



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO - UFERSA
DEPARTAMENTO DE ENGENHARIAS -
CENTRO MULTIDISCIPLINAR DE CARAÚBAS - CMC
BACHARELADO EM ENGENHARIA ELÉTRICA

ESDRAS DE SOUSA SOARES

DESENVOLVIMENTO DE MEDIDOR DE ENERGIA DE BAIXO
CUSTO APLICADO A INTERNET DAS COISAS UTILIZANDO
ESP-32

CARAÚBAS - RN

2021



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO - UFERSA
DEPARTAMENTO DE ENGENHARIAS -
CENTRO MULTIDISCIPLINAR DE CARAÚBAS - CMC
BACHARELADO EM ENGENHARIA ELÉTRICA

ESDRAS DE SOUSA SOARES

DESENVOLVIMENTO DE MEDIDOR DE ENERGIA DE BAIXO
CUSTO APLICADO A INTERNET DAS COISAS UTILIZANDO
ESP-32

Trabalho de Conclusão de Curso submetido à Universidade Federal Rural do Semi-Árido como parte dos requisitos necessários para a obtenção do Título de Bacharel em Engenharia Elétrica.

Orientador: HUGO MICHEL CAMARA DE AZEVEDO MAIA

CARAÚBAS - RN
2021

SOARES, ESDRAS DE SOUSA

DESENVOLVIMENTO DE MEDIDOR DE ENERGIA DE BAIXO CUSTO APLICADO A INTERNET DAS COISAS UTILIZANDO ESP-32/ ESDRAS DE SOUSA SOARES. – **CARAÚBAS - RN, 2021.**

75 p. : il. (algumas color.) ; 30 cm.

Orientador: HUGO MICHEL CAMARA DE AZEVEDO MAIA

Monografia – UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO - UFERSA
DEPARTAMENTO DE ENGENHARIAS -
BACHARELADO EM ENGENHARIA ELÉTRICA, **2021.**

1. medidor. 2. EPP-32. 3. *internet*. 4. *wi-fi* I. Título.

ESDRAS DE SOUSA SOARES

**DESENVOLVIMENTO DE MEDIDOR DE ENERGIA
DE BAIXO CUSTO APLICADO A INTERNET DAS
COISAS UTILIZANDO ESP-32**

Trabalho de Conclusão de Curso submetido
à Universidade Federal Rural do Semi-Árido
como parte dos requisitos necessários para a ob-
tenção do Título de Bacharel em Engenharia
Elétrica.

Defendida em: 09 de novembro de 2021

Banca Examinadora

**HUGO MICHEL CAMARA DE
AZEVEDO MAIA**
Prof. Dr. (UFERSA)
Orientador

FRANCISCO BRITO DE ASSIS FILHO
(Prof. Dr. (UFERSA))
Membro da Banca

HUNBERTO DIONÍSIO DE ANDRADE
Prof. Dr. (UFERSA)
Membro da Banca

MARCUS VINÍCIUS SILVÉRIO COSTA
Prof. Dr. (UFERSA)
Membro da Banca

**CARAÚBAS - RN
2021**

Dedico este trabalho a todo o curso de Engenharia Elétrica da UFERSA, corpo docente e discente, é uma honra fazer parte disso.

AGRADECIMENTOS

Agradeço primeiramente a mim, por ter tido forças de ter chegado até aqui, e por ter vencido cada batalha.

Agradeço grandemente ao meu orientador, o Dr. Hugo Michel, pois está do meu lado desde o início do curso, como orientador de projetos de pesquisa, depois orientador do meu TCC de CeT, ele foi a inspiração para eu seguir para engenharia elétrica. O professor também foi meu orientador de defesa de estágio e agora, na última batalha da graduação, orientador do TCC da engenharia. Agradeço por toda disponibilidade e paciência por todo esse tempo.

Agradeço à minha companheira Dayanne Bezerra que ajudou durante todo este percurso e nunca deixou que eu desanimasse por mais difícil que a situação se apresentasse.

Agradeço aos meus pais, Sinézio e Perpetua, que sempre me ajudaram no que puderam, as vezes mesmo sem poder, pois com o suor de vossos rostos é fundada essa formação.

Agradeço aos meus primos que, na verdade, são meus irmãos, Aury, Robson, Abinal, Joel Soares, Júnior e André. Cada um foi e é fundamental nesta jornada chamada vida.

Um agradecimento especial aos meus tios: Neli e Milton pois são exemplo de pessoas, e sempre sou bem recebido em sua casa.

Agradeço a toda minha família, que cada me ajudou a chegar até aqui, em especial minha avó Terezinha, que com certeza não teria chegado até aqui sem o apoio dela.

Aos amigos que a UFERSA me trouxe, Jeferson, Joel, Ozorio e Lorano um salve abraço.

RESUMO

Nos últimos anos, houve diversos aumentos na conta de energia do consumidor, que já não era barata no Brasil, país que se cobra uma das mais altas taxas de energia no mundo. Esclarecendo um pouco sobre o consumo de eletricidade no país, mais de um terço do consumo de energia são de residências, porém praticamente não existem mecanismos encontrados de forma fácil que auxiliem o consumidor a ter uma maior transparência sobre os gastos em tempo real de sua residência. Com isso, não é criado um hábito, ou ainda mais, uma cultura econômica de análise de desperdício de energia, dado que não há como saber o que está sendo gasto. Pensando nisso, este trabalho tem o intuito de construir um medidor de energia elétrica barato e também criar uma plataforma que possa ser acessada facilmente de forma remota via *internet* em qualquer dispositivo com acesso à rede de internet. O medidor aqui apresentado, pode ligar ou desligar qualquer aparelho elétrico remotamente e medir o consumo também de forma remota. Para construção desse medidor foi utilizado um sensor de corrente STC-13 e um de tensão Zmpt101b, ainda foi utilizado o microcontrolador ESP-32 que tem a função de ler, tratar e enviar os dados via *wi-fi*, para a plataforma *Node-Red*, partindo da desta plataforma as informações são armazenadas em um banco de dados em nuvem da empresa *IBM*. Esses dados serão mostrados para o usuários em uma plataforma remota construída em *NodeJS* e *ReacJs*.

Palavras-chave: medidor. ESP-32. *internet*. *wi-fi*.

ABSTRACT

In recent years, there have been several increases in the consumer's energy bill, which was no longer cheap in Brazil, a country that charges one of the highest energy rates in the world. Clarifying a little about the consumption of electricity in the country, more than a third of energy consumption is from households, but there are practically no mechanisms found in an easy way to help the consumer to have greater transparency about the real-time expenses of their residence. This does not create a habit, or even more, an economic culture of analyzing energy waste, as there is no way to know what is being spent. With that in mind, this work aims to build a cheap electricity meter and also create a platform that can be easily accessed remotely via *internet* on any device with access to the internet network. The meter presented here can turn on or off any electrical appliance remotely and measure consumption remotely as well. For the construction of this meter, an STC-13 current sensor and a Zmpt101b voltage sensor were used, and the ESP-32 microcontroller was used, which has the function of reading, processing and sending the data via Wi-Fi, to the Node-Red platform, from this platform the information is stored in a cloud database of the IBM company. This data will be shown to the users on a remote platform built in NodeJS and ReactJs.

Keywords: meter. ESP-32. intranet. wi-fi.

LISTA DE ILUSTRAÇÕES

Figura 1 – Potência Reativa (kVAr).	20
Figura 2 – Triângulo Retângulo de Potência.	22
Figura 3 – Diagrama de Blocos ESP-32.	23
Figura 4 – Configuração interna do Optoacoplador.	26
Figura 5 – Retificador de onda completa composta por 4 diodos.	27
Figura 6 – Forma de onda filtrada por um filtro capacitivo.	27
Figura 7 – Regulador de Tensão com valor de saída fixa.	28
Figura 8 – Display LCD.	29
Figura 9 – Base de funcionamento de um Relé.	30
Figura 10 – Relé com contatos Normalmente Fechados.	30
Figura 11 – Módulo Relé.	31
Figura 12 – Saída Digital ou Binária de um Sensor.	33
Figura 13 – Saída Analógica de um Sensor.	33
Figura 14 – Sensor de corrente SCT-013.	34
Figura 15 – Sensor de Tensão ZMPT 101B.	35
Figura 16 – Diagrama elétrico do Sensor de Tensão ZMPT 101B com ESP-32	38
Figura 17 – Sensor de Tensão ZMPT 101B, Teste <i>offset</i> .	39
Figura 18 – Sensor de Tensão ZMPT 101B, Ligado a rede de 220V.	40
Figura 19 – Código para calibração do sensor de tensão.	41
Figura 20 – Comparação e calibração dos valores de tensão do sensor com um multímetro.	41
Figura 21 – Diagrama do dimensionamento do resistor de carga e tensão de <i>offset</i> .	42
Figura 22 – Comparação dos valores de corrente do sensor com alicate Amperímetro.	43
Figura 23 – Fluxograma de aquisição dos dados.	44
Figura 24 – Rede <i>wi-fi</i> ESP-32.	45
Figura 25 – Tela Inicial de Configuração ESP-32.	46
Figura 26 – <i>Scan</i> de redes e potência do sinal.	47
Figura 27 – Programação em Bloco <i>Node-Red</i> .	48
Figura 28 – Plataforma de monitoramento e controle, acessada por <i>smartphone</i> .	49
Figura 29 – Plataforma de monitoramento e controle, acessada pelo computador.	49
Figura 30 – Botão para selecionar a partir de quantos minutos os dados são exibidos.	50
Figura 31 – Botões de controle quando a carga estiver acionada.	50
Figura 32 – Botões de controle quando a carga estiver "Desligada".	50
Figura 33 – Configuração <i>Broker</i> .	51
Figura 34 – Projeto completo do sistema.	52
Figura 35 – Projeto Fonte de 5V.	52
Figura 36 – PCI do Circuito.	53
Figura 37 – Placa 3D do Circuito.	53

Figura 38 – Protótipo montado dentro de uma caixa, visão interna.	54
Figura 39 – Protótipo montado dentro de uma caixa, visão externa.	55
Figura 40 – Arquivo csv com dados importados do banco de dados.	56
Figura 41 – <i>Soft</i> para salvar dados em arquivo csv	57

LISTA DE TABELAS

Tabela 1 – Detalhes Técnicos do ESP-32.	23
Tabela 2 – Detalhes técnicos do Relé JQC-3FF-S-Z.	31
Tabela 3 – Valores dos componentes utilizados e do projeto completo.	51

LISTA DE ABREVIATURAS E SIGLAS

SG	<i>Smart Grid</i>
IED	<i>Intelligent Electronic Devices</i>
IoT	<i>Internet of Things</i>
AD	<i>Analógico/Digital</i>
DA	<i>Digital/Analógico</i>
SDK	<i>Software Development Kit</i>
TCP	<i>Transmission Control Protocol</i>
IP	<i>Internet Protocol</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
JSON	<i>JavaScript Object Notation</i>
CI	<i>Circuito Integrado</i>
TC	<i>Transformador de Corrente</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
M2M	<i>Machine to Machine</i>
PCI	<i>Placa de Circuito Impresso</i>

LISTA DE SÍMBOLOS

V	<i>Volts</i>
P	<i>Potência ativa</i>
Q	<i>Potência reativa</i>
t	<i>tempo</i>
T	<i>Período de tempo</i>
VAR	<i>Volt-Ampere reativo</i>
VA	<i>Volt-Ampere</i>
S	<i>Potência aparente</i>
FP	<i>Fator de potência</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Objetivos	17
1.1.1	Objetivo Geral	17
1.1.2	Objetivos Específicos	17
2	REFERENCIAIS TEÓRICOS	18
2.1	Indústria 4.0	18
2.2	Smart Grid	18
2.3	Internet das Coisas	18
2.4	Potência: Ativa, Reativa, Aparente e Fator de Potência	19
2.5	Microcontroladores	22
2.6	ESP-32	23
2.7	TCP/IP	24
2.8	MQTT	24
2.9	NodeJS	25
2.10	ReactJS	25
2.11	Node-RED	25
2.12	Optoacopladores	26
2.13	Fonte de Alimentação	26
2.13.1	Retificador de Onda Completa	27
2.13.2	Filtro Capacitivo	27
2.13.3	Reguladores de Tensão	28
2.14	Display LCD	28
2.15	Atuadores	29
2.16	Módulo Relé	29
2.17	Sensores	32
2.18	Sensor SCT-013-000	34
2.19	Sensor ZMPT 101B	34
3	METODOLOGIA	36
3.0.1	Descrição Detalhada do Sistema	36
4	TESTES E CALIBRAÇÃO DOS SENSORES	38
4.0.1	Teste do Sensor ZMPT 101B	38
4.0.2	Teste do Sensor Sensor SCT-013	41
5	DESENVOLVIMENTO	44
5.0.1	Soft ESP-32	44
5.0.2	Configurando Node-Red	47
5.0.3	Configurando <i>Broker</i>	50
5.0.4	Custos do Projeto	51

6	RESULTADOS	52
7	CONCLUSÃO	58
8	TRABALHOS FUTUROS	59
	REFERÊNCIAS	60
	APÊNDICE A – CÓDIGO PYTHON	65
	APÊNDICE B – CÓDIGO ESP-32	68
	ANEXO A – DATASHEET SCT013-100	73
	ANEXO B – DATASHEET ZMPT101B	74

1 INTRODUÇÃO

Com o avanço tecnológico dos sistemas elétricos ao decorrer do tempo, sempre se teve a preocupação de medir o valor das grandezas elétricas, dentre estas: tensão e corrente. Com isso, a área de pesquisa em instrumentação vem buscando novos mecanismos de medição e aprimoramento diferentes dos já existentes, para que assim, esses dispositivos possam ser acessíveis a todos, (VIEIRA, 2008).

Devido ao aumento considerável na conta de energia que é acrescida do valor de bandeiras tarifárias, o consumidor procura formas de economizar a energia utilizada em sua residência, (Ochoa et al., 2015). Mas, para que o consumidor possa economizar, ele primeiramente deve entender e acompanhar o seu consumo de energia, porém a forma disponibilizada pela concessionária para isso são os boletos de pagamento que chegam na casa do consumidor mensalmente, entretanto esses boletos apresentam apenas à média mensal, (Santos et al., 2006) e não o consumo instantâneo, e isso pode ser um problema para consumidores que desejam fazer um controle mais criterioso de quanto sua residência/comércio está gastando em determinado período do dia, ou, indo além, quanto cada equipamento está consumindo.

Com isso, percebemos a importância de desenvolver mecanismos de baixo custo que ajudem esses consumidores a economizar mais, pois há poucos incentivos por parte do governo em buscar soluções que englobam todas as pessoas para criação de tecnologias que auxiliem os indivíduos a reduzirem o consumo desnecessário de energia, (SENA, 2018).

Para solucionar tal problema, há os chamados *Smart Meters*, que trazem a instrumentação convencional e possuem o contexto tecnológico, conhecido como: medidores inteligentes. Já houve interesse nesse tema pelas próprias concessionárias e distribuidoras de energia elétrica, porém esse interesse é voltado para cobranças de tarifas dos seus clientes, e não um monitoramento do consumo que deem aos usuários da rede elétrica uma transparência do seu consumo, (HAYASHI, 2018).

Novas tecnologias vêm crescendo atualmente, essas tecnologias são desenvolvidas para projetos embarcados de baixo custo, e trazem a característica de tornar aparelhos e "coisas", que são usadas no dia-a-dia, em dispositivos inteligentes que possam se comunicar com outros equipamentos, através de redes sem fio. Essa tecnologia é conhecida como Internet das Coisas, (*Internet of Things (IoT)*), (SENA, 2018).

1.1 Objetivos

1.1.1 Objetivo Geral

- Desenvolver um medidor de energia elétrica com ESP-32 onde o usuário poderá acompanhar os valores medidos de forma remota, além de poder, acionar e desacionar a carga que o equipamento está fazendo as leituras, de uma forma remota.

1.1.2 Objetivos Específicos

- Implementação e testes em laboratório residencial, dos sensores utilizados.
- Monitoramento e leitura das grandezas da rede de baixa potência de 220V.
- Armazenamento dos valores medidos em um banco de dados em nuvem.
- Utilizar a linguagem de programação *JavaScript* para desenvolver a plataforma de monitoramento.
- Acionar e desacionar uma carga usando o protocolo MQTT.
- Usar a linguagem de programação *python* para importar os dados medidos do banco de dados para um arquivo CSV.

2 REFERENCIAIS TEÓRICOS

2.1 Indústria 4.0

A indústria 4.0 surgiu em 2012, através de um projeto alemão que tinha como meta o desenvolvimento tecnológico, conhecido como *High-Tech Strategy 2020*, tendo em vista o grande ganho econômico que a tecnologia pode trazer, esta tem por objetivo aprimorar todas as etapas de produção de um produto, tal como a formulação de acervos de trabalho digitais ao longo de todos os processos de criação do produto para que todas as etapas de fabricação sejam ajustáveis e se adaptem a novos processos e especializações para criação de produtos individuais, (NETO, 2018).

Em concomitância com a indústria 4.0 vem o que ficou conhecida como quarta geração da utilização da tecnologia, possibilitando a junção de tecnologia com a distribuição de energia elétrica e otimizou o fluxo de informação disponível ao usuário sobre a qualidade da rede, a esta foi dada o nome de *smart grid*, (NETO, 2018).

2.2 Smart Grid

O termo *Smart Grid* (SG), é usado para se referir a utilização da tecnologia das áreas de computação automação, aplicados a monitorar e interagir no controle da rede elétrica, isso dar mecanismos de controle para o melhoramento da eficiência da rede elétrica usada pelos usuários, dentre as principais vantagens da implantação deste método está: uma maior conectividade do usuário com a tecnologia, e o acompanhamento da qualidade de energia de sua instalação, (NUNES *et al.*, 2019).

Isso permitirá que todas as etapas de fornecimento de eletricidade sejam beneficiadas com o uso da tecnologia, onde *Intelligent Electronic Devices* (IEDs), podem ser instalados, permitindo que qualquer área, como geração, transmissão e distribuição de energia, sejam facilmente controladas, dado que há uma troca de informações em tempo real da rede com o controlador, com isso, há a possibilidade que novas formas mais eficientes de controle e automação do sistema elétrico venham a surgir com fornecimento dessas informações, (NETO, 2018).

2.3 Internet das Coisas

A *Internet das Coisas* (IoT -*Internet of Things*), é um dos conceitos mais utilizados no presente. Atualmente, estão sendo elaborados diversos projetos com sistemas embarcados, microeletrônica, telecomunicações e sensores em geral, tanto em universidades quanto em

empresas, esse sucesso se deve a diversidade de projetos que podem ser produzidos com essa tecnologia, (Oliveira *et al.*, 2016).

O funcionamento da Internet das Coisas é uma junção de dispositivos físicos com a computação a qual permite que diversos equipamentos usuais, sejam eles residenciais ou industriais, sejam total ou parcialmente controlados ou tenham visualizados dados/informações através da internet, dado que estes estejam munidos de algum microcontrolador, que consiga interagir com a internet, permitindo que haja troca de dados pela rede. Por possuir essa característica, a maioria dos países com significativo poder de capital ao redor do mundo estão apostando em seu uso.

O conceito de smart city também é tema em estudo atualmente, pois com a utilização de dispositivo e de sensores de captação de dados é viável a implantação de cidades tecnológicas, nas quais as pessoas têm acesso a diversas informações em dispositivos que possuem acesso à internet.

Aplicando essa possibilidade de informação em tempo real aos dados das concessionárias elétricas é possível a implantação de medidores inteligentes, nos quais os usuários podem verificar todos os dados de consumo a qualquer momento através de algum equipamento com acesso à internet, se a concessionária de energia utilizar dessa abordagem com a finalidade de apresentar ao consumidor mais assertividade sobre o dados do consumo elétrico de sua residência e/ou comércio, (FREIRE *et al.*, 2018).

2.4 Potência: Ativa, Reativa, Aparente e Fator de Potência

A chamada energia útil é aquela responsável por realizar trabalho, e o fator de potência é quão próximo essa energia útil está próxima da energia total consumida, lembrando que, o ideal seria que toda energia fornecida fosse consumida em forma de energia útil, (MARCHI, 2019). Porém, grande parte dos equipamentos que consomem energia, precisam de energia reativa para seu correto funcionamento, pois há a necessidade de geração de campos magnéticos, como os motores, transformadores, *etc.* Assim, para o correto funcionamento destas máquinas é necessário o consumo de Potência Ativa (P) e Reativa (Q), (WEG Automação, 2018).

$$P(t) = v(t) \cdot i(t) \quad (2.1)$$

A equação 2.1 é a fórmula da potência ativa instantânea em relação ao tempo, sabendo que, potência ativa instantânea é igual a potência ativa média, por um determinado período de tempo T, desse modo concluímos que:

$$P_m = \frac{1}{T} \int_0^T P(t) \quad (2.2)$$

$$P_m = \frac{1}{T} \int_0^T v(t) \cdot i(t) \quad (2.3)$$

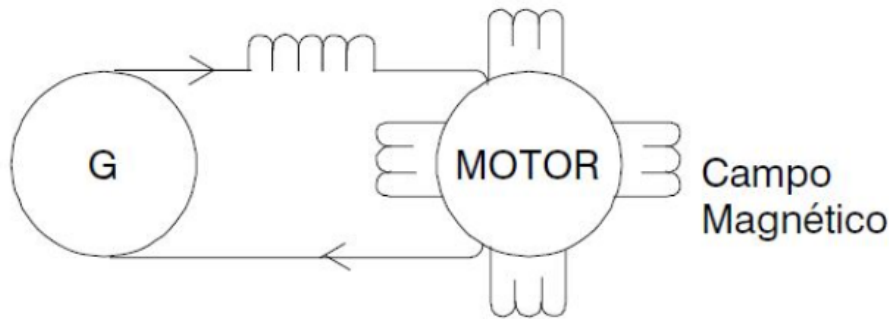
Em tempo discreto as fórmulas da potência média são dadas por:

$$P_m = \frac{1}{N} \sum_{i=1}^i P(i)$$

$$P_m = \frac{1}{N} \sum_{i=1}^i v(i) \cdot i(i) \quad (2.4)$$

A potência reativa é classificada em: potência reativa indutiva e potência reativa capacitiva, sendo a primeira necessária para geração dos campos magnéticos das bobinas de equipamentos como motores de indução, reatores, transformadores, *etc.* E a segunda, é produzida por capacitores, ou máquinas síncronas super excitadas. Por mais que a potência reativa não produza trabalho útil, ela é necessária para o funcionamento da maioria dos dispositivos. Esta é dada na literatura pela letra Q e sua unidade é Volt-Ampere reativo (VAr), (OLIVEIRA et. al. 2019).

Figura 1 – Potência Reativa (kVAr).



Fonte: WEG Automação, 2018.

As fórmulas que regem esta potência é dada por:

$$Q(t) = v(t) \cdot i(t) \cdot \sin(\theta) \quad (2.5)$$

Em tempo discreto as fórmulas da potência reativa são dadas por:

$$Q = \frac{1}{N} \sum_{k=1}^N Q(k)$$

$$Q = \frac{1}{N} \sum_{k=1}^N v(k) \cdot i(k) \cdot \sin(\theta) \quad (2.6)$$

A potência aparente é o produto da corrente e tensão eficaz, que considera a parte ativa e reativa das potência consumidas, essa potência é dada em Volt-Ampere (VA).

As equações que descrevem a Potência Aparente são dadas por:

$$S = v_{ef} \cdot i_{ef} \quad (2.7)$$

Onde:

$$v_{ef} = \frac{V_p}{\sqrt{2}} \quad (2.8)$$

$$S = \frac{V_p \cdot I_p}{2} \quad (2.9)$$

O valor eficaz, em tempo contínuo é dado por:

$$v_{ef} = \sqrt{\frac{1}{T} \int_0^T v(t)^2 dt} \quad (2.10)$$

$$i_{ef} = \sqrt{\frac{1}{T} \int_0^T i(t)^2 dt} \quad (2.11)$$

$$S = \frac{1}{T} \sqrt{\int_0^T v(t)^2 dt \cdot \int_0^T i(t)^2 dt} \quad (2.12)$$

Em tempo discreto temos:

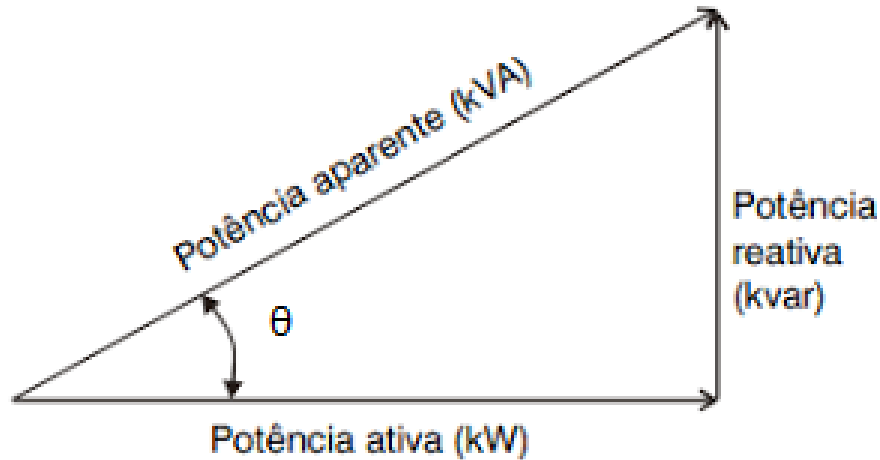
$$v_{ef} = \sqrt{\frac{1}{N} \sum_{k=1}^N v(k)^2} \quad (2.13)$$

$$i_{ef} = \sqrt{\frac{1}{N} \sum_{k=1}^N i(k)^2} \quad (2.14)$$

$$S = \frac{1}{N} \sqrt{\sum_{k=1}^N v(k)^2 \cdot \sum_{k=1}^N i(k)^2} \quad (2.15)$$

Segundo Weg (2018) o fator de potência é dado pela divisão entre potência ativa e a potência aparente, esta divisão determina o quão aproveitado está sendo a energia consumida, onde um alto fator de potência indica uma melhor eficiência, dado que a maior parte da energia gasta estará sendo transformada em potência útil, ou ativa, o contrário disso representa ineficiência no consumo de energia. Para representar as grandezas vistas anteriormente, é usualmente, utilizado um triângulo retângulo, ver Figura 2.

Figura 2 – Triângulo Retângulo de Potência.



Fonte: WEG Automação, 2018.

O cálculo do fator de potência é dado por:

$$FP = \frac{P}{S} = \cos(\theta) \quad (2.16)$$

Sendo o $\cos(\theta)$ a definição do Fator de Potência, quanto mais o valor de $\cos(\theta)$ for próximo ao valor unitário, menos potência reativa o circuito está gastando, (SOUZA, 2016).

$$FP = \frac{P}{\sqrt{P^2 + Q^2}}$$

2.5 Microcontroladores

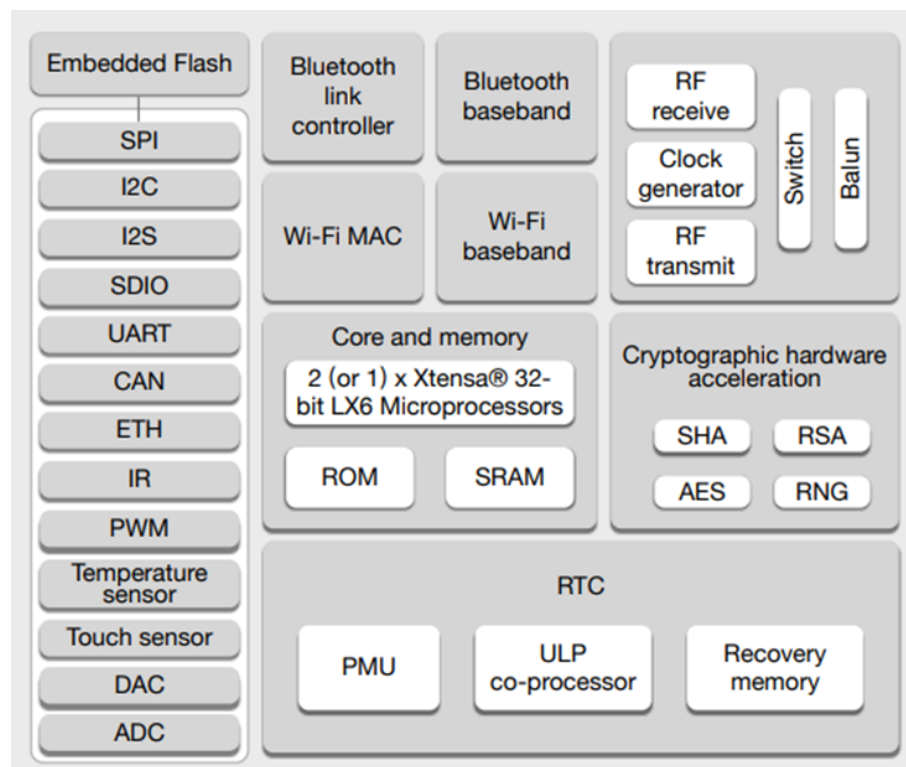
Os microcontroladores têm como principais características serem dispositivos programáveis com pequeno tamanho, que consomem pouca energia e que possuem um baixo custo de aquisição. Sua estrutura se divide em: um microprocessador; as memórias RAM e ROM; pinagens de entrada e saída; um *clock*; conversores, tanto, Digital Analógico (DA), quanto, Analógico Digital (AD). Os microcontroladores, são bastantes utilizados para controle a distância, em dispositivos embarcados, (SANTOS JUNIOR, 2019).

Com o avanço tecnológico dos dispositivos, houve a capacidade de se conectar à internet, onde pode-se, desde então, ter a comunicação e controle entre dispositivos, e isso permitiu/permite liberdade para monitoramento de equipamentos conectados que enviem seus dados para determinado dispositivo, de forma totalmente remota. Com isso tem-se a possibilidade de se fazer ajustes ou tomar decisões de forma mais rápida baseada nesta aquisição de dados, (SANTOS JUNIOR, 2019).

2.6 ESP-32

O ESP-32 foi apresentado ao público em 2016, este microcontrolador foi criado pela empresa *Espressif Systems*, que é uma empresa voltada para o desenvolvimento de tecnologia. Este dispositivo foi projetado para ter uma alta velocidade de processamento e seu maior diferencial é a conectividade com a rede wi-fi, (KOLBAN, 2018). A Figura 3 mostra o diagrama de blocos do ESP-32.

Figura 3 – Diagrama de Blocos ESP-32.



Fonte: ESPRESSIF, 2019.

Na Tabela 1 é apresentada algumas características deste microcontrolador.

Tabela 1 – Detalhes Técnicos do ESP-32.

Especificação	Detalhe
Tensão	3.3 V
Memória RAM	520 Kb
Processador	single/dual-core 32 bit
GPIOs	34
Pinos Analógico/Digital	7

Fonte: (KOLBAN, 2018).

Os conversores AD do ESP-32 possuem uma precisão de 12 *bits*, ou seja, pode-se

representar valores no intervalo de 0 a 4095, este dispositivo, foi projetado para possuir alta velocidade de processamento, chega a atingir até 240Mh, tem uma memória RAM de 520K *bytes*, e sua memória ROM é de 448K *bytes*, (CASTRO, 2019).

O ESP-32, pode ser programado nas linguagens C/C++, a qual é a mesma linguagem utilizada no Arduino. Para programá-lo, pode-se usar a plataforma *Software Development Kit (SDK)*, onde esta é mantida pela própria fabricante do microcontrolador, ou pode-se usar a IDE do Arduino, (SANTOS JUNIOR, 2019).

2.7 TCP/IP

Um dos protocolos mais utilizados para transmissão de dados pela internet é o protocolo TCP/IP, onde TCP é (*Transmission Control Protocol*) e IP é (*Internet Protocol*), esse protocolo foi criado para ter uma boa resposta, independente do tamanho da rede, além de possuir uma arquitetura *open-source*, com isso qualquer pessoa pode utilizá-lo, (TORRES, 2014).

Dispositivos conectados à rede, são reconhecidos por outros dispositivos através do endereçamento IP, com isso, cada dado dos equipamentos que trafegam pela rede podem ser entregues de uma forma confiável ao receptor, pois tanto o que está enviando, quanto o que está recebendo a informação precisam serem/estar reconhecidos por um endereço IP, para que a mensagem seja enviada e recebida, sendo este o funcionamento mais simples de uma rede TCP/IP, (TORRES, 2014).

O endereço IP é responsável pelo endereçamento em si, ou seja, ele diz onde está o servidor que receberá os dados e diz onde está o emissor que está enviando a informação. O TCP, verifica e organiza estes dados, garantindo a comunicação entre emissor e receptor, fornecendo segurança e informa se a informação está sendo enviada e recebida corretamente, o TCP também é o responsável pelas portas que serão enviados e recebidos os dados, e ainda verificar se ambas estão disponíveis, (TORRES, 2014).

2.8 MQTT

O Protocolo *Message Queuing Telemetry Transport (MQTT)* foi desenvolvido pela IBM em 1999 e criado com finalidade de comunicação *Machine-to-machine (M2M)*. O modelo de comunicação deste protocolo é o mesmo TCP, onde há o emissor e receptor, porém entre os dois existe um intermediário, que é responsável por enviar ao cliente os dados enviados pelo servidor, este intermediário é dado o nome de *Broker*. Com isso, basta que o endereço do *Broker* seja conhecido pelo cliente e servidor para que a comunicação aconteça, (TORRES; ROCHA; SOUZA, 2016).

2.9 NodeJS

O *NodeJS* é um motor do *Google Chrome* o qual foi desenvolvido para a interpretação de *JavaScript* sem o uso do navegador. Conforme a documentação (MDN, 2019) o *NodeJS* consegue executar rotinas de forma assíncrona enquanto o programa está em execução, esta plataforma foi desenvolvida no intuito de administrar de forma mais eficiente diversas conexões, (JUNIOR, 2019).

Uma das principais características do *NodeJS* é sua velocidade, e como todas as funções são executadas de forma assíncronas, não há o travamento da execução por conta da espera de algum evento terminar sua execução, isso ocorre pois, o Node usa o método de programação orientada a eventos, e assim em vez de uma programação sequencial, como é o caso da maioria das linguagens, há uma execução tipo *thread*, (FERNANDES, 2021).

2.10 ReactJS

O *react* foi criado pelos engenheiros do Facebook, é uma biblioteca baseada em *JavaScript*, que serve para o desenvolvimento de páginas web, e mostram na tela a parte da aplicação visual para o usuário. As aplicações desenvolvidas com esta ferramenta são baseadas em HTML, CSS, e JavaScript, onde as três são usadas em conjunto dentro do React, (GACKENHEIMER, 2015).

Uma das vantagens de se utilizar o *react*, conforme MOREIRA (2021), é a forma de organizar o código de programação, o qual é baseado em componentização, sendo que cada função do código possui em seus componentes funcionalidades únicas na execução do código.

Uma segunda vantagem, é que o *front-end* com o *react* transfere toda responsabilidade que não seja a parte visual, ou interface, para o *back-end*, com isso pode-se ter várias requisições sendo respondidas pelo mesmo *back-end*. A forma mais usual de resposta do *react* e *NodeJS* é o *JavaScript Object Notation* (JSON).

2.11 Node-RED

O *Node-RED* é uma plataforma de programação em blocos, desenvolvida para a programação *low code*. Nesta, por meio de uma interface, os blocos são ligados uns aos outros seguindo a lógica da aplicação a ser desenvolvida. O conjunto de blocos são divididos em grupo onde cada um possui finalidades em comum, por exemplo: protocolos de rede, comunicação sem fio, banco de dados, *etc.* (CANTÚ; MONTEZ, 2020).

A depender da aplicação, alguns blocos não vêm por padrão na plataforma, podendo serem baixados conforme a necessidade de cada projeto. O *Node-RED* possibilita a comunicação

com diversas outras plataformas, como alguns serviços do *google* e *amazon* (CANTÚ; MONTEZ, 2020).

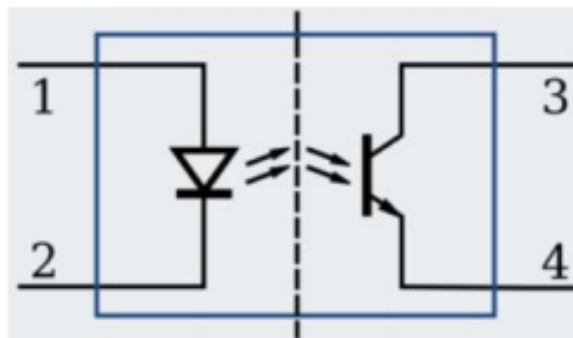
A plataforma pode ser acessada de forma local, uma vez que é *open source*, ou seja pode-se ser baixada e acessada diretamente pela máquina do usuários (CANTÚ; MONTEZ, 2020). Ou pode ser utilizada de forma *on-line* pela *IBM*, que é a empresa que produziu e mantém o projeto, (TAVARES *et al.*, 2019).

Essa empresa permite criar uma conta gratuita com alguns recursos para acesso e manipulação dos dados em nuvem, além de um banco de dados também *on-line* de armazenamento limitado, embora essa conta seja somente para testes é suficiente para um projeto de pequeno porte.

2.12 Optoacopladores

Os Optoacopladores são constituídos por um Diodo Emissor de Luz (LED), e um sensor fotodetector, sendo assim, um equipamento semi-condutor. A finalidade deste dispositivo é isolar certas partes do circuito de outras, na possibilidade de um deles passar uma corrente que possa vim a danificar um circuito mais sensível, como é o caso dos microcontroladores. Logo, como os componentes dentro do Optoacopladores não são conectados por nenhum condutor e sim por uma ligação óptica, ver Figura 4, caso haja uma variação brusca da corrente em um dos dispositivos um não afetará o outro, (SANTOS, 2019).

Figura 4 – Configuração interna do Optoacoplador.



Fonte: Vishay, 2018.

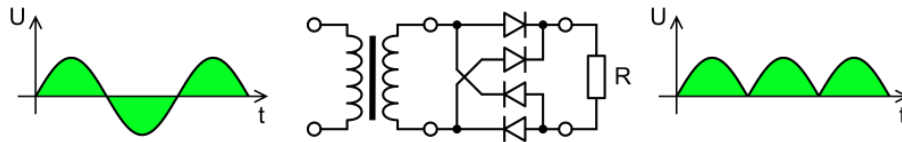
2.13 Fonte de Alimentação

A maioria dos dispositivos eletrônicos são alimentados com 5V, para essa alimentação é preciso ser dimensionado uma fonte de alimentação que atenda ao correto funcionamento do dispositivo do sistema a ser montado, para se construir essa Fonte alguns conhecimentos precisam ser abordados.

2.13.1 Retificador de Onda Completa

A função do Retificador de Onda Completa é transformar o sinal negativo da onda da rede elétrica, que vem em Corrente Alternada (CA), e transformá-la totalmente em sinais positivos, ver Figura 5. Esse procedimento é realizado através de uma ponte de diodos, (BOYLESTAD, 2004).

Figura 5 – Retificador de onda completa composta por 4 diodos.

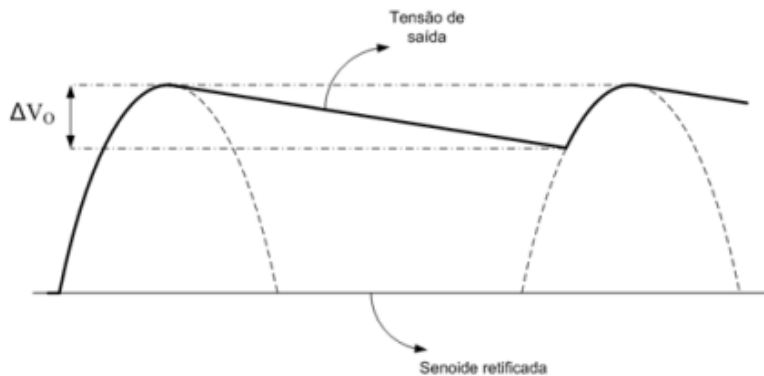


Fonte: ESPRESSIF, 2019.

2.13.2 Filtro Capacitivo

O capacitor é utilizado como filtro, com a finalidade de armazenar energia e extinguir o sinal pulsante que restou da ponte retificadora, isso ocorre, pois o capacitor não permite brusca variações de tensão, e desse modo não permite que a tensão venha a cair rapidamente, (BOYLESTAD, 2004), ver Figura 6.

Figura 6 – Forma de onda filtrada por um filtro capacitivo.



Fonte: ESPRESSIF, 2019.

A capacitância é dada pela fórmula:

$$C = \frac{Q}{V_c} \quad (2.17)$$

Onde:

$$Q = \frac{I}{f}$$

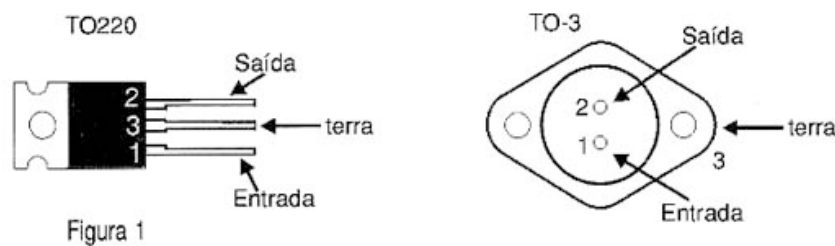
Logo a tensão de Ripple pico-a-pico é:

$$V_{rpp} = \frac{I}{fC} = \frac{V_{occ}}{fCR_L} \quad (2.18)$$

2.13.3 Reguladores de Tensão

Segundo Kuroishi (2018) os Reguladores de Tensão é um Circuito integrado (CI) cuja finalidade é manter a tensão em um determinado valor exigido para funcionamento seguro do equipamento que está se trabalhando, onde a tensão em sua entrada deve ser maior que sua tensão nominal, uma vez que não há fornecimento de energia do próprio CI para a fonte de alimentação. Os Reguladores de Tensão mais utilizados são os da série LM7800 que operam entre 5 a 24V com corrente de até 1A, na forma de CI possuindo 3 pinos, ver Figura 7.

Figura 7 – Regulador de Tensão com valor de saída fixa.

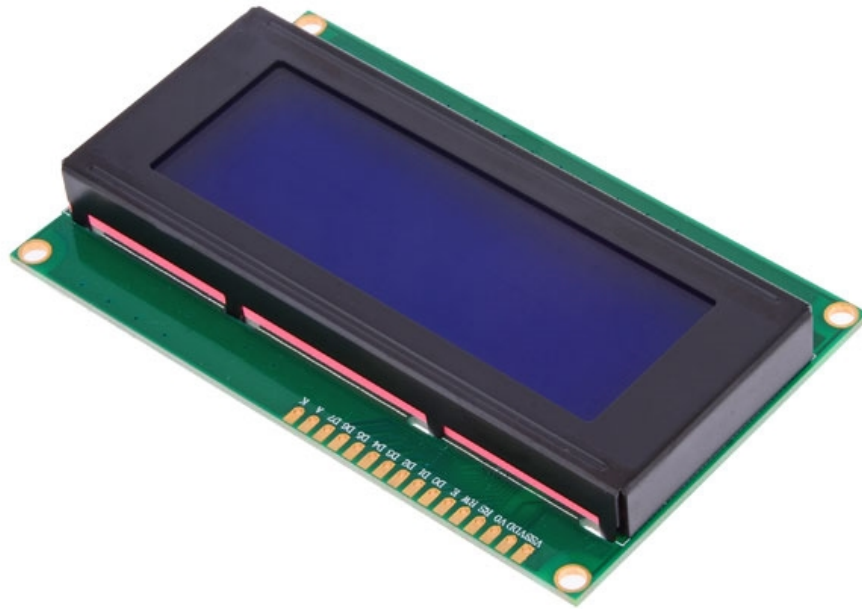


Fonte: Braga, 2010.

2.14 Display LCD

Através do display LCD pode-se apresentar um massivo número de dados por um pequeno visor, uma das principais características deste dispositivo é precisa de apenas cerca de 1mA de corrente para seu funcionamento normal com luz de fundo apagada, isso faz com que tenha um consumo baixo de energia, (FIGUEIREDO, 2017). Ver Figura 8.

Figura 8 – Display LCD.



Fonte: filipeflop, (2021).

2.15 Atuadores

Atuadores têm a função de realizar movimento físico, esse movimento é resultado da conversão de algum tipo de energia, sendo a mais comum energia elétrica, porém podem também serem do tipo hidráulica ou pneumática, esse movimento pode ser utilizado para controlar algum dispositivo se necessário, (COSTA, 2020).

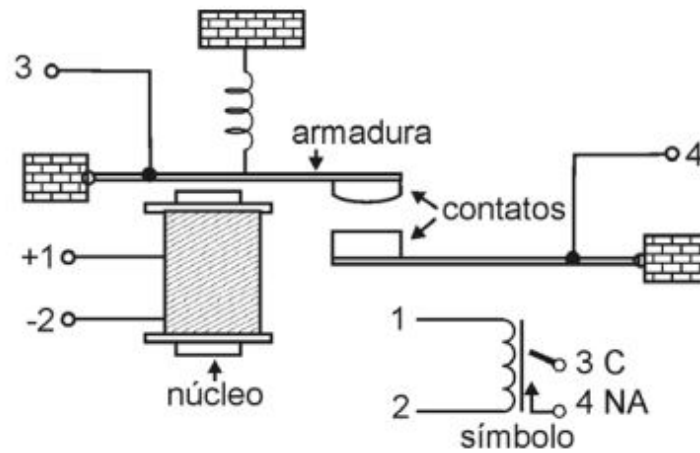
2.16 Módulo Relé

O relé tem seu funcionamento baseado na passagem de corrente elétrica em suas bobinas, essa passagem da corrente cria um campo magnético que por sua vez acionar os contatos internos no dispositivo, seu funcionamento é semelhante a uma chave liga/desliga, onde os contatos abrem-se e fecham-se a depender da circulação de corrente de sua bobina, (ARAÚJO, 2017).

Podemos ter dois estados iniciais em um relé, no primeiro visto na Figura 9 temos os contatos do tipo Normalmente Aberto (NA), onde o acionamento do relé é realizado quando uma

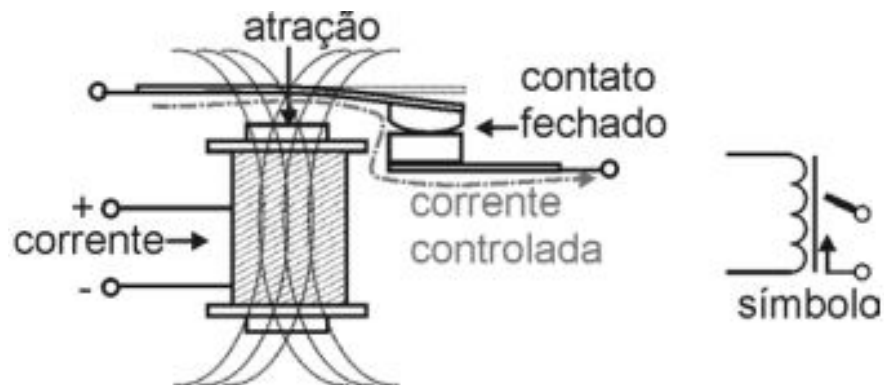
corrente percorre sua bobina a energizando, e na Figura 10 o tipo Normalmente Fechado (NF), onde o relé é desacionado quando sua bobina é energizada, (BRAGA, 2017).

Figura 9 – Base de funcionamento de um Relé.



Fonte: BRAGA, 2017.

Figura 10 – Relé com contatos Normalmente Fechados.



Fonte: BRAGA, 2017.

Pode-se encontrar facilmente no mercado, módulos relés específicos para uso com microcontroladores, como o mostrado na Figura 11, existem vários modelos, alguns com apenas um relé, e outros com vários relés, na Tabela 2 é especificado os parâmetros do modelo *JQC-3FF-S-Z* de 2 canais.

Figura 11 – Módulo Relé.



Fonte: (ELETROGATE, 2021).

Tabela 2 – Detalhes técnicos do Relé JQC-3FF-S-Z.

Parâmetros	Valores
Modelo	JQC-3FF-S-Z
Tempo de acionamento de contato	10 ms
Temperatura de funcionamento	-45 a 85 °C
Peso unitário	10 g
Proteção do relé	IP67
Umidade de operação	- 40 a 80 %
Tensão de operação	5 V
Max. tensão de comutação	277Vac/30Vdc
Max. corrente de comutação	15 A

Fonte: datasheet Adaptado.

2.17 Sensores

Segundo Foncesa (2006) Sensores são empregados na função “sentir” o meio ao qual está exposto e identificar certas grandezas físicas. Tais resultados, obtidos por esses sensores podem ser dos mais variados, por exemplo: luminosidade e calor. A obtenção desses valores são exclusivamente dependentes de qual sensor está utilizando. Após coletadas, estas grandezas, podem ser interpretadas através de um indicador ou controlador. Isto é, o sensor monitora as variações de uma grandeza física e depois pode-se mostrar essa variação, de maneira simples, por intermédio de algum equipamento.

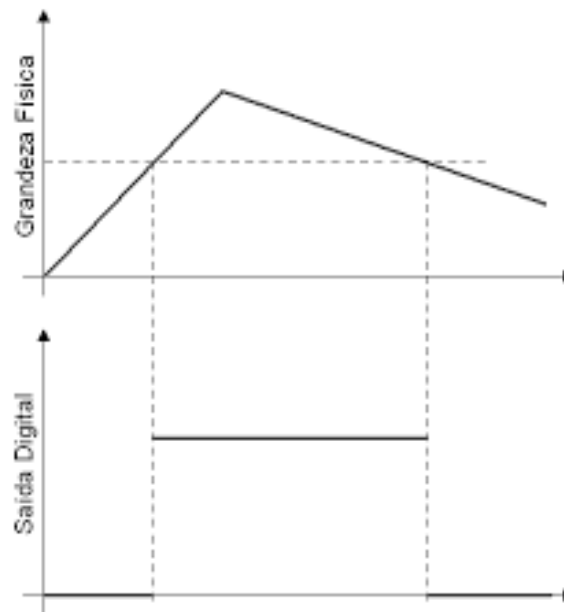
Patsko (2006) afirma que um sensor tem a função de transdutor onde este pode transformar um tipo de energia em outro, no caso, o sensor transforma uma grandeza física em variações de pulsos elétricos, com esses dados, pode-se transformá-los em uma outra categoria de sinal, podendo ser empregados para monitoramento de alguma variável ou característica do ambiente.

Roggia Fuentes (2016) dizem que sensores podem ser classificados em dois tipos: os digitais que são utilizados no monitoramento de ocorrências, ou não, de um determinado evento, e apresentam em suas saídas apenas dois estados lógicos, são eles: ligado (ON) ou desligado (OFF). E os analógicos, que são utilizados para monitoramento de grandezas físicas em uma faixa contínua, porém esta faixa de atuação tem que está estabelecida entre valores mínimos e máximos nos sinais de saídas.

Segundo Wendling (2010) várias características devem ser levadas em consideração na hora de escolher qual sensor será mais viável utilizar em um determinado projeto. Como os tipos de saída. Apresenta-se a seguir os tipos de saída: digital e analógico.

a) Tipo de saída digital nos dispositivos de controle as saídas digitais recebem apenas dois valores lógicos (0 ou 1). Esses tipos de saídas são conhecidos também por saídas binárias. Estas, assumem dois estados: ligadas, geralmente nível 1; ou desligadas, normalmente nível 0. Estes estados se alteram dependendo de grandezas físicas bem definidas, e estes valores são pré-estabelecidos de forma que o nível da saída se altere, se este valor for alcançado. A Figura 12 demonstra o comportamento da saída de um sensor digital em função da entrada de dados ao longo do tempo.

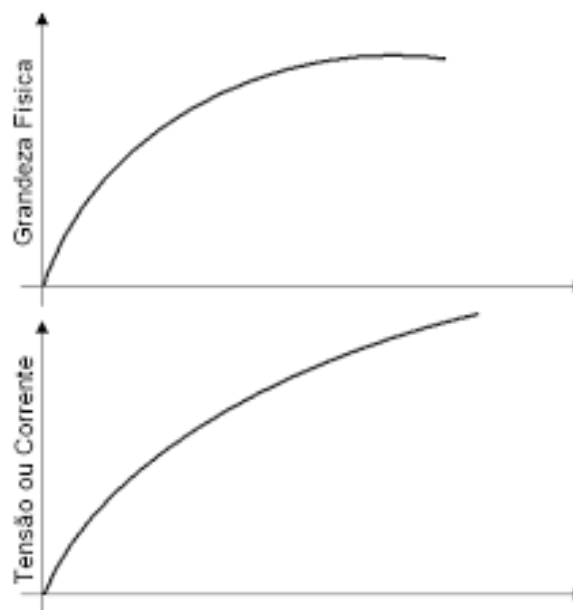
Figura 12 – Saída Digital ou Binária de um Sensor.



Fonte: Wendling, 2010.

b) Tipo de saída analógica outro tipo de saída é a analógica, sensores que utilizam essa saída interpretam diversos valores ao longo do tempo, somente se estes valores estiverem de acordo com sua faixa de funcionamento. A figura 13 demonstra o funcionamento de um sensor analógico, o gráfico da figura apresenta a tensão ou corrente de saída, que variam aproximadamente igual a uma grandeza física, em uma faixa de tempo determinada.

Figura 13 – Saída Analógica de um Sensor.

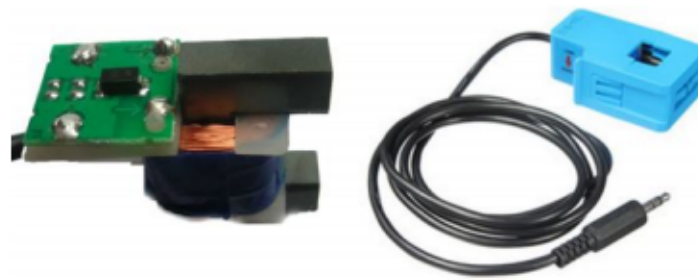


Fonte: Wendling, 2010.

2.18 Sensor SCT-013-000

Um dos dispositivos utilizados para medição da corrente alternada é o sensor SCT-013-000V. Este dispositivo consegue medir a corrente elétrica, na faixa de 0 a 100A, e uma saída de 0 a 1V. A Abreviação SCT (*Split-core Current Transformer*) significa transformador de corrente de núcleo dividido. O SCT é um transformador de corrente, que é constituído por uma bobina e um TVS (supressor transiente de tensão e um núcleo de ferrite dividido), com esse dispositivo conseguimos obter as medições de corrente, (Kuroishi, 2018). Ver Figura 14.

Figura 14 – Sensor de corrente SCT-013.

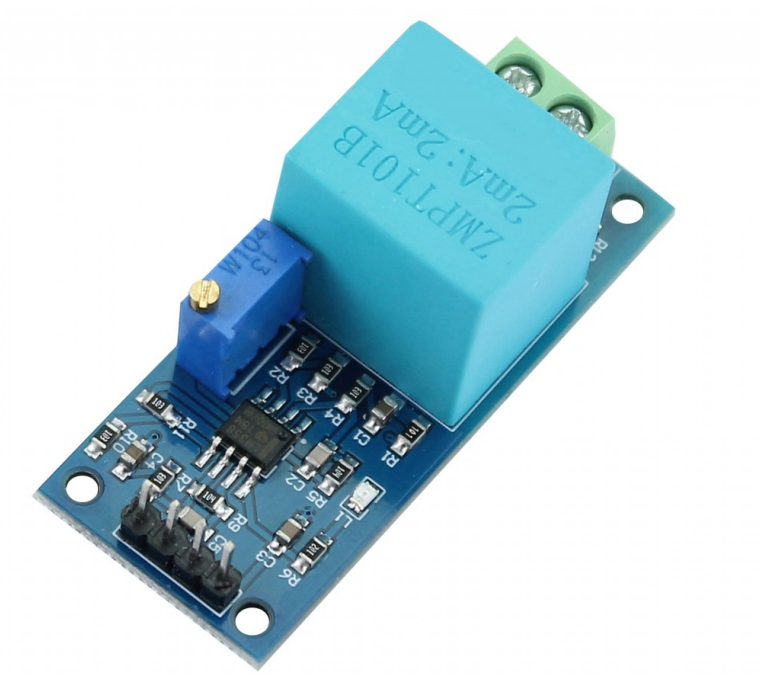


Fonte: YHDC, 2013a.

2.19 Sensor ZMPT 101B

O sensor ZMPT 101B, é um módulo utilizado para medir a tensão de um circuito elétrico, onde esta medição pode ser contínua ou corrente alternada. Embora este módulo não seja tão eficiente quanto um multímetro profissional, ele nos fornece uma medição de tensão com uma precisão satisfatória. O ZMPT 101B possui em sua estrutura um *Trimpot* que permite a calibração do sensor, assim pode-se obter uma maior precisão, (COSTA, 2019). Ver Figura 15.

Figura 15 – Sensor de Tensão ZMPT 101B.



Fonte: USINAINFO, 2021.

3 METODOLOGIA

O trabalho será guiado pela metodologia a seguir:

- Estudo teórico do ESP-32.
- Instalar a interface para programar o ESP-32 e instalação de todas as bibliotecas necessárias.
- Estudo teórico dos sensores de corrente e tensão.
- Calibrar dos sensores.
- Construir a plataforma para mostrar os gráficos dos valores obtidos pelos sensores.
- Integrar os dados dos sensores à plataforma, mostrá-los de forma remota e simultânea ao usuário.

3.0.1 Descrição Detalhada do Sistema

O projeto consiste em construir um protótipo de baixo custo, para medição de grandezas elétricas monofásicas, este poderá ser ligado em qualquer equipamento elétrico e nos fornecerá dados elétricos do equipamento. Os dados serão enviados para um banco de dados *on-line*, e poderão ser monitorados de forma simultânea ou a posteriori, e será também construído uma interface para acompanhamento via gráfico de forma remota e simultânea destes dados. Além disso, o usuário poderá acionar e desligar a carga que o equipamento está coletando os dados, também de forma remota pela mesma plataforma de aquisição de dados, pois foi adicionado um módulo relé ao sistema que será controlado via rede *wi-fi*.

O seguinte material foi utilizado para construção do sistema físico do trabalho apresentado.

- 1 ESP-32;
- 1 Sensor SCT-013-000;
- 1 Sensor ZMPT 101B;
- 1 Módulo relé de dois canais;
- 2 Resistores de 10K Ω ;
- 1 Resistor de 330 Ω ;
- 1 Capacitor de 100 μ F;

- 1 Protoboard.

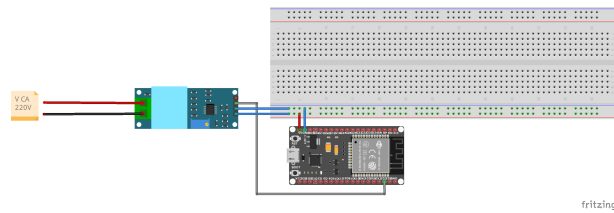
Os dados coletados pelos sensores e enviados ao banco de dados foram feitos usando o protocolo TCP/IP. Este protocolo também foi utilizado para requisição do banco de dados para plataforma de monitoramento dos dados. Já para acionamento do módulo relé foi usado o protocolo mqtt.

4 TESTES E CALIBRAÇÃO DOS SENSORES

4.0.1 Teste do Sensor ZMPT 101B

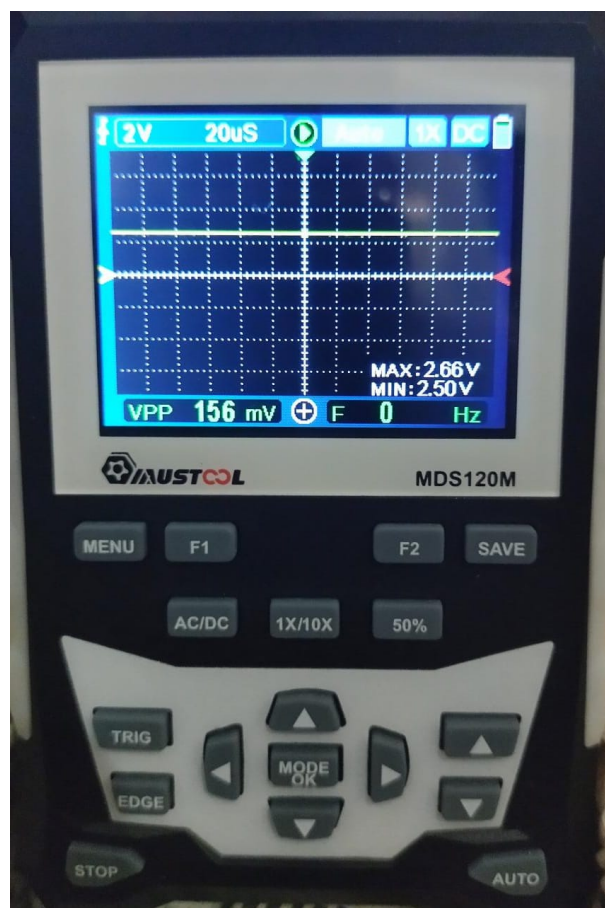
Foi montado o circuito da Figura 16, para realização da calibração deste sensor.

Figura 16 – Diagrama elétrico do Sensor de Tensão ZMPT 101B com ESP-32



Fonte: Arquivo Próprio, 2021.

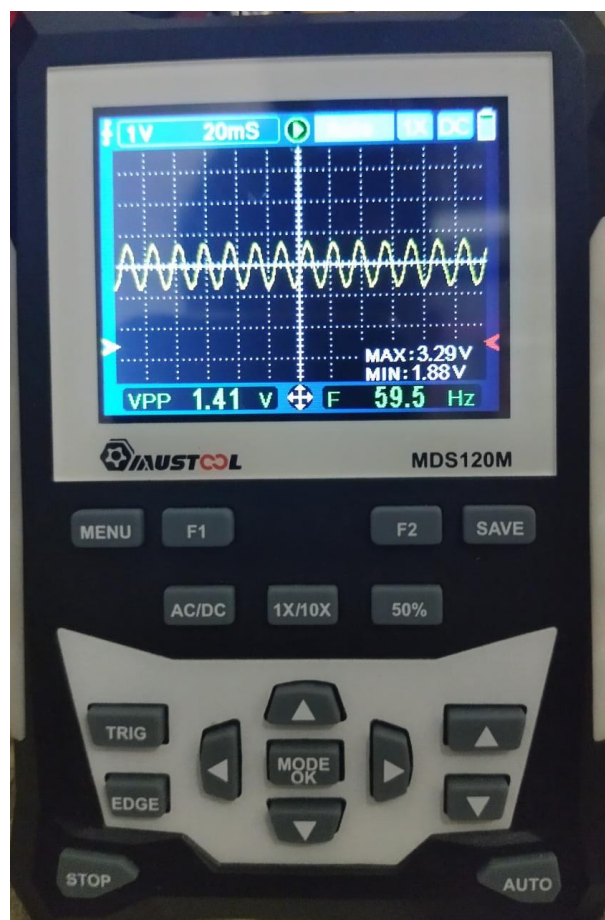
O sensor de tensão ZMPT 101B vem por padrão com um *offset* indicado pelo *datasheet* de 2.5V. Este sensor foi testado usando um osciloscópio portátil. O valor de *offset* encontrado no teste foi de 2.66V. O teste é feito sem ligar o dispositivo à rede elétrica, isto é, o valor de tensão da rede é igual a 0V, como é possível perceber na Figura 17.

Figura 17 – Sensor de Tensão ZMPT 101B, Teste *offset*.

Fonte: Arquivo Próprio, 2021.

Ao conectar o dispositivo a rede elétrica de 220V temos a máxima tensão na saída do sensor de 3.29V, ver Figura 18.

Figura 18 – Sensor de Tensão ZMPT 101B, Ligado a rede de 220V.



Fonte: Arquivo Próprio, 2021.

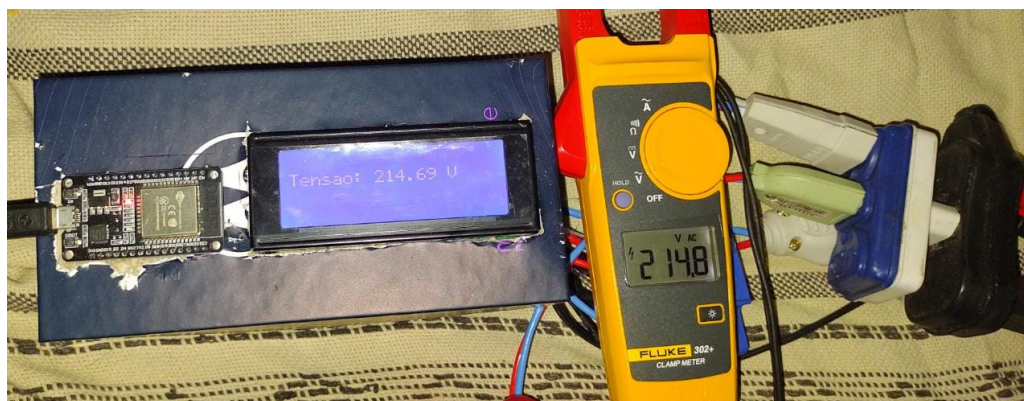
Conforme é declarado no *datasheet* do ESP-32, a faixa de operação do conversor AD lidos como valores de tensão em seus pinos é de 0 a 3.3V. Com essa informação foram feitos alguns testes, considerando a resolução de 12 *bits* deste dispositivo, e foi encontrado o valor de conversão, com a função "map" em código de programação, para nos fornecer a tensão real da rede elétrica lida pelo sensor. O código utilizado para testar a tensão da rede elétrica é mostrado na Figura 19. Os valores de tensão lidos através do sensor são mostrados em um *display LCD*, estes foram comparados e ajustados com os valores lidos por um multímetro da marca *FLUKE*, modelo 302+, conforme Figura 20.

Figura 19 – Código para calibração do sensor de tensão.

```
void loop() {  
    menorValor = 4095;  
  
    for(int i=0; i<3000 ; i++){  
        valor_inst = analogRead(pin);  
        //Serial.print("valor inst: ");  
        //Serial.println(valor_inst);  
  
        if (menorValor > valor_inst){  
            menorValor = valor_inst;  
        }  
  
        delayMicroseconds(1);  
    }  
  
    tensaoPico = map(menorValor, 3450, 2030, 0, 315);  
  
    tensaoRms = tensaoPico/1.4;  
  
    Serial.print("tensaoRms: ");  
    Serial.println(tensaoRms);  
|  
  
}
```

Fonte: Arquivo Próprio, 2021.

Figura 20 – Comparação e calibração dos valores de tensão do sensor com um multímetro.



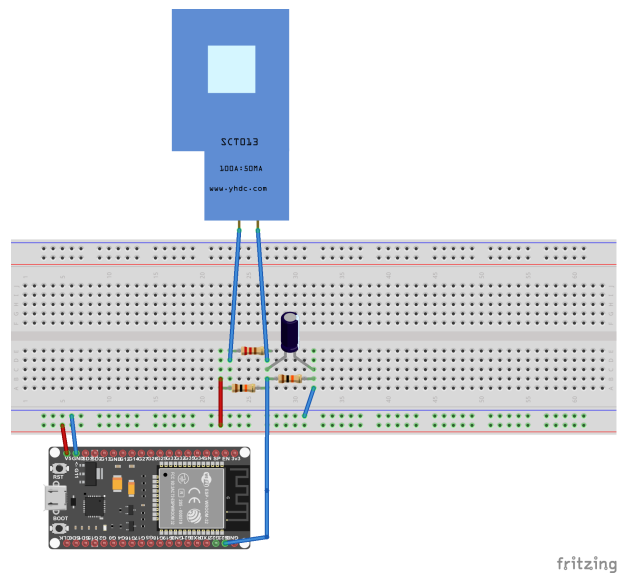
Fonte: Arquivo Próprio, 2021.

4.0.2 Teste do Sensor Sensor SCT-013

Segundo o *datasheet* do sensor SCT-013, o sensor lê no primário do Transformador de Corrente (TC), uma corrente de até 100A, e tem em sua saída, no secundário do TC, uma corrente máxima de 50mA. O AD do ESP-32 consegue ler apenas valores de tensão em seus pinos, com isso é preciso dimensionar um resistor de carga que gere uma variação de tensão em seus terminais que possa ser lido pelo microcontrolador.

O sensor não vem com um *offset* já definido, logo será necessário um circuito que desloque a onda senoidal, uma vez que o microcontrolador não consegue ler o semiciclo negativo da onda. Para definir o *offset* é feito um divisor de tensão com resistores, adicionando-se um capacitor para estabilizar a tensão, após um dos terminais do divisor de tensão é ligado ao 5V, em seguida com o outro terminal no ponto terra (referência), gera-se um *offset* de 2,5V como é possível observar na Figura 21, que mostra o circuito de *offset*, e resistor de carga explicado no parágrafo anterior.

Figura 21 – Diagrama do dimensionamento do resistor de carga e tensão de *offset*.



Fonte: Arquivo Próprio, 2021.

O cálculo do resistor de carga do circuito da Figura 21, foi feito considerando uma corrente de no máximo de 10A no primário do TC. Vale ressaltar que para este trabalho o sensor SCT-013 será mais utilizado em aparelhos domésticos, não necessitando de uma leitura de corrente tão alta quanto os 100A que é o maior valor que o sensor consegue ler. Fazendo esta adaptação melhoramos a sensibilidade do circuito à variação de corrente. A leitura máxima no secundário do TC, considerando o máximo de 10A no primário, passa a ser 5mA. Com isso, podemos dimensionar o resistor de carga pela lei de *Ohm*, dado na fórmula 4.1:

$$V_p = R \cdot I_p \quad (4.1)$$

Onde I_p e V_p são a corrente e tensão de pico, a corrente lida pelo sensor é corrente eficaz, logo é necessário multiplicar seu valor por $\sqrt{2}$ e assim temos:

$$R = \frac{2.5V}{5mA \cdot \sqrt{2}}$$

$$R = 357\Omega$$

Com isso, foi usado um resistor de valor comercial de $330\ \Omega$.

Feito o dimensionamento do resistor de carga e criado o circuito de *offset*, foi montando o modelo da Figura 21, e para assim poder-se realizar as medições e calibrações do sistema. As calibrações e comparações dos valores foram realizados, também pelo alicate amperímetro da marca *FLUKE*, modelo 302+, e os dados comparados com os valores lidos pelo sensor de corrente foram mostrados através do *display LCD*, conforme pode ser visto na Figura 22.

Figura 22 – Comparação dos valores de corrente do sensor com alicate Amperímetro.



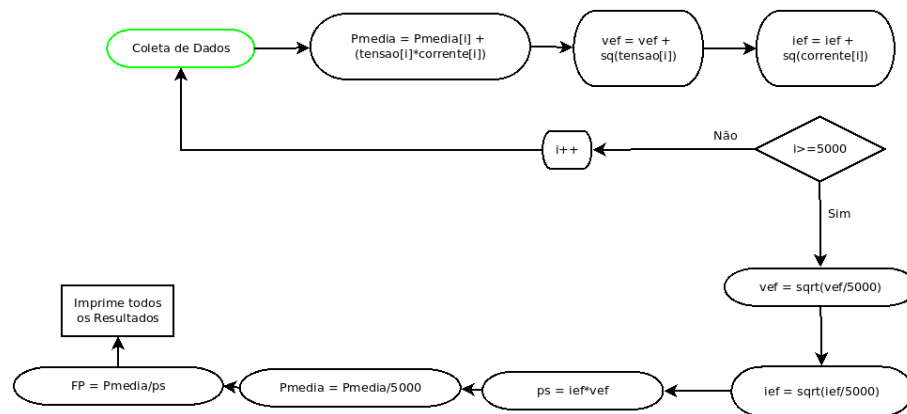
Fonte: Arquivo Próprio, 2021.

5 DESENVOLVIMENTO

5.0.1 Soft ESP-32

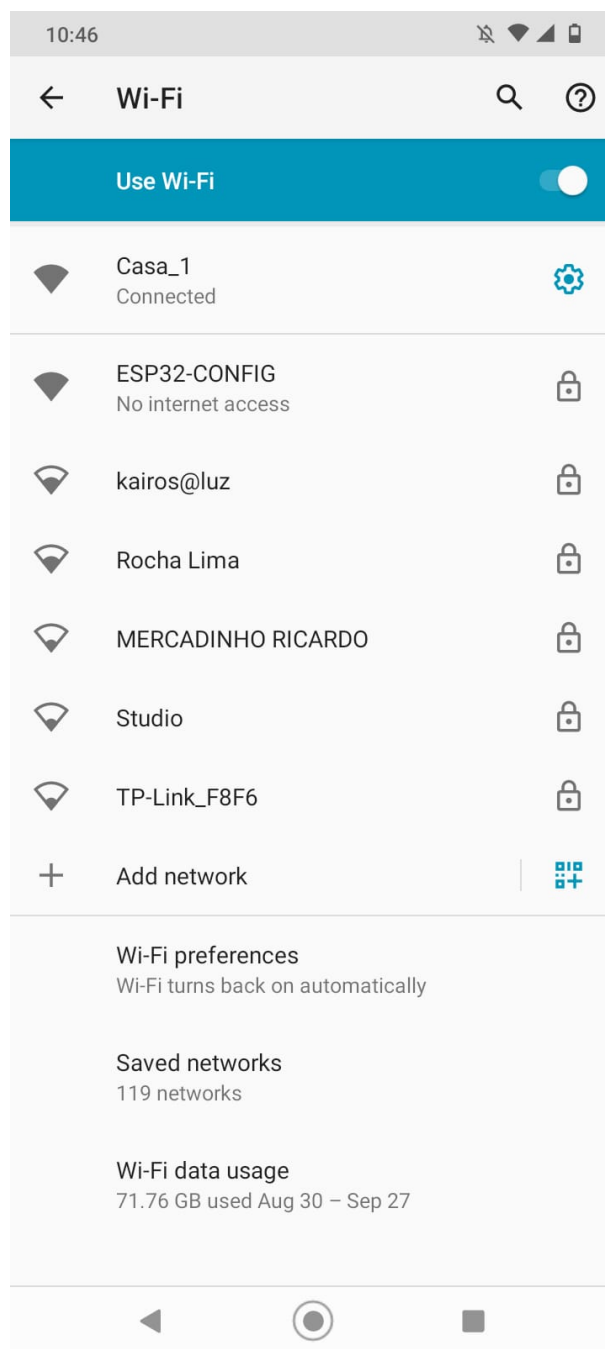
Com os sensores calibrados é feito o tratamento dos valores lidos por estes pelas portas AD do ESP-32, para depois serem enviados para armazenamento no banco de dados. São coletadas 5000 amostras para realização dos cálculos conforme mostra Figura 23, com essas informações é calculado o valor da potência média com a fórmula 2.4 e potência aparente usando a fórmula 2.15, os valores de Tensão e Corrente eficazes foram obtidos com as fórmulas 2.13 e 2.14 respectivamente. Concluído os cálculos das 5000 leituras o *soft* envia o resultado dos dados que serão mostrados ao usuário.

Figura 23 – Fluxograma de aquisição dos dados.



Fonte: Arquivo Próprio, 2021.

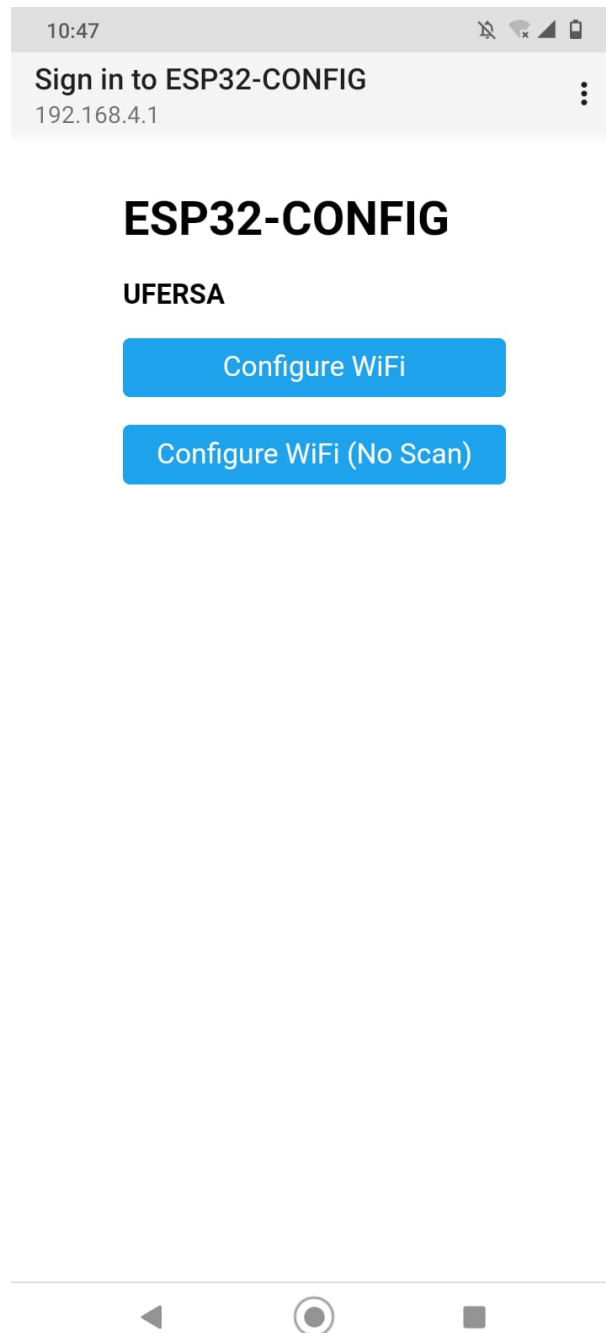
Para conectar o ESP-32 na rede *wi-fi* do usuário foi utilizado uma *interface* gráfica, utilizando a biblioteca *WiFiManager*, com algumas adaptações. Onde esta será responsável por receber usuário e senha do *wi-fi*, e conectar o ESP-32 a rede. Ao energizar o ESP-32, este gerará uma rede *wi-fi* que deverá ser acessada pelo usuário. A Figura 24 mostra a rede gerada pelo microcontrolador, essa foi acessada através de um dispositivo móvel.

Figura 24 – Rede *wi-fi* ESP-32.

Fonte: Arquivo Próprio, 2021.

Após se conectar a rede ESP32, o usuário deve abrir o navegador e acessar o IP:162.164.4.1, para ter acesso a interface de configuração, conforme mostra a Figura 25.

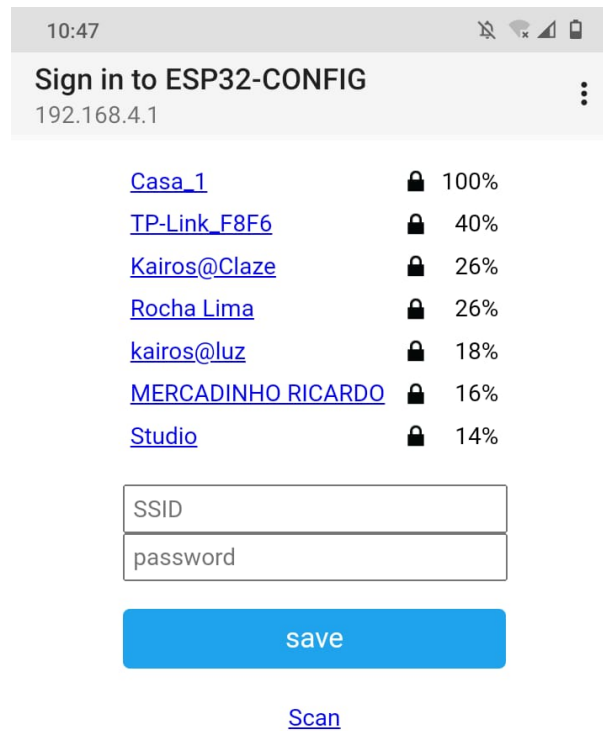
Figura 25 – Tela Inicial de Configuração ESP-32.



Fonte: Arquivo Próprio, 2021.

Temos duas opções de configurações, conforme a Figura 25, a "Configure *WiFi*", onde está fará uma varredura de todas as redes próximas ao dispositivo, dando a potência do sinal de cada uma das redes encontradas, conforme a Figura 26. E a configuração sem *scan*, como mostrado na Figura 25, essa opção deixa livre a digitação do nome da rede e senha, pelo usuário.

Figura 26 – Scan de redes e potência do sinal.



Fonte: Arquivo Próprio, 2021.

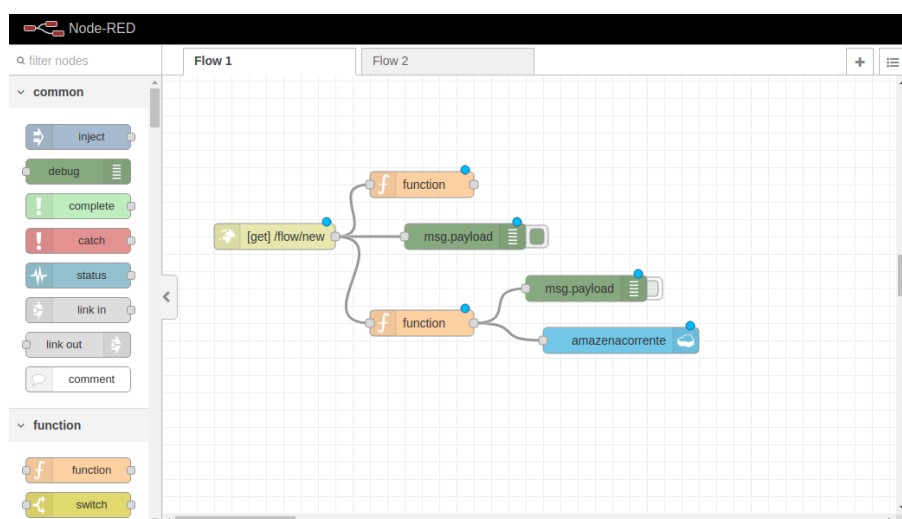
Após escolher a rede e inserir a senha, clicando em salvar o ESP-32 é inicializado e começará a enviar os dados de aquisição para a nuvem.

5.0.2 Configurando Node-Red

Depois é feita a conexão do ESP-32 com *Node-Red* em nuvem, que é para onde os dados serão enviados. O *Node-Red*, neste caso, serve como um intermédio entre o ESP-32 e o banco

de dados, pois o conjunto de dados enviados pelo ESP-32 precisam ser adicionados a data de leitura antes de serem armazenados no banco de dados, e foi usado o *Node-Red* para fazer esse tratamento. Para isso foi criada uma conta gratuita pela *IBM*, com isso temos acesso a uma plataforma do *Node-Red*, com recursos limitados, e a criação de um banco de dados de armazenamento de 1 Gb, o que para nossa aplicação é suficiente. Já com a plataforma configurada foi criada a programação em blocos para receber os dados vindos do ESP-32, e enviar para um banco de dados chamado: "armazenacorrente", conforme pode ser visto na Figura 27.

Figura 27 – Programação em Bloco *Node-Red*.



Fonte: Arquivo Próprio, 2021.

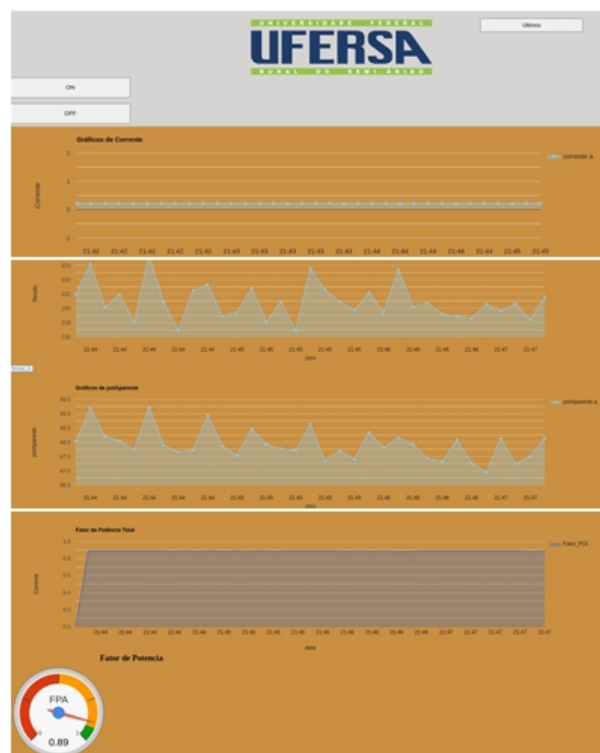
Com os dados sendo armazenados no banco de dados podemos manipulá-los com alguma linguagem de programação. Assim criou-se uma plataforma para mostrar os gráficos de forma simultânea ao seu armazenamento, utilizando *NodeJs* para o *back-end*, o qual será responsável por receber as requisições de acesso ao banco de dados, puxar os dados do banco de dados, e tratá-los para poder enviá-los para o *front-end*, este no que lhe concerne foi construído em *ReactJs*. Como o projeto é extremamente grande utilizou-se a ferramenta *GitHub* para versionamento do código, e estes podem ser acessados nos endereços: <https://github.com/Esdrassousa/TCC_Back-end> e <https://github.com/Esdrassousa/TCC_Front-End>. A plataforma foi hospedada com um subdomínio, assim pode ser acessada de forma fácil em qualquer dispositivo com acesso à *internet*. A plataforma é vista através de um *smartphone* na Figura 28 e acessada por meio de um computador na Figura 29.

Figura 28 – Plataforma de monitoramento e controle, acessada por *smartphone*.



Fonte: Arquivo Próprio, 2021.

Figura 29 – Plataforma de monitoramento e controle, acessada pelo computador.



Fonte: Arquivo Próprio, 2021.

A plataforma utilizada para hospedar o projeto foi o *Heroku*, esta disponibiliza a hospedagem de até cinco serviços de forma gratuita, o endereço da plataforma pode ser acessada pelo link: <https://tcc-ufersa-medidor-energia.herokuapp.com/esp>.

É possível visualizar nas Figuras: 28 e 29, que no canto superior direito da plataforma há um botão escrito "ultimos", ao clicar nele uma janela com alguns valores em minutos é mostrada ao usuário conforme Figura 30. Nessa mesma janela é possível selecionar até quantos minutos

atrás se deseja mostrar os valores nos gráficos, sendo que o tempo máximo configurado foi de 45 minutos e o mínimo de 5 minutos, por exemplo, se for selecionado "15 min", será exibido nos gráficos os últimos 15 minutos de coleta dos dados, e assim segue a lógica para todos os valores disponíveis.

Figura 30 – Botão para selecionar a partir de quantos minutos os dados são exibidos.



Fonte: Arquivo Próprio, 2021.

Quando o programa inicia se a carga estiver acionada o botão "ON" ficará como "Ligado", e o botão "OFF" permanecerá estático, indicando a possibilidade de desligar a carga, conforme mostra Figura 31. Caso contrário, se a carga estiver desligada, o botão "OFF" aparecerá como "Desligado" e o Botão "ON" permanecerá estático indicando que a carga pode ser acionada, conforme Figura 32.

Figura 31 – Botões de controle quando a carga estiver acionada.



Fonte: Arquivo Próprio, 2021.

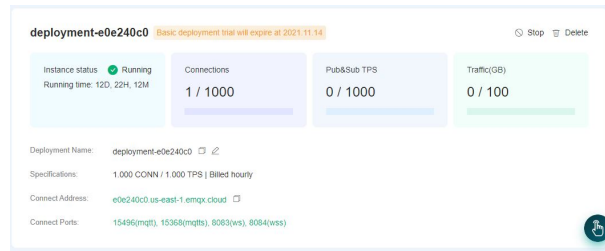
Figura 32 – Botões de controle quando a carga estiver "Desligada".



Fonte: Arquivo Próprio, 2021.

5.0.3 Configurando *Broker*

Para acionamento do relé, como já mencionado, foi utilizando o protocolo mqtt, dado que precisamos de uma comunicação M2M e para isso é preciso criar um *broker* em nuvem para intermédio de comunicação entre o ESP-32 e a plataforma de monitoramento, para tal foi utilizado uma versão gratuita de teste na plataforma *EMQ X Cloud*. Na Figura 33 é mostrado a configuração do *Broker* rodando em nuvem.

Figura 33 – Configuração *Broker*.

Fonte: Arquivo Próprio, 2021.

5.0.4 Custos do Projeto

Tabela 3 – Valores dos componentes utilizados e do projeto completo.

Componente	Quantidade	Preço por item	Preço total
ESP-32	1	R\$ 30,00	R\$ 30,00
Sensor SCT-013-000	1	R\$ 25,00	R\$ 25,00
Sensor ZMPT 101B	1	R\$ 13,00	R\$ 13,00
Resistor de 10K	2	R\$ 0,5	R\$ 1,00
Resistor de 360 Ω	1	R\$ 0,5	R\$ 0,5
Capacitor	1	R\$ 0,5	R\$ 0,5
Módulo relé de dois canais	1	R\$ 15,50	R\$ 15,50
Custo total	-	-	R\$ 85,00

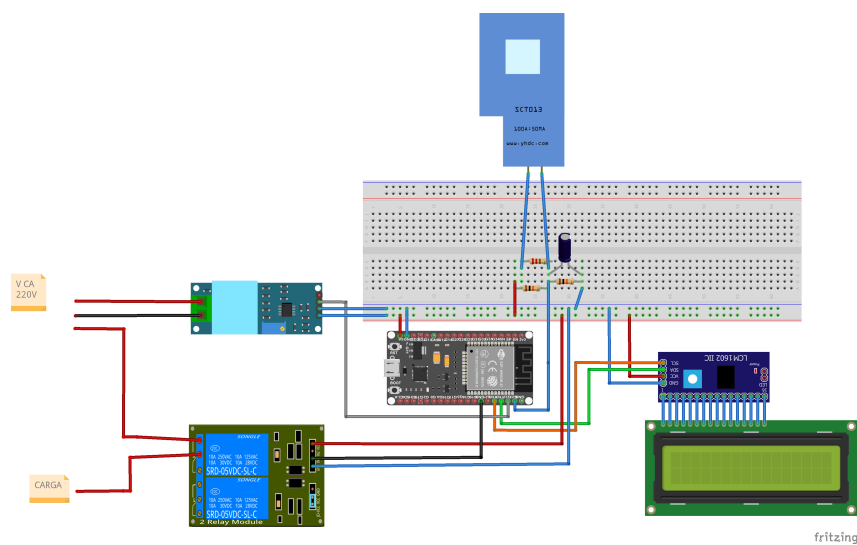
Fonte: Arquivo Próprio, 2021.

6 RESULTADOS

Após feito todos os testes de funcionamento do sistema e calibração dos sensores, foi montado o código completo disponível no Apêndice B, no ESP-32, que será responsável pelo o sistema como um todo, e está disponível no apêndice 1, foram ainda realizados vários testes e obteve-se sucesso na aquisição, manipulação e envio dos dados.

O projeto elétrico completo do sistema de medição e controle de Ligar/Desligar a carga, pode ser visto na Figura 34.

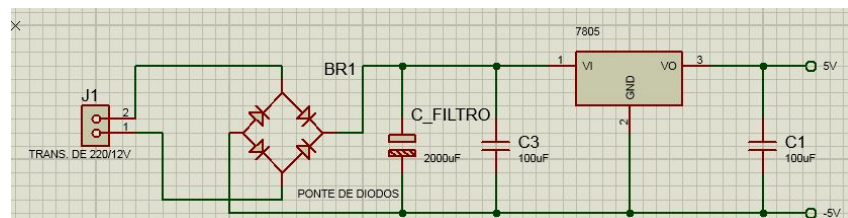
Figura 34 – Projeto completo do sistema.



Fonte: Arquivo Próprio, 2021.

Para alimentação do circuito foi dimensionado uma fonte de 5V com 1A de saída, a fonte foi simulada via *soft* é possível vê-la na Figura 35. Porém, por falta dos componentes para construção física da fonte optou-se por utilizar uma já pronta de 5V 0.6A.

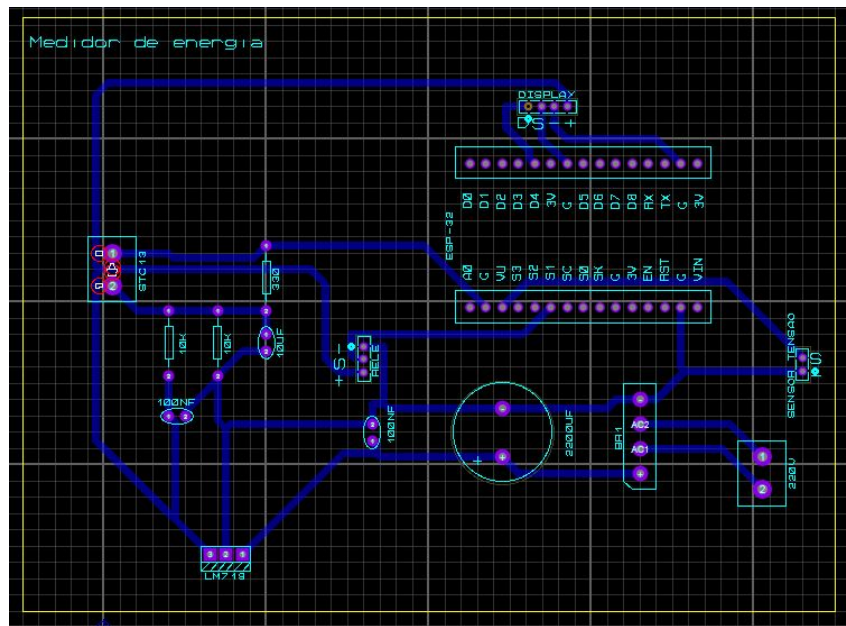
Figura 35 – Projeto Fonte de 5V.



Fonte: Arquivo Próprio, 2021.

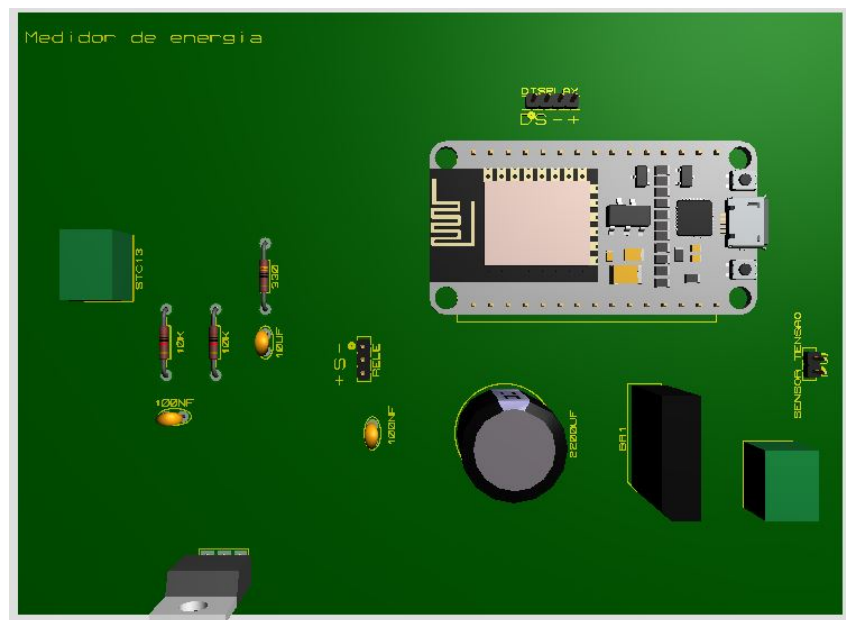
Na Figura 36, é possível visualizar a Placa de Circuito Impresso (PCI) do circuito e na Figura 37 está a placa 3D do circuito.

Figura 36 – PCI do Circuito.



Fonte: Arquivo Próprio, 2021.

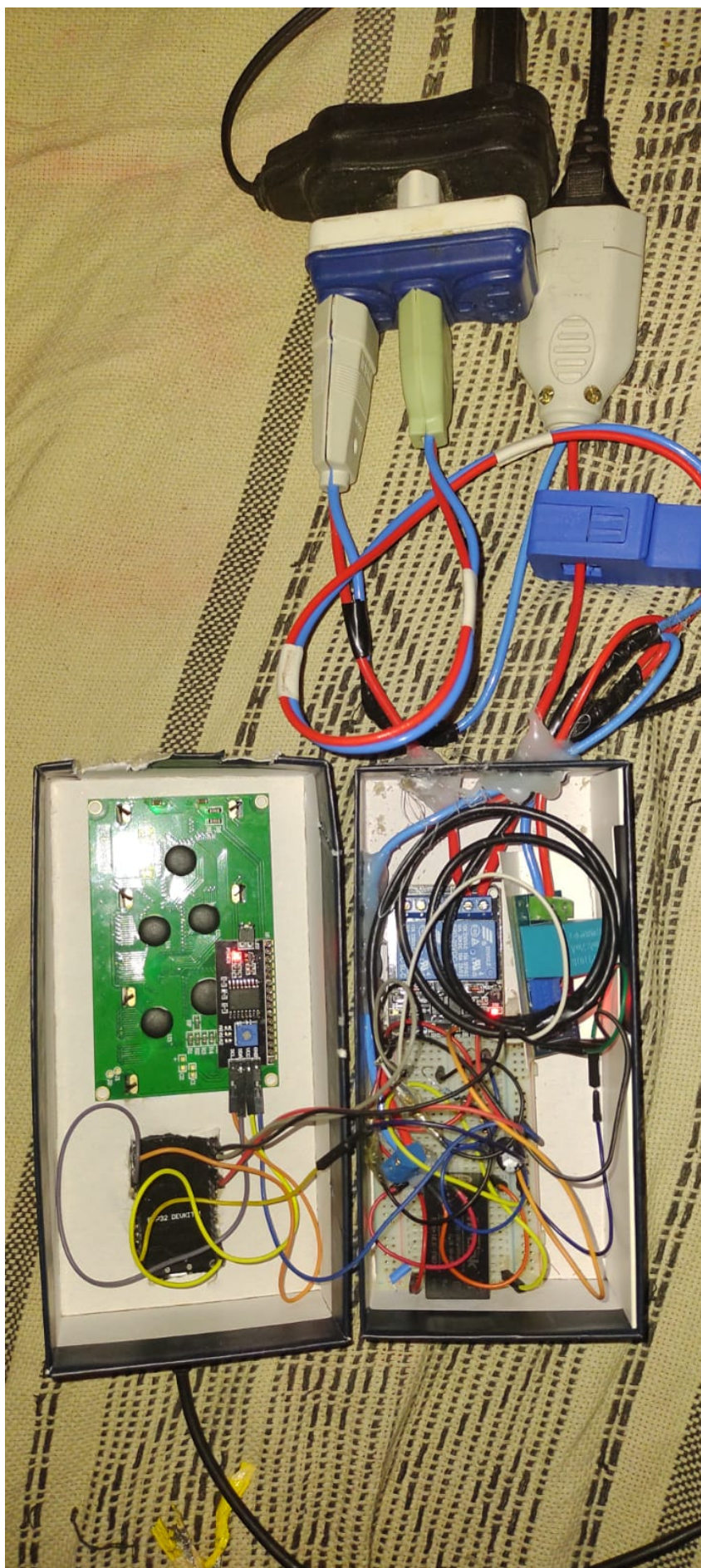
Figura 37 – Placa 3D do Circuito.



Fonte: Arquivo Próprio, 2021.

Para o protótipo final, os componentes tais como: resistores, capacitor e fonte de alimentação utilizados, foram montados em uma *protoboard*, e todo o sistema foi montado em uma caixa conforme mostra a Figura 38. A figura 39 mostra o projeto com a caixa fechada, ficando apenas o *display* e o ESP-32 amostra

Figura 38 – Protótipo montado dentro de uma caixa, visão interna.



Fonte: Arquivo Próprio, 2021.

Figura 39 – Protótipo montado dentro de uma caixa, visão externa.



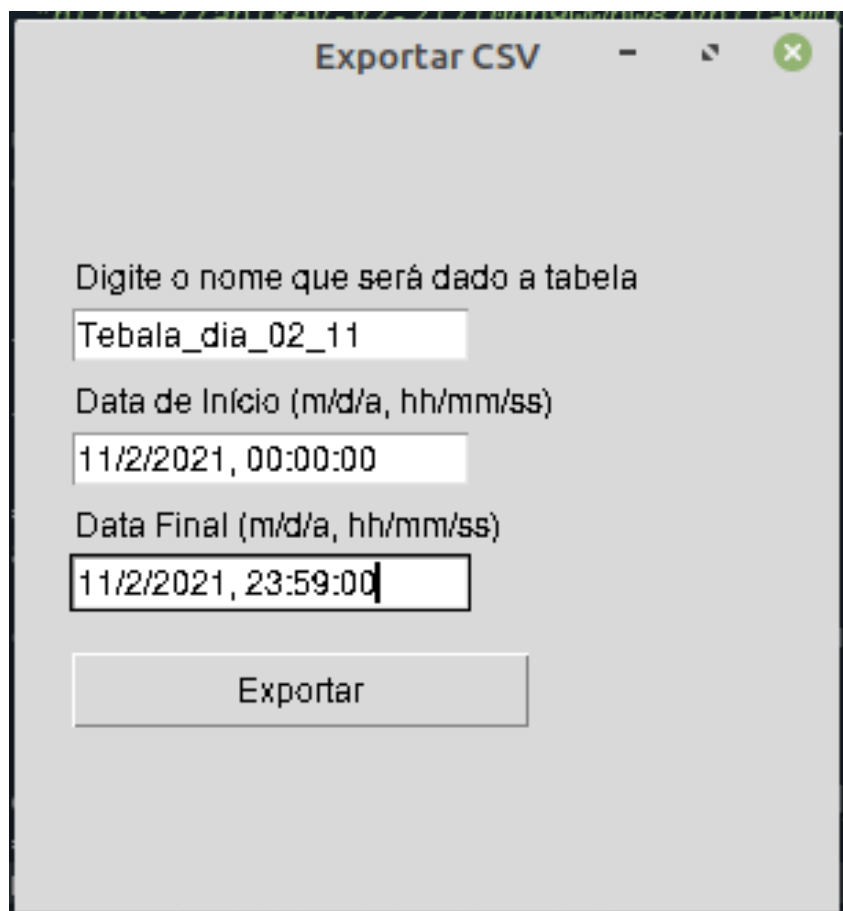
Fonte: Arquivo Próprio, 2021.

Foi criado um código em *python* visto no Apêndice A, para importar os valores armazenados no banco de dados e transferi-los para uma tabela CSV, a Figura 40 mostra uma parte da tabela, onde temos nas três primeiras colunas os campos: Data, Hora de aquisição referente aquela data, Horas, contendo: horas, minutos e segundos. Nas colunas subsequentes temos os valores de tensão, corrente, potência aparente, potência ativa e fator de potência. O *soft* criado, é mostrado na 41, e conforme pode ser visto ela permite delimitar por: data, horas, minutos e segundos o início e final do intervalo que se deseja importar os dados e o nome com que se deseja salvar a tabela. Com este *soft* pode-se transferir para uma tabela grande quantidade de dados, que poderão ser manipulados facilmente via algum *soft* de edição de planilhas como o *calc* do *libreoffice*.

Figura 40 – Arquivo csv com dados importados do banco de dados.

	A	B	C	D	E	F	G	H
1	Data	Hora	Horas	tensão	corrente	pot. aparente	pot. ativa	fp
2	02/11/2021		0 00:00:01	223,66	0,03	5,91	4,15	0,70
3	02/11/2021		0 00:00:07	224,50	0,03	6,83	4,45	0,65
4	02/11/2021		0 00:00:13	227,08	0,03	6,29	4,04	0,64
5	02/11/2021		0 00:00:19	224,50	0,03	6,85	4,43	0,65
6	02/11/2021		0 00:00:26	225,54	0,03	6,18	4,15	0,67
7	02/11/2021		0 00:00:31	223,93	0,03	6,42	4,17	0,65
8	02/11/2021		0 00:00:37	224,50	0,03	7,04	4,40	0,63
9	02/11/2021		0 00:00:43	225,95	0,03	6,61	4,31	0,65
10	02/11/2021		0 00:00:49	227,73	0,03	6,77	4,36	0,64
11	02/11/2021		0 00:00:55	224,85	0,03	6,13	3,97	0,65
12	02/11/2021		0 00:01:01	224,11	0,03	6,31	3,91	0,62
13	02/11/2021		0 00:01:07	225,34	0,03	6,39	4,22	0,66
14	02/11/2021		0 00:01:12	226,79	0,03	6,64	4,34	0,65
15	02/11/2021		0 00:01:18	224,08	0,03	6,14	4,17	0,68
16	02/11/2021		0 00:01:24	225,43	0,03	6,54	4,27	0,65
17	02/11/2021		0 00:01:30	224,45	0,03	6,01	3,55	0,59
18	02/11/2021		0 00:01:39	224,89	0,03	7,09	4,44	0,63
19	02/11/2021		0 00:01:46	225,21	0,03	7,70	6,39	0,83
20	02/11/2021		0 00:01:54	231,72	0,05	10,74	8,67	0,81
21	02/11/2021		0 00:01:57	230,45	0,04	10,29	6,82	0,66
22	02/11/2021		0 00:02:04	226,88	0,03	7,22	5,32	0,74
23	02/11/2021		0 00:02:11	227,41	0,03	6,40	4,29	0,67
24	02/11/2021		0 00:02:17	227,45	0,03	7,73	6,05	0,78
25	02/11/2021		0 00:02:22	228,45	0,03	7,25	5,57	0,77
26	02/11/2021		0 00:02:27	226,38	0,03	7,35	5,35	0,73
27	02/11/2021		0 00:02:33	226,87	0,03	7,32	5,16	0,71
28	02/11/2021		0 00:02:39	227,48	0,03	6,86	5,29	0,77
29	02/11/2021		0 00:02:46	226,63	0,03	7,36	5,24	0,71

Fonte: Arquivo Próprio, 2021.

Figura 41 – *Soft* para salvar dados em arquivo csv

Exportar CSV

Digite o nome que será dado a tabela

Tebala_dia_02_11

Data de Início (m/d/a, hh/mm/ss)

11/2/2021, 00:00:00

Data Final (m/d/a, hh/mm/ss)

11/2/2021, 23:59:00

Exportar

Fonte: Arquivo Próprio, 2021.

7 CONCLUSÃO

O trabalho apresentado teve o intuito de construir um sistema de monitoramento remoto de consumo de energia elétrica, e controle que desliga e liga a carga a qual está medindo o consumo, utilizando um microcontrolador de alta performance e de baixo custo com acesso à rede *wi-fi*, chamado ESP-32. Este dispositivo apresentou grande usabilidade e um alto poder de processamento, pois foi possível fazer diversas leituras de dois sensores em simultâneo, realizar os cálculos de tensão; corrente, potência ativa e aparente e fator de potência, enviar esses dados para um servidor, tudo isso em um curto espaço de tempo. Sua grande vantagem é já vir equipado com módulo *wi-fi* que facilita sua configuração para envio de dados para nuvem, e facilita a comunicação com equipamentos de IoT.

Com o uso dos recursos oferecidos pela plataforma de *IBM*, foi possível guardar os dados coletados pelo ESP-32 em um banco de dados, e de forma fácil manipulá-los para em seguida ser mostrado para o usuário na plataforma desenvolvida em *NodeJs* e *ReactJs*, esses dois *Framework*, formam uma excelente combinação de *Back-end* e *Front-end*, respectivamente, pois são facilmente manipuladas apresentando uma boa resposta de execução das instruções.

Com este equipamento os usuários da rede elétrica poderão ter uma melhor noção do seu consumo instantâneo de cada equipamento, podendo assim, seguir e ou criar estratégias para diminuir os gastos de consumo de sua rede. O usuário da rede elétrica poderá ainda controlar o acionamento e desacionamento do equipamento que está sendo medido, caso por exemplo: este equipamento seja esquecido ligado, o consumidor poderá desligá-lo remotamente de onde estiver, bastando ter algum equipamento conectado a *internet* que possa acessar a plataforma desenvolvida neste trabalho, evitando assim, gasto desnecessários nessas ocasiões.

8 TRABALHOS FUTUROS

Aprimorar o projeto para mostrar outros parâmetros da rede como: distorção harmônica, frequência da rede. Construir um protótipo trifásico. Implementar uma rede neural que identifique qual equipamento foi ligado ao dispositivo.

REFERÊNCIAS

Alex. F.S “**Algoritmos para Alocação de Banda em Redes de Acesso**”. 2010

ARAÚJO, Danielly Norberto. **Sistema de Monitoramento de Baterias Carregadas por Painéis Solares**. 2017. Trabalho de Conclusão de Curso. Universidade Federal de Campina Grande, 2017.

BEZERRA, Isadora Vilela; SILVA, Ellen Carmelita Capelo; ARAÚJO, Francisco José Costa. **REDES INTELIGENTES NO CONTEXTO DA INDÚSTRIA 4.0.**, IX. Simpósio de Engenharia de Produção da Região Nordeste, Joazeiro - BA, 7 set. 2020. Anais, p. 1-15.

Boylestad, Robert L. – **Introdução à Análise de Circuitos** – Prentice Hall/Pearson, 10ª. Ed, 2004

BRAGA, Newton C. **Reguladores de tensão**, 2010. Disponive em: <<https://www.newtoncbraga.com.br/index.php/como-funciona/1076-art156.html>>, Acesso em: 10 outubro de 2021.

BRAGA, Newton C. **Relés: Circuitos e aplicações**. Editora Newton C. Braga, 2017.

CANTÚ, Evandro; MONTEZ, Carlos Barros. **Protocolos, Tecnologias, Ferramentas e Laboratórios para Aplicações de Internet das Coisas**. In: XVII Congresso Latino-americano de Software Livre e Tecnologia Abertas. 2020.

CASTRO, Alisson de Souza. **Automação residencial centralizada utilizando ESP32 em conjunto com FPGA e comunicação wi-fi**. 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

COSTA, Francine de Oliveira et al. **Barreiras para a Implementação da Indústria 4.0: uma Revisão Bibliométrica e Sistêmica**. 2020. Monografia. Trabalho de Conclusão de Curso. Universidade Federal de Santa Catarina, 2020.

ELETROGATE, **Reguladores de tensão**. 2021, Disponível em: <<https://www.eletrogate.com/modulo-rele-2-canaais-5v>>, Acesso em: 11 outubro de 2021.

EMQ X Cloud. **EMQ X Cloud**. 2021. Disponível em: <<https://cloud.emqx.io>> Acesso em: 02 novembro de 2021.

ESPRESSIF, **Esp32 series datasheet**. 2019, Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>, Acesso em: 20 outubro de 2021.

FERNANDES, Ezéfferth Chlysman Araujo et al. **Desenvolvimento de uma aplicação web e android para o ramo de negócios alimentícios de açougues**. 2021.

FIGUEIREDO, FELIPE OTÁVIO PINTO. **Retrofit Do Processador Automático De Tecidos Oma Cm-69**. 2017. Monografia. Trabalho de Conclusão de Curso. UNIVERSIDADE FEDERAL DE OURO PRETO, 2017.

Filipeflop. **Display LCD 20x4 Backlight Azul**. Disponível em: <<http://www.filipeflop.com/pd-6b7e6-display-lcd-20x4-backlight-azul.html>>. Acesso em: 25 de Setembro de 2021.

FONSECA, Fabrício Ramos da. **Sensores**. 2006. Disponível em: <<http://www.adororobotica.com/Sensores.pdf>>. Acesso em: 03 Setembro de 2021.

FREIRE, Emelyn Clementino; DE ANDRADE, Lenise Souza Cardoso; DE ABREU, Carlos Alexandre Camargo. **IMPORTÂNCIA DO TRANSBORDAMENTO DE PRODUÇÕES ACADÊMICAS PARA O MERCADO DE TECNOLOGIA E INOVAÇÃO-REDES ELÉTRICAS INTELIGENTES (SMART GRIDS)**. In: 9th International Symposium on Technological Innovation. 2018.

GACKENHEIMER, C. **Introduction to React**. [S.l.]: Apress, 2015.

GitHub. **Git**Hub. Disponível em: <<https://github.com/>>. Acesso em: 01 de Setembro de 2021.

HAYASHI, R. **Artigo: Automação por meio de medidores inteligentes de energia elétrica permite adequação das distribuidoras à Tarifa Branca**. 2018. Editora Brasil Energia. Disponível em: <https://energiahoje.editorabrasilenergia.com.br/artigo-automacao-pormeio-de-medidores-inteligentes-de-energia-eletrica-permite-adequacao-das-distribuidorasa-tarifa-branca/>. Acesso em: 15 de Setembro de 2021.

Heroku. **Heroku dashboard**. 2021. Disponível em: <<https://dashboard.heroku.com/apps>>. Acesso

em: 01 setembro de 2021.

Instituto Newton Braga, **Reguladores de tensão 7800**, 20 Fevereiro 2010, Disponível em: <<https://www.newtonbraga.com.br/index.php/como-funciona/1076-art156.html>>. Acesso em: 05 de Setembro de 2021.

JUNIOR, ANTONIO RICARDO MEDRONHA STEIN; DOS SANTOS, Maicon. **Arquitetura Rest Api E Desenvolvimento De Uma Aplicação Web Service**. PROJETOS E RELATÓRIOS DE ESTÁGIOS, v. 1, n. 1, p. 1-59, 2019.

KOLBAN, Neil. Kolban's **book on ESP32**. [S.l.]: Leanpub, 2018.

MDN. Node.js. [S.l.], 2019. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/Node.js>. Acesso em: 6 ago. 2021.

MOREIRA, Ericlys de Farias. **Denuncie aqui!: Uma Aplicação para Realização de Denúncias em Tempos de Pandemia**. 2021. Trabalho de Conclusão de Curso.

NETO, Anis Assad et al. **A busca de uma identidade para a indústria 4.0**. Brazilian Journal of Development, v. 4, n. 4, p. 1379-1395, 2018.

NUNES, Leonardo HC et al. **Sistema de Baixo Custo para Monitoramento do Consumo de Energia Elétrica em Nuvem**. In: SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS, XXXVII., 2019, PETRÓPOLIS, RJ. Anais [...]. PETRÓPOLIS, RJ: SBrT, 2019. 2 p.

OCHOA, Alvaro Antonio et al. **Aplicação de uma fonte alternativa de energia termelétrica a gás natural visando reduzir o custo com energia elétrica em um edifício comercial**. Holos, v. 1, p. 72-86, 2015.

OLIVEIRA, Bruno Alberto Soares; ASSIS, Servílio; NOLLI, Carlos. **Development of a prototype electrical energy monitoring system via internet/desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via internet**. Revista de Engenharia da Universidade Católica de Petrópolis, v. 12, n. 1, p. 48-61, 2018.

OLIVEIRA, Edemilson et al. **CORREÇÃO DE FATOR DE POTÊNCIA NA EMPRESA**

SINAL RECICLAGEM DE RESÍDUOS. Revista TechnoEng-ISSN 2178-3586, v. 1, n. 1, 2019.

PATSKO, Luís Fernando. Tutorial, **Aplicações, Funcionamento e Utilização de Sensores.** 2006. Disponível em: <<http://www.maxwellbohr.com.br/downloads/robotica/mec1000kdr5000/tutorialeletronicaaplicacoesfuncionamentodesensores.pdf>>. Acesso em: 23 Setembro. 2021.

ROGGIA, Leandro; FUENTES, Rodrigo Cardozo. **Automação Industrial.** Santa Maria : Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016. Disponível em <[hTTP://ELETRO.G12.BR/ARQUIVOS/MATERIAIS/ELETRONICA6.PDF](http://eletro.g12.br/arquivos/materiais/electronica6.pdf)>. Acesso em: 02 Setembro de 2021.

SANTOS, Herman Lucas; AGULHARI, Cristiano Marcos; ENDO, Wagner. **Modelagem E Controle Robusto Por Lmi's De Planta Com Sistema De Proteção Que Imprime Não Linearidades Ao Modelo.** Congresso Brasileiro de Automática-CBA, [S. l.], v. XXII, 7 set. 2020. Anais, p. 1-8.

SANTOS, Talía Simões dos et al. **Análise da eficiência energética, ambiental e econômica entre lâmpadas de LED e convencionais.** Engenharia Sanitária e Ambiental, v. 20, p. 595-602, 2015.

SANTOS, Jean Willian; LARA JUNIOR, Renato Capelin de. **Sistema de automatização residencial de baixo custo controlado pelo microcontrolador ESP32 e monitorado via smartphone.** 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

SENA, Gerson Evandro de Oliveira. **Medidor de consumo de energia elétrica com acesso local e remoto usando plataforma ESP8266.** 2018. Trabalho de Conclusão de Curso. Universidade Federal do Pampa, 2018

SOUZA, E. da S. C. R. C. **Uma abordagem estatística para a previsão de potência reativa em sistemas elétricos.** Pesquisa Operacional, 2016. p.361-381 p.

TAVARES, R. S. et al, **Sistema De Monitoramento De Microclima De Um Ninho Da Formiga Cortadeira Sob O Paradigma Da Internet Das Coisas E Da Computação De Borda.** In: Anais do XXII ENMC – Encontro Nacional de Modelagem Computacional e X ECTM – Encontro de Ciências e Tecnologia de Materiais. 2019 .

TORRES, G. **Redes de Computadores**. 2ª. ed. Rio de Janeiro: Novaterra, 2014.

TORRES, Andrei BB; ROCHA, Atslands R.; DE SOUZA, José Neuman. **Análise de desempenho de brokers mqtt em sistema de baixo custo**. In: Anais do XV Workshop em Desempenho de Sistemas Computacionais e de Comunicação. SBC, 2016. p. 47-58.

USINAINFO, **Sensor de Tensão AC Zmpt101b/Voltímetro Arduino**. 2021. Disponível em: <https://www.usinainfo.com.br/sensor-de-tensao-arduino/sensor-de-tensao-ac-zmpt101b-voltimetro-arduino.html>. Acesso em: 25 de setembro de 2021.

VIEIRA, F. P. **Instrumento microprocessador para medição de potência ativa, reativa e fator de potência**. Universidade de São Paulo - Escola de Engenharia de São Carlos, São Carlos - SP, 2008. 16.

Vishay. Vishay **Semiconductors**. 2021. Disponível em: <http://www.vishay.com/docs/83708/appn50.pdf>, Acesso em: 10 de setembro de 2021.

WEG Automação. **Manual para correção do Fator de Potência**. 2018, Disponível em: <http://static.weg.net/medias/downloadcenter/hea/h8b/WEG-correcao-do-fator-de-potencia-958-manual-portugues-pdf>, Acesso em: 20 setembro de 2021.

WENDLING, Marcelo, **Reguladores de tensão**, 2010, Disponível em: <https://www.feg.unesp.br>, Acesso em: 20 setembro de 2021.

YHDC, **Split Core Current Transformer**, 2013. Disponível em: <http://en.yhdc.com/product1311.html?productId=401>, Acesso em: 10 setembro de 2021.

APÊNDICE A – CÓDIGO PYTHON

```

import tkinter as tk
from tkinter import *

from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
import re
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from datetime import datetime, timedelta
import datetime as dt
import pandas as pd
import csv

def importar_dados():

    serviceUsername = "apikey-v2-2i7l0qp9wwbw87vpija9mi31o7yennrevo7fd87xpz00"
    servicePassword = "974d78d495165a06f9f8a753ffe33542"

    serviceURL = "https://apikey-v2-2i7l0qp9wwbw87vpija9mi31o7yennrevo7fd87xpz00:974d78d495165a06f9f8a753ffe33542@42b86470-c6b5-4354-813b-727ba9f47dce-bluemix.cloudantnosqldb.appdomain.cloud"

    client = Cloudant(serviceUsername, servicePassword, url=serviceURL)
    client.connect()

    print (client.all_dbs())

    a = client.all_dbs()

    my_database = client[a[0]]
    print(my_database)

    result_collection = Result(my_database.all_docs, include_docs=True)

    result11 = result_collection[ed_data_inicio.get() : ed_data_final.get()]

```

```

with open(ed_nome_tabela, 'w', newline='') as file:
    writer = csv.writer(file, delimiter=";")

    writer.writerow(["Data", 'Hora', "Horas", "tensao", "corrente",
                    , "pot_aparente", "pot_ativa",
                    "fp"])
    for i in range(len(result11)):

        try:
            separadados = result11[i]
            separaemDoc = separadados['doc']

            separaDadosDoc = separaemDoc['dados']
            Pega_primeiro_separaDadosDoc = separaDadosDoc[0]
            Tensao = Pega_primeiro_separaDadosDoc['tensao'].replace('.', ',')
            Corrente = Pega_primeiro_separaDadosDoc['corrente']
                .replace('.', ',')
            Pot_Aparente = Pega_primeiro_separaDadosDoc['pot_aparente']
                .replace('.', ',')
            Pot_ativa = Pega_primeiro_separaDadosDoc['pot_ativa']
                .replace('.', ',')
            FP = Pega_primeiro_separaDadosDoc['fp'].replace('.', ',')

            separaDatas = separaemDoc['data']
            separraData_e_Horas = separaDatas.split(',')
            date = datetime.strptime(separraData_e_Horas[0], '%m/%d/%Y').date()
            data_em_texto = date.strftime('%d/%m/%Y')
            datasSetadas = pd.to_datetime(separaDatas)
            separaHora_Hora = separraData_e_Horas[1].split(':')
            writer.writerow([data_em_texto, separaHora_Hora[0],
                            separraData_e_Horas[1],
                            Tensao, Corrente, Pot_Aparente, Pot_ativa,
                            FP])

        except :
            print('error')
        finally:
            pass

janela_main = tk.Tk()

janela_main.geometry("300x300+200+200")
janela_main.title("Exportar CSV")

```

```
lb_data_nome_tabela =  
Label(janela_main, text="Digite o nome que será dado a tabela")  
lb_data_nome_tabela.place(x=20,y=60)  
  
ed_nome_tabela = tk.Entry(janela_main, width=20)  
ed_nome_tabela.place(x=20,y=80)  
  
lb_data_inicio = Label(janela_main, text="Data de Início (m/d/a, hh/mm/ss)")  
lb_data_inicio.place(x=20,y=105)  
  
ed_data_inicio = tk.Entry(janela_main, width=20)  
ed_data_inicio.place(x=20,y=125)  
  
lb_data_fim = Label(janela_main, text="Data Final (m/d/a, hh/mm/ss)")  
lb_data_fim.place(x=20,y=150)  
  
ed_data_final = tk.Entry(janela_main, width=20)  
ed_data_final.place(x=20,y=170)  
  
bt_importa =  
Button(janela_main, width=20, text="Exportar", command=importar_dados)  
bt_importa.place(x=20,y=205)  
  
janela_main.mainloop()
```

APÊNDICE B – CÓDIGO ESP-32

```

#include<WiFi.h>
#include <PubSubClient.h>
#include <HTTPClient.h>
#include "EmonLib.h"
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 3);

//////////////////Wifi//////////////////
const char* ssid = "Casa_1";
const char* password = "84996403942";
//////////////////

//////////////////mqtt//////////////////

const char* mqtt_server ="e0e240c0.us-east-1.emqx.cloud";
const int mqttPort = 15496;
const char* mqttUser = "admin";
const char* mqttPassword = "admin";
const int led = 32;
//const int led = 2;
//////////////////

int count = 0, t_count = 0, cont=0;
int pin_sensor_corrente=39;
int pin =36;

float tensaoPico;
double menorValor;
int valor_inst;
float tensaoRms;

unsigned long tempo1;
unsigned long tempo2;

unsigned long tempo11;
unsigned long tempo22;

float correnteInt;
float correntePico;
float correnteEf;
double maiorValor;
double correnteValor;

```

```
int sensorValue;

float tensao[2000], corrente[2000];
float pm, ps, vm, im, vef, ief, fp;
EnergyMonitor emon1;
EnergyMonitor SCT013;
unsigned long lastTime = 0;
unsigned long timerDelay = 5000;

String serverName = "http://fortesol-energia.mybluemix.net/flow/new";

WiFiClient espClient;
PubSubClient client(espClient);

void setup()
{
    pinMode (led, OUTPUT);
    SCT013.current(pin_sensor_corrente,2);
    pinMode(pin_sensor_tensao, INPUT);
    pinMode(pin_sensor_corrente, INPUT);
    lcd.init();
    lcd.backlight();
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print("Conectando ao WiFi:");
        Serial.println(ssid);
    }

    Serial.println("Conectado ao WiFi");
    Serial.println("");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    client.setServer(mqtt_server, mqttPort);
    client.setCallback(callback);

    while (!client.connected()) {
        Serial.println("Conectando ao MQTT...");
        String clientId = "teste";
        clientId += String(random(0xffff), HEX);
        if (client.connect(clientId.c_str(), mqttUser, mqttPassword )) {
```

```
        Serial.println("connected");
    } else {
        Serial.print("failed with state ");
        Serial.print(client.state());
        delay(2000);
    }
}

Serial.print("Tentando enviar a mensagem");
client.publish("teste", "Hello from ESP32");
client.subscribe("teste");

}

void callback(char* topic, byte* payload, unsigned int length) {

    Serial.print("Message arrived in topic: ");
    Serial.println(topic);

    Serial.print("Message:");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }

    if (payload[0] == '0'){

        Serial.println("Desligando carga");
        digitalWrite(led, HIGH);
    }

    if (payload[0] == '1'){

        Serial.println("Ligando carga");
        digitalWrite(led, LOW);
    }

    Serial.println();
    Serial.println("-----");

}

void loop()
{
```

```
client.loop();
menorValor = 4095;

maiorValor = 0;

for(int i =0; i<2000 ; i++){

    double Irms = SCT013.calcIrms(2);
    corrente[i] = Irms;

    valor_inst = analogRead(pin_sensor_tensao);
    tensaoPico = valor_inst-3450;
    if (led == HIGH) {
        tensaoPico = map(menorValor,3450,1800,0,315);
    }else{

        tensaoPico = map(menorValor,3450,1650,0,315);
    }
    tensaoRms = tensaoPico/1.4;
    tensao[i] = tensaoRms;

    sersorValue = analogRead(pin_sensor_tensao);

    tempo1=0;
    tempo2=0;
    vef = vef + sq(tensao[i]);
    ief = ief + sq(corrente[i]);
    pm = pm + (tensao[i]) * (corrente[i]);

    if (maiorValor < correnteInt){
        maiorValor = correnteInt;

    }
    if (menorValor > valor_inst){
        menorValor = valor_inst;

    }
}
//tensaoPico = map(menorValor,3550,2160,0,315);
vef = sqrt(vef / 2000);
ief = sqrt(ief / 2000);
ps = vef * ief;
pm = pm / 2000;
```



```
    fp = pm/ps;

    if(fp < 0.5){
        fp =1;
    }

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Corrente: "); lcd.print(ief); lcd.print(" A");
    lcd.setCursor(0,1);
    lcd.print("Tensao: "); lcd.print(vef); lcd.print(" V");
    lcd.setCursor(0,2);
    lcd.print("Potencia: "); lcd.print(ps); lcd.print(" W");
    lcd.setCursor(0,3);
    lcd.print("FP: "); lcd.print(fp);
    Serial.println("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX enviando dados via HTTP XXXX");
    HTTPClient http;

    String serverPath;

    serverPath = serverName + "?" + "tensao=" + vef + "&" + "corrente=" + ief + "&";

    http.begin(serverPath.c_str());

    ief =0;vef=0;pm=0;
    int httpResponseCode = http.GET();
    if (httpResponseCode>0) {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        String payload = http.getString();
        Serial.println(payload);
    }
    else {
        Serial.print("Error code: ");
        Serial.println(httpResponseCode);
    }

    //http.end();

}
```

ANEXO A – DATASHEET SCT013-100

Split core current transformer



Model: SCT-013

Rated input current: 5A/100A

Characteristics: Opening size: 13mm*13mm,

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

1m leading wire, standard $\Phi 3.5$ three core plug output.

Current output type and voltage output type (voltage output type built-in sampling resistor)

Purpose: Used for current measurement, monitor and protection for AC motor, lighting equipment, air compressor etc

Core material: ferrite

Mechanical strength: the number of switching is not less than 1000 times(test at 25°C)

Safety index: Dielectric strength(between shell and output)1000V AC/1min

Fire resistance property: In accordance with UL94-Vo

Work temperature: -25°C~+70°C

Outline size diagram: (in mm)

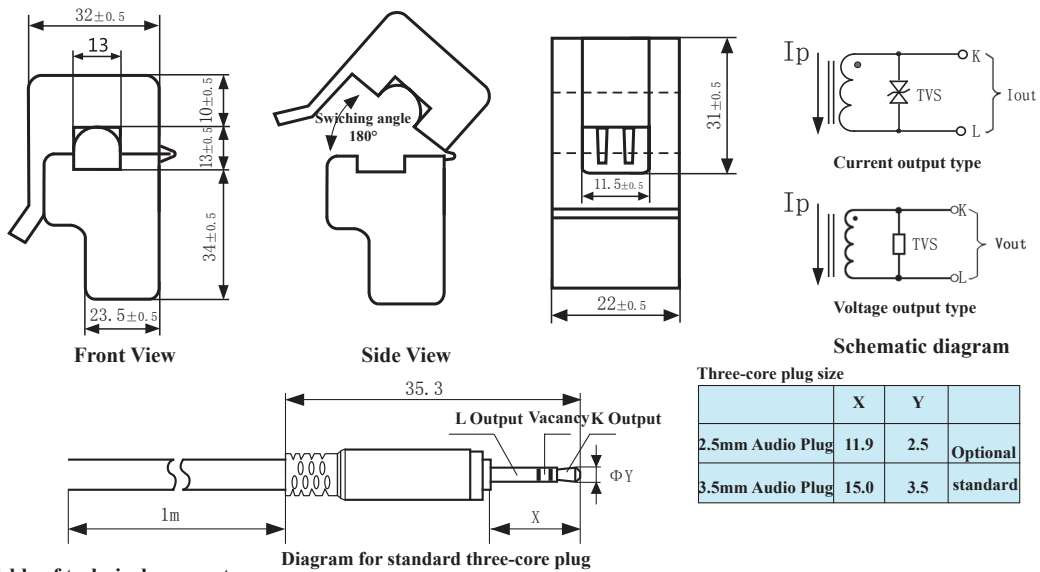


Table of technical parameter:

Model	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Input current	0-100A	0-5A	0-10A	0-15A	0-20A
Output type	0-50mA	0-1V	0-1V	0-1V	0-1V
Model	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-000V
Input current	0-25A	0-30A	0-50A	0-60A	0-100A
Output type	0-1V	0-1V	0-1V	0-1V	0-1V

※ Output type: voltage output type built-in sampling resistor, current output type built-in protective diode.

Tel: 86-13933609279 Fax: 86-7929499-804 skype:macymengl
Web (China): www.yhdc.com Web (Latvia): www.yhdc.lv

ANEXO B – DATASHEET ZMPT101B

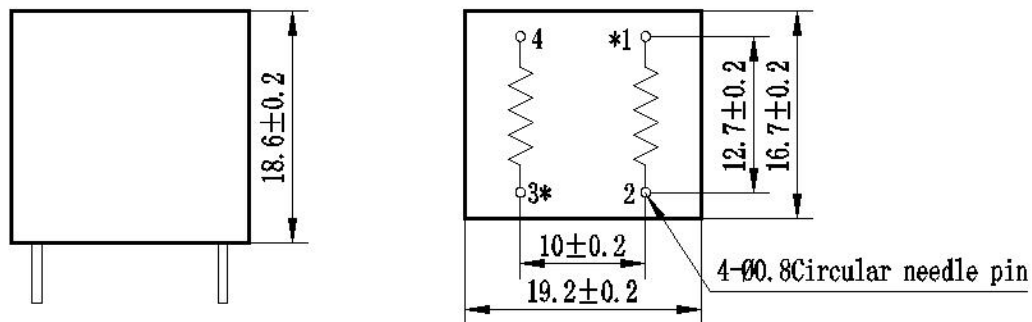
Qingxian Zeming Langxi Electronic

www.micro-transformer.com

ZMPT101B Current-type Voltage Transformer

Small size, high accuracy, good consistency, for voltage and power measurement

Structural parameters:



Remarks: primary input: 1、2 pins secondary output: 3、4pins

Or

primary input:: 3、4 pins secondary output::1、2pins

“*” Same polarity

Front view

Bottom view

The main technical parameters:

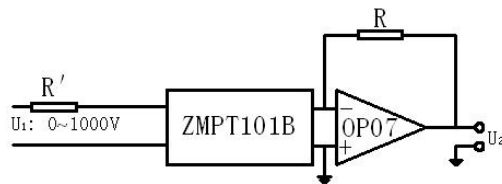
Model	ZMPT101B
Rated input current	2mA
Rated output current	2mA
turns ratio	1000:1000
phase angle error	$\leq 20'$ (input 2mA, sampling resistor 100 Ω)
operating range	0~1000V 0~10mA (sampling resistor 100Ω)
linearity	$\leq 0.2\%$ (20%~120%)
Permissible error	$-0.3\% \leq f \leq +0.2\%$ (input 2mA, sampling resistor 100 Ω)
isolation voltage	4000V
application	voltage and power measurement
Encapsulation	Epoxy
installation	PCB mounting (Pin Length>3mm)
Operating temperature	-40 $^{\circ}\text{C}$ ~+60 $^{\circ}\text{C}$
Case Material	ABS (Note: ABS CASE is NOT available for wave-soldering)

Tel: 86-25-52601870

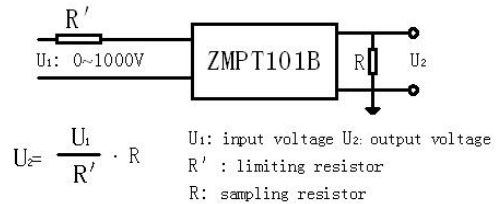
E-mail: zm@zeming-e.com

Qingxian Zeming Langxi Electronic

www.micro-transformer.com

Direction for use:**Figure I**

1. The typical usage of the product is for the active output (Figure I). R' is a limiting resistor , R is a sampling resistor.
2. The product can be directly through the resistance sampling , easy to use (Figure II).

**Figure II**