

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

ASSIS GABRIEL GRESELLE

MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA APLICADO A CARGAS
RESIDENCIAIS

Joinville
2022

ASSIS GABRIEL GRESELLE

MONITORAMENTO DO CONSUMO DE ENERGIA ELÉTRICA APLICADO A CARGAS
RESIDENCIAIS

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica no curso de Engenharia Mecatrônica, da Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Orientador: Dr. Diego Santos Greff

Joinville
2022

Esta monografia é dedicada aos meus pais por terem me proporcionado a oportunidade de estudar e realizar os meus sonhos profissionais, pilares da minha formação como ser humano.

RESUMO

A Internet das Coisas (IoT) proporciona a gestão de maneira automatizada dos recursos habitacionais que antes eram operados manualmente, conectar os ambientes físicos e virtuais representa um grande avanço. Considerando isso, este trabalho tem como objetivo proporcionar as pessoas uma maneira de medir a potência elétrica média do aparelho conectado ao módulo em uma residência, pois, mostra-se como uma forma eficiente de reduzir o consumo de energia, e intensificar os conhecimentos em temas relevantes como IoT utilizando dispositivos de baixo custo. O dispositivo deste projeto será desenvolvido utilizando um microcontrolador do tipo ESP32, um relé optoacoplador, sensor de tensão ZMPT101B e sensor de corrente ACS758. Com esse módulo é possível medir o consumo de eletrodomésticos pelo aplicativo de celular.

Palavras-chave: Sensor tensão, Sensor Corrente, IoT, Consumo de Energia, Potência Elétrica Média.

ABSTRACT

The Internet of Things (IoT) provides the automated management of housing resources that were previously operated manually, connecting the physical and virtual environments represents a great advance. Considering this, this work aims to provide people with a way to measure the average electrical power of the device connected to the module in a residence, as it is shown to be an efficient way to reduce energy consumption, and to intensify knowledge on topics relevant as IoT using low-cost devices. The device of this project will be developed using an ESP32 microcontroller, an optocoupler relay, ZMPT101B voltage sensor and ACS758 current sensor. With this module, it is possible to measure the consumption of home appliances through the mobile application.

Keywords: Voltage Sensor, Current Sensor, IoT, Energy Consumption, Average Electrical Power.

LISTA DE FIGURAS

Figura 1 – Potência elétrica	12
Figura 2 – Circuito completo do módulo	13
Figura 3 – Diagrama de casos de uso.	14
Figura 4 – ESP32 DevKit V4.	15
Figura 5 – Sensor Corrente.	15
Figura 6 – Sensor de Tensão.	16
Figura 7 – Filtro Passa baixas e Divisor de tensão.	18
Figura 8 – Modulo Relé.	19
Figura 9 – Princípio de funcionamento do relé.	19
Figura 10 – Fonte 220Vac/5Vdc.	20
Figura 11 – Placa de circuito impresso.	21
Figura 12 – Placa de circuito impresso pronta.	21
Figura 13 – Diagrama de atividades.	22
Figura 14 – Etapas para configuração inicial.	23
Figura 15 – Offset dos Sensores.	24
Figura 16 – Conversão dos códigos do ADC em valores de tensão	25
Figura 17 – Tensão na saída do regulador LM7805.	26
Figura 18 – Gráfico do sensoriamento de uma lâmpada de 12W.	27
Figura 19 – Gráfico do sensoriamento de uma lâmpada de 23W.	28
Figura 20 – Teste utilizando uma lâmpada 12W.	33

LISTA DE TABELAS

Tabela 1 – Potência obtida e erro percentual. (Autoria Própria)	29
---	----

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivo	8
1.1.1	Objetivo Geral	9
1.1.2	Objetivos Específicos	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	Potência Elétrica	11
3	ESTUDO E DESENVOLVIMENTO DO PROTÓTIPO	13
3.1	Microcontrolador	14
3.2	Sensor de corrente	15
3.3	Sensor de tensão	16
3.4	Filtros passa baixa e Divisor de tensão	17
3.5	Relé optoacoplador	19
3.6	Fonte de alimentação	20
3.7	Placa de circuito impresso	20
3.8	Interface	21
4	RESULTADOS EXPERIMENTAIS	23
4.1	Ponto de acesso	23
4.2	Offset dos sensores	24
4.3	Calculo dos valores RMS	25
4.4	Testes práticos	26
5	CONCLUSÃO	30
	REFERÊNCIAS	31
	APÊNDICE A	33

1 INTRODUÇÃO

Com a ascensão das casas inteligentes as pessoas têm cada vez mais investido em dispositivos como módulos e tomadas inteligentes, com o objetivo de trazer tecnologia, inovação e praticidade para dentro de casa, podendo controlar todos os seus aparelhos remotamente com aplicativo de celular, possibilitando a administração automatizada de recursos que antes eram operados manualmente, tendo o ser humano como fator de decisão.

A Internet das Coisas (IoT), descreve a crescente interligação entre objetos inteligentes como também a conexão desses com a Internet. Conectar os ambientes físicos e virtuais representa um grande avanço, pois sem isso todas as informações do mundo virtual são dependentes de humanos, mas esses têm tempo, precisão e atenção limitadas. A IoT permite aos computadores terem acesso aos dados do ambiente sem ajuda dos seres humanos, o que permite enorme quantidade de dados ser adquirida e analisada fornecendo percepções mais rápidas e precisas do ambiente (ASHTON, 2009).

Este dispositivo tem objetivo de permitir o acesso de informações sobre os aparelhos a ele conectados. De acordo com a *American Council for an Energy-Efficient Economy* (ACEEE), em uma pesquisa realizada entre 1995 e 2010 o acesso às informações de consumo de eletricidade causa uma redução do consumo. A pesquisa encontrou uma redução média de 12% no consumo quando há o acesso a informações em tempo real discriminadas por aparelho (EHRHARDT-MARTINEZ et al., 2010).

Esses resultados demonstram como o desenvolvimento de dispositivos de medição podem impactar no consumo de recursos em uma habitação. Nessa perspectiva, esse trabalho visa desenvolver um módulo para medir e proporcionar acesso às informações de consumo. Para isso, será utilizado o hardware ESP32 juntamente com sensores de tensão ZMPT101B, sensor de corrente ACS758 e um relé, como plataforma de desenvolvimento para o software será utilizado o Integrated Development Environment (IDE) do Arduino.

1.1 OBJETIVO

Desenvolver um módulo para medir a potência elétrica média de eletrodomésticos conectados a ele, propõe-se neste trabalho os seguintes objetivos.

1.1.1 Objetivo Geral

Propiciar a ambientes a capacidade de medir e informar o consumo de energia elétrica em tempo real.

1.1.2 Objetivos Específicos

- Desenvolver um módulo para medir potência elétrica média;
- Projetar e construir um protótipo do dispositivo;
- Analisar erros associados as medições;
- Examinar a viabilidade do dispositivo.

2 FUNDAMENTAÇÃO TEÓRICA

Entre os anos de 2019 e 2022, através de uma pesquisa na base de dados do Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE Xplore Digital Library), foram publicados mais de 3000 artigos relacionados com medidores de energia. Isso revela a importância do tema, juntamente com a expansão da comunicação sem fio e automação residencial, gerando avanços e descobertas nestas áreas.

No artigo "Implementation of wifi-based single phase smart meter for internet of things (iot)"(HLAING et al., 2017) os autores sugerem a implementação de um sistema de monitoramento para o uso diário da energia elétrica, sendo uma solução barata e fácil de instalar, trazendo a proposta de eliminar o trabalho manual, evitar erros humanos na leitura de medidores antigos e a redução de custos no consumo de energia, pois os usuários estarão cientes do uso de eletricidade em suas casas para reduzir o desperdício de energia. O trabalho é focado, principalmente, em monitoramento de energia com o conceito IoT, implementação de uma rede e protocolo de sensores sem fio de custo muito baixo, capaz de ler e enviar automaticamente os dados para que os usuários vejam a leitura atual do medidor de energia.

Os autores de "An IoT-Based Smart Utility Meter" (AL-ALI et al., 2018), propõem o design, desenvolvimento e testes de um sistema Smart Meter de utilidade que reúne em um único dispositivo o monitoramento do consumo de eletricidade, água e gás, baseado em IoT. O medidor possui uma rede única e pode ser programado remotamente para ler os três tipos de consumo pelos prestadores de serviços públicos. Os consumidores finais podem acessar através de um computador pessoal ou um telefone móvel. Também foi utilizado o serviço do Google Map para exibir o status do medidor mostrando o consumo de cada concessionária e se está conectado ou desconectado.

Tiwari e Matta discutem em (TIWARI; MATTA, 2019) estabelecer uma arquitetura eficiente para uma rede smart home. Neste artigo também é estabelecida uma forma de gerenciar recursos em IoT com base em diferentes camadas de implementação, a arquitetura proposta é dada pelo mapeamento de dispositivos inteligentes com base em sua localização dentro de uma residência, como por exemplo a ação de ligar a televisão deve ser mapeada para uma instrução que percorrerá o endereço deste aparelho definido pelo cômodo onde ele se encontra.

O artigo (AMRI; SETIAWAN, 2018) propõe implementar um sistema de smart grid baseado em IoT que monitora o uso de energia elétrica de dispositivos domésticos, como lâmpadas. Os dados são obtidos utilizando o microcontrolador Arduino, juntamente com o microcontrolador e módulo Wi-Fi ESP8266, e enviados para

um broker utilizando o protocolo de troca de mensagens MQTT. Esses dados são por fins armazenados e disponibilizados por acesso via smartphone. O usuário pode então manter um controle de seu gasto de energia elétrica e também realizar o pagamento da conta utilizando RFID.

2.1 POTÊNCIA ELÉTRICA

A grandeza de interesse neste trabalho é a potência elétrica, com ela é possível monitorar a quantidade de energia gasta por um equipamento, sendo um dos dados de maior interesse para a aquisição com dispositivos de IoT. O conceito de carga elétrica é a base para descrever todos os fenômenos elétricos, os conceitos de tensão e corrente são úteis porque podem ser expressos quantitativamente, assim sendo, a definição da tensão é a energia (ω) por unidade de carga (q) criada pela separação entre cargas e é expressa em volt [V] (HAYT; KEMMERLY; DURBIN, 2014).

$$v = \frac{d\omega}{dq} \quad (1)$$

A taxa de fluxo dessas cargas, em função do tempo (t), é a definição de corrente e sua unidade é o ampere [A].

$$i = \frac{dq}{dt} \quad (2)$$

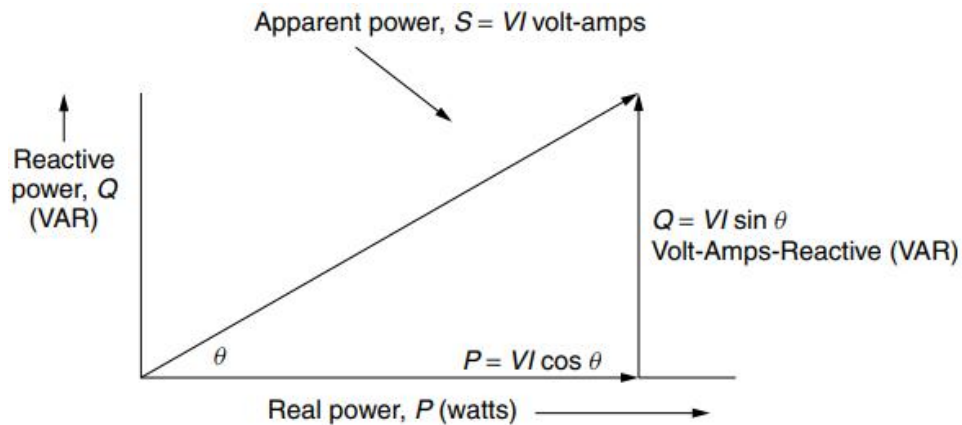
Embora a tensão e corrente sejam úteis na análise e projeto de sistemas que utilizam eletricidade, muitas vezes o resultado que realmente interessa não é expresso em termos elétricos, mas em potência (HAYT; KEMMERLY; DURBIN, 2014).

Potência é a energia por unidade de tempo e é igual ao produto da tensão e da corrente, sendo expressa em watt [W].

$$p = \frac{d\omega}{dt} = vi \quad (3)$$

A potência em sistemas que trabalham com tensão alternada é complexa sendo essa composta de uma componente real chamada de potência ativa e uma componente imaginária chamada de potência reativa. A Figura 1 ilustra esse conceito, sendo P a potência média, S a potência aparente e Q a potência reativa (DORF; SVOBODA, 2016).

Figura 1 – Potência elétrica



Fonte: MASTERS (2004, p.63).

Destaca-se que no contexto desse trabalho considerou-se a potência média mais significativa ao usuário, e no decorrer do trabalho a potência média foi tratada apenas como potência. Para cálculo da potência média, é utilizado a tensão e corrente eficaz.

A tensão eficaz (VRMS) é equivalente a área da senoide e devido o fato de a mesma possuir um semiciclo positivo e outro negativo, para evitar um resultado nulo, utilizamos a técnica de integrar pela raiz da média quadrática (SADIKU; MUSA, 2014).

$$V_{RMS} = \sqrt{\frac{1}{T} \int_0^T V^2(t) dt} \quad (4)$$

De maneira análoga, o valor eficaz de uma corrente de sinal periódico é dado pelo valor em corrente contínua que ao passar por um resistor dissiparia a mesma quantidade de calor em um intervalo de tempo, isto é, que teria a mesma potência média (SADIKU; MUSA, 2014).

$$I_{RMS} = \sqrt{\frac{1}{T} \int_0^T I^2(t) dt} \quad (5)$$

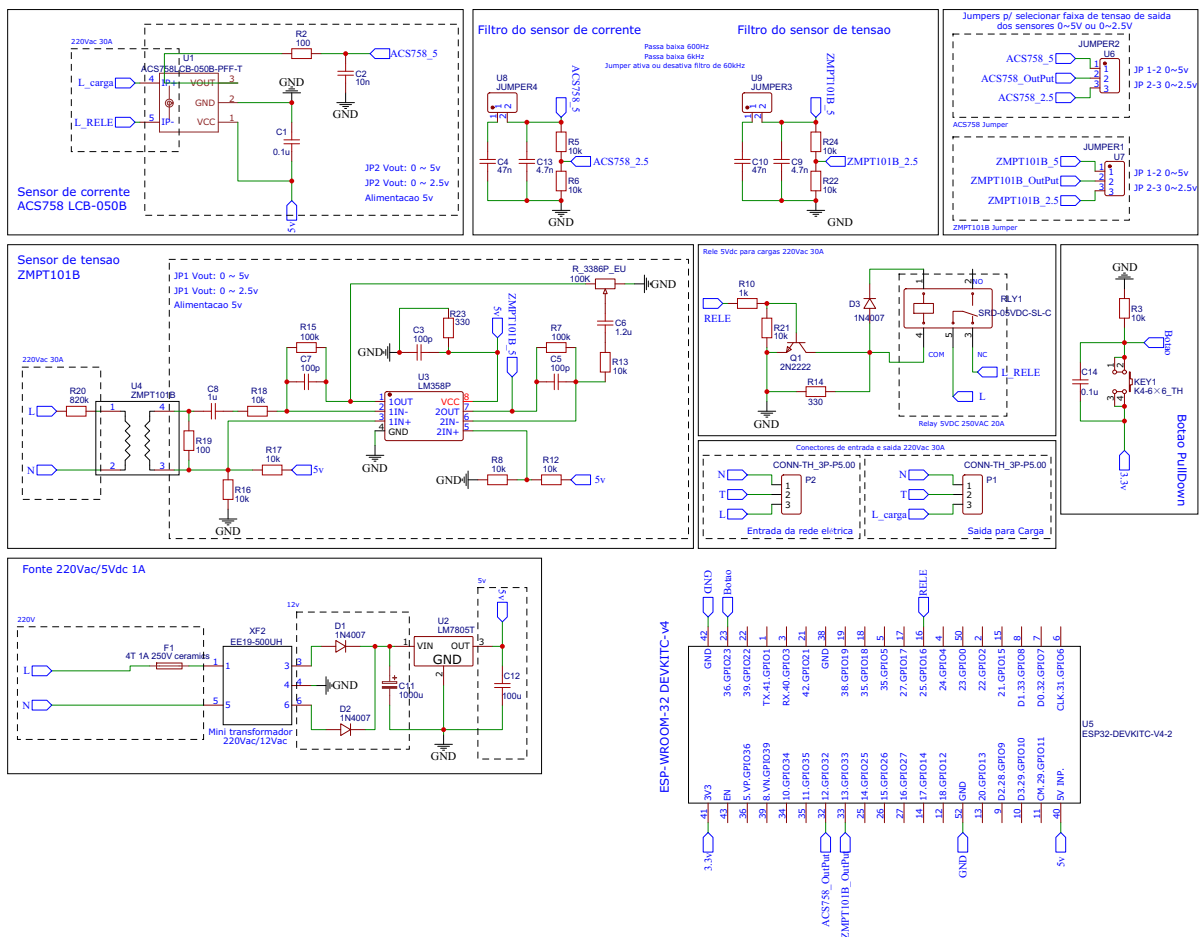
Portanto para o cálculo da potência média apresentado pelo dispositivo foi utilizado a equação 6, onde θ é o ângulo entre a tensão e corrente, e $\cos(\theta)$ é denominado fator de potência.

$$P = V_{RMS} I_{RMS} \cos(\theta) \quad (6)$$

3 ESTUDO E DESENVOLVIMENTO DO PROTÓTIPO

Um módulo de monitoramento pode ser aqui definido como um sistema embarcado conectado diretamente a Internet visando gerar informações em tempo real da carga acionada (AVCU, 2021). O sistema proposto é a integração entre hardware e software, onde o ESP32 recebe informações de um sensor de corrente ACS758, o qual transmite suas informações para um aplicativo de celular. Ainda ligado ao ESP32, tem-se um relé optoacoplador que interliga a parte de potência com a parte de controle do protótipo, para alimentar o sistema é utilizado uma fonte de 5 Vcc, a imagem 2 demonstra o circuito completo do módulo.

Figura 2 – Circuito completo do módulo



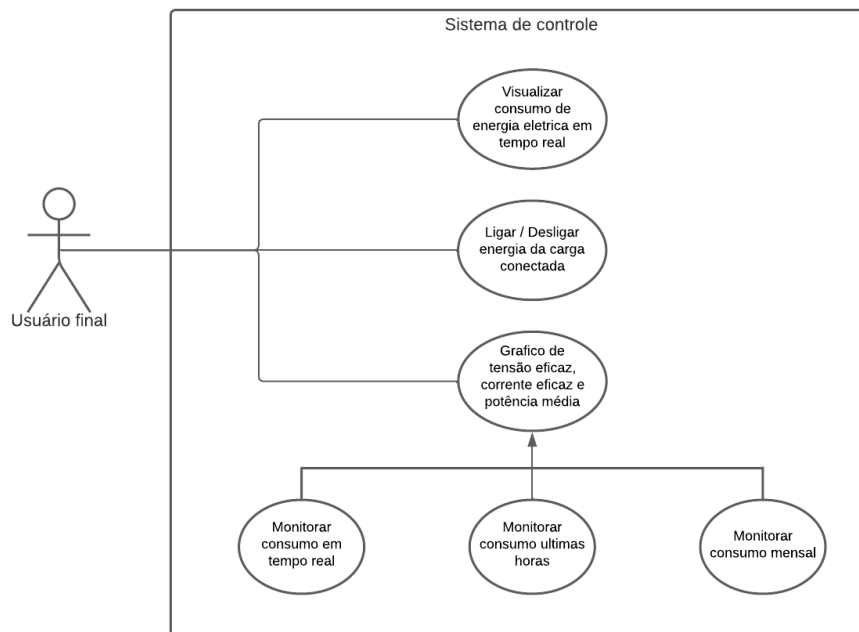
Fonte: Autoria própria.

Para o desenvolvimento do projeto foi realizado um levantamento de requisitos, dessa forma, essa seção analisa os requisitos necessários e a adequação da solução

implementada e as fundamentações teóricas necessárias.

- O sistema deve ser capaz de medir a potência elétrica média consumida pela carga conectada;
- O sistema deve operar em redes de tensões monofásicas 110/220 VAC;
- O usuário deve ser capaz de atuar (ON/OFF) sobre a energia fornecida para a carga;
- O sistema deve prover uma interface em que o usuário possa acessar seus dados de consumo utilizando a Internet a partir de um smartphone.

Figura 3 – Diagrama de casos de uso.



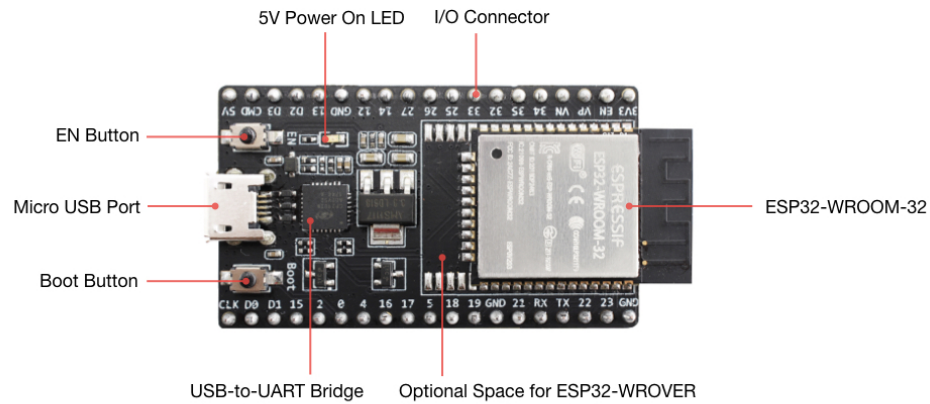
Fonte: Autoria própria.

Nas seções seguintes cada parte do circuito será detalhada.

3.1 MICROCONTROLADOR

O ESP32 DevKit4 (ESPRESSIF, 2022) é uma placa que junta o chip ESP32-WROOM-32 (chip de arquitetura 32 bits com Wi-Fi e Bluetooth de 2,4 GHz), interface USB (Universal Serial Bus) e regulador de tensão 3,3V. A programação pode ser feita usando o IDE (Integrated Development Environment) do Arduino e a comunicação via cabo micro-usb.

Figura 4 – ESP32 DevKit V4.

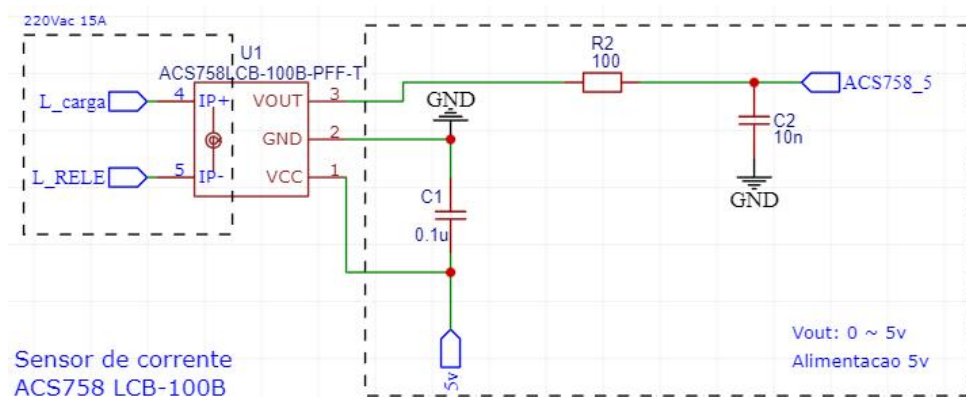


Fonte: Espressif (2022).

3.2 SENSOR DE CORRENTE

Para realizar a medição do sinal da corrente elétrica, é necessário disponibilizar os níveis da corrente que circulam no circuito através de sensores adequados, o sensor de corrente utilizado é o ACS758 100A que usa o efeito hall para detectar o campo magnético gerado pela passagem de corrente, gerando na saída (pino OUT) uma tensão proporcional ao valor da corrente e com uma curva de saída linear.

Figura 5 – Sensor Corrente.



Fonte: Autoria própria.

A tensão de saída no do pino V_{out} varia de 0v até V_{cc} e a tensão de saída sem carga é $\frac{V_{cc}}{2}$. O capacitor entre V_{cc} e GND e o filtro RC em V_{out} são recomendados pelo datasheet (ALLEGRO, 2022) do fabricante.

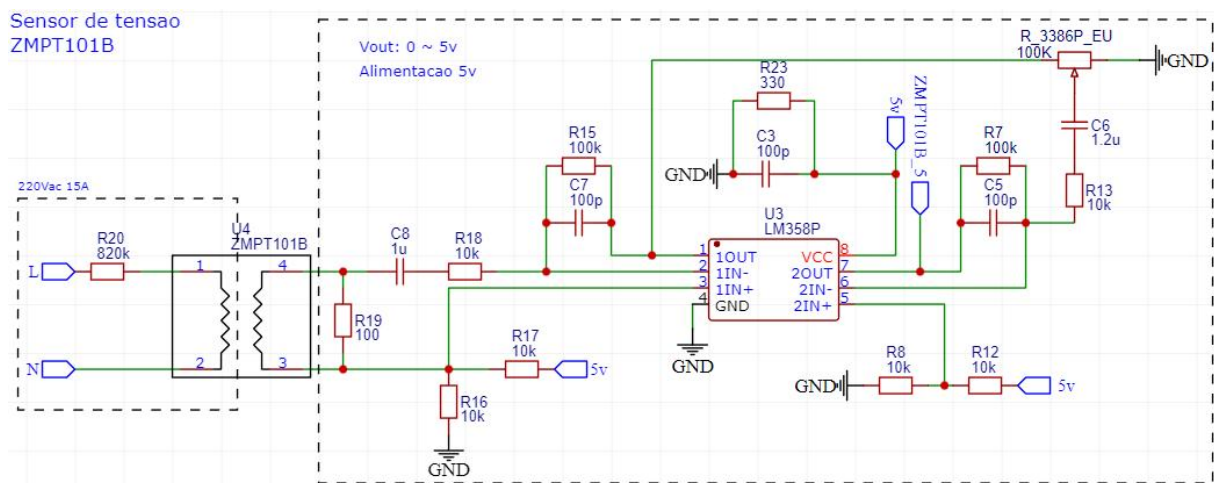
3.3 SENSOR DE TENSÃO

Para realizar o sensoriamento da tensão elétrica se faz necessária a utilização de dispositivos ou métodos para adequar os níveis de tensão que estão sendo medidos, pois, muitas vezes não é possível ser aplicada diretamente aos circuitos que realizam o processamento desta grandeza.

O sensor de tensão utilizado é um micro transformador de precisão ZMPT101B, sendo possível o monitoramento da tensão da rede AC de até 1000 volts, suporta até 4kV de tensão de ruptura, possui uma relação de voltagens de 1000:1000 e corrente máxima de 2mA:2mA. Para limitar a tensão de saída em uma faixa ideal, foi adicionado um resistor em série no primário do transformador e um resistor em paralelo no secundário, assim obtendo uma tensão e corrente na saída do transformador adequada para o ESP32.

Ao selecionar o resistor R_{20} que está em série com o primário do transformador deve ser levado em consideração que a corrente através do enrolamento não exceda 2mA pois a saída se torna não linear, de acordo com o datasheet (QINGXIAN, 2022) do fabricante uma corrente na faixa de 1mA deve ser adotada para uma melhor linearidade e precisão nas medidas.

Figura 6 – Sensor de Tensão.



Fonte: Autoria própria.

A tensão máxima de saída V_{Outmax} é limitada pela tensão de pico do ADC no circuito de amostragem do microcontrolador. Para este projeto estamos utilizando o microcontrolador ESP32 onde a tensão de pico do ADC é 1.1v, porém o módulo ESP32 Devkit4 (ESPRESSIF, 2022) já contém um divisor de tensão de modo que o ADC possa trabalhar com tensões até 3.3v. Portanto para tensão de 0 a 3.3V, a tensão RMS máxima do transformador é dada por:

$$V_{Outmax} = \frac{V_p}{2\sqrt{2}} = \frac{3.3V}{2\sqrt{2}} = 1,16V \quad (7)$$

Para determinar o resistor limitador de corrente, é utilizado equação 8, onde V_{in} é a tensão de entrada nominal e I é a corrente de operação nominal. Conforme datasheet (QINGXIAN, 2022) do fabricante o ZMPT101B tem uma faixa de melhor precisão com corrente nominal de 1mA, por esse motivo essa foi a faixa de corrente escolhida.

$$R_{20} = \frac{V_{in}}{I} = \frac{220}{0.001} = 220k\Omega * 4 = 880k\Omega \quad (8)$$

Seguindo as recomendações do fabricante para melhorar a confiabilidade, o resistor limitador de corrente selecionado é geralmente quatro vezes maior. Portanto adotamos 820k Ω que é um valor comercial de resistor próximo ao valor encontrado na equação 8, obtendo uma tensão de saída de 25mV aproximadamente. Ao determinar o resistor amostrador de corrente de saída é utilizado a equação 9.

$$R_{19} = \frac{V_{Outmax}}{I} = \frac{V_{Outmax}}{V_{in}} * R_{20} = \frac{0.025}{220} * 880k = 100\Omega \quad (9)$$

Onde V_{in} é a tensão de entrada nominal e I é a corrente de operação nominal, V_{Outmax} tensão máxima de saída e R_{20} resistor limitador na entrada.

3.4 FILTROS PASSA BAIXA E DIVISOR DE TENSÃO

Os filtros passivos passa-baixa são circuitos que permitem a passagem de sinais de baixa frequência e reduzem a intensidade de sinais de alta frequência, ou seja, a partir de uma frequência de referência ele permite que frequências mais baixas que ela passem livremente e frequências mais altas sejam atenuadas.

Os circuitos dos filtros passivos passa-baixa são compostos por uma resistência e um capacitor, que são conectados em série. O sinal de saída já filtrado fica em paralelo com o capacitor, assim como um divisor de tensão. Quando o capacitor é submetido à sinais alternados ele apresenta uma resistência, que é definida como reatância capacitiva, com a fórmula 10 da reatância capacitiva e considerando o valor da capacitância constante é o suficiente para entendermos o funcionamento do filtro passa-baixa (LATHI, 2006).

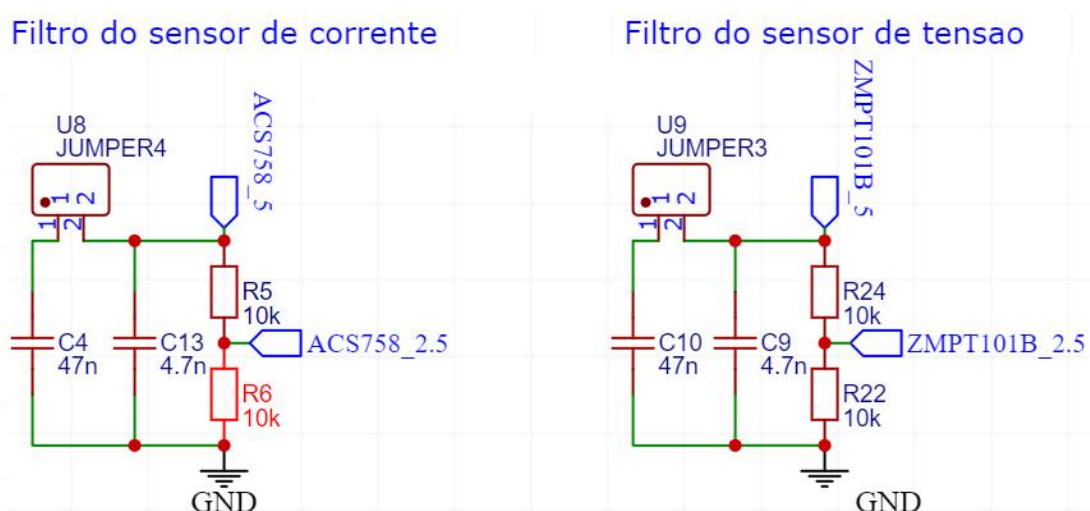
$$X_c = \frac{1}{2\pi FC} \quad (10)$$

Onde X_c é a reatância capacitiva, F a frequência e C a capacitância. Pela fórmula podemos observar que se o valor da frequência for baixo, a reatância capacitiva será alta e caso a frequência seja alta a sua reatância capacitiva será baixa. Ou seja, sinais de baixa frequência o capacitor tende a aumentar sua reatância capacitiva e bloqueá-lo, e sinais de alta frequência o capacitor tende a abaixar sua reatância capacitiva e permitir que o sinal passe. A fórmula 11 representa a frequência de corte do filtro passivo passa-baixa.

$$F_c = \frac{1}{2\pi RC} \quad (11)$$

De acordo com a análise feita pode se observar que apenas com o capacitor é possível cortar os sinais de maior frequência e deixar passar os de menor frequência, porém é necessário definir a frequência de corte do filtro e para isso é utilizado o resistor, sem ele não seria possível selecionar o maior valor de frequência permitido passar. A frequência de corte também é definida como a frequência em que o sinal sofre uma atenuação, sendo o valor dessa atenuação aproximadamente 30%, ou seja, o sinal de saída possui aproximadamente 70% da intensidade do sinal de entrada (LATHI, 2006).

Figura 7 – Filtro Passa baixas e Divisor de tensão.

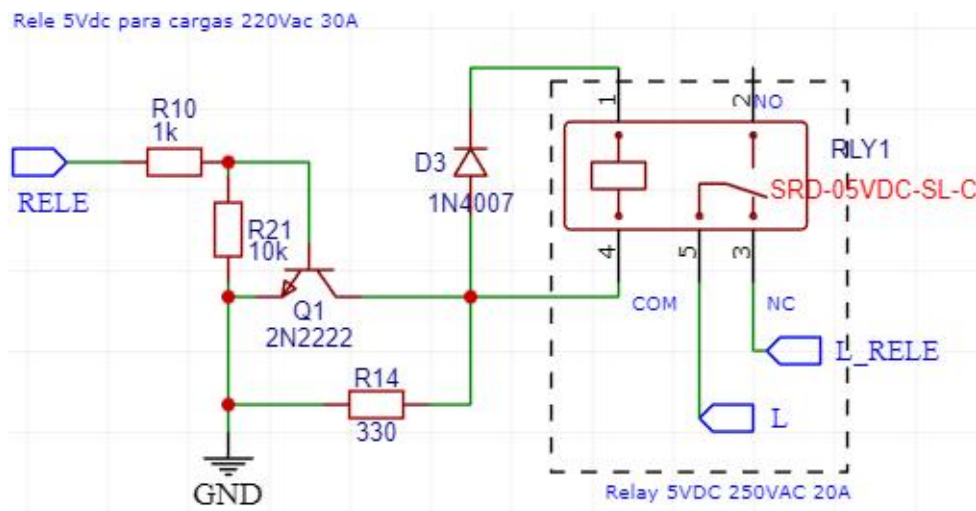


Fonte: Autoria própria.

3.5 RELÉ OPTOACOPLADOR

O relé é capaz de chavear vários dispositivos elétricos, como por exemplo, lâmpadas, tomadas, motores entre outros. Ele pode ser controlado por diversos tipos de microcontroladores, o controle desse modulo é feito a partir de um envio de sinal digital do microcontrolador, fazendo com que acione ou não o dispositivo acoplado ao relé, o relé utilizado nesse projeto é do tipo eletromecânico.

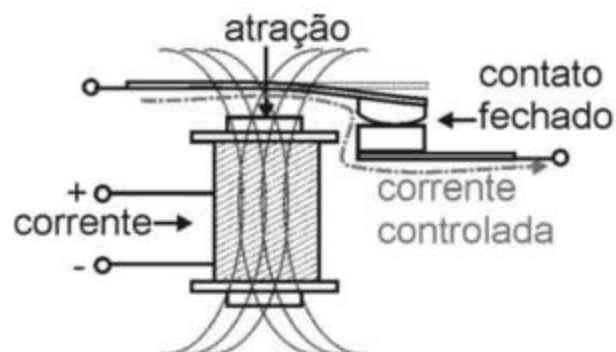
Figura 8 – Modulo Relé.



Fonte: Autoria própria.

O princípio de funcionamento do relé eletromecânico se dá a pôr efeitos eletromagnéticos, dentro dele possui um eletroímã e próximo a ele há uma armadura de metal, o eletroímã tem o intuito de controlar um jogo de contatos, quando há uma corrente percorrendo esse eletroímã, um campo magnético é criado, e isso faz com que haja uma atração magnética, ocorrendo a junção dos contatos e por consequência a corrente pode fluir através da armadura de metal (BRAGA, 2017).

Figura 9 – Princípio de funcionamento do relé.



Fonte: BRAGA (2017, p. 135).

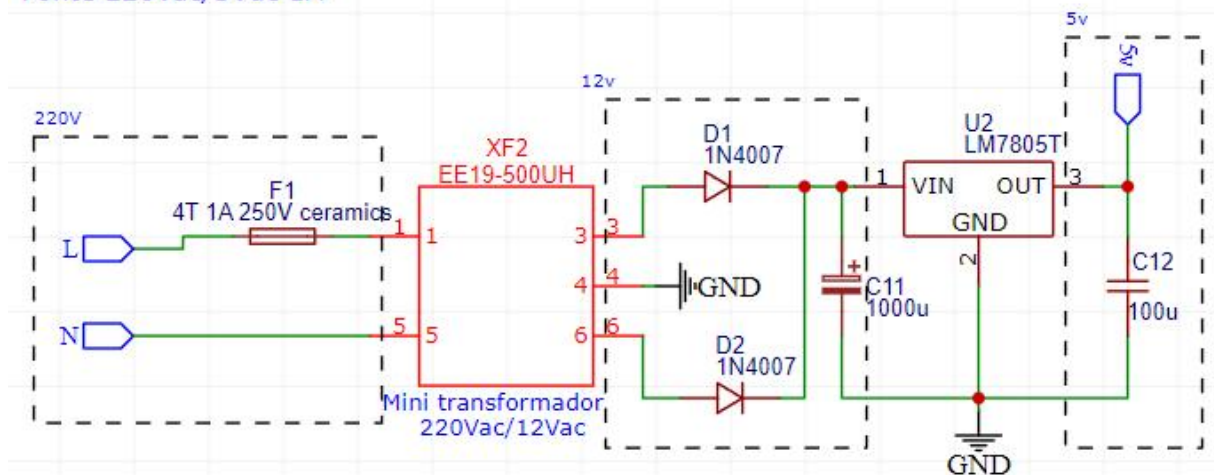
Quando a corrente que percorre o eletroímã é interrompida não há mais a atração magnética, por consequência abrindo o circuito, dessa maneira não haverá mais corrente fluindo através da armadura.

3.6 FONTE DE ALIMENTAÇÃO

Para alimentação do módulo foi projetado uma fonte AC-DC que é capaz de converter tensão alternada (AC) em tensão contínua (DC). A topologia escolhida foi um transformador com derivação central, uma ponte simples a diodo e um regulador de tensão, sem necessidade de utilizar componentes complexos visando o baixo custo, de forma a alimentar o módulo com tensão de 5VDC.

Figura 10 – Fonte 220Vac/5Vdc.

Fonte 220Vac/5Vdc 1A



Fonte: Autoria própria.

No projeto da fonte foi utilizado um transformador para baixar a tensão de 220VAC para 12VAC, essa tensão é então retificada com dois diodos de silício 1N4007 para um nível 10VDC e logo após regulada para 5VDC com o regulador de tensão LM7805, estabilizando e deixando a tensão com o nível necessário para a alimentação dos componentes. O capacitor eletrolítico posicionado antes do regulador de tensão tem a finalidade de reduzir o ripple na saída da ponte retificadora e o capacitor após o retificador tem a finalidade de e fazer o desacoplamento da fonte para carga.

3.7 PLACA DE CIRCUITO IMPRESSO

Para este projeto foi desenvolvido uma placa de circuito impresso utilizando a máquina prototipadora LPKF ProtoMat E34 do fabricante LPKF Laser Electronics (LPKF, 2022).

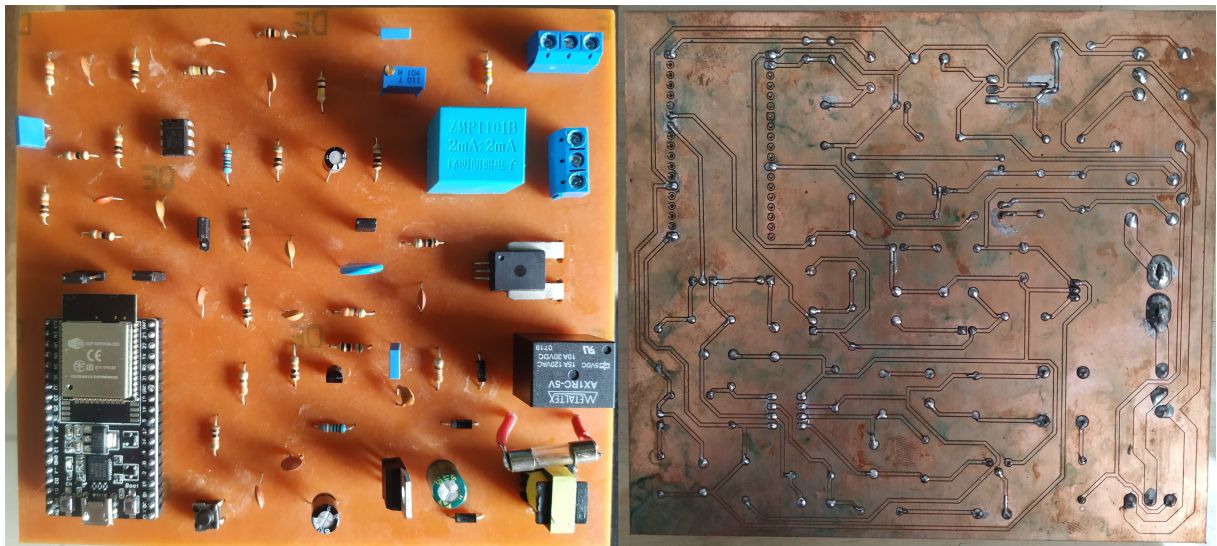
Figura 11 – Placa de circuito impresso.



Fonte: Autoria própria.

Após a confecção das trilhas na placa fenolite, foi soldado os componentes em seus respectivos lugares conforme demonstrado na figura 12.

Figura 12 – Placa de circuito impresso pronta.



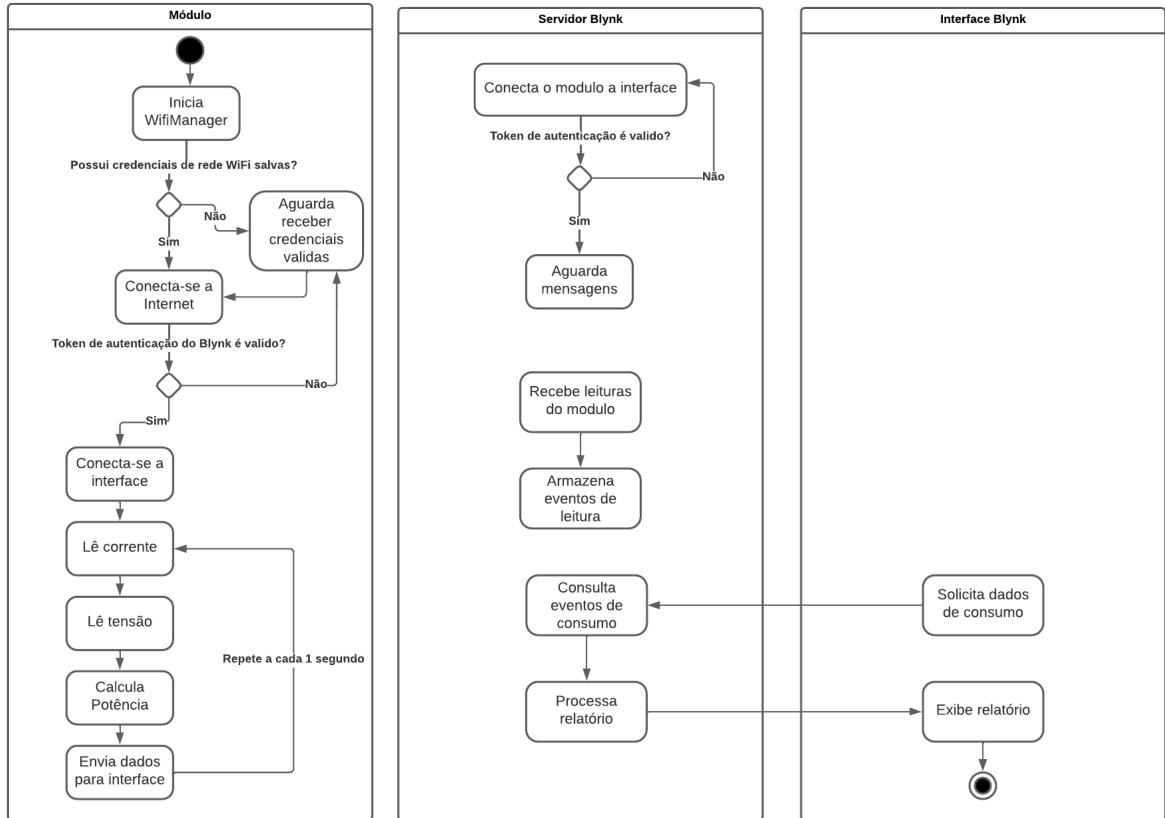
Fonte: Autoria própria.

3.8 INTERFACE

A interface para controle e monitoramento da carga é gerada utilizando o app gratuito Blynk (SHYMANSKY, 2022), onde podemos observar a corrente, tensão, potência e fazer o controle do relé para ligar ou desligar a energia fornecida para a

carga conectada ao módulo utilizando o botão da interface como demonstrado na figura 20, ainda podemos observar o gráfico da corrente, tensão e potência e definir a janela de tempo desejada para visualização das informações nos gráficos.

Figura 13 – Diagrama de atividades.



Fonte: Autoria própria.

4 RESULTADOS EXPERIMENTAIS

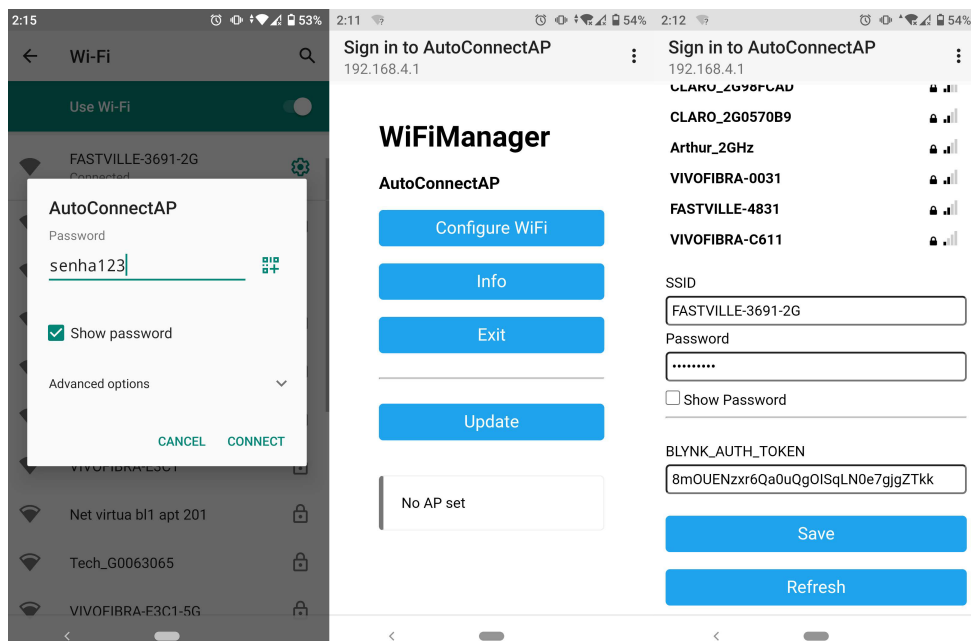
O trabalho objetiva proporcionar a ambientes residenciais a capacidade de medir e informar o consumo elétrico, a fim de atingir esse objetivo é proposto a criação de um módulo capaz de medir o consumo de cargas a ele conectadas.

4.1 PONTO DE ACESSO

Para podermos acessar e controlar o dispositivo localmente ou de maneira remota, conectamos o ESP32 a um roteador sem fio conectado a Internet. Para realizar o primeiro acesso e receber as credenciais da rede Wi-Fi que será conectado, é necessário definir o ESP32 como um ponto de acesso (AP), então se conectar ao ESP32 usando qualquer dispositivo com recursos Wi-Fi sem a necessidade de se conectar ao seu roteador, portando ao definir o ESP32 como um ponto de acesso, você cria sua própria rede Wi-Fi e dispositivos Wi-Fi próximos (estações) podem se conectar a ela (como seu smartphone ou computador).

Foi utilizada a biblioteca Open Source WiFiManager (TZAPU, 2022) para criar a rede e interface de configuração inicial e adaptada para incluir o código de autenticação necessário para configurar o app Blynk (SHYMANSKYY, 2022), utilizado para realizar o controle e monitoramento da carga acoplada.

Figura 14 – Etapas para configuração inicial.



Fonte: Autoria própria.

4.2 OFFSET DOS SENSORES

Ao realizar medições com os sensores, deve ser levado em consideração que sua saída mantém um offset de metade da tensão da alimentação, neste caso $\frac{5V}{2} = 2,5V$. Esses 2,5V passam em seguida por um divisor de tensão de resistores iguais, logo, o sinal de saída é novamente dividido por 2, portanto quando não houver carga conectada ao módulo há uma tensão de 1,25V aplicada ao ADC do ESP32.

Para determinar a tensão de offset, basta ler o valor do ADC quando não houver carga conectada ao sensor. A figura 15 mostra uma parte dos dados capturados e o valor médio obtido para o código de offset dos sensores.

Figura 15 – Offset dos Sensores.

Sem nenhuma CORRENTE AC aplicada Códigos de ADC do sensor de CORRENTE para o offset após o divisor		Sem nenhuma TENSÃO AC aplicada Códigos de ADC do sensor de TENSÃO para o offset após o divisor
1395	1395,393991	1396
1394	offset médio	1392
1395		1407
1395		1401
1395		1397
1395		1392
1397		1392
1398		1392
1395		1391
1395		1392
1396		1395
1395		1397
1394		1398
1393		1395
1395		1392
1395		1407
1396		1401
1394		1397

Fonte: Autoria própria.

Os valores de offset serão importantes para a conversão dos sinais lidos no ADC em valores de tensão e será utilizada como uma constante no código-fonte de captura, da mesma forma que o sensor de tensão, a saída do sensor de corrente também apresenta um offset de metade da tensão de alimentação, que por sua vez é aplicado a um divisor de tensão de resistores iguais, resultando em um código de offset do ADC do sensor de corrente, vale ressaltar que o valor obtido de offset do sensor de corrente é diferente do sensor de tensão devido a variações entre os componentes dos dois circuitos.

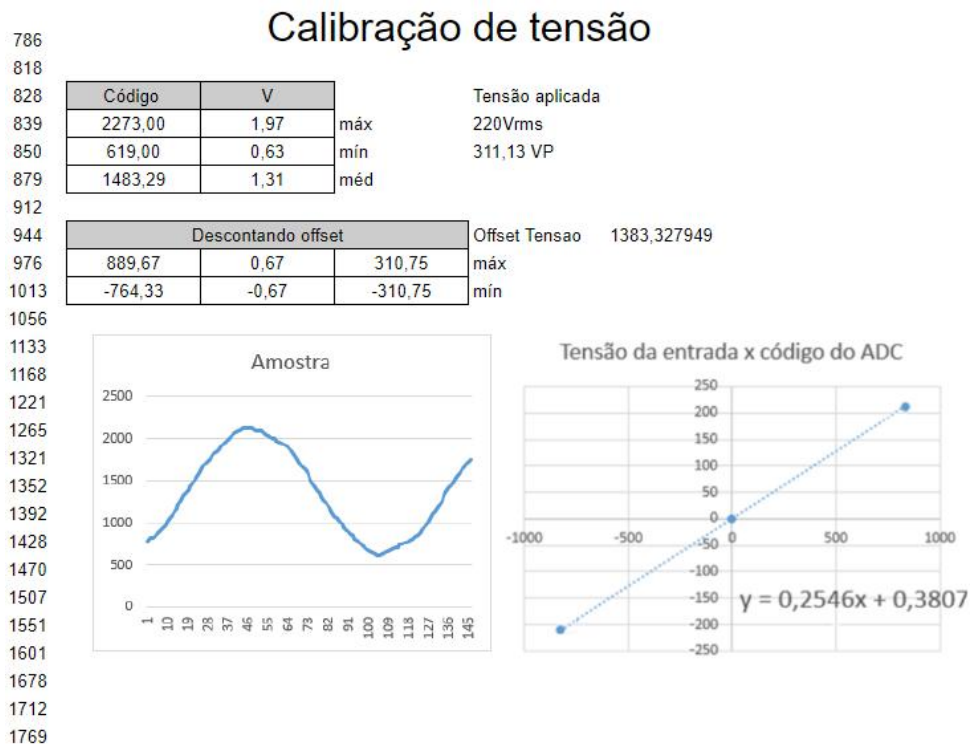
Uma vez determinado o offset, é aplicada uma tensão senoidal conhecida e através dela e comparado os códigos obtidos. Neste exemplo uma tensão senoidal

de 220Vrms é aplicada ao modulo. Capturando os códigos gerados e comparando os valores máximos e mínimos obtidos (descartando os picos causados por ruídos eventuais), obtém-se uma relação direta entre o código do ADC e os valores instantâneos, como mostrado na figura 16.

A função da reta de regressão é utilizada como função de conversão, seu coeficiente angular e linear (A e B) são utilizados no código fonte. O mesmo princípio pode ser aplicado ao sensor de corrente. Nele, é aplicado uma corrente senoidal de 66mArms, capturando os códigos gerados e comparando os valores máximos e mínimos obtidos (descartando os picos causados por ruídos eventuais), é obtido uma relação direta entre o código do ADC e os valores de corrente.

As retas de regressão usadas nas conversões foram obtidas usando apenas três pontos para simplificar a aplicação e por conhecermos a priori o comportamento e o grau de linearidade dos sensores de tensão e corrente e do restante do circuito.

Figura 16 – Conversão dos códigos do ADC em valores de tensão



Fonte: Autoria própria.

4.3 CALCULO DOS VALORES RMS

Para calcular o valor RMS (Root Mean Square), também conhecido como valor eficaz, para uma coleção de medidas discretas, podemos usar a expressão 12.

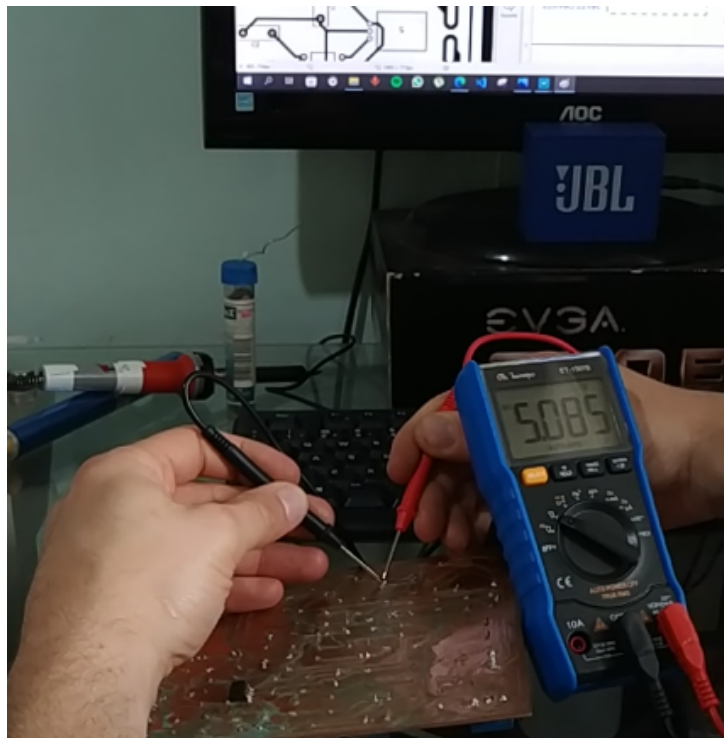
$$X_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N X_i^2} \quad (12)$$

Para realizar esta operação, primeiro é acumulado o valor do quadrado de cada medida, bastando para isso, multiplicar cada medida realizada por ela mesma e, em seguida, adicioná-la a uma variável acumuladora. Depois de todas as medidas serem realizadas, o resultado do acumulador é dividido pelo número de medidas efetuadas, a raiz quadrada é aplicada ao resultado da divisão, o resultado é então o valor RMS da coleção de medidas.

4.4 TESTES PRÁTICOS

Após finalizar a montagem do circuito, iniciou-se a fase de testes, o primeiro teste tem intuito de verificar o devido funcionamento do circuito de potência e garantir que o valor fornecido seja 5V, pois sabemos que ambos os sensores tem uma saída de $\frac{V_{cc}}{2}$, logo com oscilação da tensão fornecida há uma oscilação na tensão lida implicando em erros de leitura. Como é possível observar na imagem 17 a fonte de tensão projetada atende os requisitos do projeto e fornece 5v na saída do regulador de tensão LM7805.

Figura 17 – Tensão na saída do regulador LM7805.



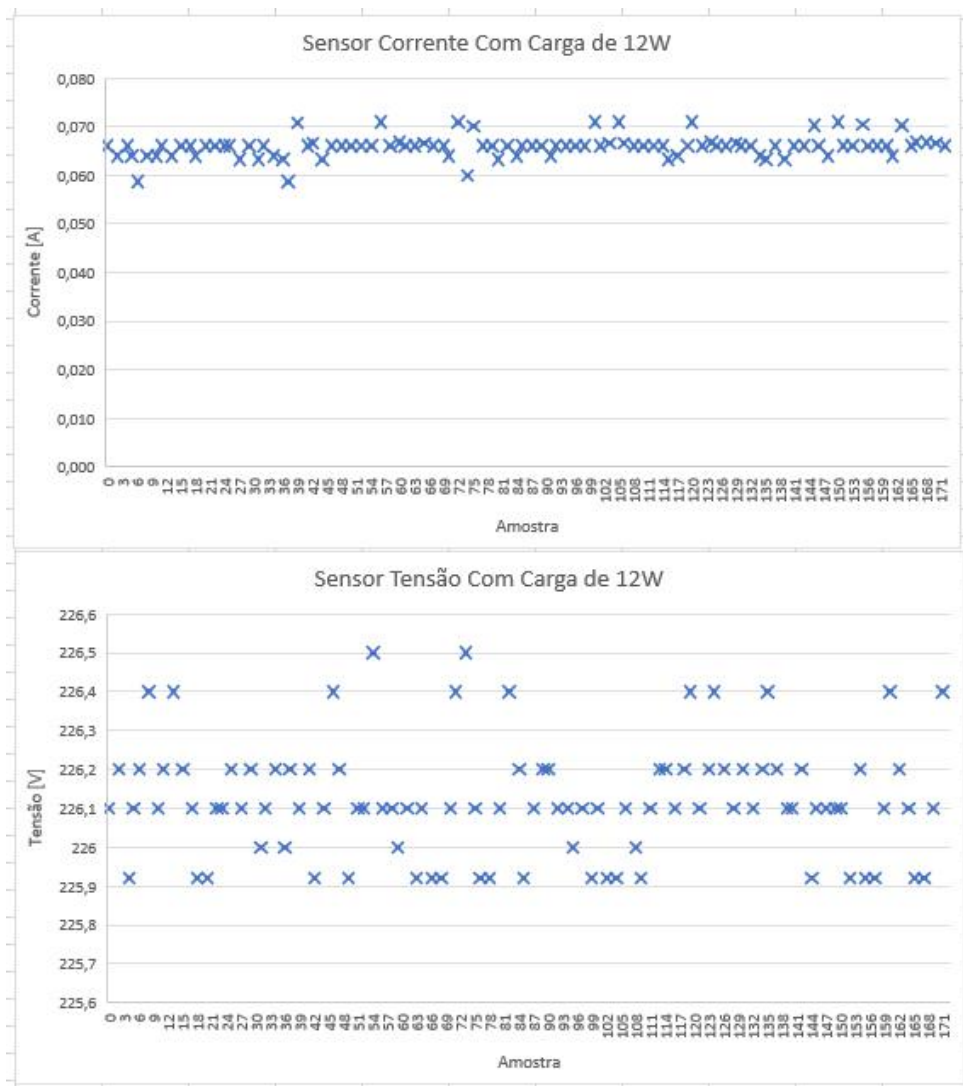
Fonte: Autoria própria.

Em seguida é verificado o sensor de corrente ACS758 e de tensão ZMPT101B, para isso foram efetuadas medição de tensão, corrente e potência e analisado o erro em relação ao multímetro Minipa ET-1507B (MINIPA, 2022), vale ressaltar que o mesmo tem uma incerteza de medição de 0,8% na faixa de tensão utilizada e 1,2% na faixa de corrente utilizada. O erro do modulo em relação ao multímetro foi calculado a partir da equação 13.

$$Erro = \frac{[CorrenteMultmetro - ACS758]}{CorrenteMultmetro} * 100 \quad (13)$$

O primeiro teste foi realizado utilizando uma carga resistiva de 12W como demonstrado na figura 18, a leitura da corrente se manteve em média 58mA com um erro percentual de 11,97% em relação a mesma carga medida com multímetro.

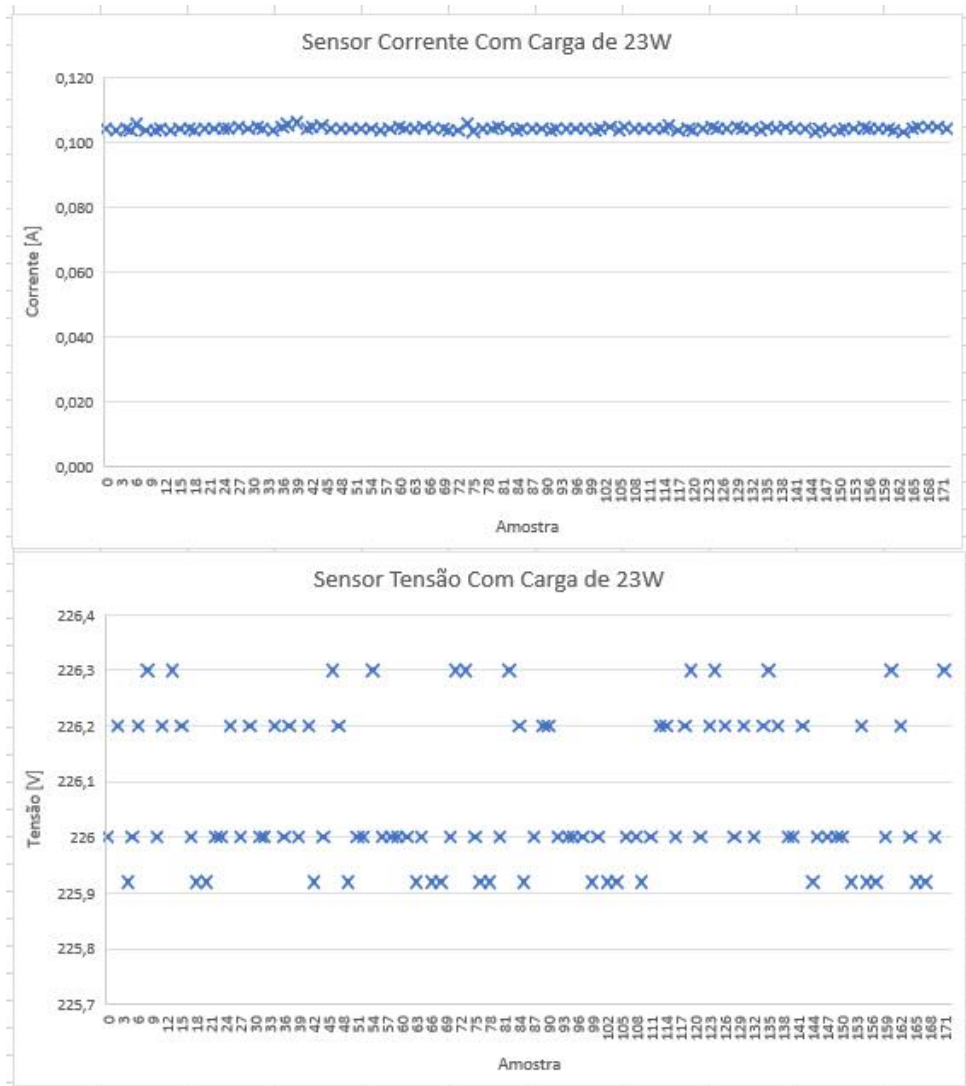
Figura 18 – Gráfico do sensoriamento de uma lâmpada de 12W.



Fonte: Autoria própria.

A carga foi substituída por uma carga de 23W, figura 19, para verificar a reprodutibilidade das medições, houve uma menor variação nas leituras da corrente, se mantendo em média 103mA com um erro percentual de 0,87% em relação ao multímetro.

Figura 19 – Gráfico do sensoriamento de uma lâmpada de 23W.



Fonte: Autoria própria.

O mesmo teste foi repetido com outros equipamentos, a tabela 1 mostra os valores obtidos da corrente para cada equipamento junto com seu erro percentual em relação ao multímetro.

Com base nos resultados encontrados os erros foram menores para correntes mais altas, exceto para o liquidificador, o fato se deve ao tipo da carga, pois o liquidificador é uma carga resistiva e indutiva portanto existe um atraso da corrente em relação a tensão e maiores ruídos ocasionando em erros na medição.

Tabela 1 – Potência obtida e erro percentual. (Autoria Própria)

Equipamento	Multímetro	ACS758	Erro Percentual
Lâmpada	0,058A	0,066A	11,97%
Ferro de solda	0,103A	0,104A	0,87%
Televisão	0,148A	0,1462A	1,42%
Monitor	0,113A	0,114A	0,80%
Computador	1,271A	1,263A	0,63%
Liquidificador	2,008A	2,105A	4,83%
Grill	2,309A	2,302A	0,31%

De modo geral os resultados foram satisfatórios pois os erros percentuais foram pequenos em relação ao multímetro utilizado, porém deve-se levar em consideração que sempre irá existir erros devido a medições incorretas, provenientes de falha humana ou até mesmo do multímetro Minipa ET-1507B (MINIPA, 2022) que foi utilizado como referência, pois o mesmo tem uma incerteza de medição de 0,8% na faixa de tensão utilizada e 1,2% na faixa de corrente utilizada conforme descrito no datasheet do fabricante.

Outro fator importante que deve ser levado em consideração é de que o sensor de corrente ACS758 possui um ruído de aproximadamente 10mA, para medições com corrente mais altas esse ruído pode ser desconsiderado, porém ao medir correntes baixas esse ruído é relevante, vale lembrar que medições em mA representam menos de 1% da faixa de medição do sensor ACS758 que é de 100A.

5 CONCLUSÃO

O projeto teve o objetivo de construir um aparelho capaz de monitorar a potência elétrica de um eletrodoméstico conectado ao módulo e permitir o controle (ON/OFF) da energia para a carga. De forma geral, o módulo atingiu os resultados esperados, apresentando informações importantes sobre a utilização da energia elétrica, com dados instantâneos e históricos de consumo, cumprindo os requisitos aqui estabelecidos, mas como todo projeto sempre existem alterações, inovações e incrementos que agregam mais características e evoluem o produto.

Um dos problemas do projeto foi que mesmo o sensor suportando 100A o relé utilizado suporta 10A e a placa de circuito impresso teve o dimensionamento das trilhas para 3A, limitando a corrente dos dispositivos conectados a 3A, uma solução é soldar um reforço de cobre sobre as trilhas ou utilizar uma placa dupla face com uma camada mais grossa de cobre, outro ponto foi o tamanho do protótipo, como sugestão de solução que pode ser empregada é a utilizar componentes SMD e placa dupla face reduzindo bastante seu tamanho e tornando viável sua utilização. Outro ponto passível de melhora é o seu preço final que não obteve uma resposta muito satisfatória, já que é bastante semelhante à de um smart plug comercial, porém existem maneiras de deixar o protótipo ainda mais barato, como por exemplo comprar os componentes dos circuitos e a placa de circuito impresso em lotes e com fornecedores internacionais.

A plataforma Blynk utilizada para interface foi uma excelente plataforma e o plano gratuito foi suficiente para as necessidades do projeto, um outro fator excelente dessa plataforma é de que o aplicativo é responsivo fazendo com que o seja acessível em diferentes tipos de telas, dessa forma o usuário tem inúmeras maneiras de controlar seu projeto. A plataforma permite acessar os dados em tempo real em horas, dias ou meses, o que é interessante para visualizar o perfil de consumo.

Como sugestão para trabalhos futuros, recomenda-se adicionar novas funções ao protótipo, como por exemplo agendamentos para ligar ou desligar a carga conectada. Outra sugestão é utilizar o ESP32 wroom ao invés da placa de desenvolvimento pelo menor custo associado. E por fim uma última sugestão para futuras melhorias é reduzir o consumo do módulo implementando o desligamento automático, sendo uma maneira bastante inteligente para reduzir o custo de energia elétrica quando não tem carga conectada ao módulo, visto que uma das maneiras mais eficiente para reduzir o consumo é eliminar totalmente o gasto proveniente da função standby.

REFERÊNCIAS

AL-ALI, A.-R. et al. An iot-based smart utility meter. **2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC)**, p. 80–83, 2018.

ALLEGRO, M. **Current Sensor ACS758**. Allegro Microsystems, 2022. Acessado em: 13/12/2022. Disponível em: <https://www.allegromicro.com/-/media/files/datasheets/acs758-datasheet.pdf>.

AMRI, Y.; SETIAWAN, M. A. Improving smart home concept with the internet of things concept using RaspberryPi and NodeMCU. In: **Proceedings of the IOP Publishing**. IOP Publishing, 2018. (Materials Science and Engineering, V.325), p. 5–7. Acessado em: 07/03/2022. Disponível em: <https://doi.org/10.1088/1757-899x/325/1/012021>.

ASHTON, K. That 'internet of things' thing. **RFID Journal**, V.22, n. 7, june 2009.

AVCU, M. **Arduino measure any ac current with ACS sensor**. [S.l.]: Arduino Instructor, 2021.

BRAGA, N. **Relés: Circuitos e aplicações**. Editora NCB, 2017. Disponível em: <https://books.google.com.br/books?id=DXAEDgAAQBAJ>.

DORF, R.; SVOBODA, J. **Introdução aos circuitos elétricos 9ª edição**. Grupo Gen - LTC, 2016. Disponível em: <https://books.google.com.br/books?id=niwPtAEACAAJ>.

EHRHARDT-MARTINEZ, K. et al. **Advanced metering initiatives and residential feedback programs: a meta-review for household electricity-saving opportunities**. American Council for an Energy-Efficient Economy (ACEEE), 2010. Disponível em: <https://books.google.com.br/books?id=11LRjwEACAAJ>.

ESPRESSIF. **ESP32 DevKit V4**. Espressif, 2022. Acessado em: 13/12/2022. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>.

HAYT, W.; KEMMERLY, J.; DURBIN, S. **Análise de Circuitos em Engenharia - 8.ed.** AMGH Editora, 2014. ISBN 9788580553840. Disponível em: <https://books.google.com.br/books?id=L7kLBAAQBAJ>.

HLAING, W. et al. Implementation of wifi-based single phase smart meter for internet of things (iot). In: **2017 International Electrical Engineering Congress (iEECON)**. [S.l.: s.n.], 2017. p. 1–4.

LATHI, B. **Sinais e Sistemas Lineares - 2.ed.** Bookman, 2006. Acessado em: 13/12/2022. ISBN 9788577803910. Disponível em: <https://books.google.com.br/books?id=S806DQAAQBAJ>.

LPKF, L. E. **ProtoMat E34**. LPKF, 2022. Acessado em: 13/12/2022. Disponível em: <https://www.lpkf.com/en/industries-technologies/research-in-house-pcb-prototyping/products/lpkf-protomat-e44>.

MASTERS, G. **Renewable and Efficient Electric Power Systems**. Wiley, 2005. (IEEE Press). ISBN 9780471668831. Disponível em: <https://books.google.com.br/books?id=dZG7EFaOPSMC>.

MINIPA. **Multímetro Minipa ET-1507B**. [s.n.], 2022. Disponível em: <https://www.minipa.com.br/images/Manual/ET-1507B-1109-BR.PDF>.

QINGXIAN, Z. L. E. **MICRO PRECISION VOLTAGE TRANSFORMERS: Zmpt101b**. Innovators Guru, 2022. Acessado em: 13/12/2022. Disponível em: <https://innovatorsguru.com/wp-content/uploads/2019/02/ZMPT101B.pdf>.

MATTHEW N. O. SADIKU, C.; MUSA, S. **Análise de Circuitos Elétricos com Aplicações**. AMGH Editora, 2014. ISBN 9788580553031. Disponível em: <https://books.google.com.br/books?id=d2Q3AgAAQBAJ>.

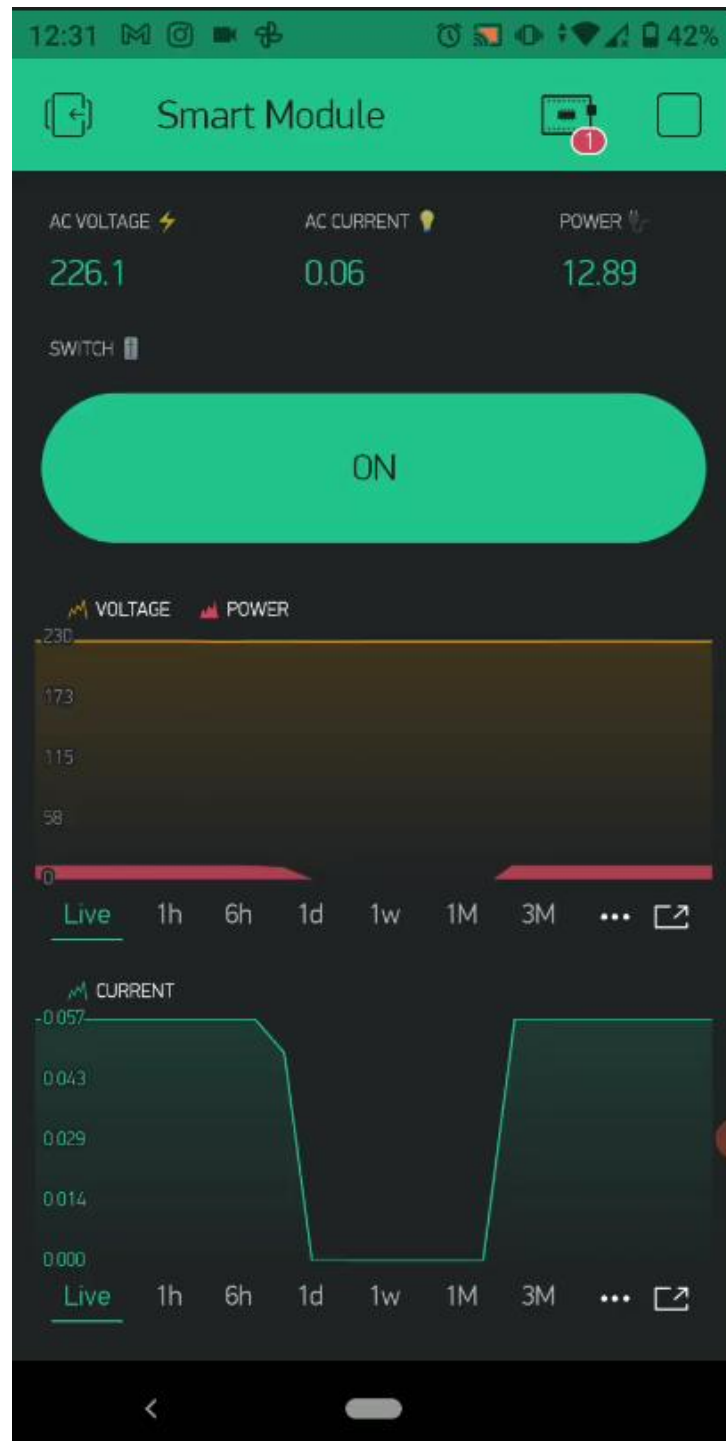
SHYMANSKYY, V. **Blynk smartphone app**. [s.n.], 2022. Disponível em: <https://www.arduino-libraries.info/libraries/blynk>.

TIWARI, U. K.; MATTA, P. Efficient smart-home architecture: An application of internet of things. **Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019**, 2019.

TZAPU. **WiFi Configuration manager**. [s.n.], 2022. Disponível em: <https://www.arduino-libraries.info/libraries/wi-fi-manager>.

APÊNDICE A

Figura 20 – Teste utilizando uma lâmpada 12W.



Fonte: Autoria própria.

```

1 #include <FS.h>
2 #include <ArduinoOTA.h>
3 #include <DNSServer.h> // Local DNS Server used for redirecting all
   requests to the config
4 #include <Ethernet.h>
5 #include <SPI.h>
6 #include <WiFi.h>
7 #include <WiFiClient.h>
8 #include <BlynkSimpleEsp32.h>
9 #include <ArduinoJson.h> // Gerencia a gravação de dados em arquivo.
   json
10 #include "settings.h"
11
12 /*      MODO de operação:
13         0 - Transmite os valores brutos dos ADCs
14         1 - Transmite os valores instantâneos de tensão e corrente, em
   volts e amperes
15         2 - Transmite os valores RMS de tensão e corrente, em volts e
   amperes
16 */
17 #define MODO 2
18
19 //Pinos de entrada utilizados
20 const uint8_t pinAdcTensao = 32;
21 const uint8_t pinAdcCorrente = 33;
22
23 //Pinos de entrada utilizados
24 const uint8_t pinAdcTensao = 32;
25 const uint8_t pinAdcCorrente = 33;
26
27 //Caracter separador dos dados
28 const char separador = '\\t';
29
30 //valores de offset de cada sensor (determinados estatisticamente
   através de capturas)
31 const float offsetVAC = 1383.328;
32 const float offsetIAC = 1395.393991;
33
34 // Flag para salvar os dados
35 bool precisaSalvar = false;
36
37 // Função de retorno para notificar sobre a necessidade de salvar as
   configurações
38 void precisaSalvarCallback()
39 {
40     imprimirSerial(true, "As configurações tem que ser salvas.");
41     precisaSalvar = true;

```

```
42 }
43
44 // Função para imprimir na porta serial
45 void imprimirSerial(bool linha, String mensagem)
46 {
47 #ifndef DEBUG
48     if (linha)
49     {
50         Serial.println(mensagem);
51     }
52     else
53     {
54         Serial.print(mensagem);
55     }
56 #endif
57 }
58
59 BlynkTimer timer;
60 double Vrms = 0, Irms = 0, Power = 0;
61
62 //Função para atualizar leituras
63 void updateBlynk() {
64     Blynk.virtualWrite(vPIN_VOLTAGE, Vrms);
65     Blynk.virtualWrite(vPIN_CURRENT_USAGE, Irms);
66     Blynk.virtualWrite(vPIN_ACTIVE_POWER, Power);
67 }
68
69 // Callback que indica que o ESP entrou no modo AP
70 void configModeCallback (WiFiManager *myWiFiManager) {
71     Serial.println("Entrou no modo de configuração");
72     Serial.println(WiFi.softAPIP()); //imprime o IP do AP
73     Serial.println(myWiFiManager->getConfigPortalSSID()); //imprime o SSID
74     criado da rede
75 }
76
77 // Callback que indica que salvamos uma nova rede para se conectar (modo
78     esta o)
79 void saveConfigCallback () {
80     Serial.println("Configuração salva");
81 }
82
83 void setup() {
84     Serial.begin(115200);
85
86     //Define as funções dos pinos
87     pinMode(pinAdcTensao, INPUT);
```

```

87 pinMode(pinAdcCorrente, INPUT);
88 pinMode(PIN_RELAY, OUTPUT);
89
90 //Formatando a memoria interna
91 //(descomente a linha abaixo enquanto estiver testando e comente ou
   apague quando estiver pronto)
92 //SPIFFS.format();
93
94 // Iniciando o SPIFSS (SPI Flash File System) para salvar o AUTH TOKEN
   do Blynk na memoria flash
95 imprimirSerial(true, "Iniciando o SPIFSS (SPI Flash File System)");
96 if (SPIFFS.begin())
97 {
98     imprimirSerial(true, "Sistema de arquivos SPIFSS montado!");
99     if (SPIFFS.exists("/config.json"))
100    {
101        // Arquivo de configura o existe e ser lido.
102        imprimirSerial(true, "Abrindo o arquivo de configuracao...");
103        File configFile = SPIFFS.open("/config.json", "r");
104        if (configFile)
105        {
106            imprimirSerial(true, "Arquivo de configuracao aberto.");
107            size_t size = configFile.size();
108
109            // Alocando um buffer para armazenar o conte do do arquivo.
110            std::unique_ptr<char[]> buf(new char[size]);
111
112            configFile.readBytes(buf.get(), size);
113            DynamicJsonBuffer jsonBuffer;
114            JsonObject &json = jsonBuffer.parseObject(buf.get());
115            json.printTo(Serial);
116            if (json.success())
117            {
118                // Copiando as vari veis salvas previamente no aquivo json
   para a memoria do ESP.
119                imprimirSerial(true, "arquivo json analisado.");
120                strcpy(AUTH, json["AUTH"]);
121            }
122            else
123            {
124                imprimirSerial(true, "Falha ao ler as configuracoes do arquivo
   json.");
125            }
126        }
127    }
128 }
129 else

```

```

130 {
131     imprimirSerial(true, "Falha ao montar o sistema de arquivos SPIFFS."
132     );
133 }
134 // Fim da leitura do sistema de arquivos SPIFFS
135 // Par metros extras para configura o
136 // Depois de conectar, parameter.getValue() vai pegar o valor
137 // configurado.
138 // Os campos do WiFiManagerParameter s o: id do par metro , nome ,
139 // valor padr o , comprimento
140 WiFiManagerParameter AUTH_TOKEN("AUTH", "BLYNK_AUTH_TOKEN", AUTH, 35);
141
142 // Inicializa o do WiFiManager. Uma vez iniciado n o
143 // necess rio mant -lo em mem ria .
144 WiFiManager wifiManager;
145
146 // Para resetar as configura es da WiFi descomentar essa linha
147 wifiManager.resetSettings();
148
149 // Callback para informar a necessidade de salvar as configura es
150 wifiManager.setSaveConfigCallback(precisaSalvarCallback);
151 // Callback para quando entra em modo de configura o AP
152 wifiManager.setAPCallback(configModeCallback);
153 // Callback para quando se conecta em uma rede, ou seja, quando passa
154 // a trabalhar em modo esta o
155 wifiManager.setSaveConfigCallback(saveConfigCallback);
156
157 // Adicionando os par metros para conectar ao servidor Blynk
158 wifiManager.addParameter(&AUTH_TOKEN);
159
160 // Busca o ID e senha da rede wifi e tenta conectar.
161 // Caso n o consiga conectar ou n o exista ID e senha,
162 // cria um access point com o nome "AutoConnectAP" e a senha "senha123
163 "
164 // E entra em loop aguardando a configura o de uma rede WiFi
165 // v lida .
166 if (!wifiManager.autoConnect("AutoConnectAP", "senha123"))
167 {
168     imprimirSerial(true, "Falha ao conectar. Excedeu o tempo limite para
169     conexao.");
170     delay(3000);
171     // Reinicia o ESP e tenta novamente ou entra em sono profundo (
172     DeepSleep)
173     ESP.reset();
174     delay(5000);
175 }

```

```
168
169 // Se chegou at aqui porque conectou na WiFi!
170 imprimirSerial(true, "Conectado!! :)");
171
172 //DEBUG
173 //imprimirSerial(true, AUTH);
174
175 // Lendo os par metros atualizados
176 strcpy(AUTH, AUTH_TOKEN.getValue());
177
178 //DEBUG
179 //imprimirSerial(true, AUTH);
180
181 // Salvando os par metros informados na tela web do WiFiManager
182 if (precisaSalvar)
183 {
184     imprimirSerial(true, "Salvando as configuracoes");
185     DynamicJsonBuffer jsonBuffer;
186     JsonObject &json = jsonBuffer.createObject();
187     json["AUTH"] = AUTH;
188
189     File configFile = SPIFFS.open("/config.json", "w");
190     if (!configFile)
191     {
192         imprimirSerial(true, "Houve uma falha ao abrir o arquivo de
193         configuracao para incluir/alterar as configuracoes.");
194     }
195
196     json.printTo(Serial);
197     json.printTo(configFile);
198     configFile.close();
199 }
200 // Converte a string de credenciais da internet para char para ser
201 // utilizada pelo Blynk
202 String ssid_str;
203 String pass_str;
204 ssid_str = wifiManager.getWiFiSSID();
205 pass_str = wifiManager.getWiFiPass();
206 ssid_str.toCharArray(ssid, 50);
207 //DEBUG
208 imprimirSerial(true, ssid);
209 pass_str.toCharArray(pass, 50);
210 //DEBUG
211 imprimirSerial(true, pass);
212
```

```

213 WiFi.hostname(myhostname);
214 WiFi.mode(WIFI_STA);
215 Blynk.begin(AUTH, ssid, pass);
216 while (Blynk.connect() == false) {}
217 ArduinoOTA.setHostname(OTA_HOSTNAME);
218 ArduinoOTA.begin();
219
220 //Inicia timer para chamar a fun o updateBlynk e atualizar as
    leituras
221 timer.setInterval(1000L, updateBlynk);
222 }
223
224 void loop() {
225   Blynk.run();
226   timer.run();
227   ArduinoOTA.handle();
228
229 #if MODO == 0
230   //Envio direto dos c digos do ADC sem nenhum tratamento
231
232   //Vari veis
233   int32_t tensao = 0;
234   int32_t corrente = 0;
235
236   //Captura
237   tensao = analogRead(pinAdcTensao);
238   corrente = analogRead(pinAdcCorrente);
239
240   //Transmiss o
241   Serial.print(tensao);
242   Serial.print(separador);
243   Serial.print(corrente);
244   Serial.println();
245 #endif
246
247 #if MODO == 1
248   //Envio dos c digos convertidos em tens o e corrente instat neos
249
250   //coeficientes das regress es lineares. Usados para a convers o dos
    c digos
251   const float coefA_VAC = 0.2546;
252   const float coefB_VAC = 0.3807;
253   const float coefA_IAC = 0.117;
254   const float coefB_IAC = 0.1299;
255
256   //Vari veis
257   int32_t tensao = 0;

```



```

258 int32_t corrente = 0;
259
260 //Captura
261 tensao = analogRead(pinAdcTensao);
262 corrente = analogRead(pinAdcCorrente);
263
264 //Transmiss o
265 Serial.print((float(tensao) - offsetVAC)*coefA_VAC + coefB_VAC);
266 Serial.print(separador);
267 Serial.print((float(corrente) - offsetIAC)*coefA_IAC + coefB_IAC);
268 Serial.println();
269 #endif
270
271 #if MODO == 2
272 //Envio dos c digos convertidos da tens o e corrente RMS
273
274 //coeficientes das regress es lineares. Usados para a convers o dos
    c digos
275 const float coefA_VAC = 0.2652;
276 const float coefB_VAC = 0.0;
277 const float coefA_IAC = 0.1191;
278 const float coefB_IAC = 0.0;
279
280 //Per odo m nimo de captura de 16,7ms para garantir que ao menos um
    ciclo seja capturado
281 const int32_t duracaoDaAmostragem = (int32_t)((1000.0 / 60.0) * 6.0);
    //6 ciclos de 1000ms/60 (60Hz)
282
283 //Vari veis
284 int32_t tensao = 0;
285 int32_t corrente = 0;
286 uint32_t qtdDeAmostras = 0;
287 float acumuladorTensao = 0.0;
288 float acumuladorCorrente = 0.0;
289
290 //Captura
291 unsigned long inicio = millis();
292 do
293 {
294     tensao = analogRead(pinAdcTensao) - offsetVAC;
295     corrente = analogRead(pinAdcCorrente) - offsetIAC;
296
297     acumuladorTensao = acumuladorTensao + (tensao * tensao); //Soma dos
        quadrados
298     acumuladorCorrente = acumuladorCorrente + (corrente * corrente);//
        Soma dos quadrados
299

```

```

300     qtdDeAmostras++;
301 } while ((millis() - inicio) < duracaoDaAmostragem);
302
303 //C lculo dos valores RMS
304 acumuladorTensao = acumuladorTensao / float(qtdDeAmostras); //M dia
    dos quadrados
305 acumuladorCorrente = acumuladorCorrente / float(qtdDeAmostras); //
    M dia dos quadrados
306
307 acumuladorTensao = sqrt(acumuladorTensao); //RMS da captura
308 acumuladorCorrente = sqrt(acumuladorCorrente); //RMS da captura
309
310 acumuladorTensao = acumuladorTensao * coefA_VAC + coefB_VAC; //
    Convers o em volts
311 acumuladorCorrente = acumuladorCorrente * coefA_IAC + coefB_IAC; //
    Convers o em amp res
312
313 //Transmiss o
314 Serial.print(acumuladorTensao);
315 Serial.print(separador);
316 Serial.print(acumuladorCorrente);
317 Serial.println();
318 #endif
319
320 #ifndef DEBUG
321 Serial.print("Vrms:      "); Serial.print(acumuladorTensao); Serial.
    println(" V"); // V
322 Serial.print("Irms:      "); Serial.print(acumuladorCorrente, 3);
    Serial.println(" A"); // A
323 Serial.print("Power:     "); Serial.print(Power); Serial.println(" W");
    // W
324 Serial.println("=====");
    ;
325 #endif
326 }

```

```

1 //Over The Air Hostname
2 #define OTA_HOSTNAME                "Smart Module"
3
4 //WiFi data
5 char ssid[] = "FASTVILLE-3691-2G"; //WiFi Credential
6 char pass[] = "ABCDEFGHI"; //WiFi Password
7 char server[] = "blynk.cloud"; //Blynk local server
    IP address
8 int port = 8080; //Blynk local port
9 #define AUTH "8m0UENzxr6QasFqcAAqerN0e7gjjgZTkk" //Auth code for
    Blynk Local Server

```

```
10
11 //Virtual Pins - Base
12 #define vPIN_VOLTAGE          V0
13 #define vPIN_CURRENT_USAGE    V1
14 #define vPIN_ACTIVE_POWER     V2
15
16 //Relay pin
17 #define PIN_RELAY 16
18
19 //Debug. Change to 0 when you are finished debugging.
20 const int DEBUG              =          1;
```