

REDE DE ENSINO DOCTUM - CAMPUS DOM ORIONE
BACHARELADO EM ENGENHARIA ELÉTRICA

PAULO AUGUSTO FERREIRA DE SOUZA

DESENVOLVIMENTO DE DISPOSITIVO AFERIDOR DE ENERGIA IOT

JUIZ DE FORA
2024

REDE DE ENSINO DOCTUM - CAMPUS DOM ORIONE
BACHARELADO EM ENGENHARIA ELÉTRICA

PAULO AUGUSTO FERREIRA DE SOUZA

DESENVOLVIMENTO DE DISPOSITIVO AFERIDOR DE ENERGIA IOT

Trabalho de Conclusão de Curso submetido à Rede de Ensino Doctum como parte dos requisitos necessários para a obtenção do Título de Bacharel em Engenharia Elétrica.

Orientador: LUIS GUSTAVO SCHRODER E BRAGA

JUIZ DE FORA
2024

PAULO AUGUSTO FERREIRA DE SOUZA

DESENVOLVIMENTO DE DISPOSITIVO AFERIDOR DE ENERGIA IOT

Trabalho de Conclusão de Curso submetido à Rede de Ensino Doctum como parte dos requisitos necessários para a obtenção do Título de Bacharel em Engenharia Elétrica.

Orientador: LUIS GUSTAVO SCHRODER E BRAGA

Defendida em:

Banca Examinadora

Prof.^a Daniele Alcantara

**JUIZ DE FORA
2024**

“A essência do conhecimento consiste em aplicá-lo, uma vez possuído.”

(Confúcio)

RESUMO

Este trabalho de conclusão de curso, teve como tema escolhido “O desenvolvimento de um dispositivo aferidor de energia IoT”, devido à perspectiva de aumento de consumo de energia e do interesse em eficiência energética. Foi definido como problema a escolha das tecnologias embarcadas tanto no hardware quanto no software e mostrar quais dificuldades encontradas no processo. Os objetivos contam também com a compreensão do funcionamento, aferição da precisão e variações na construção. A metodologia utilizada foi de pesquisa exploratória, o estudo dos resultados teve abordagem qualitativa com método dedutivo. A etapa de desenvolvimento conta com detalhamento da construção, calibração e teste do dispositivo, durante o desenvolvimento do dispositivo, foram encontradas dificuldades descritas na conclusão, estas por sua vez impactaram nos resultados, mas ainda sim foi possível atingir quase que completamente os objetivos propostos. Ao analisar a construção do dispositivo, foi considerado que o mesmo obteve resultado geral satisfatório, mas deixando pontos para melhoria.

Palavras-chave: desenvolvimento; aferidor; IoT.

ABSTRACT

This course completion work had as its chosen theme "The development of an IoT energy gauging device", due to the perspective of increased energy consumption and interest in energy efficiency. The choice of embedded technologies in both hardware and software was defined as a problem and to show what difficulties were encountered in the process. The objectives also include understanding the operation, gauging the accuracy and variations in construction. The methodology used was exploratory research, the study of the results had a qualitative approach with a deductive method. The development stage includes detailing the construction, calibration and testing of the device, during the development of the device, difficulties were encountered described in the conclusion, these in turn impacted the results, but it was still possible to almost completely achieve the proposed objectives. When analyzing the construction of the device, it was considered that it obtained a satisfactory overall result, but leaving points for improvement.

Keywords: development; gauge; IoT.

LISTA DE ILUSTRAÇÕES

Figura 1 -	Tensão alternada	12
Figura 2 -	Potência Complexa	13
Figura 3 -	Ilustração Transformador	15
Figura 4 -	ESP-32	16
Figura 5 -	ADC	17
Figura 6 -	ADS1115	17
Figura 7 -	Exemplos de Soma de Riemann	19
Figura 8 -	Esquema elétrico ilustrado do dispositivo	21
Figura 9 -	Dispositivo aferidor de energia IoT	21
Figura 10 -	Divisão do tempo	23
Figura 11 -	Fluxograma	24
Figura 12 -	Medição com multímetro e alicate amperímetro	25
Figura 13 -	Recorte da aplicação web	25

LISTA DE ABREVIATURAS E SIGLAS

AC	- Alternating current [Corrente alternada].
ADC	- Analog-to-Digital converters [Conversor analógico digital].
CSS	- Cascading Style Sheets [Folhas de estilo em cascata].
DDP	- Diferença de Potencial.
EPE	- Empresa de Pesquisa Energética.
HTML	- HyperText Markup Language [Linguagem de Marcação de Hipertexto].
IDE	- Integrated Development Environment[Ambiente de desenvolvimento integrado]
INMETRO	- Instituto Nacional de Metrologia, Qualidade e Tecnologia.
IoT	- Internet of Things [Internet das coisas].
I2C	- Inter-Integrated Circuit [Circuito Inter-integrado].
MQTT	- Message Queuing Telemetry Transport.
RMS	- Root mean square [Valor eficaz].
SEP	- Sistema elétrico de potência.
SI	- Sistema Internacional de Unidades.
TC	- Transformador de corrente.
USB	- Universal Serial Bus [Porta serial universal].
Wi-Fi	- Wireless field.

LISTA DE SÍMBOLOS

V	- Tensão elétrica.
T	- Trabalho.
Q	- Carga elétrica.
I	- Corrente elétrica.
Δq	- Variação da carga elétrica.
Δt	- Variação do tempo.
ω	- Velocidade angular.
θ	- Defasagem angular.
f	- Frequência.
T	- Período de tempo.
P	- Potência ativa.
Q	- Potência reativa.
S	- Potência aparente.
FP	- Fator de potência.

SUMÁRIO

1	INTRODUÇÃO	10
2	REFERENCIAL TEÓRICO	11
	2.1 Conceitos da Eletricidade	11
	2.1.1 Tensão	11
	2.1.2 Corrente	11
	2.1.3 Tensão e Corrente alternada	11
	2.1.4 Potência Complexa	13
	2.2 Benefícios da monitoração de energia	15
	2.3 Transformador	15
	2.3.1 Talema AC1050	16
	2.4 Microcontrolador	16
	2.4.1 ESP-32	16
	2.5 ADC	17
	2.5.1 ADS1115	17
	2.6 IoT	18
	2.7 MQTT	18
	2.7.1 Broker MQTT	18
	2.7.1.1 EMQX	18
	2.8 Arduino IDE	18
	2.9 Aplicação Web	19
	2.10 Soma de Riemann	19
3	METODOLOGIA	20
4	DESENVOLVIMENTO	20
	4.1 Descrição detalhada do Hardware	20
	4.2 Descrição detalhada do Software	22
	4.3 Calibração e testes	24
5	RESULTADOS	26
6	CONCLUSÃO	26
	REFERÊNCIAS	27
	APÊNDICE A	29
	APÊNDICE B	38

1. INTRODUÇÃO

Devido a perspectiva de aumento de consumo de energia, o interesse em eficiência energética também tem aumentado substancialmente (PDE, 2022). Para ser possível monitorar a eficiência energética de um determinado equipamento ou circuito, é necessário mensurar os valores de energia a serem analisados (SENRA, 2015).

Todo dispositivo de medição tem a necessidade de apresentar os valores para serem aferidos e interpretados pelo usuário. Devido ao interesse em realizar as verificações de forma remota e a possibilidade de se obter maior eficiência elétrica de uma instalação, foi definido a utilização dos smartphones para apresentar os valores lidos pelo dispositivo. Em razão do dispositivo necessitar estar conectado à internet para seu funcionamento, o mesmo se enquadra na descrição de dispositivo IoT (MADAKAM, 2015).

Após feitas todas as observações, foi definido o tema do trabalho como: desenvolvimento de dispositivo aferidor de energia elétrica IoT. O dito tema foi escolhido para ter um aperfeiçoamento do conhecimento pessoal na área em questão, com foco na utilização de microcontroladores, e em seu hardware agregado, assim como em toda a programação e tecnologias embarcadas no projeto. A relevância do tema considero como alta, por contemplar aspectos como medição de energia, microcontrolador, eficiência energética e tecnologias que estão tendo alta utilização. Sua utilidade para a comunidade científica tende a ser boa, pois o trabalho poderá ser consultado não só por completo, mas também por partes separadas, podendo assim aumentar sua utilidade.

Com o tema escolhido, foi enfrentado o seguinte problema: como escolher as tecnologias para criar o dispositivo e quais as dificuldades encontradas no processo de criação do mesmo?

O objetivo geral deste trabalho é investigar e propor uma seleção de tecnologias necessárias no desenvolvimento de um dispositivo aferidor de energia IoT, bem como identificar e analisar as dificuldades encontradas durante o processo de criação deste dispositivo.

Os objetivos específicos são:

- Compreender os conceitos de funcionamento do dispositivo;
- Analisar e definir as tecnologias embarcadas visando custo benefício;
- Aferir sua precisão;
- Demonstrar possíveis variações na criação e suas características.

2. REFERENCIAL TEÓRICO

2.1 Conceitos da Eletricidade

2.1.1 Tensão

Tensão elétrica (V ou U), grandeza física também denominada DDP, é uma medida importante para o entendimento de circuitos elétricos (FROES, 2017). Sua unidade de medida de acordo com o (SI) é o Volt (V) (INMETRO, 2021). A tensão é a quantidade de energia potencial elétrica por unidade de carga elétrica (HALLIDAY, 2013).

$$U = \frac{T}{Q} \quad (2.1)$$

Onde:

U → Tensão elétrica medida em Volt (V).

T → Trabalho medido em Joule (J).

Q → Carga elétrica medida em Coulomb (C).

A tensão elétrica é responsável por criar o campo elétrico que movimenta os elétricos em um elemento condutor em um circuito fechado (FROES, 2017).

2.1.2 Corrente

A corrente elétrica (I), é a movimentação de carga elétrica em um elemento condutor, ou o fluxo de partículas carregadas eletricamente, sua intensidade é a razão do módulo da quantidade de carga que atravessa uma seção em um intervalo de tempo. (FROES, 2017).

$$I = \frac{|\Delta q|}{\Delta t} \quad (2.2)$$

Onde:

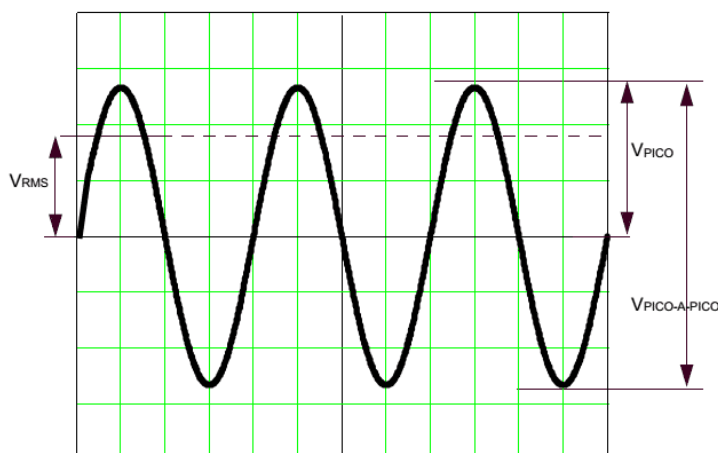
I → Corrente elétrica medida em Ampere (A).

Δq → Variação da Carga elétrica medida em Coulomb (C).

Δt → Intervalo de tempo medido em segundos (s).

2.1.3 Tensão e Corrente alternada

Nos sistemas elétricos de potência (SEP) em todo o mundo, é comumente utilizada a tensão alternada, que também proporciona a corrente alternada. Essas por sua vez tem esse nome pois alteram sua amplitude no decorrer do tempo, alterando a polaridade a cada semiciclo (HALLIDAY, 2013).

Figura 1 - Tensão alternada

Fonte: EMBARCADOS 2016

Abaixo está a função no qual representa a tensão alternada:

$$f(t) = A \times \text{sen}(\omega t + \Theta) \quad (2.3)$$

Onde:

$A \rightarrow$ Constante de amplitude da senoide no qual pode representar o valor de pico.

$t \rightarrow$ Variável associada à função.

$\omega \rightarrow$ Velocidade angular em rad/s.

$\Theta \rightarrow$ Defasagem angular da função.

Uma grandeza no qual pode ser aferido nesse tipo de sistema é a frequência, que é a relação de quantos ciclos ocorrem em um segundo.

$$f = \frac{1}{T} \quad (2.4)$$

Onde:

$f \rightarrow$ É a frequência em Hertz (Hz).

$T \rightarrow$ Período em segundos (s).

Para a realização de cálculos com valores de tensão e corrente alternados, não se utiliza os valores de pico da senoide, mas sim valores eficazes ou RMS, estes por sua vez são a representação de um valor contínuo que possua mesma potência média em uma carga resistiva (HALLIDAY, 2013).

Abaixo está a fórmula simplificada no qual pode-se converter valores de tensão e corrente de pico de uma senóide para os valores eficazes ou RMS.

$$V_{rms} = \frac{V_{pico}}{\sqrt{2}} \quad (2.5)$$

$$I_{rms} = \frac{I_{pico}}{\sqrt{2}} \quad (2.6)$$

Onde:

V_{rms} → É o valor da tensão eficaz ou RMS.

I_{rms} → É o valor da corrente eficaz ou RMS.

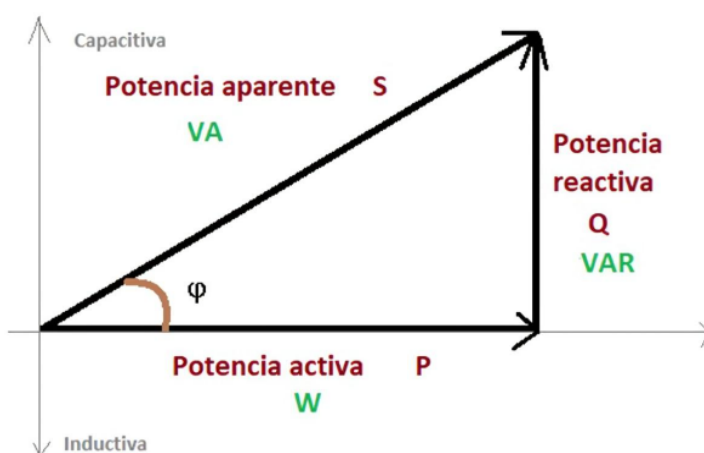
V_{pico} → É o valor da tensão de pico da senóide.

I_{pico} → É o valor da corrente de pico da senóide.

2.1.4 Potência Complexa

Potência complexa é a definição de toda a potência circulante em um sistema elétrico em corrente alternada (AC). Ela é composta por: Potência ativa (P), potência reativa (Q) e potência aparente (S). Estas por sua vez possuem uma relação, podendo ser exemplificada como um triângulo retângulo, este por sua vez é denominado triângulo de potências. (BICHELS, 2018)

Figura 2 - Potência Complexa



Fonte: FULLWAT BLOG 2018

● Potência Ativa

Esta é a potência que efetivamente realiza o trabalho em um circuito, sua unidade de medida é o Watt (W), também é conhecida como potência real.

$$P = V \times I \times \cos(\phi) \quad (2.7)$$

Onde:

$P \rightarrow$ É o valor da Potência Ativa

$V \rightarrow$ É o valor da Tensão eficaz (RMS)

$I \rightarrow$ É o valor da corrente eficaz (RMS)

$\cos(\phi) \rightarrow$ É o Fator de Potência (FP)

● Potência Reativa

É a potência que alterna entre a fonte e a carga, não realizando trabalho útil, mas necessária para manter os campos elétricos e magnéticos no circuito. Ela é medida em volt-ampere reativo (VAR) e é dada pela fórmula:

$$Q = V \times I \times \sin(\phi) \quad (2.8)$$

Onde:

$Q \rightarrow$ É o valor da Potência Reativa

$V \rightarrow$ É o valor da Tensão eficaz (RMS)

$I \rightarrow$ É o valor da corrente eficaz (RMS)

$\sin(\phi) \rightarrow$ Seno do ângulo de defasagem da tensão e corrente.

● Potência Aparente

É a combinação vetorial da potência ativa e reativa, representada como um número complexo. Ela é medida em volt-ampères (VA) e é dada pelas fórmulas:

$$S = V \times I \quad (2.9)$$

$$S = P + jQ \quad (2.10)$$

Onde:

$S \rightarrow$ É o valor da Potência Aparente

$V \rightarrow$ É o valor da Tensão eficaz (RMS)

$I \rightarrow$ É o valor da corrente eficaz (RMS)

$P \rightarrow$ É o valor da Potência Ativa.

$Q \rightarrow$ É o valor da Potência Reativa.

2.2 Benefícios da monitoração de energia.

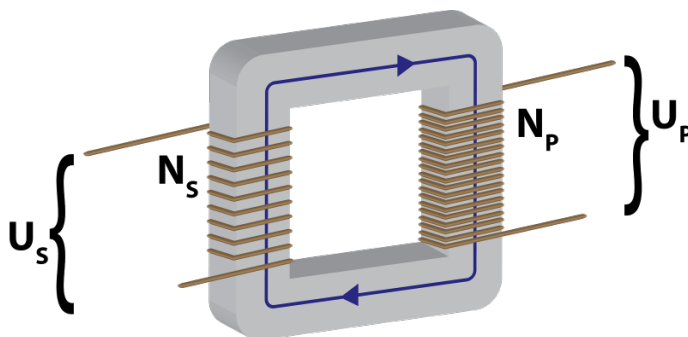
A monitoração de energia pode-se ter inúmeros benefícios (PASCALICCHIO, 2011), dentre eles estão:

- **Benefício econômico:** Indicar desperdícios de energia, e possibilitar a otimização do consumo; otimização de processos para reduzir os gastos com manutenção e substituição de equipamentos.
- **Benefício ambiental:** Com o aumento da eficiência energética, também é reduzida a demanda por energia de fontes poluentes. Além disso, ajuda a aumentar a vida útil dos aparelhos, prevenindo defeitos, o que diminui a necessidade de novos equipamentos e, consequentemente, reduz o uso de recursos naturais e a geração de resíduos.
- **Benefício operacional:** Diagnosticar defeitos ao identificar padrões irregulares de consumo e facilitando a manutenção preditiva e corretiva; permite otimizar o uso de equipamentos e processos, reduzindo o tempo de inatividade e aumentando a produtividade.

2.3 Transformador

Um transformador é um dispositivo elétrico que altera a tensão e corrente de um circuito de corrente alternada (CA) por meio de indução eletromagnética. Composto por duas bobinas, chamadas de primário e secundário, e por um núcleo magnético, o transformador opera quando uma corrente alternada no enrolamento primário gera um campo magnético variável, que induz uma tensão no enrolamento secundário. A relação entre o número de espiras de cada bobina determina se a tensão será aumentada (transformador elevador) ou reduzida (transformador rebaixador) no secundário, mantendo a potência aproximadamente constante, exceto por pequenas perdas (JORDÃO, 2002).

Figura 3 - Ilustração Transformador



Fonte: FÍSICA ILUSTRADA 2017

2.3.1 Talema AC1050

Transformador de forma toroidal, fabricado pela Talema Group, utilizado para medição de corrente (TC), possui relação de transformação de 1000:1, o mesmo é projetado para corrente nominal de 50A e máxima de 125A para frequências de 50 e 60Hz (AC1050 DATA SHEET, 2018).

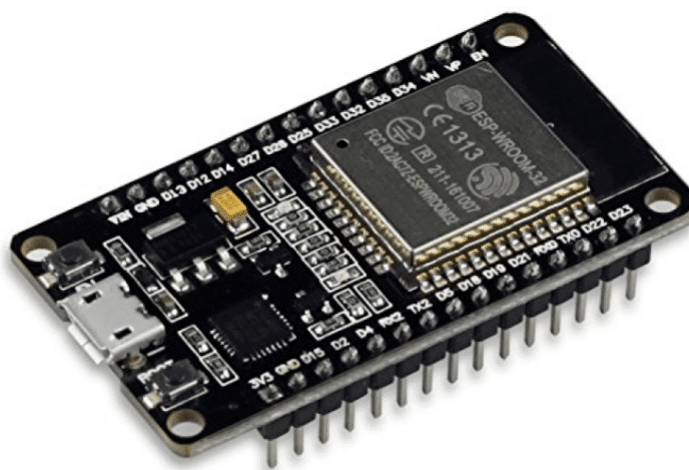
2.4 Microcontrolador

Um microcontrolador é um circuito integrado, que contém um processador, memória e periféricos de entrada/saída em um único chip. Pode ser considerado um "computador em um chip" projetado para controlar dispositivos eletrônicos. Microcontroladores são amplamente utilizados em sistemas embarcados, onde desempenham funções específicas em produtos como eletrodomésticos, automóveis, brinquedos e muitos outros. (DA SILVA, 2019)

2.4.1 ESP-32

O ESP-32 é um microcontrolador de baixo custo e alto desempenho desenvolvido pela Espressif Systems. Ele é amplamente utilizado em projetos de Internet das Coisas (IoT) devido às suas capacidades de conectividade Wi-Fi e Bluetooth, além de ser altamente versátil e potente em termos de recursos de hardware. (ESPRESSIF, 2024)

Figura 4 - ESP-32

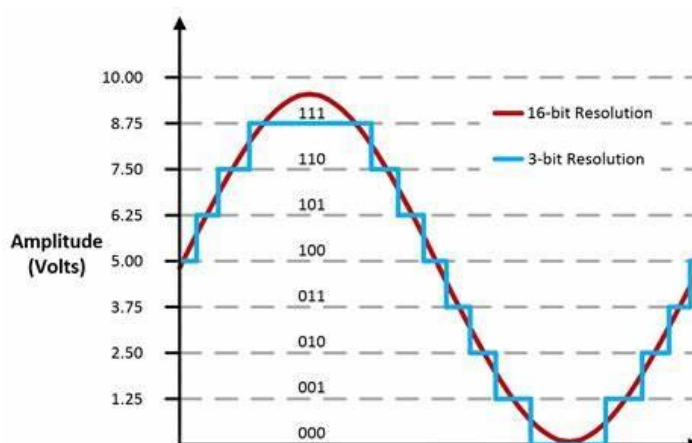


Fonte: EMBARCADOS 2018

2.5 ADC

Um ADC é um dispositivo ou circuito eletrônico que converte um sinal analógico (contínuo) em um sinal digital (discreto), que pode ser processado por sistemas digitais como microcontroladores, processadores e computadores. Suas características principais se dão pela resolução que é a precisão da representação da amplitude e taxa de amostragem, que é a quantidade de medições em um determinado período (ALBUQUERQUE, 2013).

Figura 5 - ADC

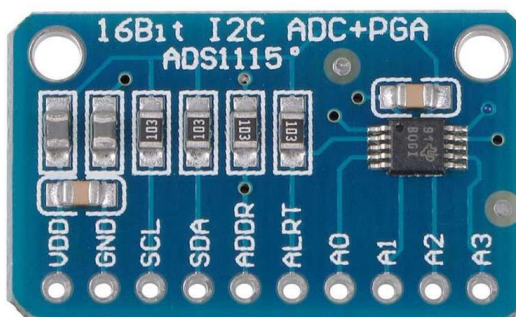


Fonte: CODE BLOGUES 2018

2.5.1 ADS1115

O ADS1115 é um conversor analógico-digital (ADC) de 16 bits de resolução com interface I2C, taxa de amostragem máxima de 860 amostras por segundo, possui 4 entradas analógicas, podendo ser utilizadas individualmente ou no modo diferencial, que utiliza duas entradas combinadas, fabricado pela Texas Instruments. Este conversor é facilmente encontrado em forma de módulo, facilitando sua utilização em protoboards (ADS1115 DATA SHEET, 2019)

Figura 6 - ADS1115



Fonte: COMPONENTS101 2021

2.6 IoT

IoT, ou Internet das Coisas, refere-se à conexão de objetos físicos à internet, permitindo que eles colem e troquem dados entre si e com sistemas. Esses objetos podem ser desde dispositivos simples, como lâmpadas e eletrodomésticos, até equipamentos industriais complexos. A IoT permite automação, monitoramento remoto e tomada de decisões com base em dados em tempo real, beneficiando áreas como saúde, transporte, agricultura e indústria (GOKHALE, 2018).

2.7 MQTT

MQTT (Message Queuing Telemetry Transport) é um protocolo de comunicação leve e eficiente, projetado para facilitar a troca de mensagens em redes de dispositivos com poucos recursos. Ele opera no modelo publish/subscribe (publicação/assinatura), que permite que dispositivos se comuniquem entre si por meio de tópicos. Devido às suas características, o MQTT é amplamente utilizado em aplicações de IoT, onde uma grande quantidade de dispositivos precisa se comunicar eficientemente, como em monitoramento remoto, controle de dispositivos, cidades inteligentes, e sistemas industriais (MQTT, 2022).

2.7.1 Broker MQTT

Um broker no MQTT é o núcleo central que gerencia a comunicação entre os clientes (dispositivos ou aplicativos). Ele recebe mensagens de clientes que atuam como publicadores e as distribui para os assinantes interessados nessas mensagens (MQTT, 2022).

2.7.1.1 EMQX

EMQX é um broker MQTT disponibilizado pela EMQ, possui código aberto e é considerado o broker mais escalável e confiável do mundo, possui baixíssima latência e tem atualmente mais de 100 milhões de dispositivos conectados. O broker é utilizado por grandes empresas, aumentando ainda mais sua credibilidade, este por sua vez disponibiliza o serviço em nuvem sem custo para aplicações limitadas. (EMQ TECHNOLOGIES INC, 2024).

2.8 Arduino IDE

Arduino IDE é uma plataforma de código aberto totalmente gratuita, no qual é utilizada para programar e gravar microcontroladores, como as placas Arduino, placas ESP, dentre outros modelos. Sua interface é leve e bem eficiente, e possui uma enorme quantidade de bibliotecas disponíveis, podendo ser utilizada com diferentes linguagens de programação. (GUSE, 2024).

2.9 Aplicação Web

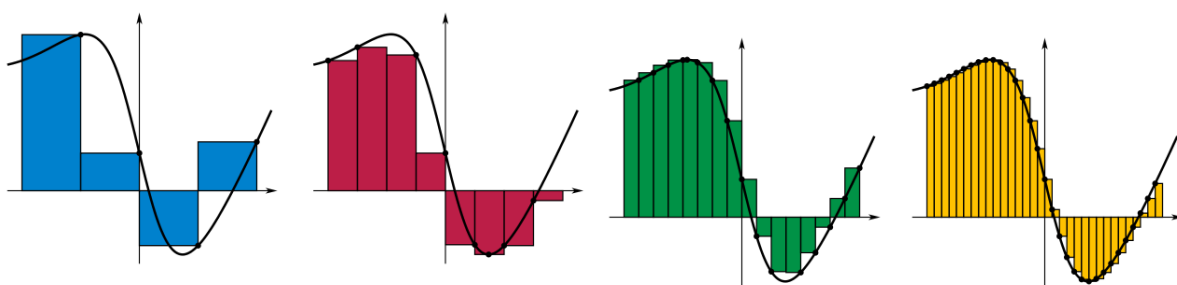
Uma aplicação web, é um software que roda diretamente em navegadores, sendo disponibilizado pela internet e pode ser acessado de diversas plataformas, pode ter integração com servidores, banco de dados dentre outras coisas. Normalmente dividido em front-end e back-end, podem ser utilizadas diversas tecnologias, mas as mais comuns são: HTML, CSS, PHP e JavaScript (MARTINS, 2023).

2.10 Soma de Riemann

A soma de Riemann é um método de se obter a aproximação da área de uma curva com intervalo definido. Para uma função contínua $f(x)$ definida em um intervalo $[a,b]$, a área sob a curva pode ser aproximada dividindo o intervalo em subintervalos menores e somando as áreas dos retângulos formados por esses subintervalos.

Ao se utilizar este método com subdivisões infinitas, o seu resultado tende a se aproximar de uma integral da mesma função (SCUCUGLIA, 2011).

Figura 7 - Exemplos de Soma de Riemann



Fonte: Wikimedia Commons 2007

Veja abaixo a equação para se utilizar da soma de Riemann:

$$S = \sum_{i=1}^n f(y_i)(x_i - x_{i-1}) \quad (2.11)$$

Onde:

$y_i \rightarrow$ É um ponto arbitrário do intervalo.

$n \rightarrow$ Número de subdivisões.

$S \rightarrow$ Soma da Área.

3 METODOLOGIA

A metodologia deste trabalho será com o método de pesquisa simples e exploratória a fim de conseguir definir a construção do dispositivo, as tecnologias embarcadas e as características do seu funcionamento. O estudo teve abordagem qualitativa com método dedutivo, para descrever a viabilidade de construir um dispositivo como o criado neste artigo.

4 DESENVOLVIMENTO

4.1 Descrição detalhada do Hardware

Para a criação do projeto atingindo seus objetivos, inicialmente foi definido quais tecnologias estariam empregadas no hardware do dispositivo. Partindo do pressuposto que o dispositivo seria microcontrolado, iniciou-se a pesquisa por qual microcontrolador utilizar.

Ao analisar o microcontrolador a ser utilizado no projeto, viu-se que o ESP-32 é a melhor opção, tendo em vista que possui um bom desempenho com um baixo custo, conta com ampla documentação disponível na internet devido sua comunidade ativa, e o mesmo possui comunicação WI-FI nativa, que é uma tecnologia indispensável para o projeto.

Devido a estudos realizados comprovando a baixa precisão do ADC interno do ESP-32, foi definida a utilização de um módulo ADC externo. A opção escolhida de ADC externo foi o ADS1115, devido seu baixo custo e boa resolução, apesar de ter baixa taxa de amostragem para a aplicação, essa desvantagem se tornou um desafio no decorrer do desenvolvimento do projeto.

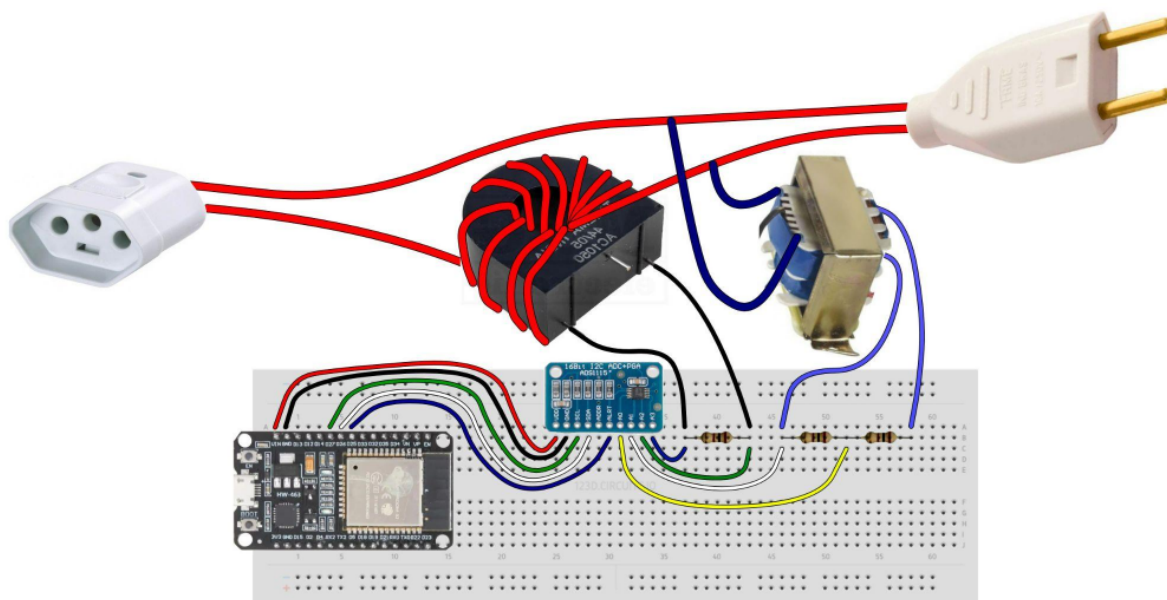
Para o funcionamento do dispositivo, é necessário que ele faça medição de valores de tensão e corrente, mas para isso é preciso a adequação dos valores para leitura do módulo ADC, este por sua vez possui uma entrada máxima de $\pm 6,144V$. A medição de tensão se dá por meio da utilização de um transformador abaixador de 220V para 12V, e em sua saída é utilizado um divisor de tensão com dois resistores de 100Ω . Para a medição de corrente, foi utilizado um transformador toroidal de relação 1000:1, ao se utilizar 10 espiras no primário sua relação fica 100:1 sendo utilizado com um resistor de amostragem de 100Ω .

O dispositivo conta com plugues macho e fêmea, para ser utilizado em qualquer tomada residencial, sendo necessário somente ligar a carga a ser monitorada em sua saída.

Devido o dispositivo ainda ser um protótipo, parte dos componentes foram montados em uma placa protoboard.

Segue abaixo o diagrama elétrico ilustrado com cada parte do hardware do dispositivo e suas devidas conexões.

Figura 8 - Esquema elétrico ilustrado do dispositivo

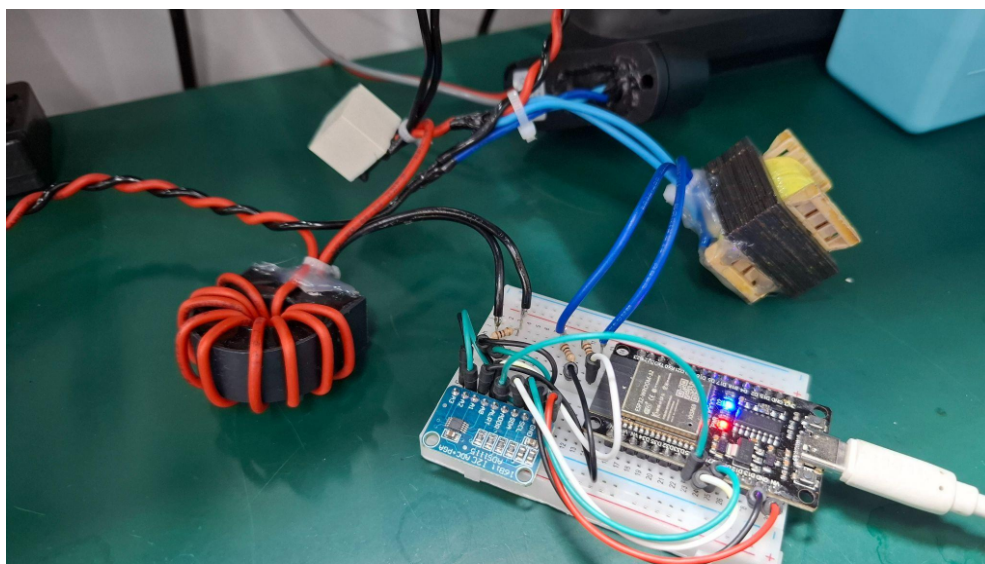


Fonte: Arquivo Próprio 2024

O hardware do dispositivo ainda está passando por melhorias, mas como não foi possível concluí-las, devem ficar para melhorias futuras. As melhorias seriam: Adicionar um relé para interromper o funcionamento da carga remotamente, e adicionar um fonte de 5V para alimentar o circuito, pois até então ele depende da alimentação oriunda do cabo USB conectado no ESP.

Segue abaixo a foto real do dispositivo.

Figura 9 - Dispositivo aferidor de energia IoT



Fonte: Arquivo Próprio 2024

4.2 Descrição detalhada do Software

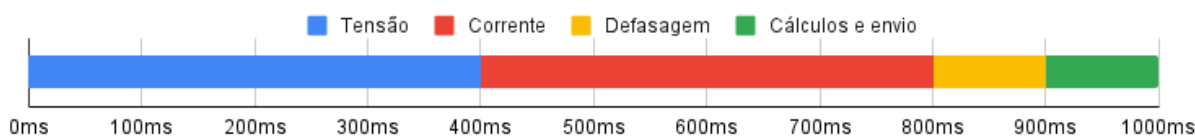
O software embarcado no projeto basicamente está dividido em duas partes, sendo o código do microcontrolador e o código da aplicação web para monitoração. Abaixo estão os detalhes do desenvolvimento e funcionamento de cada parte, respectivamente.

O código gravado no ESP-32, foi escrito na linguagem C++ (Apêndice A). Em seu funcionamento, inicialmente carrega as bibliotecas ou contêineres necessários para o funcionamento, sendo eles:

- **WiFi.h e WiFiClientSecure.h** - Bibliotecas para realizar a comunicação wifi.
- **ADS1X15.h** - Utilizada para configurar o funcionamento do módulo ADS1115 conforme a necessidade (ROBTILLAART, 2024).
- **PubSubClient.h** - Biblioteca necessária para configurar e utilizar o protocolo MQTT no microcontrolador ESP-32 (O'LEARY, 2020).
- **Wire.h** - Necessário para configurar o protocolo de comunicação I2C, entre o módulo e o microcontrolador.
- **vector** - Utilizado para trabalhar com uma lista de variáveis com tamanho flexível.
- **cmath** - Possibilita a utilização de operações matemáticas.

Ao iniciar o microcontrolador, em seu código o mesmo irá fazer uma sequência de passos para seu funcionamento. Inicialmente o mesmo irá conectar na rede WiFi previamente configurada em seu código, logo em seguida irá realizar a comunicação I2C com o módulo, posteriormente fará a conexão com o broker mqtt, este também deve ser configurado no código com os parâmetros do broker utilizado, que neste projeto está sendo utilizado o EMQX. Após essa parte inicial, o código inicia o loop principal, que no qual executa propriamente as medições.

Para estruturar o loop principal do código, foi definido que o dispositivo faria a publicação de uma medição por segundo no broker. Ao analisar a forma de funcionamento do ADS1115, viu-se que o mesmo não consegue realizar medições simultâneas para tensão e corrente mantendo sua taxa de amostragem máxima, o desempenho do mesmo também é afetado quando se tenta realizar cálculos complexos durante a medição. Devido a esta queda da performance do módulo, foi necessário definir um tempo para cada etapa da medição, para se obter sua máxima performance. A definição do tempo ficou definido da seguinte forma:

Figura 10 - Divisão do tempo

Fonte: Arquivo Próprio 2024

Inicialmente ao iniciar o desenvolvimento da medição de tensão, foi tentado utilizar a soma de riemann para chegar no valor RMS, aparentemente devido à baixa amostragem do ADC utilizado, a medição se apresentou inconsistente, tendo grandes variações no decorrer do tempo, fazendo com que desistisse de utilizar este método. Com isso foi adotada a medição utilizando o valor de pico para chegar no valor eficaz, este método se apresentou significativamente melhor no quesito estabilidade, mas ainda sim foi necessário aprimorar o código, para que a medição se apresentasse ainda mais estável. Juntamente na parte de medição, existe um fator de multiplicação utilizado posteriormente para calibrar o dispositivo.

O desenvolvimento do código para medição da corrente foi replicado da medição de tensão, com exceção da calibração realizada, que se dá por um fator de multiplicação que será citado posteriormente.

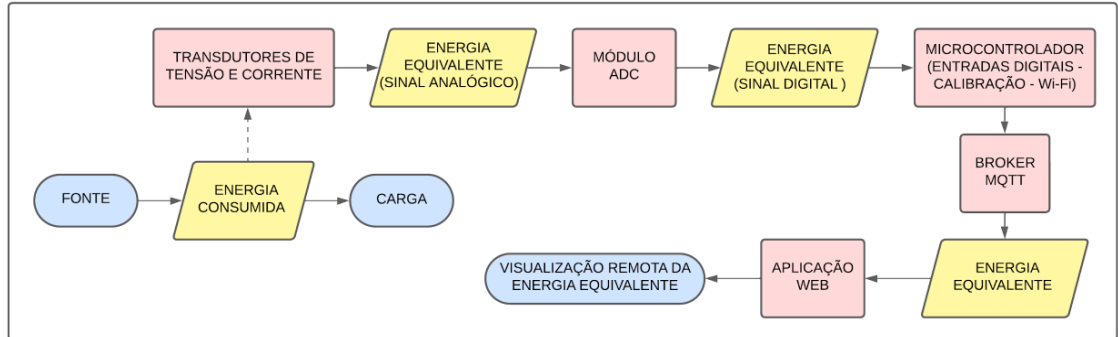
Para se obter o FP que também seria uma grandeza a ser medida, é preciso realizar a medição da defasagem entre tensão e corrente, para isso o ADS1115 realiza simultaneamente as medições, mas sua taxa de amostragem cai pela metade, chegando a 430 sps. Ao se considerar a frequência da rede de 60Hz e a taxa de amostragem, se obtém aproximadamente 7 medições por ciclo, sendo insuficiente para conseguir medir a defasagem com eficiência. Mesmo com este empecilho foi feita a tentativa de realizar esta medição utilizando aproximações, mas infelizmente não foi obtido sucesso na tentativa.

Na etapa final do loop, são realizados todos cálculos necessários para que sejam obtidos os valores de tensão e corrente eficazes, para só assim serem enviados para o broker. E assim o loop se repete de forma contínua, realizando as medições, calculando e as enviando a cada segundo.

O código da aplicação web para monitoração se baseia na utilização de uma página web desenvolvida com HTML e CSS, e para adicionar a funcionalidade de receber as medições via protocolo MQTT, foi utilizada a linguagem JavaScript (Apêndice B), no qual a mesma carrega uma biblioteca chamada MQTT.js. Com os parâmetros do broker mqtt definidos no código, o JavaScript começa a receber os dados enviados pelo dispositivo, os tratando se necessário, e os apresentando na tela para o usuário visualizar.

Após todas as definições de funcionamento do dispositivo realizadas, foi criado um fluxograma que retrata o funcionamento do mesmo, veja logo abaixo.

Figura 11 - Fluxograma



Fonte: Arquivo Próprio 2024

4.3 Calibração e Testes

Para o correto funcionamento do dispositivo, o mesmo necessita ser calibrado, e para isso na parte de medição foi adicionado um fator de multiplicação tanto na medição de tensão quanto na medição de corrente, no qual ao ser alterado altera o valor medido. Veja abaixo um exemplo da equação utilizada.

$$V_{rms} = \frac{V_{pico}}{\sqrt{2} \times k} \quad (3.1)$$

Onde:

V_{pico} → Valor de medição de pico oriunda do ADC.

V_{rms} → Valor eficaz ou RMS após o cálculo.

k → Fator de Multiplicação utilizado para calibração.

Para realizar a calibração foi necessário definir o fator de multiplicação como 1, para o dispositivo entregar os valores sem correção. Após isso foi aferido o valor de tensão no dispositivo com um multímetro, e foi feita uma relação entre o valor obtido pelo dispositivo e o valor obtido pelo multímetro para se obter o coeficiente k . Veja abaixo a equação utilizada.

$$\frac{V_{dis.}}{V_{mul.}} = k \quad (3.2)$$

Onde:

V_{dis} → Tensão obtida pelo dispositivo.

V_{mul} → Tensão obtida pelo multímetro.

k → Fator de Multiplicação utilizado para calibração.

Da mesma forma que foi realizada a calibração de tensão, a calibração de corrente segue a mesma lógica, mas substituindo o multímetro por um alicate amperímetro, e definindo o coeficiente k também para a corrente.

Segue abaixo um teste realizado após feita a calibração, onde é apresentado valores de tensão e corrente simultaneamente, na primeira imagem se tem a medição de comparação utilizando multímetro para aferir tensão e alicate amperímetro para aferir a corrente. Na segunda imagem se tem os valores lidos pelo dispositivo, apresentados na aplicação web.

Figura 12 - Medição com multímetro e alicate amperímetro



Fonte: Arquivo Próprio 2024

Figura 13 - Recorte da aplicação web



Fonte: Arquivo Próprio 2024

5 RESULTADOS

Ao se concluir a etapa de desenvolvimento, foram feitos testes do dispositivo, com os dados gerados foi visto que para medição de tensão, é possível se obter boa precisão tendo baixa variação. Para medição de corrente mesmo após calibração, o dispositivo ainda apresenta grande variação do valor apresentado, mostrando que ainda sim ele precisa de melhorias para entregar uma medição mais estável e precisa. Já para medição do fator de potência não foi possível obter esta medição devido a baixa taxa de amostragem do módulo utilizado.

6 CONCLUSÃO

Este trabalho contribuiu para meu aprendizado como um todo, apesar de não ter atingido todos os objetivos definidos, considero que o dispositivo construído teve um resultado geral satisfatório, mas deixando pontos para melhoria do mesmo.

Analisando cada objetivo deste trabalho, conclui-se que foi possível compreender todo o conceito de funcionamento do dispositivo, assim como a definição das tecnologias embarcadas visando custo benefício. No decorrer do desenvolvimento foram encontradas dificuldades que impactam diretamente nos resultados obtidos, sendo elas: alta complexidade do desenvolvimento do código (considerando a minha experiência na área) e sensor utilizado com baixa taxa de amostragem. Não foi possível aferir a precisão de forma quantitativa, somente qualitativa, mas foi possível deduzir que o dispositivo tem um resultado satisfatório por se considerar toda sua construção, mas ainda precisa de melhorias para funcionar da forma que foi proposta. As variações possíveis na construção do dispositivo, principalmente no intuito de sanar os problemas encontrados são: utilizar módulos e/ou CI's específicos para medição de tensão e corrente alternada, utilização de um módulo ADC com maior taxa de amostragem e a utilização de um relé no dispositivo para interromper a alimentação da carga de forma remota.

REFERÊNCIAS

AC1050 DATA SHEET, Talema Group. Disponível em:

<<https://talema.com/wp-content/uploads/datasheets/AC-1050.pdf>>. Acesso em 01 out. 2024.

ADS1115 DATA SHEET, product information and support | TI.com. Disponível em:

<https://www.ti.com/product/ADS1115?keyMatch=ADS1115&tisearch=universal_search&usecase=GPN-ALT#support-training>. Acesso em: 27 out. 2024.

ALBUQUERQUE, BRUNNA DE SOUZA. Descrição de um conversor analógico-digital de 12 Bits. João Pessoa, 2013. Disponível em

<<https://repositorio.ufpb.br/jspui/handle/123456789/493>>. Acesso em 02 nov. 2024.

BICHELS, ARLEI. Sistemas elétricos de potência: Métodos de análise e solução. Curitiba, EDUTFPR. 2018. ISBN: 978-85-7014-208-5 (E-book) Disponível em:

<<http://repositorio.utfpr.edu.br/jspui/>>

CODE BLOGUES, 2018. Disponível em:

<https://codebloges.blogspot.com/2018/07/dacdigital-to-analog-conversion-using.html#google_vignette>.

COMPONENTS101, 2021. Disponível em:

<<https://components101.com/modules/ads1115-module-with-programmable-gain-amplifier>>

DA SILVA, R. O., Araújo, W. M. & Cavalcante, M. M. (2019). Visão Geral Sobre Microcontroladores e Prototipagem com Arduino. TECNOLOGIAS EM PROJEÇÃO, 10. Disponível em: <<https://projecaociencia.com.br/index.php/Projecao4/article/view/1357>>

EMBARCADOS, 2016. Disponível em:

<<https://embarcados.com.br/wp-content/uploads/2016/01/04-potencias-senoide-medidas.png>>

EMBARCADOS, 2018. Disponível em:

<<https://embarcados.com.br/ambiente-esp32-no-windows/>>.

EMQ TECHNOLOGIES INC. About Us. Disponível em: <<https://www.emqx.com/en/about>>. Acesso em: 9 nov. 2024.

ESPRESSIF SYSTEMS. ESP32. 2024. Disponível em:

<<https://www.espressif.com/en/products/socs/esp32>>. Acesso em 20 out. 2024.

FÍSICA ILUSTRADA, 2017. Disponível em:

<<https://fisicailustrada.blogspot.com/2017/03/transformador.html>>.

FRENKEL, Josif. Princípios de Eletrodinâmica Clássica. 1996.

FRÓES, André Luís Delvas. Princípios de eletricidade e magnetismo. Editora e Distribuidora Educacional S.A. 2017.

FULLWAT BLOG, 2018. Disponível em:

<<http://blog.fullwat.com/que-es-el-factor-de-potencia/>>

GOKHALE, Pradyumna; BHAT, Omkar; BHAT, Sagar. Introduction to IOT. International Advanced Research Journal in Science, Engineering and Technology, v. 5, n. 1, 2018.

GUSE, R. O que é Arduino IDE? Como baixar e instalar a nova versão. Disponível em:

<<https://www.makerhero.com/blog/arduino-ide/>>. Acesso em: 9 out. 2024.

HALLIDAY, David; RESNICK, Robert; WALKER, Jearl. Fundamentos de Física. Volume 3. LTC Editora, 2013.

JORDÃO, Rubens Guedes. Transformadores. Editora Blucher, 2002.

MARTINS, D. Aplicação web: o que é, diferença para website, como funciona e mais!

Disponível em: <<https://rockcontent.com/br/blog/aplicacao-web/>>.

MAKADAM, S.; RAMASWAMY, R.;TRIPATHI, S. Internet of Things (IoT): A Literature Review. Journal of Computer and Communications, 2015.

MQTT. MQTT - The Standard for IoT Messaging. Disponível em: <<https://mqtt.org/>>.

MQTTJS. GitHub - mqttjs/MQTT.js: The MQTT client for Node.js and the browser, 2023.

Disponível em: <<https://github.com/mqttjs/MQTT.js>>. Acesso em: 15 nov. 2024.

O'LEARY, N. Arduino Client for MQTT, 2020. Disponível em:

<<https://github.com/knolleary/pubsubclient>>. Acesso em: 12 jul. 24

PASCALICCHIO, A. C. Perspectiva econômica e modelo de negócio da tecnologia de telecomunicação nas redes de distribuição de energia elétrica no Brasil. Tese de Doutorado, Programa de Pós-graduação em Energia da Universidade de São Paulo, SP, Brasil, 2011.

PDE 2031. Plano decenal de expansão de energia 2031, 2022. Disponível em:

<<https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/plano-decenal-de-expansao-de-energia-2031>>. Acesso em 01 mar. 24.

ROBTILLAART. GitHub - RobTillaart/ADS1X15: Arduino library for ADS1015 = I2C 12 bit ADC and ADS1115 = I2C 16 bit ADC, 2024. Disponível em:

<<https://github.com/RobTillaart/ADS1X15>>. Acesso em: 15 nov. 2024.

SCUCUGLIA, Ricardo; RICHIT, Andriceli. O papel das tecnologias informáticas na investigação do conceito de soma de riemann. 2011.

SENRA, Renato. Energia Elétrica - Medição, Qualidade e Eficiência, 2015.

SISTEMA INTERNACIONAL DE UNIDADES. Tradução do Grupo de Trabalho luso-brasileiro do Inmetro e IPQ. Brasília, DF. 2021.

WIKIMEDIA, 2021. Disponível em:

<https://commons.wikimedia.org/wiki/File:Riemann_sum_convergence.png>.

APÊNDICE A - CÓDIGO MICROCONTROLADOR

```
#include "WiFi.h"
#include <WiFiClientSecure.h>
#include <ADS1X15.h>
#include <PubSubClient.h>
#include "Wire.h"
#include <vector>
#include <cmath>
#include <Preferences.h>

Preferences preferences;

ADS1115 ADS(0x48);

bool printrms = true;
volatile bool RDY, print = false;
bool adReadOK[2] = {false, false};
int lastRequestAD = 0;
uint8_t pair = 01;
std::vector<uint16_t> vector_01, vector_23;
std::vector<int> vector_ph01, vector_ph23;
std::vector<unsigned long> micros_01, micros_23, microscroszero_01, microscroszero_23;
int16_t val_01, val_23, lastval_01, lastval_23 = 0;
int samples, mode = 0;
int sps, period_01, period_23 = 0;
uint32_t now, last, timerMode, timeStartCycle_01, timeStartCycle_23 = 0;
float frequency, sumperiod, RMS_01, RMS_23, prevRMS_01, prevRMS_23 = 0.00;
float croszero;
static int countCycle = 0;
int tickModeChange = 400;
int microsSample_01, microsSample_23 = 0;
float k1, k2;
char rms01char[6], rms23char[6];
const char* ssid = "Giovana"; //NOME DA REDE WIFI
const char* password = "souzapio103"; //SENHA
const char *mqtt_broker = "m4b00291.ala.eu-central-1.emqxsl.com";
const char *mqtt_topic = "testtopic"; // MQTT topic
const char *mqtt_topic1 = "testtopic/tensao";
const char *mqtt_topic2 = "testtopic/corrente";
const char *mqtt_topick1 = "testtopic/k1";
const char *mqtt_topick2 = "testtopic/k2";
const char *mqtt_username = "paulo"; // MQTT username for authentication
const char *mqtt_password = "1234"; // MQTT password for authentication
```

```

const int mqtt_port = 8883; // MQTT port (TCP)
const char *ca_cert = R"EOF(
-----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIQCDvgVpBCRRrGhdWrJWZHHSjANBgkqhkiG9w0BAQUFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQ
QLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCB
D
QTAEFw0wNjExMTAwMDAwMDBaFw0zMTEwMTAwMDAwMDBaMGExCzAJBgNVBA
YTAIVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2Vy
dC5j
b20xIDAeBgNVBAMTF0RpZ2l0eXJ0IEVsb2JhbCBSb290IENBMiIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4jvhEXLeqKTTo1eqUUKKPC3eQyaKl7hLollsB
CSDMAZOnTjC3U/dDxGkAV53ijSLdhwZAAIEJzs4bg7/fzTxRuLWZscFs3YnFo97
nh6Vfe63SKMI2tavegw5BmV/Sl0fvBf4q77uKNd0f3p4mVmFaG5cIzJLv07A6Fpt
43C/dxC//AH2hdmoRBBYMql1GNXRor5H4idq9Joz+EkIYIvUX7Q6hL+hqkpMfT7P
T19sdl6gSzeRntwi5m3OFBqOasv+zbMUZBfHWymeMr/y7vrTC0LUq7dBMtoM1O/4
gdW7jVg/tRvoSSiicNoxBN33shbyTApOB6jtSj1etX+jkMOvJwIDAQABo2MwYTAO
BgNVHQ8BAf8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUA95
QNVbR
TLtm8KPiGxvDI7I90VUwHwYDVR0jBBgwFoAUA95QNVbRTLtm8KPiGxvDI7I90VUw
DQYJKoZIhvcNAQEFBQADggEBAMucN6pIExIK+t1EnE9SsPTfrgT1eXkIoyQY/Esr
hMATudXH/vTBH1jLuG2cenTnmCmrEbXjcKChzUyImZOMkXDiqw8cvpOp/2PV5Adg
06O/nVsJ8dWO41P0jmP6P6fbtGbfYmbW0W5Bjflttep3Sp+dWOIrWcBAI+0tKIJF
PnlUkiaY4IBIqDfv8NZ5YBberOgOzW6sRBc4L0na4UU+Krk2U886UAb3LujEV0ls
YSEY1QSteDwsOoBrp+uvFRtp2InBuThs4pFsis9kuXclVzDAGySj4dzp30d8tbQk
CAUw7C29C79Fv1C5qfPrmAESrcilxpg0X40KPMbp1ZWVbd4=
-----END CERTIFICATE-----
)EOF";

WiFiClientSecure espClient;
PubSubClient mqtt_client(espClient);

void setWireClock(uint32_t speed = 1000000);

void setup() {

    // initialize digital pin LED_BUILTIN as an output.
    pinMode(2, OUTPUT);
    Serial.begin(500000);
    connectToWiFi();
    espClient.setCACert(ca_cert);

```

```

mqtt_client.setServer(mqtt_broker, mqtt_port);
mqtt_client.setCallback(callback);
connectToMQTTBroker();
Wire.setPins(26, 27);
Wire.begin();
ADS.begin();
Serial.print("ADS1115 connected: ");
Serial.println(ADS.isConnected() ? "true": "false");
ADS.setMode(0);
ADS.setDataRate(7);
ADS.setWireClock(1000000);
xTaskCreatePinnedToCore(loop2,"adcLoop",1000,NULL,1,&Task1,0);
ADS.setComparatorThresholdHigh(0x8000);
ADS.setComparatorThresholdLow(0x0000);
ADS.setComparatorQueConvert(0);
pinMode(25, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(25), adsReady, RISING);
ADS.requestADC_Differential_0_1();
preferences.begin("my-app", false);
k1 = preferences.getFloat("k1", 167.0);
k2 = preferences.getFloat("k2", 2000.0);
preferences.end();
}

}

void adsReady(){
  RDY = true;
}

bool handleConversion()
{
  if (RDY)
  {
    RDY = false;
    if (mode == 0){ //medição de tensão
      val_01 = ADS.getValue();
      vector_01.push_back(abs(val_01));
      ADS.requestADC_Differential_0_1();
      lastRequestAD = 0;
      return true;
    }
    if (mode == 1){ //medição de corrente
      val_23 = ADS.getValue();

```



```

vector_23.push_back(abs(val_23));
ADS.requestADC_Differential_2_3();
lastRequestAD = 1;
return true;
}
if (mode == 2){
    if(lastRequestAD == 0){
        val_01 = ADS.getValue();
        vector_ph01.push_back(val_01);
        micros_01.push_back(micros());
        ADS.requestADC_Differential_2_3();
        lastRequestAD = 1;
        adReadOK[0] = true;
        return false;
    }
    if(lastRequestAD == 1){
        val_23 = ADS.getValue();
        vector_ph23.push_back(val_23);
        micros_23.push_back(micros());
        ADS.requestADC_Differential_0_1();
        lastRequestAD = 0;
        adReadOK[1] = true;
        //return false;
    }
    return (adReadOK[0] & adReadOK[1]);
}
}
if(mode == 3 && printrms && vector_01.size() > 0 && vector_23.size() > 0){
    sort(vector_01.begin(), vector_01.end(), std::greater<uint16_t>());
    sort(vector_23.begin(), vector_23.end(), std::greater<uint16_t>());
    vector_01.resize(30);
    vector_23.resize(30);
    prevRMS_01 = RMS_01;
    prevRMS_23 = RMS_23;
    RMS_01 = static_cast<float>(vector_01[15]) / (sqrt(2) * k1);
    RMS_23 = static_cast<float>(vector_23[15]) / (sqrt(2) * k2);
    RMS_01 = static_cast<float>(prevRMS_01 + RMS_01) / 2;
    RMS_23 = static_cast<float>(prevRMS_23 + RMS_23) / 2;
    vector_01.clear();
    vector_23.clear();

    for(int i = 1; i <= 40; i++){
        if(vector_ph01[i - 1] < 0 & vector_ph01[i] > 0){

```

```

        croszero = static_cast<float>(abs(vector_ph01[i - 1])) / (abs(vector_ph01[i - 1]) +
vector_ph01[i]);
        microscroszero_01.push_back(((micros_01[i] - micros_01[i - 1]) * croszero) +
micros_01[i - 1]);
    }
    if(vector_ph23[i - 1] < 0 & vector_ph23[i] > 0){
        croszero = static_cast<float>(abs(vector_ph23[i - 1])) / (abs(vector_ph23[i - 1]) +
vector_ph23[i]);
        microscroszero_23.push_back(((micros_23[i] - micros_23[i - 1]) * croszero) +
micros_23[i - 1]);
    }
}

```

```

int smallperiod = 16666;
int n = 0;
for(int i = 0; i < 5; i++){
    int period = microscroszero_01[2] - microscroszero_23[i];
    period = abs(period);
    if(period < smallperiod){
        smallperiod = period;
        n = i;
    }
}
int num = microscroszero_01[2] - microscroszero_23[n];

```

```

rms01char[6] = {0};
rms23char[6] = {0};
sprintf(rms01char, "%.2f", RMS_01);
sprintf(rms23char, "%.2f", RMS_23);
mqtt_client.publish(mqtt_topic1, rms01char);
mqtt_client.publish(mqtt_topic2, rms23char);

```

```

microscroszero_01.clear();
microscroszero_23.clear();
vector_ph01.clear();
vector_ph23.clear();
micros_01.clear();
micros_23.clear();

```

```

    printrms = false;
}
return false;

```

```
}
```

```
void connectToWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("\nConnecting to WiFi Network ..");
  while(WiFi.status() != WL_CONNECTED){
    Serial.println(".");
    delay(1000);
  }
  Serial.println("\nConnected to the WiFi network");
  Serial.print("Local ESP32 IP: ");
  Serial.println(WiFi.localIP());
}
```

```
void connectToMQTTBroker() {
  while (!mqtt_client.connected()) {
    String client_id = "esp8266-client-" + String(WiFi.macAddress());
    Serial.printf("Connecting to MQTT Broker as %s.....\n", client_id.c_str());
    if (mqtt_client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
      Serial.println("Connected to MQTT broker");
      mqtt_client.subscribe(mqtt_topic);
      mqtt_client.subscribe(mqtt_topick1);
      mqtt_client.subscribe(mqtt_topick2);
      // Publish message upon successful connection
      mqtt_client.publish(mqtt_topic, "Hi EMQX I'm ESP8266 ^^");
    } else {
      Serial.print("Failed to connect to MQTT broker, rc=");
      Serial.print(mqtt_client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}
```

```
void callback(char* topic, byte* message, unsigned int length) {
  Serial.print("Message arrived on topic: ");
  Serial.print(topic);
  Serial.print(". Message: ");
  String messageTemp;

  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
```

```

    messageTemp += (char)message[i];
}
Serial.println();

```

```

if (String(topic) == "testtopic/k1") {
    Serial.println("Changing voltage k calibration");
    Serial.println(messageTemp.toFloat());
    k1 = messageTemp.toFloat();
    preferences.begin("my-app", false);
    preferences.putFloat("k1", k1);
    preferences.end();
}
if (String(topic) == "testtopic/k2") {
    Serial.println("Changing current k calibration");
    Serial.println(messageTemp.toFloat());
    k2 = messageTemp.toFloat();
    preferences.begin("my-app", false);
    preferences.putFloat("k2", k2);
    preferences.end();
}
}

```

```

void sampleCount(){
    now = micros();
    sps = 1000000 / (now - last);
    last = now;
}

```

```

void modeTick(){
    now = micros();
    if((now - timerMode) > (tickModeChange * 1000)){
        mode++;
        if(mode == 4)
        {
            tickModeChange = 400;
            mode = 0;
            printrms = true;
        }
        if(mode == 2)
        {
            tickModeChange = 100;
        }
        timerMode = now;
    }
}

```

```

    }
    else if(timerMode > now){
        timerMode = now;
    }
}

void frequency_01() {

    if(lastval_01 <= 0 & val_01 > 0){
        period_01 = now - timeStartCycle_01;
        timeStartCycle_01 = now;
    }
}

void frequency_23() {

    if(lastval_23 <= 0 & val_23 > 0){
        period_23 = now - timeStartCycle_23;
        timeStartCycle_23 = now;
        sumperiod += period_23;
        countCycle ++;
        if(countCycle >= 10){
            countCycle = 0;
            frequency = (10000000/sumperiod);
            sumperiod = 0;
        }
    }
}

void loop() {
    mqtt_client.loop();
    modeTick();
    if (Serial.available()){
        String serialk = Serial.readString();
        if(serialk.substring(0,2) == "k1"){
            serialk.remove(0,3);
            k1 = serialk.toFloat();
            preferences.begin("my-app", false);
            preferences.putFloat("k1", k1);
            preferences.end();
        }
        if(serialk.substring(0,2) == "k2"){

```

```
        serialk.remove(0,3);
        k2 = serialk.toFloat();
        preferences.begin("my-app", false);
        preferences.putFloat("k2", k2);
        preferences.end();
    }

}
if (handleConversion() == true)
{
    sampleCount();

    print = false;

}
else if(print){

    lastval_01 = val_01;
    lastval_23 = val_23;
    val_01 = 0;
    val_23 = 0;
    adReadOK[0] = false;
    adReadOK[1] = false;
    print = false;
}
}
```

APÊNDICE B - CÓDIGO APLICAÇÃO WEB

index.html

```
<head>
  <meta charset="utf-8">
  <title>Iot Power Meas</title>
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <header>
    <h1>IoT Power Measuring</h1>
  </header>
  <main>
    <section class="dash conection">
      <h2>Status de conexão</h2>
      <span class="brokerstatus">Broker MQTT Desconectado</span>
      <span class="espstatus">ESP-32 Desconectado</span>
    </section>
    <section class="dash realmeas">
      <h2>Medição em tempo real</h2>
      <div class="values">
        <div>
          <h3>Tensão:</h3>
          <p class="voltage"></p>
        </div>
        <div>
          <h3>Corrente:</h3>
          <p class="current"></p>
        </div>
        <div>
          <h3>FP:</h3>
          <p>XXX</p>
        </div>
      </div>
    </section>
    <section class="dash realpower">
      <h2>Potência</h2>
      <div class="values">
        <div>
          <h3>Aparente:</h3>
          <p class="VA"></p>
        </div>
      </div>
    </section>
  </main>
</body>
```

```

    <div>
      <h3>Ativa:</h3>
      <p class="P">XXX</p>
    </div>
    <div>
      <h3>Reativa:</h3>
      <p class="Q">XXX</p>
    </div>
  </div>
</section>
<section class="dash graphpower">
  <h2>Triângulo de potências</h2>
  <p class="current">XXXX</p>

</section>
<section class="dash graphenergy">
  <h2>Energia</h2>
  <p class="current">XXXX</p>
  <input class="topic" type="text">
  <input class="message" type="text">
  <input class="btn" type="button">
</section>
</main>
<script defer src="script.js"></script>
<script src="https://unpkg.com/mqtt/dist/mqtt.min.js"></script>
</body>

</html>

```

script.js

```

var txt = document.querySelector("span.brokerstatus")
var espstatus = document.querySelector("span.espstatus")
var voltage = document.querySelector("p.voltage")
var current = document.querySelector("p.current")
var va = document.querySelector("p.VA")
var btn = document.querySelector("input.btn")
var topic = document.querySelector("input.topic")
var message = document.querySelector("input.message")
var tensao, corrente
function send() {
  client.publish(topic.value, message.value)
}
btn.addEventListener("click", send, false)

```



```

txt.innerText = "";
const clientId = 'mqttjs_' + Math.random().toString(16).substr(2, 8)
const host = 'wss://m4b00291.ala.eu-central-1.emqxsl.com:8084/mqtt'
//const host = "wss://broker.emqx.io:8084"
const topicsub1 = 'testtopic/tensao'
const topicsub2 = 'testtopic/corrente'
const options = {
  username: 'paulo',
  password: '1234',
  keepalive: 60,
  clientId: clientId,
  protocolId: 'MQTT',
  protocolVersion: 4,
  clean: true,
  reconnectPeriod: 1000,
  connectTimeout: 30 * 1000,
  will: {
    topic: 'WillMsg',
    payload: 'Connection Closed abnormally..!',
    qos: 0,
    retain: false
  },
}
txt.innerText = 'Conectando ao Broker ...'
const client = mqtt.connect(host, options)
client.on('error', (err) => {
txt.innerText = 'Connection error'
client.end()
})
client.on('reconnect', () => {
txt.innerText = 'Reconnecting...'
})
client.on('connect', () => { txt.innerText = "Conectado ao broker EMQX"
//Subscribe
client.subscribe(topicsub1, { qos: 0 })
client.subscribe(topicsub2, { qos: 0 }) }) /* Unsubscribe client.unsubscribe('testtopic', ()
=> { alert('Unsubscribed'); })*/
client.on("message", (topic, message) => {
  if(topic == topicsub1){
    tensao = message + 0
    voltage.innerText = message.toString() + "V";
    va.innerText = (tensao * corrente).toFixed(2).toString() + "VA"
  }
  if(topic == topicsub2){

```

```

    corrente = message + 0
    current.innerText = message.toString() + "A";
    //client.publish('presence', 'bin hier')

}

espstatus.innerText = "ESP-32 Conectado!"
//console.log(topic)
// message is Buffer
//console.log(message.toString())
})

style.css

@import
url('https://fonts.googleapis.com/css2?family=Kdam+Thmor+Pro&display=swap');
html,
body {
    height: 100%;
    width: 100%;
    max-width: 100%;
    overflow-x: hidden;
}
* {
    font-family: 'Kdam Thmor Pro',sans-serif;
    margin: 0px;
    padding: 0px;
    box-sizing: border-box;
}
h1 {
    font-size: 50px;
}

header {
    display: flex;
    padding: 20px;
    align-items: center;
    justify-content: center;
}

main {
    padding: 50px 150px;
    font-size: 20px;
    display: grid;

```

```

    grid-template: 170px 170px 170px/ 1fr 1fr 1fr;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: space-between;
    gap: 30px;
    height: 100vh;
}
main * {
    text-align: center;
}
footer.about{
    display: flex;
    gap: 20px;
    justify-content: center;
    align-items: center;
    border-radius: 10px;
    height: 140px;
    background-color: #ddd;
}

footer.about span{
    height: 50px;
    width: 50px;
    background-color: #fff;
    border: 1px solid black;
    border-radius: 10px;
    box-shadow: 0px 0px 10px black;
}

section.dash{
    display: flex;
    flex-direction: column;
    border-radius: 20px 0px;
    border: 1px solid #000;
    box-shadow: 5px 40px 0px #ddd inset, -5px -5px 0px #ddd inset, 5px 5px 10px #888;
    padding: 0px 30px 0px;
}

section.dash h2{
    align-self: flex-start;
}

section.conection {
    flex-direction: column;

```

```

    gap: 20px;
    justify-content: flex-start;
    align-items: flex-start;
}

section h2 {
    font-size: 1.4rem;
    line-height: 40px;
}

section.realmeas, section.realpower {
    grid-column: 2/span 2;
}

.dash div.values{
    height: 100%;
    width: 100%;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    gap: 10px;
    padding: 10px 0px 14px 0px;
}

.dash .values div{
    background-color: #e0e0e0;
}

.values p{
    font-size: 30px;
}

section.realpower {
    grid-column: 1/span 2;
}

section.graphpower {
    grid-area: 2/3/4/4;
}

section.graphenergy {
    grid-column: 1/span 2;
}

```