforma é dado por meio de entradas (*inputs*), onde o valor enviado pelo ESP07, respeitando o limite de 2.4 segundos entre dados enviados (Emoncms.org), é lido e introduzido em um nó/grupo designado pelo próprio usuário. Para ocorrer o armazenamento de dados, é necessário criar um *feed* na plataforma, que serve para como histórico para os dados enviados, que por sua vez é utilizado pelo *Dashboard* criado pelo usuário na plataforma e também utilizado pelo aplicativo, para fornecer os gráficos desejados nesse presente trabalho.

O aplicativo *Open Energy Monitor* pode ser baixado pela *google play*, e para o envio de dados para ele, é necessário a API KEY de leitura, como representado na figura 12, pois ele somente lê os dados que já estão tratados e gravados no *feed* e os representa por forma de gráfico para fácil visualização.

3 RESULTADOS

Na Figura (13) é mostrado o medidor de potência desenvolvido. O circuito foi produzido correndo-se uma placa de fenolite revestida de cobre, utilizando percloroeto de ferro para

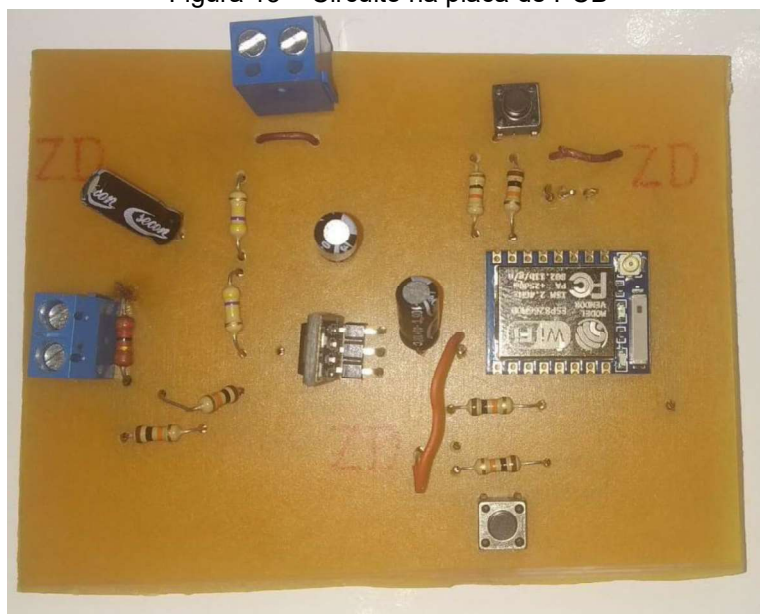
produzir as trilhas condutoras, perfurando-a e soldando-se os componentes. O custo total do protótipo está descrito na tabela (1).

Tabela 1 – Custo Componentes

Componente	Custo (R\$)
ESP07	29,9
Conversor USB-UART FT232	19,9
Sensor SCT013	44,75
Placa fenolite 15x15	15
Percloroeto de ferro	18
Frete total	16,9
Total	144,45

Fonte: elaborado pelos autores

Figura 13 – Circuito na placa de PCB



Fonte: elaborado pelos autores

Ao ligar-se a chave, os equipamentos são energizados e, após cerca de 10 s, ele conecta-se à rede WiFi e inicia o envio de dados ao servidor. O dispositivo ainda não oferece um mecanismo para configurar SSID e senha da rede WiFi, tendo estes parâmetros tratados como constantes e gravados em código.

Com o objetivo de testar o sistema quanto a seu funcionamento e as informações que ele é capaz de fornecer, foi monitorado o consumo de energia de uma lâmpada de filamento de carbono ligada durante 24 horas.

A lâmpada tem as seguintes especificações técnicas:

Lâmpada de filamento de carbono G95 *StarLux*

Tensão em Volts: 110/130 V

Potência em Watts: 40 W

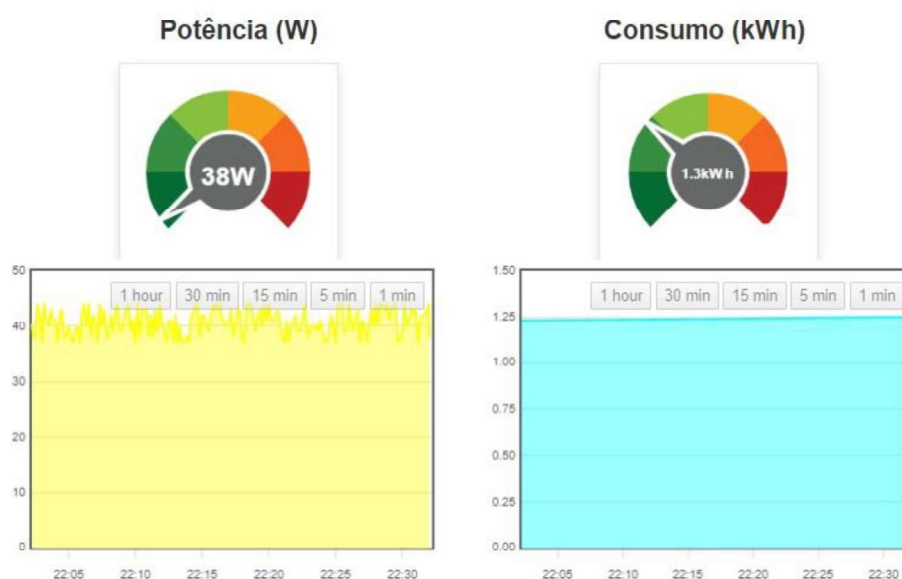
Fator de potência: 1

No gráfico de consumo instantâneo da figura 15 podemos perceber que o consumo da lâmpada obteve uma variação de aproximadamente 37 para 45 Watts em 24 horas de monitoramento. Seu consumo dentro de 24 horas, calculado utilizando os parâmetros de *feed* em conjunto com o elemento visual de *dashboard* estabelecidos pela própria plataforma Emoncms, foi de 1,3kWh.

Calculando manualmente o consumo e considerando a variação dada pelo gráfico, obtêm-se uma média de consumo de 41 W e multiplicando por 24 horas, temos o resultado de 0,98kWh, ou seja, uma diferença de 0,34kWh utilizando o dashboard do Emoncms. Aqui pode-se constatar que houve algum erro na medição o qual necessita de uma análise mais profunda que não será abordada nesse presente trabalho.

O medidor de potência criado na plataforma de Dashboard do Emoncms varia de 0 W (verde) a 1kW (vermelho), e o de consumo varia de 0 kWh (verde) a 4kWh (vermelho).

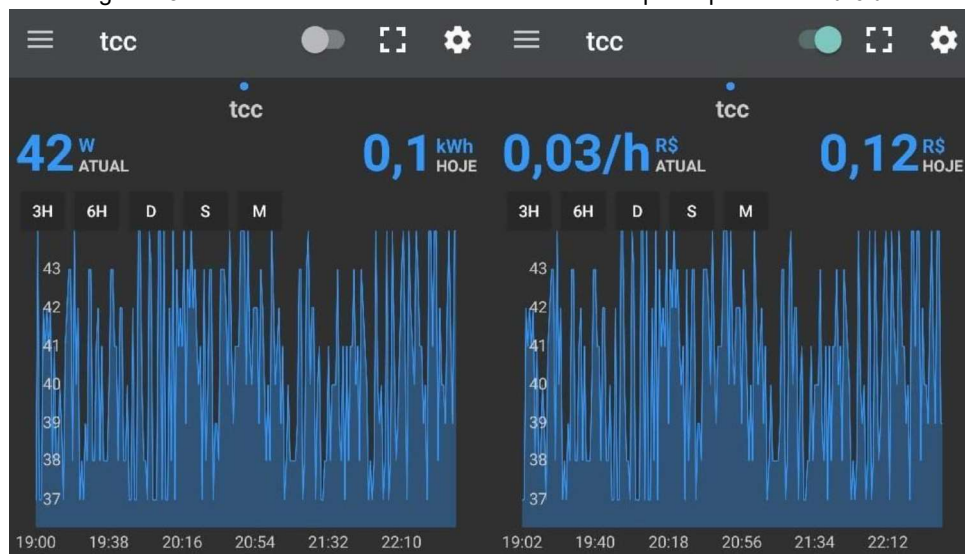
Figura 14 – Dashboard Emoncms.org



Fonte: elaborado pelos autores

No gráfico de consumo instantâneo gerado pelo aplicativo de *smartphone Android* escolhido e considerando um valor simbólico de R\$0,8/kWh, foi calculado que em 24 horas a lâmpada custou R\$ 0,12 e consumiu 0,1kWh e ainda temos que ela custa em média 0,03R\$/h ligada.

Utilizando o cálculo de consumo feito anteriormente (aproximadamente 0,98kWh), há uma diferença de apenas 0,04kWh utilizando o aplicativo para *Android*, o que prova que o aplicativo é menos propenso a erros do que a plataforma de dashboard fornecido pela Emoncms.

Figura 15 – Leitura das entradas do Emoncms feitos pelo aplicativo *Android*

Fonte: elaborado pelos autores

4 CONCLUSÃO

Como proposto, nesse trabalho de conclusão de curso foi desenvolvido um sistema de monitoramento de consumo de energia elétrica não invasivo de equipamentos elétricos e eletrônicos com conexão à Internet, alcançando-se resultados satisfatórios. Medições de corrente eficaz são realizadas pelo equipamento, transformada em potência ativa pelo código embarcado e enviados para nuvem.

Os dados enviados são analisados e disponibilizados através de uma API. Gráficos são alimentados pelas informações do feed e então são disponibilizados através de navegador *WEB* ou para mais praticidade, em um aplicativo Android.

Pelos testes realizados pôde-se obter algumas informações interessantes sobre o equipamento monitorado como a variação do consumo energético para qualquer período de tempo analisado, seu custo por hora, e o total gasto por dia.

É necessário aprimoramento no hardware, software embarcado e processamento de dados, como: utilização de protocolos de comunicação mais eficientes, diminuição do consumo de energia do próprio módulo a fim de deixá-lo somente com uma bateria 5 V e incluir um aumento de portas lógicas para a conexão de múltiplos sensores. O custo do protótipo de cento e quarenta e quatro reais pode diminuir bastante se os equipamentos fossem importados, por exemplo da China, em grande quantidade, tornando-se mais atrativo comercialmente.

A Internet das coisas está gerando novas ideias que irão impactar a vida das pessoas. É imprescindível que pesquisas sejam feitas, produtos sejam desenvolvidos e a infraestrutura preparada, para que oportunidades não sejam perdidas nos campos empresarial, industrial e acadêmico.

REFERENCIAL TEÓRICO

BRASIL. Implantação de medição eletrônica em baixa tensão, Documento Anexo à nota técnica nº 0013/2009 – SRD/ANEEL, janeiro 2009

CEMIG. **O que são as redes inteligentes de energia?**. 20??. On-line. Disponível em: <<http://www.cemig.com.br>>. Acesso em: 16 junho. 2018

CPLF. **Smart Grids, o que é?**. 20??. Online. Disponível em: <<https://www.cplf.com.br>>. Acesso em: 17 junho. 2018

FALCÃO, M. DJALMA. **smart grids e microrredes o futuro já é presente**. 2009. On-line. Disponível em: <<https://www.researchgate.net>>. Acesso em: 19 maio. 2018

LAKATOS, E.M; MARCONI, M.A. **Fundamentos de metodologia científica**. 5. Ed. São Paulo: Atlas S.A, 2003.

SIEMENS. **Power Engineering Guide**. 8.Ed. Erlangen: Publicis Pixelpark, 2017.

SIEMENS. **Why microgrids are the future of energy management**. 20??. On-line. Disponível em: <<https://www.siemens.com>>. Acesso em: 15 junho. 2018

TOLEDO, FÁBIO. **Desvendando as Redes Elétricas Inteligentes: Smart Grid Handbook**. 1. Ed. São Paulo: Brasport, 2012. 336 p.

WERANGA,K.S.K; KUMARAWADU, SISIL; CHANDIMA,D. P. **Smart Metering Design and Applications**. 1. Ed. Moratuwa: Springer, 2014.

AI-THINKER. **ESP-07 WiFi Module** 2015. On-line. Disponível em: <https://www.mikrocontroller.net/attachment/338570/Ai-thinker_ESP-07_WIFI_Module-EN.pdf>. Acesso em: 20 novembro. 2019

FTDI. **FT232R USB UART IC Datasheet** 2019. On-line. Disponível em: <https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf>. Acesso em: 20 novembro. 2019

XIDI TECHNOLOGY. **SCT-013-000** 2011. On-line. Disponível em: <https://www.elecrow.com/download/SCT013-000_datasheet.pdf>. Acesso em: 20 novembro. 2019

TEXAS INSTRUMENTS. **LM1117 800-mA Low-Dropout Linear Regulator** 2016. On-line. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm1117.pdf>>. Acesso em: 20 novembro. 2019

ALVES, JOSEMAR. **O poderoso FTDI FT232R** 2015. On-line. Disponível em: <<https://www.embarcados.com.br/ftdi-ft232r/>>. Acesso em: 20 novembro. 2019

DEMETRAS, EZEQUIEL. **SCT-013 - Sensor de corrente alternada com Arduino** 2017. On-line. Disponível em: <<https://portal.vidadesilicio.com.br/sct-013-sensor-de-corrente-alternada/>>. Acesso em: 20 novembro. 2019

OPENSTAXCOLLEGE. **Magnetic Fields Produced by Currents: Ampere's Law** 20??. On-line. Disponível em: <<https://opentextbc.ca/physicstestbook2/chapter/magnetic-fields-produced-by-currents-amperes-law/>>. Acesso em: 22 novembro. 2019

ELETRICIDADE. **Lei de faraday** 2013. On-line. Disponível em: <<http://eletricidadeaz.blogspot.com/2013/11/lei-de-faraday.html>>. Acesso em: 22 novembro. 2019

TYLER ANAIRO. **Open Energy Monitor current transformer (CT) value calculators** 2013. On-line. Disponível em: <<https://tyler.anairo.com/projects/open-energy-monitor-calculator>>. Acesso em: 22 novembro. 2019

GE. **Você sabe o que é Smart Grid? Tire suas dúvidas no GE Reports Brasil!** 2013. On-line. Disponível em: <<https://gereportsbrasil.com.br/voc%C3%AA-sabe-o-que-%C3%A9-smart-grid-tire-suas-d%C3%BAvidas-no-ge-reports-brasil-191fc22998c4>>. Acesso em: 22 novembro. 2019

RODRIGUES, F. FÁBIO. **Estudo de Aplicações da Internet das Coisas em um Ambiente Acadêmico** 2014. On-line. Disponível em: <<https://www.revistaespacios.com/a14v35n13/14351309.html>>. Acesso em: 22 novembro. 2019

GITHUB. **Estudo de Aplicações da Internet das Coisas em um Ambiente Acadêmico** 2014. On-line. Disponível em: <<https://github.com/emoncms/emoncms>>. Acesso em: 22 novembro. 2019

TRAVI, RODRIGO. **Indústria 4.0 e o setor elétrico** 2019. On-line. Disponível em: <<https://www.hojeemdia.com.br/opini%C3%A3o/blogs/opini%C3%A3o1.363900/ind%C3%BAstria-4-0-e-o-setor-el%C3%A9trico-1.744613>>. Acesso em: 23 novembro 2019.

ANEXO A

```

#include <ESP8266WiFi.h>
#include <SPI.h>
#include "EmonLib.h"
EnergyMonitor emon1;
#define pin A0
#define calib 90.9
#define volt 127
const char* ssid      = "SSID";
const char* password  = "SENHA";
const char* host      = "emoncms.org";
int port              = 80;
const char* apikey    = "APIKEY";
int updatedata        = 2400;
const char* inputname = "power";
int nodedata          = 10;
void setup()
{
  Serial.begin(115200);
  Serial.println("ESP8266 in normal mode");
  WiFi.begin(ssid, password); while
  (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());
  emon1.current(pin, calib);
  Void
  loop()
  {
    Serial.print("Connecting to ");
    Serial.println(host);
    WiFiClient client;
    if (!client.connect(host, port)) {
      Serial.println("connection failed");
      return;
    }
    double Irms = emon1.calcIrms(1480);
    Serial.print(Irms*volt);
    Serial.println(Irms);
    int value= (Irms*volt);
    long rssi = WiFi.RSSI();
    client.print("GET
/emoncms/input/post.json?node="+String(nodedata)+"&json={" +inputname+": "+
String(value)+", "+ "RSSI:"+String(rssi)+"}&apikey="+apikey + " HTTP/1.1\r\n"+
"Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
    unsigned long timeout = millis(); while
    (client.available() == 0) { if (millis()
    - timeout > 5000) { Serial.println(">>>
Client Timeout !"); client.stop();
    return;}}
    while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
    Serial.println();
    Serial.println("closing connection");
    delay(updatedata); }
  }

```

ANEXO B

