

UNIVERSIDADE FEDERAL DO PARANÁ  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**DESENVOLVIMENTO DE UM RELÉ DE PROTEÇÃO PARA SISTEMAS  
ELÉTRICOS DE POTÊNCIA UTILIZANDO MICROCONTROLADOR  
COMERCIAL**

**GUSTAVO LAROCA MICHELI  
LUCAS FILUS RAMOS**

**CURITIBA**

**2024**

**DESENVOLVIMENTO DE UM RELÉ DE PROTEÇÃO PARA SISTEMAS  
ELÉTRICOS DE POTÊNCIA UTILIZANDO MICROCONTROLADOR  
COMERCIAL**

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica.

Área de conhecimento: Engenharia Elétrica

Orientador: Prof. Dr. Mateus Duarte Teixeira

CURITIBA

2024

*Dedicamos este trabalho aos nossos pais, por todo o tempo dedicado à educação, pelos conselhos sempre sábios que nos guiaram e, acima de tudo, pelos valores e princípios que nos formaram e nos trouxeram até aqui. Sem vocês, nada disso seria possível.*

*Agradecemos também aos nossos professores, por compartilharem seu conhecimento e nos orientarem durante a jornada acadêmica.*

*E aos amigos, companheiros de tantas piadas e momentos inesquecíveis, que tornaram essa caminhada memorável.*

*Com reverências, Gustavo e Lucas.*

“O mundo se desvia para deixar passar qualquer homem que sabe aonde vai”.

Epicteto

## RESUMO

A proteção de sistemas elétricos de potência é essencial para garantir a operação segura e contínua de redes de transmissão e distribuição de energia. Este trabalho apresenta o desenvolvimento de um protótipo de relé de proteção utilizando o microcontrolador ESP32, com foco em funções de monitoramento e proteção contra anomalias de corrente e tensão, como subtensão, sobretensão, sobrecorrente instantânea e sobrecorrente temporizada. O projeto incorpora sensores comerciais ACS712 e ZMPT101B, integrados ao ESP32, e utiliza a plataforma Arduino IDE para programação. O desenvolvimento envolveu a calibração de sensores para garantir a precisão das medições, a implementação de algoritmos baseados em normas técnicas e a realização de ensaios laboratoriais para validação. O sistema demonstrou um desempenho confiável, com detecção eficiente de condições anômalas e tempos de resposta compatíveis com aplicações práticas. Além disso, o protótipo mostrou-se uma solução de baixo custo, acessível para fins educacionais, e com potencial de expansão, incluindo funcionalidades de teleproteção e integração com sistemas inteligentes de controle. Por fim, o trabalho contribui para a área de proteção elétrica ao oferecer uma alternativa acessível para estudos e aplicações práticas, aliando simplicidade de implementação e alta funcionalidade.

**Palavras-chave:** relés de proteção, microcontroladores, ESP32, sobrecorrente, sobretensão.

## **Abstract**

This work presents the development of a protection relay for power systems using the ESP32 microcontroller, with the objective of monitoring and protecting circuits against abnormal conditions of current and voltage, such as undervoltage, overvoltage, instantaneous overcurrent, and time-delayed overcurrent. For this purpose, commercial sensors ACS712 for current and ZMPT101B for voltage were employed, integrated with the ESP32. The microcontroller was programmed using the Arduino IDE. Laboratory tests validated the prototype's operation, demonstrating the accuracy and efficiency of the implemented protection functions. The developed system proved to be an accessible, reliable solution with potential for educational applications and future expansions in teleprotection.

**Keywords:** protection relays, microcontrollers, ESP32, overcurrent, overvoltage.

## LISTA DE ILUSTRAÇÕES

Figura 1 – ESP32 . . . . .	26
Figura 2 – Esquemático ACS712 . . . . .	27
Figura 3 – Esquemático ZMPT101B . . . . .	28
Figura 4 – Interface SerialPlot . . . . .	29
Figura 5 – Display OLED . . . . .	31
Figura 6 – Diagrama do circuito . . . . .	33
Figura 7 – Linearidade do sensor ZMPT101B . . . . .	36
Figura 8 – Linearidade do sensor ACS712 . . . . .	37
Figura 9 – Leitura de tensão e valor filtrado . . . . .	38
Figura 10 – Leitura de corrente e detecção do pico . . . . .	41
Figura 11 – Variador de Tensão . . . . .	50
Figura 12 – Carga variável . . . . .	51
Figura 13 – Circuito de testes . . . . .	52
Figura 14 – Protótipo final . . . . .	55
Figura 15 – Sinal de Subtensão Imposta no Sistema . . . . .	56
Figura 16 – Teste de subtensão . . . . .	57
Figura 17 – Sinal de Sobretensão Imposta no Sistema . . . . .	58
Figura 18 – Teste de sobretensão . . . . .	59
Figura 19 – Sinal de Sobrecorrente Instantânea Imposta no Sistema . . . . .	59
Figura 20 – Teste de sobrecorrente instantânea . . . . .	60
Figura 21 – Teste de Pick-up . . . . .	61
Figura 22 – Teste de sobrecorrente temporizada . . . . .	62
Figura 23 – Teste F51 com $K=0,1$ . . . . .	63
Figura 24 – Teste F51 com $K=1$ . . . . .	63

## LISTA DE TABELAS

Tabela 1 – Valores de $\alpha$ e $\beta$ para as curvas IEC. . . . .	19
Tabela 2 – Valores de $A$ , $B$ e $C$ para as curvas IEEE. . . . .	20
Tabela 3 – Custos de componentes utilizados no projeto. . . . .	32
Tabela 4 – Comparação dos tempos de atuação (IEC vs. Obtido) para diferentes valores de corrente e múltiplos de tempo ( $k = 1$ e $k = 0,1$ ). . . . .	62

## LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Conversor Analógico-Digital</i>
ANSI	<i>American National Standards Institute</i>
CPU	<i>Unidade Central de Processamento</i>
ESP32	<i>Microcontrolador ESP32</i>
I/O	<i>Entrada e Saída (Input/Output)</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
LED	<i>Diodo Emissor de Luz (Light Emitting Diode)</i>
RMS	<i>Valor Eficaz (Root Mean Square)</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SEP	<i>Sistema Elétrico de Potência</i>
TC	<i>Transformador de Corrente</i>
TP	<i>Transformador de Potencial</i>
$\alpha$	<i>Constante das curvas IEC para cálculo do tempo de atuação</i>
$\beta$	<i>Constante das curvas IEC para cálculo do tempo de atuação</i>
$t$	<i>Tempo de atuação do relé (s)</i>
$M$	<i>Relação entre a corrente de curto-circuito e a corrente de ajuste</i>
$k$	<i>Multiplicador de tempo ajustável</i>
$A, B, P$	<i>Constantes das curvas IEEE para cálculo do tempo de atuação</i>
$V_{\text{out}}$	<i>Tensão de saída do divisor de tensão (V)</i>
$V_{\text{in}}$	<i>Tensão de entrada do sensor (V)</i>
$R_1, R_2$	<i>Resistores utilizados no divisor de tensão (<math>\Omega</math>)</i>

$I_{\text{RMS}}$       *Corrente RMS (A)*

$V_{\text{RMS}}$       *Tensão RMS (V)*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
1.1	Objetivos gerais . . . . .	14
1.2	Objetivos específicos . . . . .	14
1.3	Estrutura . . . . .	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>15</b>
2.1	Sistemas Elétricos de Potência . . . . .	15
2.2	Relés de Proteção . . . . .	15
2.2.1	Evolução dos Relés de Proteção . . . . .	15
2.2.2	Relés de Proteção Microprocessados . . . . .	16
2.3	Sistema IEC/IEEE . . . . .	16
2.4	Função 27 - Subtensão . . . . .	17
2.5	Função 59 - Sobretensão . . . . .	17
2.6	Funções 50 - Sobrecorrente Instantânea . . . . .	17
2.7	Funções 51 - Sobrecorrente de tempo inverso . . . . .	18
2.8	Curvas Características . . . . .	18
2.8.1	Curvas de Tempo Definido . . . . .	18
2.8.2	Curvas de Tempo Inverso IEC . . . . .	18
2.8.3	Curvas de Tempo Inverso IEEE . . . . .	19
2.9	Microcontroladores . . . . .	20
2.10	Conversor Analógico-Digital . . . . .	21
2.11	Sensores de Tensão . . . . .	21
2.12	Sensores de Corrente . . . . .	22
<b>3</b>	<b>REVISÃO BIBLIOGRÁFICA . . . . .</b>	<b>23</b>
<b>4</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>25</b>
4.1	Materiais . . . . .	25
4.1.1	Microcontrolador ESP32 . . . . .	25
4.1.2	Arduino IDE . . . . .	26
4.1.3	Módulo Sensor de Corrente ACS712 . . . . .	26

4.1.4	Módulo Sensor de Tensão ZMPT101B . . . . .	28
4.1.5	SerialPlot . . . . .	29
4.1.6	LEDs . . . . .	29
4.1.7	Display . . . . .	30
4.1.8	Custo dos Componentes . . . . .	32
4.2	Diagrama do Circuito e Conexões dos Componentes . . . . .	32
4.3	Operação do Circuito . . . . .	34
4.4	Adequação dos Módulos Sensores . . . . .	35
4.5	Descrição do Código . . . . .	36
4.5.1	Leitura e Adequação do Sinal . . . . .	37
4.5.2	Verificação de Corrente de Pico . . . . .	40
4.5.3	Verificação das Condições de Operação do Sistema . . . . .	41
4.5.4	Atuação . . . . .	44
4.5.5	Display . . . . .	47
4.6	Sistema de testes . . . . .	50
4.6.1	Montagem do Sistema de Testes . . . . .	51
4.6.2	Condições Iniciais do Sistema de Testes . . . . .	51
4.6.3	Tempo de Atuação . . . . .	52
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>54</b>
5.1	Protótipo final . . . . .	54
5.2	Subtensão - Função 27 . . . . .	54
5.3	Sobretensão - Função 59 . . . . .	57
5.4	Sobrecorrente Instantânea - Função 50 . . . . .	58
5.5	Sobrecorrente Temporizada - Função 51 . . . . .	60
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>64</b>
<b>A</b>	<b>Código Completo do Sistema de Proteção . . . . .</b>	<b>67</b>
	<b>BIBLIOGRAFIA UTILIZADA . . . . .</b>	<b>81</b>

# 1 INTRODUÇÃO

O fornecimento de energia elétrica desempenha um papel fundamental na sociedade moderna, sendo essencial para o funcionamento de praticamente todos os aspectos da vida cotidiana. Desde a iluminação e climatização de ambientes até a operação de indústrias e sistemas de comunicação, a energia elétrica é o alicerce sobre o qual se constrói o desenvolvimento econômico e social. A confiabilidade e a continuidade do fornecimento de energia são, portanto, cruciais para garantir a estabilidade das atividades humanas, e qualquer interrupção pode resultar em impactos significativos, como perdas financeiras, riscos à segurança e até mesmo crises sociais.

A proteção de sistemas elétricos de potência é um aspecto essencial para a operação segura e confiável das redes de transmissão e distribuição de energia elétrica. Esses sistemas, compostos por linhas de transmissão, transformadores e outros equipamentos, são frequentemente expostos a condições adversas, como curtos-circuitos, que podem causar danos severos e interrupções no fornecimento de energia. Para mitigar esses riscos, os relés de proteção são dispositivos críticos, projetados para detectar anomalias e isolar partes do sistema afetadas por falhas. No entanto, os relés de proteção tradicionais podem ser caros e complexos, dificultando seu acesso e estudo em ambientes educacionais e de pesquisa.

A evolução das plataformas de desenvolvimento para sistemas embarcados tem proporcionado avanços significativos na construção de projetos eletrônicos complexos. O ESP32, desenvolvido pela Espressif Systems, destaca-se como uma solução robusta e versátil para a criação de sistemas integrados. Segundo o datasheet da Espressif Systems, o ESP32 é um microcontrolador de baixo consumo que combina capacidades de processamento avançado com conectividade sem fio, como Wi-Fi e Bluetooth. Essa característica torna o ESP32 particularmente atraente para aplicações que demandam comunicação em rede e processamento em tempo real.

Complementando essa perspectiva, Fábio Pereira, em seu trabalho sobre microcontroladores MSP430, oferece uma visão prática sobre o uso de microcontroladores em geral, ressaltando a importância da compreensão dos recursos e das aplicações práticas desses dispositivos para um aproveitamento efetivo em 5 projetos de engenharia elétrica (PEREIRA, 2005). Embora o foco de Pereira seja o MSP430, suas observações sobre a

teoria e a prática dos microcontroladores podem ser estendidas para o contexto do ESP32, evidenciando a relevância e as potencialidades dessas plataformas na engenharia moderna.

Neste contexto, devido à flexibilidade e ao baixo custo, o uso de microcontroladores como o ESP32 surge como uma alternativa acessível e eficaz para substituir relés de proteção em sistemas de potência.

## 1.1 Objetivos gerais

O objetivo geral deste trabalho é desenvolver e implementar um relé de proteção para condições anormais de corrente e tensão em sistemas elétricos utilizando um microcontrolador de baixo custo, como o ESP32.

## 1.2 Objetivos específicos

- A. Monitorar tensão e corrente de um circuito utilizando módulos eletrônicos comerciais;
- B. Desenvolver a arquitetura do hardware e montagem do protótipo laboratorial;
- C. Desenvolver algoritmo para implementação de funções de proteção: Algoritmos baseados em normas técnicas para programar o microcontrolador com funções de proteção contra surtos de tensão e corrente;
- D. Elaborar testes de bancada para testar as funções de sobretensão programadas;
- E. Realizar ensaios e comparar com as curvas teóricas.

## 1.3 Estrutura

Este trabalho está dividido em seis capítulos, organizados da seguinte forma:

O Capítulo 1 - Introdução apresenta a contextualização do tema, justificativa do trabalho, relevância do estudo e os desafios associados à proteção de sistemas elétricos de potência. No Capítulo 2 - Objetivos, são descritos os objetivos gerais e específicos do projeto, bem como sua organização estrutural. O Capítulo 3 - Fundamentação Teórica aborda os conceitos essenciais para o desenvolvimento do projeto, incluindo sistemas elétricos de potência, relés de proteção, microcontroladores, sensores de corrente e tensão, além das normas técnicas IEC/IEEE. O Capítulo 4 - Materiais e Métodos detalha os componentes utilizados, o projeto do circuito, os métodos de calibração dos sensores e a programação do ESP32. O Capítulo 5 - Resultados apresenta os testes realizados em laboratório e as análises das funcionalidades do relé de proteção. Por fim, o Capítulo 6 - Conclusão sintetiza os principais resultados alcançados, avalia o cumprimento dos objetivos e sugere possibilidades para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Sistemas Elétricos de Potência

Os Sistemas Elétricos de Potência (SEP) formam a infraestrutura crítica para geração, transmissão e distribuição de energia elétrica, interligando geradores, transformadores, linhas de transmissão e consumidores finais. A confiabilidade e a continuidade do fornecimento de energia são essenciais para o funcionamento da sociedade moderna, influenciando diretamente setores como a indústria, telecomunicações e serviços essenciais. Qualquer falha no SEP pode gerar prejuízos financeiros e riscos à segurança. Nesse contexto, os sistemas de proteção desempenham um papel fundamental, sendo os relés de proteção os principais dispositivos encarregados de detectar falhas e isolar partes comprometidas, garantindo a estabilidade e a segurança do sistema (PEREIRA, 2024).

### 2.2 Relés de Proteção

Os relés de proteção são dispositivos fundamentais para garantir a operação segura e confiável dos sistemas elétricos de potência. Sua função principal é monitorar parâmetros como corrente e tensão, detectando anomalias que possam comprometer a estabilidade do sistema. Ao identificar falhas como sobrecorrente e subtensão, os relés atuam automaticamente, desconectando partes defeituosas da rede. Essa atuação rápida minimiza danos aos equipamentos e evita apagões generalizados, preservando a continuidade do fornecimento de energia (BREDA, 2009).

#### 2.2.1 Evolução dos Relés de Proteção

A história dos relés de proteção remonta ao início do século XX, quando as primeiras redes de transmissão e distribuição de energia começaram a se expandir e a necessidade de proteção contra falhas se tornou evidente. Os primeiros relés eram dispositivos eletromecânicos simples, compostos por bobinas, contatos e mecanismos que respondiam às variações de corrente e tensão. Esses relés baseavam-se em princípios de indução e eletromagnetismo para detectar anomalias e acionar mecanismos de desconexão do circuito em caso de falha. Embora robustos, esses relés apresentavam limitações em termos de precisão, tempo de resposta e flexibilidade de configuração.

Nas décadas de 1970 e 1980, com o avanço da eletrônica, os relés começaram a incorporar componentes eletrônicos, tornando-se mais sensíveis e precisos. Essa evolução permitiu o desenvolvimento dos relés eletrônicos, que melhoraram a confiabilidade e reduziram o desgaste mecânico. No entanto, foi apenas com a introdução dos relés digitais, a partir da década de 1990, que se deu uma verdadeira revolução na área de proteção (XAVIER, 2013). Esses relés, equipados com microprocessadores, possibilitaram a integração de diversas funções de proteção em um único dispositivo, além de oferecerem maior capacidade de comunicação e automação. Hoje, os relés digitais são amplamente utilizados devido à sua flexibilidade, alta precisão e capacidade de diagnóstico.

### 2.2.2 Relés de Proteção Microprocessados

Os relés de proteção modernos, também conhecidos como relés digitais ou microprocessados, operam utilizando microprocessadores para monitorar continuamente as variáveis elétricas do sistema, como corrente, tensão e frequência. Diferentemente dos relés eletromecânicos, que atuavam com base em componentes físicos, os relés modernos processam sinais digitais provenientes de transformadores de corrente (TCs) e transformadores de potencial (TPs), comparando-os com valores predefinidos de operação. Quando uma falha é detectada, o relé envia um comando para dispositivos de interrupção, como disjuntores, isolando a parte defeituosa do sistema (RUSH, 2009).

Além de sua função de proteção, os relés digitais possuem capacidades avançadas de comunicação, permitindo sua integração com sistemas de automação e controle, como SCADA (*Supervisory Control and Data Acquisition*). Isso permite que os operadores monitorem e ajustem remotamente os parâmetros de proteção em tempo real. Os relés modernos também oferecem funções adicionais, como monitoramento contínuo da condição dos equipamentos, registro de eventos e análise de falhas, fornecendo informações detalhadas que auxiliam na manutenção preditiva e na melhoria da confiabilidade do sistema elétrico (RUSH, 2009).

## 2.3 Sistema IEC/IEEE

As funções de proteção em sistemas elétricos são identificadas por números padronizados como parte de um sistema estabelecido pela ANSI (*American National Standards*) e pela IEEE (*Institute of Electrical and Electronics Engineers*). Esses números, conhecidos

como ANSI device numbers, foram criados para associar de forma clara e concisa cada função de proteção a um código numérico específico. Essa padronização facilita a comunicação e a documentação técnica, permitindo que engenheiros e operadores reconheçam rapidamente a função de um relé ou dispositivo de proteção sem a necessidade de descrições detalhadas (PEREIRA, 2024).

#### 2.4 Função 27 - Subtensão

A função 27 é utilizada para detectar condições em que a tensão de operação do sistema cai abaixo de um valor pré-determinado (KINDERMANN, 2005). Em sistemas elétricos de potência, a operação em baixa tensão pode ser tão prejudicial quanto em alta tensão. Quando um sistema opera com níveis de tensão inferiores ao necessário, isso pode causar um desempenho ineficiente de motores, aquecimento excessivo de transformadores e, em casos mais graves, falhas nos dispositivos conectados à rede.

#### 2.5 Função 59 - Sobretensão

A função 59 é responsável por proteger os sistemas elétricos contra elevações na tensão que ultrapassam os valores nominais permitidos. Esse tipo de proteção é fundamental para evitar danos aos equipamentos, especialmente aqueles mais sensíveis, cujos isolamentos podem ser comprometidos por exposições prolongadas a tensões elevadas (XAVIER, 2013).

#### 2.6 Funções 50 - Sobrecorrente Instantânea

A função 50 é projetada para detectar falhas de sobrecorrente que ocorrem instantaneamente, ou seja, sem qualquer atraso intencional. Essa função é essencial em situações de curto-circuito severo, onde a corrente aumenta de maneira abrupta e substancial, ultrapassando em várias vezes o valor nominal de operação. O objetivo dessa proteção é atuar de forma rápida e eficaz, desligando o circuito o mais rápido possível para minimizar os danos ao sistema (PEREIRA, 2024).

## 2.7 Funções 51 - Sobrecorrente de tempo inverso

A função 51 atua de forma a proteger o sistema contra sobrecorrentes com um tempo de atuação que varia inversamente à magnitude da corrente. Isso significa que, quanto maior a corrente de falha, mais rápido será o tempo de atuação do relé. A lógica por trás dessa característica de tempo inverso é permitir que o sistema tenha mais tempo para operar em situações de sobrecorrente moderada e, ao mesmo tempo, responder de forma rápida a falhas severas, como curtos-circuitos. A função 51 é amplamente utilizada para a proteção de alimentadores, transformadores e motores, sendo uma proteção crucial em sistemas de distribuição de energia (XAVIER, 2013).

## 2.8 Curvas Características

### 2.8.1 Curvas de Tempo Definido

As curvas de tempo definido são aquelas em que o relé de proteção atua após um tempo fixo, independentemente da magnitude da corrente de falha. Uma vez que a corrente de falha excede o valor ajustado de pickup, o tempo de atuação do relé é sempre o mesmo, não importando o quão grande seja a corrente de sobrecarga. Essa característica torna as curvas de tempo definido bastante simples de configurar e úteis em aplicações em que o tempo de atuação deve ser sempre o mesmo, como em sistemas de proteção onde a seletividade entre diferentes dispositivos não é um fator crítico (XAVIER, 2013).

### 2.8.2 Curvas de Tempo Inverso IEC

As curvas de tempo inverso definidas pela norma IEC 60255-3 seguem a lógica de que, quanto maior a corrente de curto-circuito (ou sobrecorrente), menor é o tempo de resposta do relé. Isso proporciona uma atuação mais rápida em condições de falha severa, minimizando os danos ao sistema.

As curvas IEC são divididas em três tipos principais:

- Normalmente Inversa (NI),
- Muito Inversa (VI),
- Extremamente Inversa (EI).

A equação geral que descreve o tempo de atuação das curvas IEC é dada por:

$$t = \frac{\beta}{M^\alpha - 1} \cdot k$$

Onde:

- $t$  é o tempo de atuação do relé,
- $M$  é a relação entre a corrente de curto-circuito e a corrente de ajuste,
- $k$  é o multiplicador de tempo, ajustável de acordo com o projeto,
- $\alpha$  e  $\beta$  são constantes específicas de cada tipo de curva.

Os valores típicos de  $\alpha$  e  $\beta$  para as curvas IEC são:

Tabela 1 – Valores de  $\alpha$  e  $\beta$  para as curvas IEC.

Curva	$\alpha$	$\beta$
Normalmente Inversa (NI)	0,02	0,14
Muito Inversa (VI)	1,0	13,5
Extremamente Inversa (EI)	2,0	80,0

Fonte: Kindermannn, 2005.

Esses valores de  $\alpha$  e  $\beta$  determinam como a relação entre a corrente de curto-circuito e a corrente de ajuste influencia o tempo de resposta, com as curvas mais inversas resultando em tempos de atuação mais curtos para correntes elevadas (XAVIER, 2013).

### 2.8.3 Curvas de Tempo Inverso IEEE

As curvas de tempo inverso IEEE (também conhecidas como curvas ANSI) são semelhantes às curvas IEC, mas apresentam algumas diferenças nos parâmetros de ajuste e na aplicação. Elas são regidas pela norma IEEE C37.112 e são amplamente utilizadas em sistemas norte-americanos e em várias outras regiões (KINDERMANN, 2005).

A equação geral para as curvas IEEE é expressa da seguinte forma:

$$t = \frac{A}{M^P - 1} + B$$

Onde:

- $t$  é o tempo de operação do relé,
- $M$  é a relação entre a corrente de curto-circuito e a corrente de ajuste,
- $A$ ,  $B$ , e  $P$  são constantes que variam de acordo com o tipo de curva.

Os valores típicos de  $A$ ,  $B$  e  $P$  para as curvas IEEE são:

<b>Curva</b>	$A$	$B$	$P$
Inversa	0,0086	0,0250	0,02
Muito Inversa	2,8550	0,0712	2,00
Extremamente Inversa	6,4070	0,0185	2,00

Tabela 2 – Valores de  $A$ ,  $B$  e  $C$  para as curvas IEEE.

Assim como nas curvas IEC, os valores de  $A$ ,  $B$  e  $C$  determinam a sensibilidade da curva e o tempo de resposta do relé para diferentes níveis de sobrecorrente (PEREIRA, 2024).

## 2.9 Microcontroladores

O funcionamento básico de um microcontrolador pode ser descrito pela interação entre seus três principais componentes: a unidade central de processamento (CPU), a memória e os periféricos. A CPU é responsável por executar instruções, que são armazenadas na memória. Essas instruções podem ser o resultado de um programa desenvolvido pelo engenheiro de sistemas, o qual, geralmente, é escrito em linguagens como C ou C++. A memória de um microcontrolador, por sua vez, é subdividida em memória volátil (RAM), usada para armazenamento temporário de dados, e memória não volátil (ROM ou Flash), onde o código do programa é armazenado (PEREIRA, 2005).

Além disso, os microcontroladores possuem uma série de interfaces para comunicação com o ambiente externo, conhecidas como portas de entrada e saída (I/O). Essas portas permitem que o microcontrolador interaja com sensores e atuadores, monitorando variáveis como temperatura, pressão e corrente elétrica, e tomando decisões com base nessas leituras.

A compreensão do funcionamento e das características desses dispositivos é essencial para sua correta aplicação em projetos de engenharia (PEREIRA, 2005). Em particular, o uso de microcontroladores para simulação de relés de proteção, como proposto neste trabalho, permite a criação de sistemas mais acessíveis, flexíveis e adaptáveis, ao mesmo

tempo em que oferece uma alternativa de baixo custo aos relés convencionais utilizados em sistemas de potência.

## 2.10 Conversor Analógico-Digital

O conversor analógico-digital (ADC) é um componente essencial em sistemas de controle baseados em microcontroladores, pois realiza a interface entre sinais analógicos e o processamento digital. Ele converte grandezas físicas representadas como sinais analógicos em valores digitais, permitindo que sejam processados por dispositivos eletrônicos programáveis.

O princípio de funcionamento do ADC consiste em amostrar o sinal analógico em intervalos regulares, quantificando os níveis de tensão em valores discretos representados em formato binário. Esse processo envolve duas etapas principais: a amostragem, que captura o valor instantâneo do sinal analógico em momentos específicos, e a quantização, que converte cada valor amostrado em um nível digital dentro de uma faixa de valores predefinida. A precisão dessa quantificação depende da resolução do ADC, geralmente expressa em bits, a qual define o número de níveis disponíveis para representar o sinal. Por exemplo, um ADC de 12 bits pode discriminar 4096 níveis, enquanto um ADC de 10 bits discrimina 1024 níveis (GOMES, 2021).

## 2.11 Sensores de Tensão

A medição de tensão em sistemas de potência é crucial para garantir a proteção adequada de equipamentos e a operação segura de redes elétricas. Sensores de tensão desempenham um papel fundamental em relés de proteção, fornecendo dados essenciais para a detecção de condições de sobretensão e subtensão, que podem comprometer a integridade do sistema. Esses sensores convertem a tensão de linha em valores adequados para monitoramento por sistemas de controle baseados em microcontroladores.

Sensores de tensão comerciais são projetados para realizar a medição precisa da tensão alternada (AC) de um sistema elétrico, oferecendo uma interface analógica segura e compatível com dispositivos digitais de controle. O princípio básico de funcionamento desses sensores está relacionado à transformação da tensão de entrada em um sinal de baixa

amplitude que pode ser processado por conversores analógico-digitais (ADC) integrados em microcontroladores.

Esses sensores utilizam circuitos internos de isolamento que garantem a segurança na medição de tensões elevadas sem a necessidade de um contato direto com o sistema de potência. A maioria dos sensores comerciais de tensão é projetada com amplificadores e filtros, que garantem que o sinal de saída seja linearmente proporcional à tensão medida, além de eliminar ruídos e distorções que possam prejudicar a precisão das medições (OLIVEIRA, 2021).

## 2.12 Sensores de Corrente

A medição de corrente elétrica em sistemas é crucial para o monitoramento e controle de dispositivos, bem como para a proteção de equipamentos e redes elétricas. Sensores de corrente desempenham um papel fundamental ao converter a corrente elétrica de um circuito em um sinal proporcional, que pode ser facilmente monitorado e processado por sistemas digitais, como microcontroladores.

Os sensores de corrente utilizam diferentes tecnologias para medir a intensidade da corrente elétrica, como o efeito Hall, transformadores de corrente (TCs) ou resistores de shunt. Entre essas tecnologias, os sensores baseados no efeito Hall destacam-se por sua capacidade de medir correntes contínuas (DC) e alternadas (AC) sem necessidade de conexão direta ao circuito principal. Esses sensores funcionam detectando o campo magnético gerado pela corrente elétrica e convertendo-o em uma tensão proporcional, que pode ser processada posteriormente (OLIVEIRA, 2021).

### 3 REVISÃO BIBLIOGRÁFICA

A proteção de sistemas elétricos de potência (SEP) é essencial para garantir a integridade das redes de distribuição e transmissão, bem como a segurança de equipamentos e usuários. De acordo com Lima Filho (1997), os dispositivos de proteção são projetados para detectar anomalias elétricas, como sobrecorrentes, curtos-circuitos e sobretensões, e atuar de maneira seletiva e rápida. Relés de proteção e disjuntores são os principais dispositivos utilizados, e sua operação conjunta promove a confiabilidade das redes ao isolar partes comprometidas do sistema.

A evolução tecnológica trouxe melhorias significativas no campo dos relés de proteção. Enquanto os relés eletromecânicos utilizam componentes móveis e princípios magnéticos, os relés digitais têm se consolidado como uma solução mais avançada. Esses dispositivos integram maior flexibilidade na configuração, monitoramento remoto e a capacidade de executar múltiplas funções de proteção em um único equipamento. Estudos como o de Manassero Júnior (2016) destacam que os relés digitais são amplamente utilizados em sistemas industriais e comerciais devido à sua precisão e robustez, além de suportarem a implementação de funções padronizadas pelas normas ANSI e IEC.

O trabalho de Romano e Xavier (2013) aborda especificamente a aplicação de relés de sobrecorrente e sua relevância em redes de baixa e média tensão. Esses dispositivos utilizam curvas características, como as definidas pelas normas IEC 60255 e IEEE C37.112, que relacionam o tempo de atuação à magnitude da corrente de falha. As curvas Normalmente Inversa, Muito Inversa e Extremamente Inversa são ajustadas com base nos parâmetros de partida, tempo de operação e fatores específicos da instalação, garantindo seletividade e rapidez na proteção.

No campo da medição elétrica, o trabalho de Silva et al. (2014) discute a utilização de conversores analógico-digitais (ADC) para medições precisas de corrente e tensão em sistemas monofásicos. Componentes como o ADE7753 são amplamente utilizados devido à sua capacidade de medir potência ativa, reativa e aparente, além de integrarem filtros internos e amplificadores programáveis para melhorar a qualidade do sinal processado. Esses dispositivos são frequentemente associados a sistemas baseados em microcontroladores para facilitar a coleta e o processamento dos dados.

A integração de dispositivos de medição e proteção com protocolos de comunicação é discutida por Falcão (2010), que destaca a importância de soluções conectadas em redes inteligentes (Smart Grids). A utilização de protocolos como Modbus e MQTT viabiliza o monitoramento remoto e o ajuste dinâmico de parâmetros, aumentando a confiabilidade dos sistemas. Além disso, a comunicação em tempo real permite identificar falhas e eventos críticos com maior eficiência, reduzindo o tempo de resposta em cenários adversos.

Os avanços em sensores comerciais também são amplamente discutidos na literatura. Sensores como o ZMPT101B, para medições de tensão, e o ACS712, para corrente, são frequentemente utilizados em projetos de baixo custo. Contudo, no contexto deste trabalho, esses sensores foram integrados a um sistema de proteção utilizando o microcontrolador ESP32, explorando seu potencial para implementar funções de proteção em sistemas monofásicos. Uma característica marcante do protótipo desenvolvido é a adaptação dos sensores aos algoritmos desenvolvidos, o que possibilitou melhorar a confiabilidade das medições mesmo considerando suas limitações inerentes, como a necessidade de calibração manual e sensibilidade a ruídos.

Diferentemente das abordagens convencionais encontradas na literatura, o protótipo desenvolvido neste trabalho de conclusão de curso integra funções de proteção contra sobrecorrente e sobretensão em um único sistema compacto, com base em componentes comerciais acessíveis. Enquanto outros trabalhos discutem sistemas de proteção voltados para aplicações industriais ou acadêmicas específicas, o presente projeto destaca-se por aliar simplicidade e eficiência, sendo uma solução viável tanto para aplicações educacionais quanto para a introdução de sistemas de proteção elétricos. Além disso, o uso do ESP32, com suas capacidades de processamento e conectividade, diferencia este trabalho ao abrir possibilidades para futuras expansões, como integração com protocolos de comunicação e monitoramento remoto.

## 4 MATERIAIS E MÉTODOS

Este trabalho se propõe ao desenvolvimento de um relé de proteção capaz de identificar faltas de subtensão, sobretensão e sobrecorrente, portanto no presente capítulo serão descritos os materiais utilizados no desenvolvimento do relé de proteção, bem como a programação inserida no microcontrolador que atua como unidade central de processamento do relé

### 4.1 Materiais

Nesta seção são descritos os principais componentes e estruturas eletrônicas que foram utilizados para construção do relé de proteção. Cada componente foi selecionado visando sua confiabilidade na função que se propõe, sendo considerado também o custo benefício da sua aplicação no sistema.

#### 4.1.1 Microcontrolador ESP32

O ESP32 foi escolhido como a unidade central de processamento do protótipo de relé de proteção desenvolvido neste trabalho. Este microcontrolador se destaca por suas capacidades de processamento e flexibilidade de interfaces, características que tornam possível sua aplicação neste projeto.

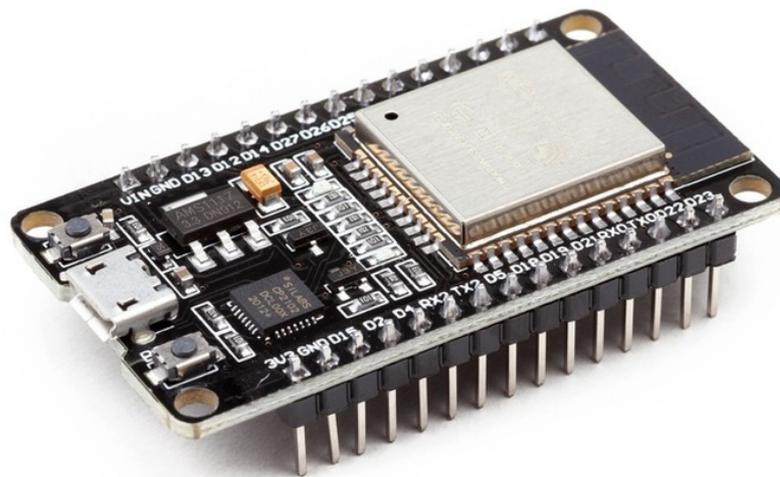
Entre os recursos mais relevantes do ESP para o presente trabalho, destacam-se seus pinos de entrada analógica, com conversores ADC de 12 bits. Estes são essenciais para a aquisição de dados dos sensores de corrente e tensão, permitindo uma conversão precisa dos sinais analógicos gerados pelos sensores.

Além disso, o ESP32 possui conectividade Wi-Fi e Bluetooth, o que pode ser explorado para a comunicação remota ou integração com sistemas de monitoramento e controle em redes inteligentes (*smart grids*), permitindo a expansão das funcionalidades deste relé de proteção em trabalhos futuros.

Outra vantagem significativa é a presença de timers de hardware, que garantem um controle preciso dos tempos de resposta necessários em um relé de proteção. Isso facilita a implementação de funções como a sobrecorrente temporizada, onde o tempo de atuação deve ser ajustado conforme as curvas de proteção adotadas.

A escolha do ESP32 neste trabalho deve-se à sua versatilidade, ao baixo custo, e à sua capacidade de realizar processamento de sinais e controle em tempo real. Sua capacidade de integrar sensores analógicos, além da facilidade de programação via a plataforma Arduino IDE, faz do ESP32 uma excelente opção para a prototipagem rápida e eficiente do sistema proposto.

Figura 1 – ESP32



Fonte: Adaptado de Vida de Silício (2024)

#### 4.1.2 Arduino IDE

A Arduino IDE foi utilizada para o desenvolvimento e programação do microcontrolador ESP32, na linguagem de programação C++, devido à sua simplicidade, integração com o ESP32 e ampla documentação disponível.

Além disso, a Arduino IDE é familiar aos autores devido sua ampla aplicação durante a graduação de engenharia elétrica.

#### 4.1.3 Módulo Sensor de Corrente ACS712

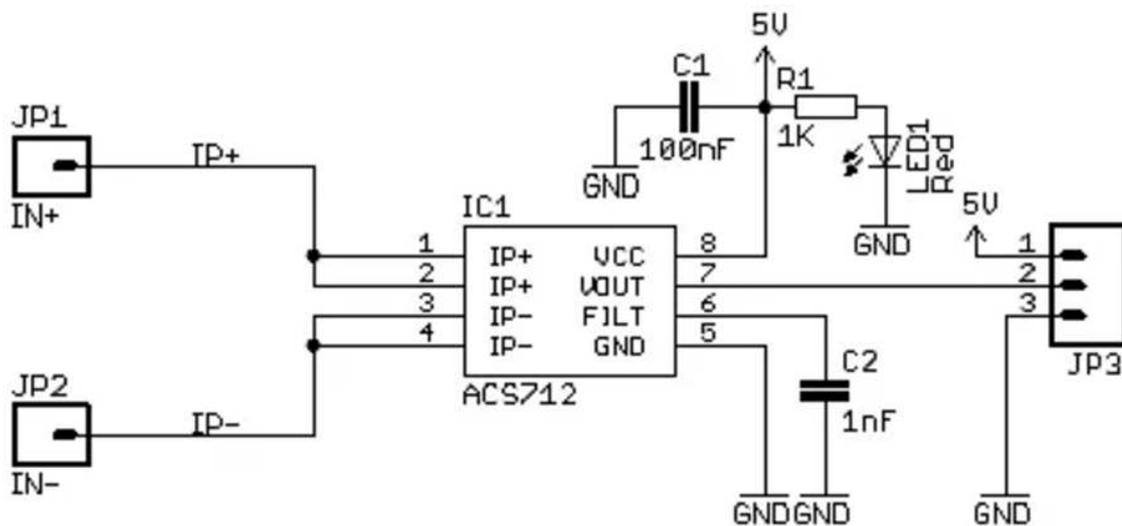
O sensor de corrente ACS712 é utilizado no projeto para medir correntes alternadas e contínuas em circuitos de baixa potência. Esse sensor foi escolhido por sua capacidade

de fornecer leituras corrente com boa resolução para faixa de corrente que interessa a esta aplicação, além de sua fácil integração com o microcontrolador ESP32.

O ACS712 é um sensor baseado no efeito Hall, então o campo magnético gerado pela corrente circulando no sensor pode ser detectada pelo sensor de efeito Hall interno, que converte esse campo em uma tensão de saída analógica proporcional à corrente medida. Essa tensão de saída pode ser facilmente lida por um conversor analógico-digital (ADC), como os presentes no ESP32. Na aplicação de relés em sistemas elétricos de potência os transformadores de corrente são comumente dimensionados para o maior nível de curto ser transformado para 5 A no secundário do transformador de corrente, devido a este fato, o modelo do ACS712 foi escolhido com capacidade máxima de corrente de 20 A, que é o modelo comercial disponível com menor capacidade máxima de corrente acima de 5 A.

Na Figura 2 é apresentado o esquemático do módulo sensor de corrente baseado ACS712.

Figura 2 – Esquemático ACS712



Fonte: Adaptado de ElectroSchematics (2024)

Como pode ser observado no esquemático, o módulo sensor precisa ser conectado em série com o sistema ao qual esta sendo monitorado, de forma que o campo magnético da corrente que circula pelo sistema cause a diferença de potencial necessária para formação do campo de Hall.

O ACS712 é alimentado com tensão de 5 V e fornece em sua saída um sinal entre

0 V e 5 V. O microcontrolador ESP32 é projetado para receber sinais de até 3,3 V e tensões superiores a essa podem afetar o funcionamento da unidade central de processamento do projeto. Para extinguir este risco, será aplicado um divisor de tensão com resistores de 1 k $\Omega$  e precisão de 1% na saída do módulo do sensor, desta forma o sinal de saída do módulo ACS712 fica limitado a 2.5 V.

#### 4.1.4 Módulo Sensor de Tensão ZMPT101B

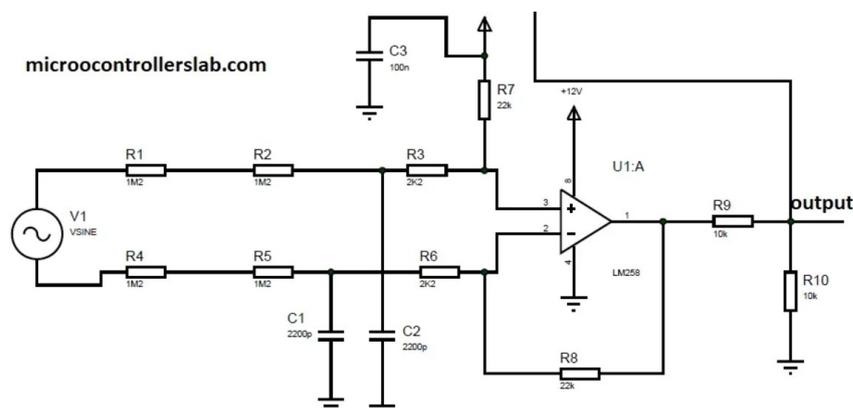
O sensor de tensão ZMPT101B é utilizado neste trabalho para realizar a medição de tensão no sistema monitorado. Ele foi escolhido devido à sua alta aplicação já documentada, sendo uma solução adequada para aplicações que exigem medições de tensão entre 0 V e 250 V. Esse sensor é baseado em um transformador de tensão que converte a tensão em um sinal analógico, adequado para ser lido pelo conversor analógico-digital presente no microcontrolador ESP32.

Este sensor opera de maneira confiável em uma faixa de tensões de até 250 V, e sua corrente nominal de operação é de aproximadamente 1 mA a 2 mA, portanto será suficiente para aplicação no relé, visto que a tensão que interessa ser monitorada está na faixa de 100 V a 130 V.

A faixa de aplicação desse módulo sensor foi definida entre 100 V e 130 V, pois a atuação do relé de proteção desenvolvido neste projeto acontece a partir de uma variação maior que 10% da tensão de referência. Sendo definida a tensão de referência como 115 V, os limites de variação de 10% a partir deste valor são 103,5 V e 126,5 V.

Na Figura 3, está o esquemático do módulo sensor baseado no ZMPT101B.

Figura 3 – Esquemático ZMPT101B



Fonte: Adaptado de Microcontrollerslab (2024)

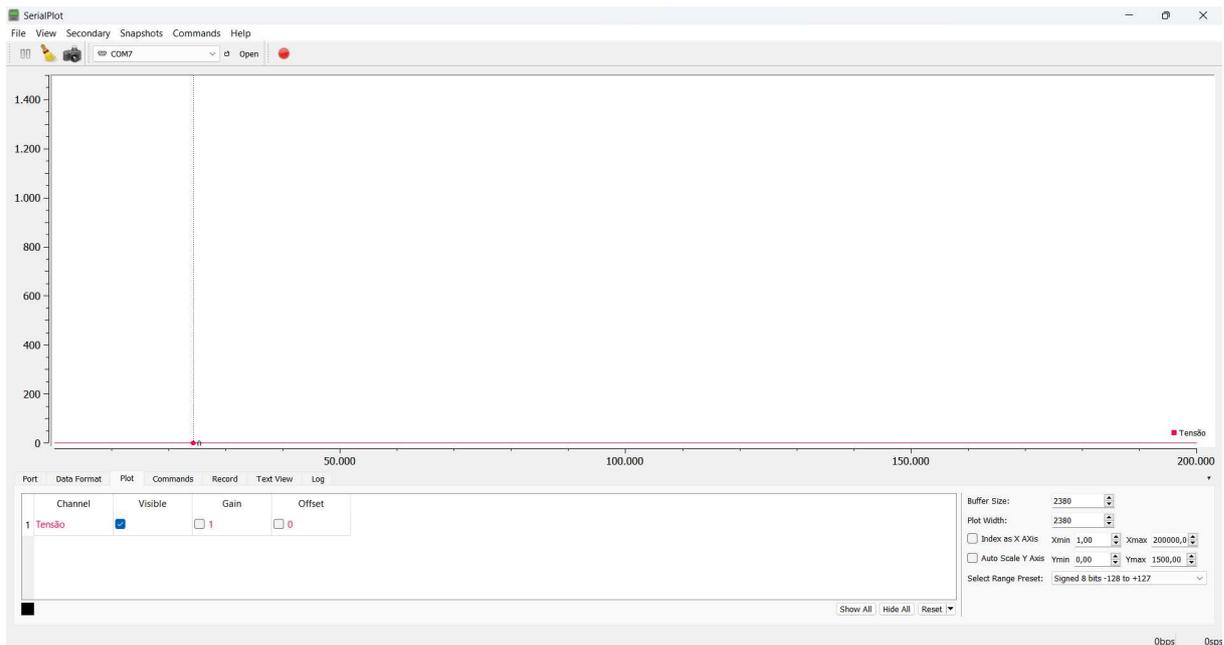
A saída deste módulo, assim como o módulo ACS712, tem tensão máxima de 5 V. Portanto a mesma solução adotada para o módulo de leitura de corrente, foi utilizada para possibilitar a conexão segura entre módulo ZMPT101B e o ESP32.

#### 4.1.5 SerialPlot

O SerialPlot é um software utilizado para a visualização em tempo real dos dados dos sensores, permitindo a análise das formas de onda de corrente e tensão transmitidas via comunicação serial pelo ESP32.

O SerialPlot é um software familiar aos autores e foi utilizado ao longo da graduação, junto a isso foi escolhido pela sua eficiência na visualização gráfica de dados seriais em tempo real, o que auxiliou na calibração dos sensores e ajustes no sistema de proteção.

Figura 4 – Interface SerialPlot



Fonte: Os Autores, (2024).

#### 4.1.6 LEDs

Como uma das formas sinalizar os estados de operação do relé de proteção, optou-se por utilizar LEDs de diferentes cores a depender do estado de operação do sistema que está sendo monitorado. A escolha foi a seguinte:

- Funcionamento normal do sistema monitorado pelo relé: LED verde;

- Atuação por sobretensão: LED vermelho;
- Atuação por subtensão: LED vermelho;
- Corrente de ajuste de pickup alcançada: LED amarelo;
- Atuação por sobrecorrente temporizada: LED vermelho;
- Atuação por sobrecorrente instantânea: LED vermelho.

É importante destacar que no trabalho desenvolvido e descrito nesse documento, o relé de proteção não contempla um dispositivo capaz de interromper o funcionamento do sistema monitorado. Optou-se pelo desenvolvimento do trabalho desta forma pelo fato de que em sistemas elétricos de potência as unidades de aquisição e processamento de sinais do sistema e envio de comando de atuação são diferentes das unidades responsáveis por interromper o funcionamento do sistema.

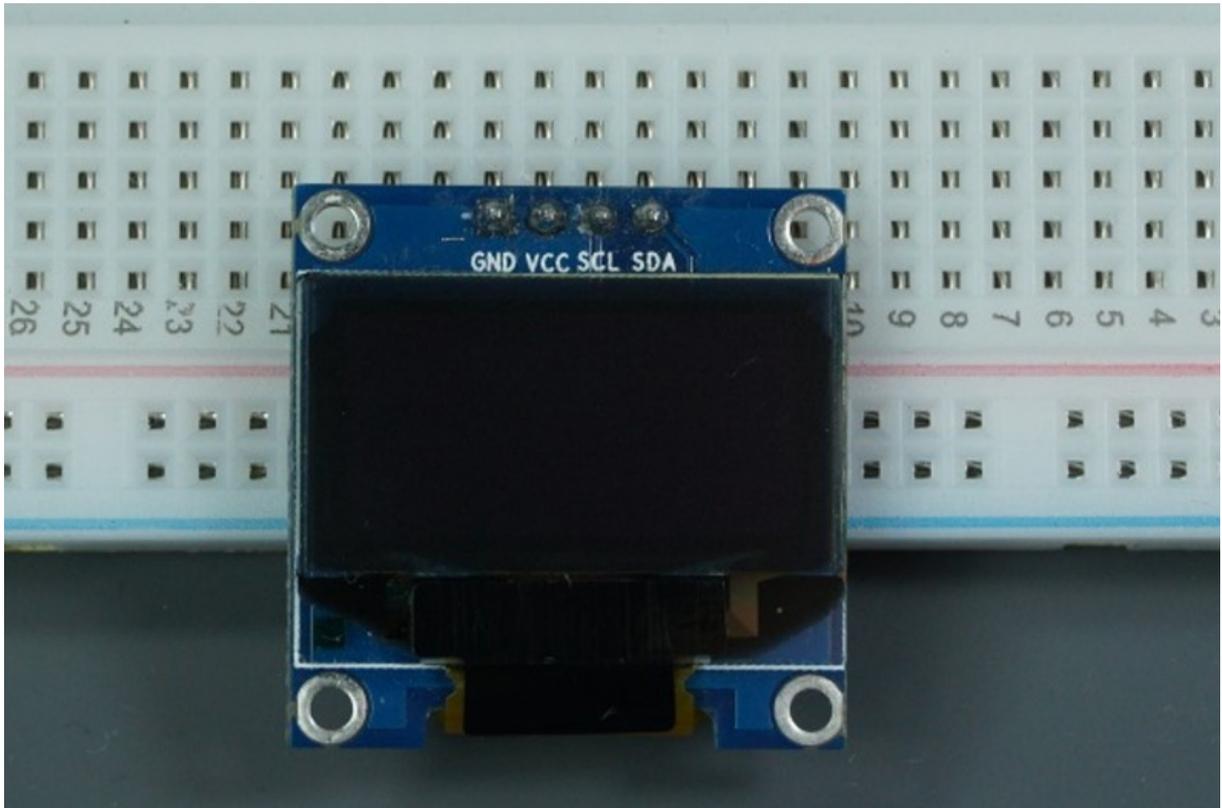
#### 4.1.7 Display

Complementar a função dos LEDs, descritos na seção anterior, será utilizado um display OLED com a finalidade de apresentar dados relevantes ao funcionamento do sistema monitorado. O modelo do display utilizado é apresentado na Figura 5.

Os displays OLED (Diodo Emissor de Luz Orgânico) são amplamente utilizados em sistemas embarcados devido à sua eficiência energética, alta qualidade de exibição e capacidade de apresentar informações em ambientes com baixa luminosidade. Diferentemente dos displays LCD tradicionais, os OLEDs não requerem iluminação de fundo, pois cada pixel emite luz individualmente. Essa característica resulta em contrastes mais elevados e menor consumo de energia, especialmente ao exibir conteúdos predominantemente escuros. Esses displays estão disponíveis em diferentes tamanhos e resoluções, sendo o modelo SSD1306 um dos mais comuns para projetos baseados em microcontroladores, como o ESP32. Por sua compatibilidade com interfaces como I2C e SPI, o SSD1306 é uma escolha versátil e amplamente adotada em projetos embarcados para exibição de informações dinâmicas.

A comunicação I2C (Inter-Integrated Circuit) é amplamente utilizada para interligar o display SSD1306 ao microcontrolador, operando como um barramento serial síncrono

Figura 5 – Display OLED



Fonte: Adaptado de Random Nerd Tutorials.

que utiliza apenas dois fios para transmissão de dados: SDA (linha de dados) e SCL (linha de clock). O protocolo segue uma hierarquia mestre-escravo, em que o dispositivo mestre (geralmente o microcontrolador) inicia a comunicação e controla o fluxo de dados. Cada dispositivo no barramento possui um endereço único, permitindo que o mestre envie comandos ou dados para dispositivos específicos sem interferências. O funcionamento do protocolo baseia-se em ciclos de clock para sincronizar o envio e a recepção dos bits, enquanto a linha SDA transmite os dados em formato serial. Esse mecanismo simplificado, combinado com a capacidade de conectar múltiplos dispositivos no mesmo barramento, faz do I2C uma solução eficiente para aplicações embarcadas que requerem comunicação confiável e com consumo reduzido de recursos físicos.

Utilizando dos recursos de apresentação do display OLED, nele serão apresentadas quatro diferentes informações, sendo elas:

- Status: similarmente a função dos LEDs, o indicador de "Status" informa sobre o funcionamento normal do sistema ou em caso de atuação do relé é apontada a função qual foi atuada;

- Tensão RMS: valor de tensão RMS do sistema em Volts;
- Corrente RMS: valor de corrente RMS do sistema em Amperes;
- Corrente PKUP: é apresentado no display quando a corrente de ajuste de pickup é alcançada, então o valor valor de corrente utilizado para cálculo de tempo de atuação da função 51 é apresentado;
- Atuação: é apresentado quando o relé de proteção atua pela função 51, então o display apresenta o tempo entre o instante que a corrente de ajuste de pickup foi ultrapassada até o instante da atuação do relé de proteção;

#### 4.1.8 Custo dos Componentes

Como este trabalho se propõe ao desenvolvimento de um relé comercial utilizando componentes eletrônicos comerciais, na tabela a seguir estão apresentados os custos dos principais componentes desse projeto no momento da compra dos mesmos.

Tabela 3 – Custos de componentes utilizados no projeto.

<b>Item</b>	<b>Preço (R\$)</b>
ESP32	42,00
ACS712	18,00
ZMPT101B	24,00
Display	28,00
LEDs	6,00
Outros	30,00
Total	148,00

Fonte: Os Autores, (2024).

O item outros representa o custo de materiais utilizados para montagem do circuito, como por exemplo, estanho para solda e a placa perfurada na qual foi montado o circuito.

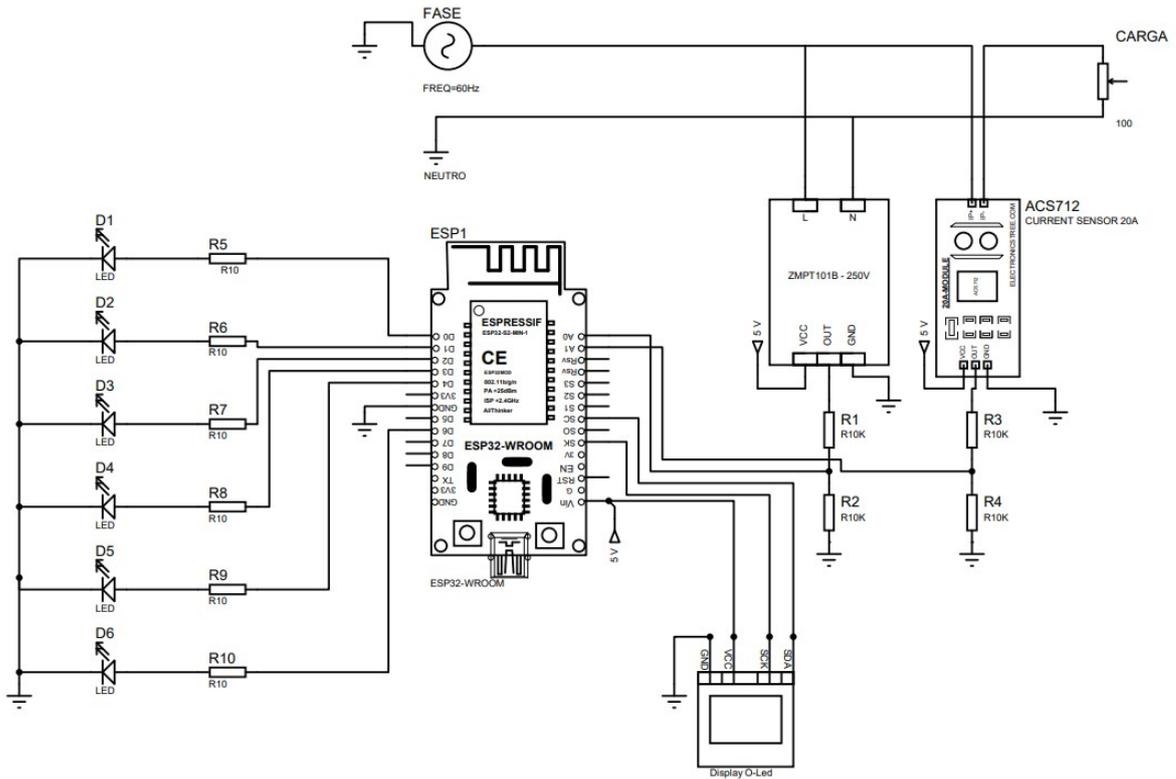
#### 4.2 Diagrama do Circuito e Conexões dos Componentes

Nesta seção, detalhamos a montagem física do circuito desenvolvido para o relé de proteção, descrevendo a disposição e conexão dos componentes utilizados. Apresentamos o diagrama do circuito, a configuração dos componentes e o funcionamento operacional do sistema.

O diagrama esquemático do circuito é fundamental para compreender a interligação entre os diversos componentes utilizados no projeto. O microcontrolador ESP32 atua como

a unidade central de processamento, responsável pela leitura dos sensores e controle do relé de proteção. Os sensores de corrente ACS712 e de tensão ZMPT101B são conectados ao ESP32 para monitorar as grandezas elétricas do sistema de potência.

Figura 6 – Diagrama do circuito



Fonte: Os Autores, (2024).

A alimentação do microcontrolador ESP32 é realizada através de entrada USB tipo C presente na placa do microcontrolador. O mesmo ponto da alimentação, em 5 V, fornece energia para os módulos sensores de tensão e corrente. Os pinos GND do ESP32 são utilizados como referencial de terra para todo o circuito, garantindo um ponto comum de referência para as medições e conexões dos sensores.

Para a conexão dos sensores de corrente e tensão, o sensor ACS712 (corrente) tem seu pino VCC conectado ao pino 5V do ESP32, o pino GND ao GND do ESP32, e o pino OUT é conectado ao pino analógico 34 do ESP32, passando pelo divisor de tensão.

De forma semelhante, o sensor ZMPT101B (tensão) tem seu pino VCC conectado ao pino 5V do ESP32, o pino GND ao GND do ESP32, e o pino OUT é conectado ao pino analógico 13 do ESP32, passando pelo seu divisor de tensão.

Os divisores de tensão são essenciais para reduzir a amplitude dos sinais analógicos provenientes dos sensores, garantindo que os valores lidos pelo ADC do ESP32 estejam dentro da faixa operacional segura de 0V a 3,3V. A equação utilizada para calcular a tensão de saída do divisor de tensão é:

$$V_{\text{out}} = V_{\text{in}} \times \frac{R2}{R1 + R2} \quad (4.1)$$

onde  $V_{\text{out}}$  é a tensão de saída do divisor de tensão,  $V_{\text{in}}$  é a tensão de entrada do sensor e  $R1$  e  $R2$  são os resistores de  $1\text{ k}\Omega$  utilizados.

Os indicadores de atuação do relé, formados pelos 6 LEDs, são conectados cada um a uma porta com função de saída digital do ESP32. Dessa forma é possível controlar individualmente cada um dos LEDs, formando uma maneira simples de identificar os estados do sistema.

De forma diferente dos módulos sensores, o display é alimentado pelo pino de 3.3 V, o GND do display é conectado ao GND do ESP32. Para comunicação entre display e o ESP32, os pinos 21 e 22 do ESP32 são conectados aos pinos SDA e SCK do display, respectivamente.

### 4.3 Operação do Circuito

O funcionamento operacional do circuito baseia-se na interação contínua entre os módulos sensores e o microcontrolador ESP32. Os módulos sensores de corrente e tensão monitoram as grandezas elétricas do sistema, convertendo as variações em sinais de tensão analógica proporcional. O conversor ADC integrado ao ESP32 converte esses sinais analógicos em valores digitais que representam as grandezas elétricas monitoradas. O microcontrolador processa esses valores para determinar se condições de sobrecorrente ou sobretensão estão presentes, ao mesmo tempo, a cada 1 segundo os dados apresentados no display são atualizados. Em caso de detecção de uma condição anômala, o ESP32 aciona o LED indicador da falta correspondente e atualiza os dados do display.

#### 4.4 Adequação dos Módulos Sensores

Uma etapa importante para correta conversão dos sinais adquiridos pelos módulos sensores é a calibração dos mesmos. Com o processo de calibração é possível garantir que a amplitude dos sinais adquiridos pelos módulos representará corretamente os valores de tensão e corrente reais do sistema.

O processo de calibração consiste em realizar o seguinte procedimento para diferentes valores de corrente e tensão: coletar um valor da saída do ADC do ESP32, que é representado por valores entre 0 e 4095. O valor fornecido no ADC é comparado ao respectivo valor de tensão ou corrente real do sistema, que são obtidos através de multímetros digitais, nas funções de voltímetro e amperímetro.

Para concluir o processo de calibração de um módulo sensor, os valores coletados são inseridos na função `multimap()` presente no código de programação do relé de proteção, que realiza a interpolação de valores em um conjunto pré-definido de pontos de referência. A função `multimap()` associa um valor de entrada a um valor correspondente em uma tabela de mapeamento, permitindo converter a leitura bruta do sensor (em tensão ou corrente, por exemplo) para o valor físico equivalente, como amperes ou volts. Quando o valor de entrada não corresponde exatamente a um dos pontos da tabela, a função interpola linearmente entre os dois pontos mais próximos, garantindo maior precisão na interpretação dos dados do sensor.

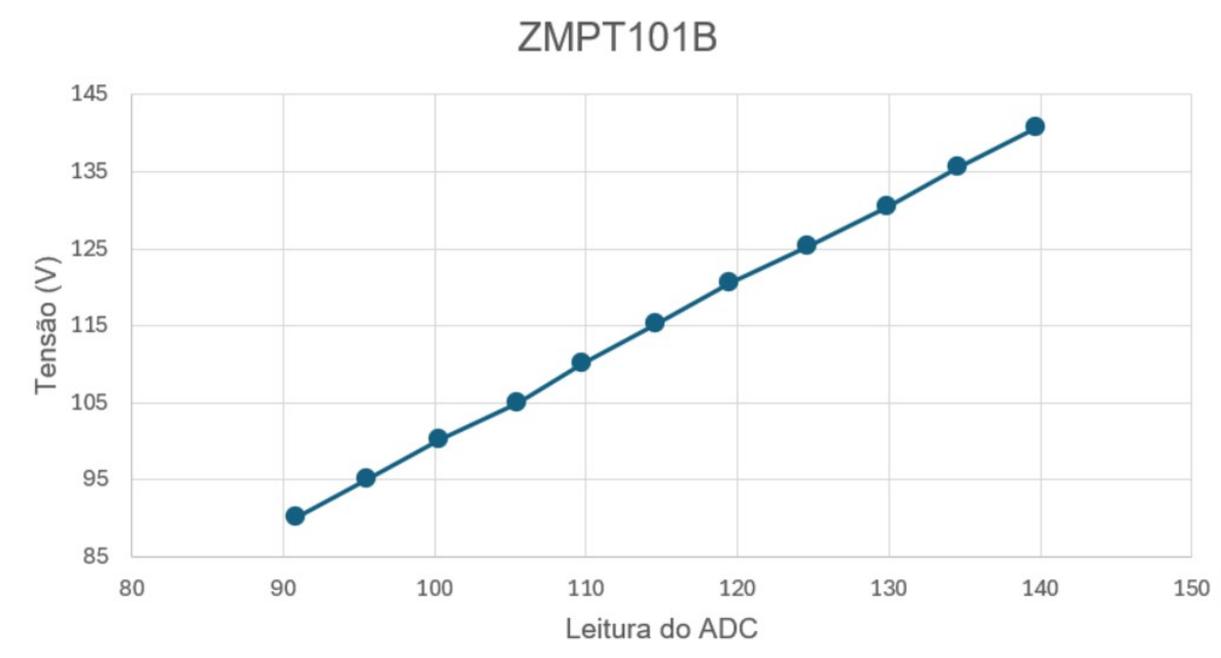
Essa funcionalidade é essencial para ajustar as leituras às características não-lineares do sensor e corrigir discrepâncias que possam ocorrer devido a fatores como tolerâncias nos componentes ou ruídos na medição. No contexto do relé de proteção, a `multimap()` permite que os valores medidos pelo sensor sejam convertidos para grandezas normalizadas e precisas, possibilitando o processamento correto das variáveis e a tomada de decisão confiável pelo sistema de proteção.

Com estes dados coletados é possível apresentar a linearidade de cada um dos módulos sensores dentro da faixa de interesse de monitoramento de corrente e tensão. Para corrente, os valores para calibração do sensor foram obtidos com valores de corrente entre 0 A e 5 A. Para tensão, os valores para calibração do sensor foram obtidos com valores de tensão entre 90 V e 140 V.

As faixas de interesse para calibração de corrente e tensão foram definidas conforme a relevância dos níveis de tensão e corrente medidos. Para o caso da tensão, o interesse é uma variação de 10% a partir do valor de 115 V, que é definido devido ao fato de 115 V ser o valor usual para transformadores de potência aplicados na função de proteção de sistemas elétricos de potência. O mesmo critério de aplicação para transformadores de corrente em SEPs foi utilizado para definição da faixa de interesse da calibração do sensor de corrente, pois TCs são dimensionados para que o maior nível de curto seja apresentado com no máximo 5 A no secundário do transformador.

A linearidade dos módulos sensores ACS712 e ZMPT101B pode ser observada respectivamente na Figura 8 e Figura 7, apresentados a seguir:

Figura 7 – Linearidade do sensor ZMPT101B

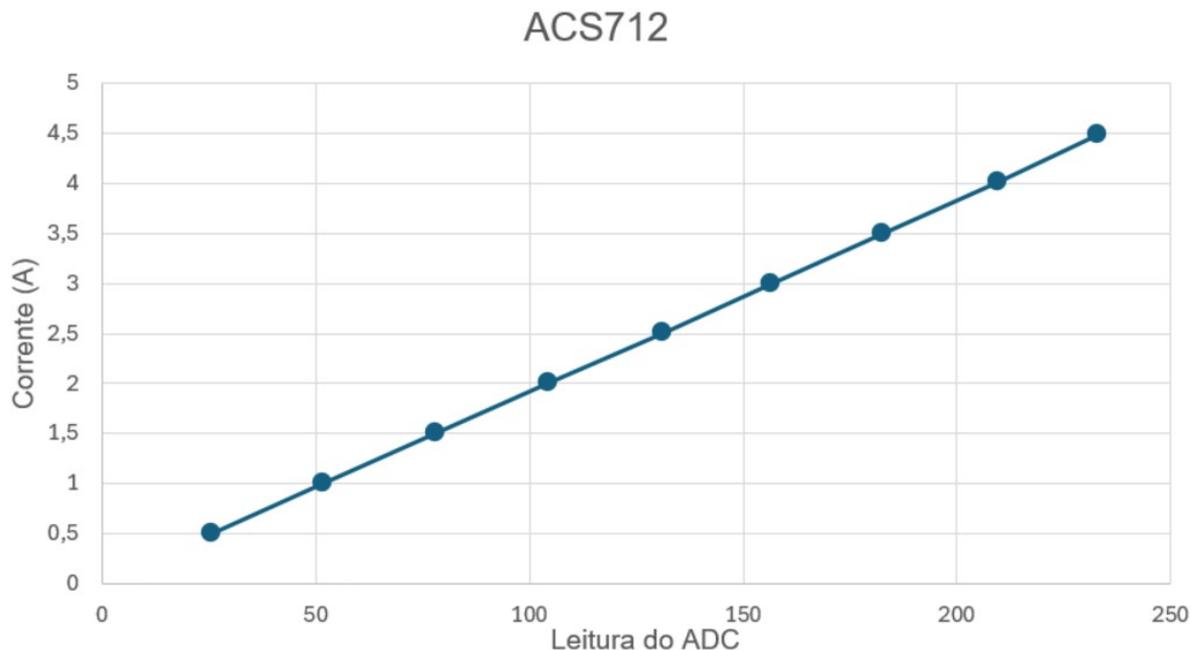


Fonte: Os Autores, (2024).

#### 4.5 Descrição do Código

Nesta seção, detalha-se o desenvolvimento do código de controle para o microcontrolador ESP32, implementado para realizar a leitura de grandezas elétricas (tensão e corrente), calcular os valores RMS, monitorar picos de corrente e atuar na proteção do circuito em caso de condições de falta. O código é escrito na linguagem C++ e programado na Arduino IDE, com uma taxa de comunicação serial configurada para visualização dos dados via SerialPlot.

Figura 8 – Linearidade do sensor ACS712



Fonte: Os Autores, (2024).

#### 4.5.1 Leitura e Adequação do Sinal

A leitura e adequação do sinal é uma etapa essencial para a obtenção de valores precisos das grandezas elétricas, como tensão e corrente, no relé de proteção. A partir dos sensores de tensão e corrente, os sinais analógicos são convertidos para valores digitais utilizando o ADC do microcontrolador. Esses valores são então processados para calcular o valor RMS das grandezas, garantindo que as leituras representem adequadamente as condições reais do sistema.

O código implementado realiza a leitura dos sinais e aplica técnicas de ajuste, como um filtro considerando uma média de 20 amostras, compensação de *offset* e cálculo RMS. As funções `lerTensao()` e `lerCorrente()` são responsáveis por coletar as leituras do ADC, filtrar e calcular os valores médios:

```
1 double lerTensao(int numAmostras) {
2     double soma = 0;
3     for (int i = 0; i < numAmostras; i++) {
4         int valorADC = analogRead(SENSOR_T) - off_set_tensao;
5         soma += valorADC;
6     }
7     return ((soma / numAmostras) + off_set_tensao);
```

```

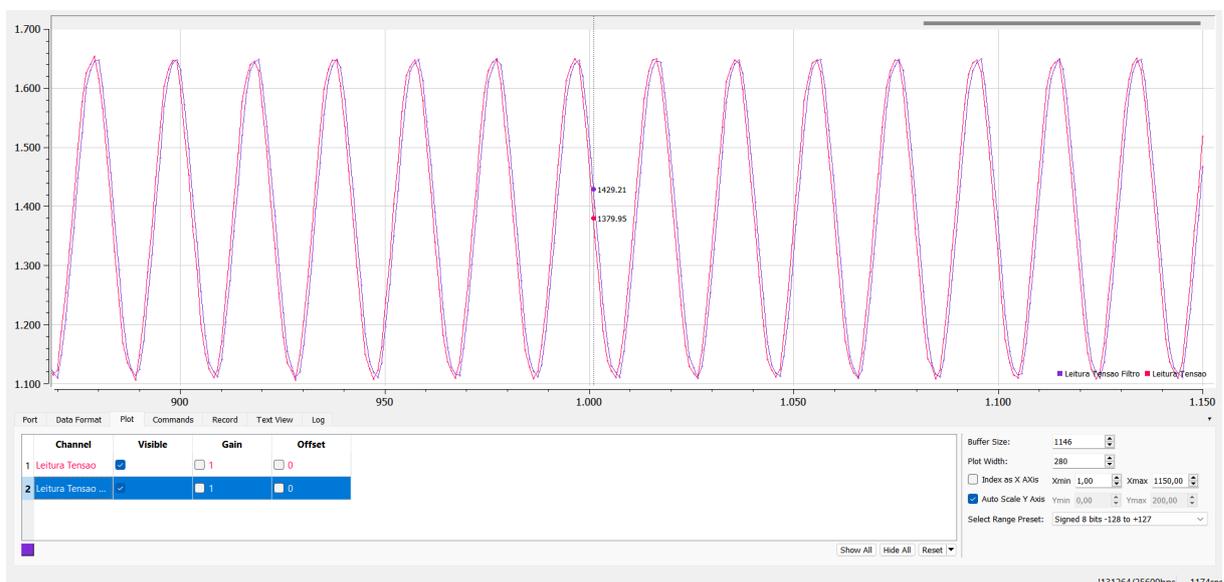
8 }
9
10 double lerCorrente(int numAmostras) {
11     double soma = 0;
12     for (int i = 0; i < numAmostras; i++) {
13         int valorADC = analogRead(SENSOR_C) - off_set_corrente;
14         soma += valorADC;
15     }
16     return ((soma / numAmostras) + off_set_corrente);
17 }

```

Código 4.1 – Função de leitura de tensão e corrente

A Figura 9 apresenta a leitura de tensão e também o valor filtrado após a média de 20 amostras.

Figura 9 – Leitura de tensão e valor filtrado



Fonte: Os Autores, (2024).

O *offset* é uma correção aplicada às leituras do ADC para eliminar componentes de ruído ou valores de base indesejados introduzidos pelos sensores. Para determinar o *offset* de tensão, o sistema calcula a média quadrática das leituras iniciais do ADC:

```

1 if(amostras < 1000) {
2     soma_quadrados = ((leitura_tensao) * (leitura_tensao)) +
3     soma_quadrados;

```

```

3     amostras++;
4 } else {
5     off_set_tensao = sqrt(soma_quadrados / amostras);
6     soma_quadrados = 0;
7     amostras = 0;
8 }

```

Código 4.2 – Cálculo do offset da tensão

Após ajustar o *offset*, o código executa o cálculo do valor RMS da tensão, levando em consideração as variações no sinal lido. A seguinte rotina implementa esse cálculo:

```

1  if(amostras_2 < 500) {
2      soma_quadrados_2 = ((leitura_tensao - off_set_tensao) *
3                          (leitura_tensao - off_set_tensao)) +
4                          soma_quadrados_2;
5      amostras_2++;
6  } else {
7      tensao = sqrt(soma_quadrados_2 / amostras_2); // Calcula o
8              valor RMS
9      soma_quadrados_2 = 0;
10     amostras_2 = 0;
11 }
12 // Ajusta o valor RMS para a escala de medicao
13 tensao_RMS = multiMap<float, float>(tensao, in, out, 11);

```

Código 4.3 – Cálculo do valor RMS da tensão

O valor RMS ajustado é então processado pela função `multiMap()`, que realiza a interpolação para mapear o valor medido para uma escala realista, compatível com as características do sistema monitorado. Isso garante a precisão dos valores utilizados pelo relé de proteção. O método foi utilizado para calibração e cálculo RMS da corrente.

Essa abordagem combina técnicas de calibração e processamento para garantir que as leituras dos sensores sejam confiáveis e adequadas para as análises realizadas pelo sistema de proteção.

## 4.5.2 Verificação de Corrente de Pico

A verificação de corrente de pico foi desenvolvida devido a velocidade de atuação ser mais rápida com esse método e não considerando o RMS, por não ser necessário realizar média. O objetivo é identificar e registrar o maior valor instantâneo de corrente durante um ciclo, que pode ser utilizado para análise e tomadas de decisão no sistema de proteção. O código implementado realiza essa verificação utilizando comparações sucessivas entre os valores lidos do sensor de corrente.

O trecho de código abaixo mostra a lógica utilizada para detectar o maior valor de corrente de pico:

```
1  if (leitura_corrente > off_set_corrente && leitura_corrente <
    leitura_corrente_fase_anterior) {
2      corrente_pico = leitura_corrente_fase_anterior; // Detector
        de pico
3      if (corrente_pico > maior_corrente_pico) {
4          maior_corrente_pico = corrente_pico; // Detector do
            maior valor de pico
5      }
6  }
7  if (leitura_corrente < off_set_corrente) {
8      if (maior_corrente_pico != 0) {
9          maior_corrente_pico_anterior = maior_corrente_pico;
10     }
11     maior_corrente_pico = 0; // Reinicia o valor de maior
        corrente de pico
12 }
13 leitura_corrente_fase_anterior = leitura_corrente; // Atualiza a
    corrente anterior
```

Código 4.4 – Verificação de corrente de pico

Descrição do Funcionamento:

1. Detecção de pico de corrente: O código verifica se o valor atual de leitura da corrente (*leitura\_corrente*) está acima do *offset* (*off\_set\_corrente*) e é menor do que a leitura anterior (*leitura\_corrente\_fase\_anterior*). Quando essas condições são

satisfeitas, o sistema considera que um pico foi detectado e armazena o valor da corrente anterior como `corrente_pico`.

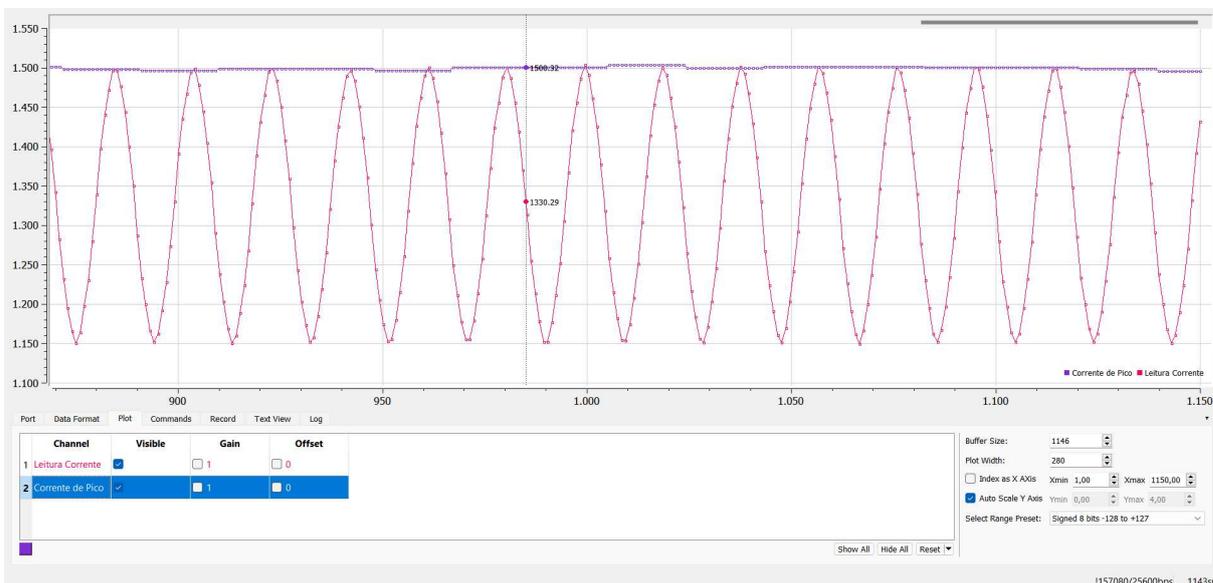
2. Registro do maior valor de pico: Após detectar um pico, o código verifica se o valor atual de `corrente_pico` é maior que o maior valor registrado anteriormente (`maior_corrente_pico`). Caso seja, o maior valor é atualizado.

3. Reinício ao cruzar o zero: Quando a leitura da corrente fica abaixo do `offset`, indicando que o sinal passou pelo zero, o sistema registra o maior valor de pico atual como o `maior_corrente_pico_anterior` e reinicia o valor de `maior_corrente_pico` para zero, preparando-se para o próximo ciclo.

4. Atualização da corrente anterior: No final de cada ciclo, o valor atual de `leitura_corrente` é armazenado em `leitura_corrente_fase_anterior` para ser utilizado na próxima verificação.

A Figura 10 demonstra o funcionamento desta função.

Figura 10 – Leitura de corrente e detecção do pico



Fonte: Os Autores, (2024).

#### 4.5.3 Verificação das Condições de Operação do Sistema

A verificação das condições de operação do sistema é responsável por monitorar as grandezas elétricas (tensão e corrente) e determinar a atuação das funções de proteção. Esse processo avalia as condições de sobrecorrente (instantânea e temporizada), sobretensão

e subtensão, sinalizando o status do sistema e acionando os relés de proteção quando necessário.

O código abaixo apresenta a lógica implementada para verificar e atuar nas diferentes condições:

```
1  if (corrente_RMS > CORRENTE_AJUSTE) {
2      status = 1;
3
4      if (!monitorandoSobrecorrente) {
5          instanteInicioSobrecorrente = millis(); // Armazena o
6              instante em que a sobrecorrente comecou
7          monitorandoSobrecorrente = true;
8          corrente_PK = corrente_RMS;
9      }
10
11     tempoAtuacao = (0.14 / (pow((corrente_RMS / CORRENTE_AJUSTE),
12         0.02) - 1)) * DIALTIME;
13     tempoAtuacao *= 1000; // Convertendo para milissegundos
14
15     // Prote o de sobrecorrente instantanea
16     if (corrente_RMS >= CORRENTE_INSTANTANEA_LIMITE) {
17         atuarSobrecorrenteInstantanea(); // Atua instantaneamente
18             por sobrecorrente
19         status = 5;
20     }
21     else if (millis() - instanteInicioSobrecorrente >=
22         tempoAtuacao) {
23         status = 6;
24         atuarSobrecorrenteTemporizada(); // Atua por sobrecorrente
25             temporizada
26     }
27     else {
28         digitalWrite(PINO_LED_PICKUP, HIGH); // Sinaliza que a
29             corrente de ajuste foi ultrapassada
30         status = 4;
31     }
32 }
```

```

25     }
26 }
27 else {
28     // Corrente voltou ao normal antes de atuar
29     monitorandoSobrecorrente = false;
30     digitalWrite(PINO_LED_PICKUP, LOW); // Desliga LED de pickup
31 }
32
33 if (tensao_RMS > TENSAO_LIMITE_SUPERIOR) {
34     instanteInicioSobretensao = millis();
35     atuarSobretensao(); // Atua por sobretens o
36     status = 2;
37 }
38 else if (tensao_RMS < TENSAO_LIMITE_INFERIOR) {
39     instanteInicioSubtensao = millis();
40     atuarSubtensao(); // Atua por subtens o
41     status = 3;
42 }
43
44 if ((corrente_RMS <= CORRENTE_AJUSTE) &&
45     (tensao_RMS >= TENSAO_LIMITE_INFERIOR) &&
46     (tensao_RMS <= TENSAO_LIMITE_SUPERIOR)) {
47     digitalWrite(PINO_LED_VERDE, HIGH);
48     digitalWrite(PINO_LED_SOBRETENSAO, LOW);
49     digitalWrite(PINO_LED_SUBTENSAO, LOW);
50     digitalWrite(PINO_LED_PICKUP, LOW);
51     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
52     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
53 }

```

Código 4.5 – Verificação das condições de operação do sistema

#### Descrição do Funcionamento

##### 1. Monitoramento de sobrecorrente:

- Quando a corrente RMS excede o valor de ajuste (`CORRENTE_AJUSTE`), a variável

`status` é ajustada para indicar que uma condição de sobrecorrente foi detectada.

- O instante em que a sobrecorrente foi detectada é armazenado com `millis()`, e a corrente RMS é comparada com o limite de sobrecorrente instantânea.

- Se a corrente ultrapassar esse limite, a função `atuarSobrecorrenteInstantanea()` é chamada, e o sistema sinaliza a atuação instantânea.

- Caso contrário, a lógica verifica se o tempo calculado para atuação temporizada foi excedido e, se sim, chama a função `atuarSobrecorrenteTemporizada()`.

## 2. Monitoramento de sobretensão e subtensão:

- Se a tensão RMS (`tensao_RMS`) ultrapassar o limite superior (`TENSAO_LIMITE_SUPERIOR`), a função `atuarSobretensao()` é acionada.

- De forma semelhante, se a tensão RMS estiver abaixo do limite inferior, a função `atuarSubtensao()` é chamada.

## 3. Condição normal do sistema:

- Se os valores de corrente e tensão estiverem dentro dos limites normais, os LEDs indicativos de falhas (sobrecorrente, sobretensão e subtensão) são desligados, e o LED verde é ativado para indicar que o sistema está operando normalmente.

A lógica implementada permite que o sistema de proteção identifique, classifique e atue em diferentes condições de operação, como sobrecorrente, subtensão e sobretensão. Essa abordagem garante a confiabilidade do sistema e protege os equipamentos contra falhas elétricas, além de fornecer indicações visuais do estado atual do sistema por meio dos LEDs.

### 4.5.4 Atuação

As funções de atuação são responsáveis por sinalizar e pausar o sistema em caso de detecção de condições anormais, como sobretensão, subtensão, sobrecorrente instantânea e sobrecorrente temporizada. Cada condição possui uma função dedicada que realiza as seguintes ações: - Sinaliza a falha através de LEDs indicativos. - Desativa os LEDs de funcionamento normal ou de outras falhas. - Pausa o sistema até que um reset manual seja realizado.

Abaixo, as funções de atuação implementadas:

```
1 void atuarSobretensao() {
2     digitalWrite(PINO_LED_SOBRETENSAO, HIGH); // Acende LED de
3         sobretens o
4     digitalWrite(PINO_LED_VERDE, LOW);
5     digitalWrite(PINO_LED_SUBTENSAO, LOW);
6     digitalWrite(PINO_LED_PICKUP, LOW);
7     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
8     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
9     while (true); // Sistema pausado at reset
10 }
11 void atuarSubtensao() {
12     digitalWrite(PINO_LED_SUBTENSAO, HIGH); // Acende LED de
13         subtens o
14     digitalWrite(PINO_LED_VERDE, LOW);
15     digitalWrite(PINO_LED_SOBRETENSAO, LOW);
16     digitalWrite(PINO_LED_PICKUP, LOW);
17     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
18     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
19     while (true); // Sistema pausado at reset
20 }
21 void atuarSobrecorrenteInstantanea() {
22     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, HIGH); //
23         Acende LED de sobrecorrente instant nea
24     digitalWrite(PINO_LED_VERDE, LOW);
25     digitalWrite(PINO_LED_SOBRETENSAO, LOW);
26     digitalWrite(PINO_LED_SUBTENSAO, LOW);
27     digitalWrite(PINO_LED_PICKUP, LOW);
28     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
29     while (true); // Sistema pausado at reset
30 }
31 void atuarSobrecorrenteTemporizada() {
```

```

32     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA , HIGH); //
        Acende LED de sobrecorrente temporizada
33     digitalWrite(PINO_LED_VERDE , LOW);
34     digitalWrite(PINO_LED_SOBRETENSAO , LOW);
35     digitalWrite(PINO_LED_SUBTENSAO , LOW);
36     digitalWrite(PINO_LED_PICKUP , LOW);
37     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA , LOW);
38     instanteFimSobrecorrente = millis(); // Armazena o instante
        do fim
39     tempoAtuacaoReal = (instanteFimSobrecorrente -
        instanteInicioSobrecorrente) / 1000.0; // Converte para
        segundos
40     mostrarDisplay();
41     while (true); // Sistema pausado at reset
42 }

```

Código 4.6 – Funções de atuação para falhas no sistema

1. `atuarSobretensao()`: - Sinaliza a ocorrência de uma sobretensão acendendo o LED correspondente (`PINO_LED_SOBRETENSAO`). - Desliga os demais LEDs indicativos e pausa o sistema até que um reset manual seja realizado.

2. `atuarSubtensao()`: - Similar à função anterior, mas destinada a sinalizar uma condição de subtensão.

3. `atuarSobrecorrenteInstantanea()`: - É acionada quando a corrente RMS ultrapassa o limite definido para sobrecorrente instantânea. - Acende o LED de sobrecorrente instantânea (`PINO_LED_SOBRECORRENTE_INSTANTANEA`) e pausa o sistema.

4. `atuarSobrecorrenteTemporizada()`: - Calcula o tempo de atuação real com base no intervalo entre o início e o fim da sobrecorrente temporizada. - Acende o LED correspondente (`PINO_LED_SOBRECORRENTE_TEMPORIZADA`) e exibe os dados no display utilizando a função `mostrarDisplay()`. - O sistema é pausado após a sinalização.

As funções de atuação garantem que o sistema tome as medidas necessárias para proteger os equipamentos conectados, evitando falhas maiores. Além disso, a sinalização

visual permite que o operador identifique rapidamente o tipo de falha, facilitando a tomada de ações corretivas.

#### 4.5.5 Display

A função `mostrarDisplay()` é responsável por exibir as informações mais relevantes sobre o estado do sistema no display OLED. Essa interface fornece uma visualização clara das grandezas elétricas monitoradas (tensão e corrente RMS), do status do sistema e, quando aplicável, das condições de atuação, como corrente de *pick-up* e tempo de atuação.

Abaixo está o código da função implementada:

```
1 void mostrarDisplay() {
2
3     // Limpa e configura o display
4     display.clearDisplay();
5     display.setTextColor(WHITE);
6
7     // Exibi o da Tens o RMS
8     display.setCursor(0,8);
9     display.print("Tensao RMS:");
10    display.setCursor(70,8);
11    display.print(tensao_RMS,1);
12    display.setCursor(105,8);
13    display.print("V");
14
15    // Exibi o da Corrente RMS
16    display.setCursor(0,16);
17    display.print("Corrente RMS:");
18    display.setCursor(82,16);
19    display.print(corrente_RMS,1);
20    display.setCursor(105,16);
21    display.print("A");
22
23    // Exibi o do Status do Sistema
24    display.setTextSize(1);
25    display.setCursor(0,0);
```

```

26     display.print("Status:");
27     display.setCursor(43,0);
28     if (status == 1) {
29         display.print("Normal");
30     } else if (status == 2) {
31         display.print("F 59");
32     } else if (status == 3) {
33         display.print("F 27");
34     } else if (status == 4) {
35         display.print("Pick-up");
36         // Exibi o da Corrente de Pick-up
37         display.setCursor(0,24);
38         display.print("Corrente PKUP:");
39         display.setCursor(85,24);
40         display.print(corrente_PK,3);
41         display.setCursor(120,24);
42         display.print("A");
43     } else if (status == 5) {
44         display.print("F 50");
45     } else if (status == 6) {
46         display.print("F 51");
47         // Exibi o do Tempo de Atua o
48         display.setCursor(0,24);
49         display.print("Atuacao:");
50         display.setCursor(50,24);
51         display.print(tempoAtuacaoReal,3);
52         display.setCursor(89,24);
53         display.print("s");
54         display.display();
55         while (true);
56     }
57
58     display.display();
59 }

```

---

#### Código 4.7 – Função para exibição no display OLED

1. Exibição das grandezas elétricas: - A tensão RMS (`tensao_RMS`) e a corrente RMS (`corrente_RMS`) são exibidas nas linhas correspondentes do display, com uma precisão de uma casa decimal para tensão e corrente.

2. Exibição do status do sistema: - O status do sistema é exibido no canto superior esquerdo do display, indicando condições normais ou de falha. Cada falha é representada por um código:

- F 59: Sobretensão.
- F 27: Subtensão.
- F 50: Sobrecorrente instantânea.
- F 51: Sobrecorrente temporizada.
- Pick-up: Corrente acima do ajuste, mas sem atuação.

3. Exibição de parâmetros adicionais: - Durante condições específicas, como o estado Pick-up, o display mostra a corrente de *pick-up* registrada (`corrente_PK`). - Em caso de sobrecorrente temporizada (F 51), o tempo de atuação real (`tempoAtuacaoReal`) é exibido junto com o código da falha.

4. Atualização do display: - A função `display.display()` atualiza a tela com as informações processadas.

5. Pausa do sistema em falhas críticas: - Em falhas que exigem atuação (F 50, F 51, etc.), o sistema entra em um estado de pausa (`while (true)`), aguardando um reset manual.

A função `mostrarDisplay()` desempenha um papel essencial no fornecimento de informações em tempo real sobre o estado do sistema, auxiliando na operação e manutenção. A organização clara das informações no display permite que operadores identifiquem rapidamente condições normais e anormais.

**Nota:** O código completo está no Apêndice A.

#### 4.6 Sistema de testes

Todos os testes, para qualquer uma das funções testadas, foram realizados a partir do mesmo sistema, este estando estável nas mesmas condições iniciais.

Por se tratar de testes que dependem da variação da tensão e da corrente, foram utilizados elementos que possibilitassem o controle dessas grandezas.

Para tensão, utilizou-se um variador de tensão, que permite ajuste de tensão entre 0 V e 250 V. O variador de tensão utilizado apresenta-se na Figura 11.

Figura 11 – Variador de Tensão



Fonte: Os Autores, (2024).

Para realizar o controle da corrente que circula pelo sistema, foram utilizadas cargas puramente resistivas de resistência variável, dessa forma, o controle da resistência total da carga permitiu controlar a corrente que circulava pelo sistema. Foram utilizadas dois equipamentos de resistência variável conectados em série, dessa forma foi possível simular uma carga entre  $0\ \Omega$  e  $332\ \Omega$ .

Figura 12 – Carga variável



Fonte: Os Autores, (2024).

#### 4.6.1 Montagem do Sistema de Testes

Como citado na seção anterior, o variador de tensão e cargas variáveis foram utilizados para controlar corrente e tensão do sistema de testes, a seguir na Figura 13, é apresentada a montagem do sistema de testes.

As conexões do sistema com o relé de proteção foram realizadas conforme o esquemático apresentado na Figura 6, permitindo a realização dos testes

Pode-se afirmar que o variador de tensão simula a conexão com o secundário de um transformador de potencial, e que a carga variável permite simular o secundário de um transformador de corrente.

A alimentação do relé de proteção no momento dos testes foi feita através do ESP32 conectado a uma bateria portátil capaz de operar na tensão de funcionamento do ESP32.

#### 4.6.2 Condições Iniciais do Sistema de Testes

Como condição inicial de cada um dos testes, o variador de tensão era ajustado de forma que fornecesse tensão de interesse de cada teste.

Figura 13 – Circuito de testes



Fonte: Os Autores, (2024).

Para o caso da carga variável, esta era ajustada de forma que fornecesse a corrente necessária para cada um dos testes.

#### 4.6.3 Tempo de Atuação

Nesta seção fica definido que o termo "tempo de atuação" irá se referir ao tempo entre o instante de identificação da condição anormal do sistema pelo relé, e o instante logo após o comando de atuação do relé. A atuação do relé é representada pelo acender dos LEDs de cada função do relé de proteção.

A medição da passagem do tempo foi realizada com a função `millis()`, que registra o instante de tempo em milissegundos desde que o microcontrolador foi iniciado. A lógica para medição entre o tempo que o relé de proteção identifica a falta e o instante que o relé envia o sinal da atuação é descrita da seguinte forma:

1. Quando o relé de proteção identifica que a corrente de ajuste foi ultrapassada, este armazena este instante de tempo presente através da função `millis()`, esse instante de

tempo é armazenado na variável "instanteInicioSobrecorrente";

2. O tempo exigido para atuação pela curva IEC Normal Inversa é calculado através de sua equação e armazenado na variável "tempoAtuacao";

3. O código então verifica continuamente que a diferença entre o instante de tempo atual e o instante de identificação da falta é menor ou igual ao tempo de atuação calculado para curva Normal Inversa, no código, essa verificação é declarada como "millis () - instanteInicioSobrecorrente >= tempoAtuacao";

4. Caso a condição "millis () - instanteInicioSobrecorrente >= tempoAtuacao" seja verdadeira, junto aos demais comandos de atuação, a função millis() é novamente utilizada para calcular a diferença "millis () - instanteInicioSobrecorrente", o resultado desta equação é atribuído na variável "tempoAtuacaoReal".

A variável "tempoAtuacaoReal" fornece então, o tempo em milissegundos entre identificação da corrente de ajuste de pickup e o instante de tempo logo após todos os sinais para atuação terem sido enviados, este tempo é referido como tempo de atuação no presente capítulo e no capítulo a seguir, que apresenta os resultados de testes efetuados com o relé de proteção.

## 5 RESULTADOS

Com a realização dos testes em laboratório foi possível entender o funcionamento do relé de proteção desenvolvido bem como suas limitações. A seguir serão apresentados os resultados para cada uma das situações que pode causar a atuação do relé de proteção.

### 5.1 Protótipo final

O protótipo final foi desenvolvido visando a integração dos componentes eletrônicos utilizados no sistema de proteção em uma montagem compacta, confiável e organizada. Para isso, foi utilizada uma placa de circuito impresso de fenolite, que permitiu a soldagem dos componentes de forma permanente e segura, garantindo a robustez mecânica e elétrica do circuito.

A escolha da placa de fenolite ilhada foi motivada por sua versatilidade e simplicidade no processo de montagem, especialmente em projetos de protótipos. Cada ilha na superfície da placa atua como um ponto de conexão para os terminais dos componentes, permitindo a interligação dos elementos por meio de fios ou pontes de solda. Esse método proporciona maior liberdade para alterações e ajustes durante o desenvolvimento, além de facilitar o diagnóstico e reparo em caso de falhas.

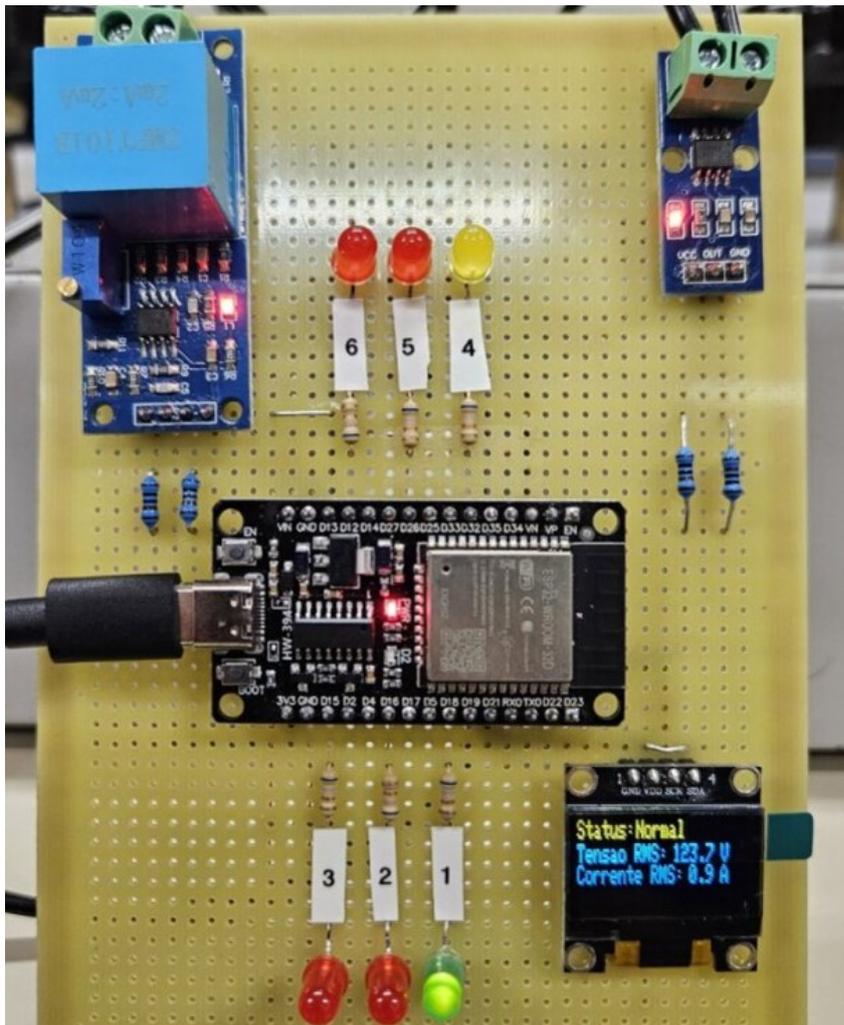
Os sensores de corrente e tensão, bem como o microcontrolador ESP32 e os LEDs indicativos de status, foram posicionados estrategicamente na placa para minimizar o comprimento das conexões e reduzir possíveis interferências eletromagnéticas. As trilhas e conexões foram cuidadosamente planejadas para garantir a integridade dos sinais medidos e processados pelo sistema. O protótipo final está demonstrado na Figura 14.

### 5.2 Subtensão - Função 27

Como definido anteriormente, a atuação do sistema por subtensão, função 27, deve acontecer quando a tensão do sistema estiver abaixo 90% do valor de referência. Como a tensão de referência foi definida como 115 V, o relé deve atuar para função 27, com tensões abaixo de 103,5 V.

Em algumas aplicações em sistemas elétricos de potência, os relés de subtensão ou sobretensão são ajustados de forma que haja um atraso definido no tempo de atuação.

Figura 14 – Protótipo final



Fonte: Os Autores, (2024).

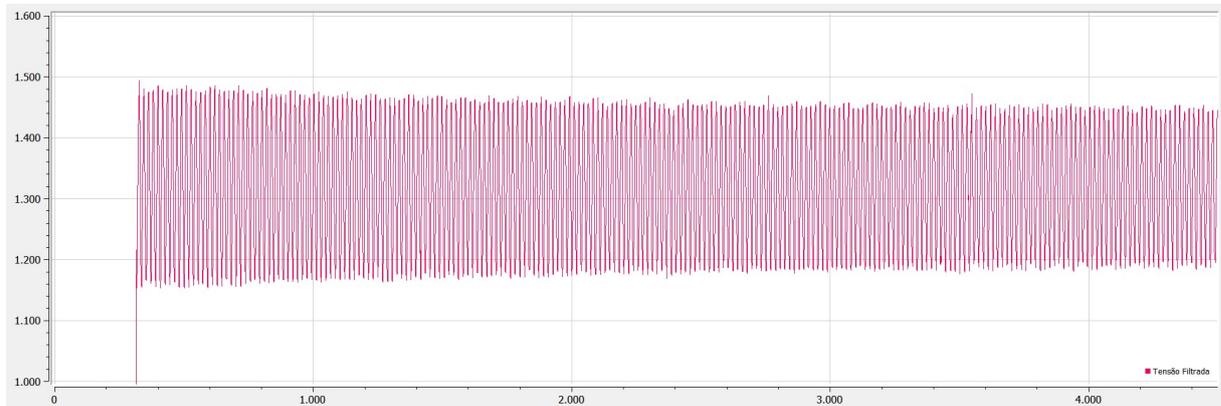
No caso deste relé, o atraso de atuação citado não foi implementado, ou seja, quando um evento de subtensão o sobretensão é identificado pelo relé de proteção, este manda instantaneamente o sinal para atuação.

Dois diferentes testes foram realizados para função de subtensão. O teste 1 com a tensão do sistema inicialmente na tensão de referência, com diminuição contínua no nível de tensão até atuação do relé de proteção. No teste 2 o sistema era ajustado para iniciar com valor de 103,5 V. Em ambos os testes, a carga do sistema era ajustada para a mais alta possível, impondo a circulação de baixos níveis de corrente e evitando o aquecimento da carga.

Para realização do teste 1, a tensão inicial foi ajustada em 115 V e então o nível de tensão era continuamente diminuído até que a atuação do relé acontecesse, após a indicação

de atuação por subtensão ser indicada pelo relé, a tensão continuou a ser diminuída até os limites utilizados na calibração dos módulos para assegurar que não haveria funcionamento anormal nestas situações.

Figura 15 – Sinal de Subtensão Imposta no Sistema



Fonte: Os Autores, (2024).

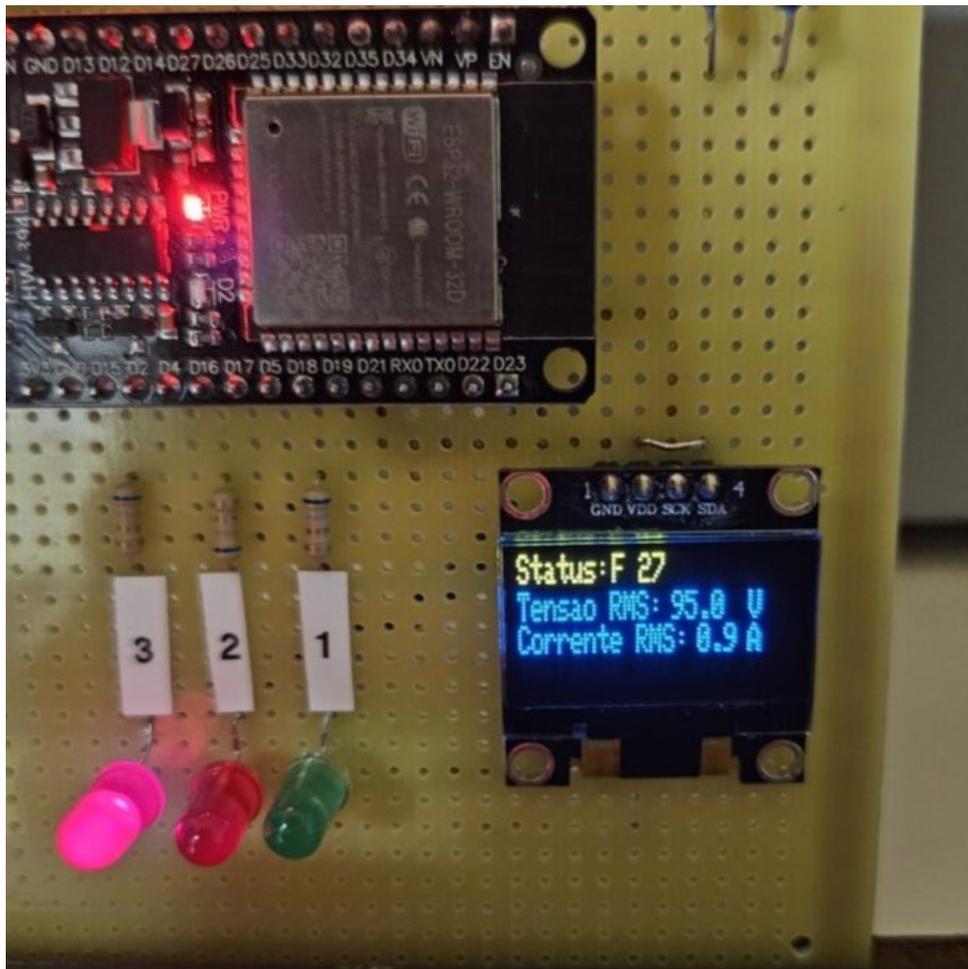
A Figura 15 apresenta a queda da tensão imposta no sistema durante este teste. A forma de onda foi obtida através do próprio relé de proteção e apresentada no SerialPlot, o eixo horizontal representa a quantidade de amostras obtidas pelo módulo sensor, enquanto o eixo vertical representa o valor da saída do ADC, que é representados por números de 0 a 4095.

A Figura 16 apresenta as informações apresentadas no display durante um dos testes de subtensão.

Para realização do teste 2, como citado, o sistema era ajustado com tensão de 103,5 V e então o relé era reiniciado para que fosse possível observar o comportamento no limiar da atuação. Neste caso, houve variação no comportamento do relé de proteção. Para a tensão do sistema de 103,5 V, cada vez que o relé de proteção era reiniciado, a leitura de tensão realizada por ele iniciava-se em valores entre 103,2 V e 103,8 V.

Os resultados do teste 2, podem ser utilizados para questionar a exatidão da atuação do teste 1 em 103,5 V. Apesar disso a variação de 0,3 V em relação aos 103,5 V, representa um erro de 0,3% na precisão da leitura de tensão do sistema no ponto de atuação de subtensão. Isso representa que, no relé desenvolvido, para um ajuste de atuação de subtensão de 90% da tensão de referência, no pior caso, o relé de proteção irá atuar para subtensão com 89,7% da tensão de referência.

Figura 16 – Teste de subtensão



Fonte: Os Autores, (2024).

### 5.3 Sobretensão - Função 59

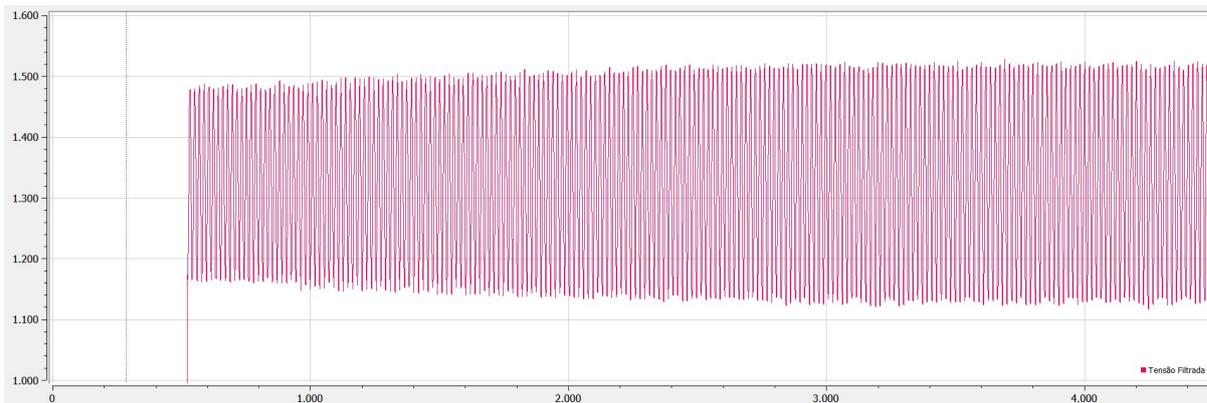
Como definido anteriormente, a atuação do sistema por sobretensão, função 59, deve acontecer quando a tensão do sistema ultrapassar 110% do valor de referência. Como a tensão de referência foi definida como 115 V, o relé deve atuar para função 59, com tensões acima de 126,5 V.

De forma similar ao realizado para função 27, foram executados os mesmos 2 testes, com a diferença que para sobretensão a atuação deve ocorrer em 126,5 V.

Para o teste 1, em que o relé de proteção era inicializado com o sistema com tensão de 115 V e então a tensão era aumentada até que acontecesse a sinalização de atuação, os resultados mostraram-se positivos e todas as atuações ocorreram, assim como nos testes equivalentes para subtensão.

A Figura 17 apresenta a sobretensão imposta no sistema durante este teste. A

Figura 17 – Sinal de Sobretensão Imposta no Sistema



Fonte: Os Autores, (2024).

forma de onda foi obtida através do próprio relé de proteção e apresentada no SerialPlot. Como apresentado na figura 15, o eixo horizontal representa a quantidade de amostras obtidas pelo módulo sensor, enquanto o eixo vertical representa o valor da saída do ADC, que é representados por números de 0 a 4095.

A Figura 18 foi obtida no momento de um dos testes de sobretensão.

Para o teste 2, em que o relé foi inicializado quando a tensão do sistema era de 126,5 V, o resultado foi que a leitura realizada pelo relé variou de 126,3 V até 126,5 V. Portanto, apesar do erro de 0,2% na aquisição de tensão, não houve atuação do relé de proteção já que a tensão não ultrapassou os 126,5 V.

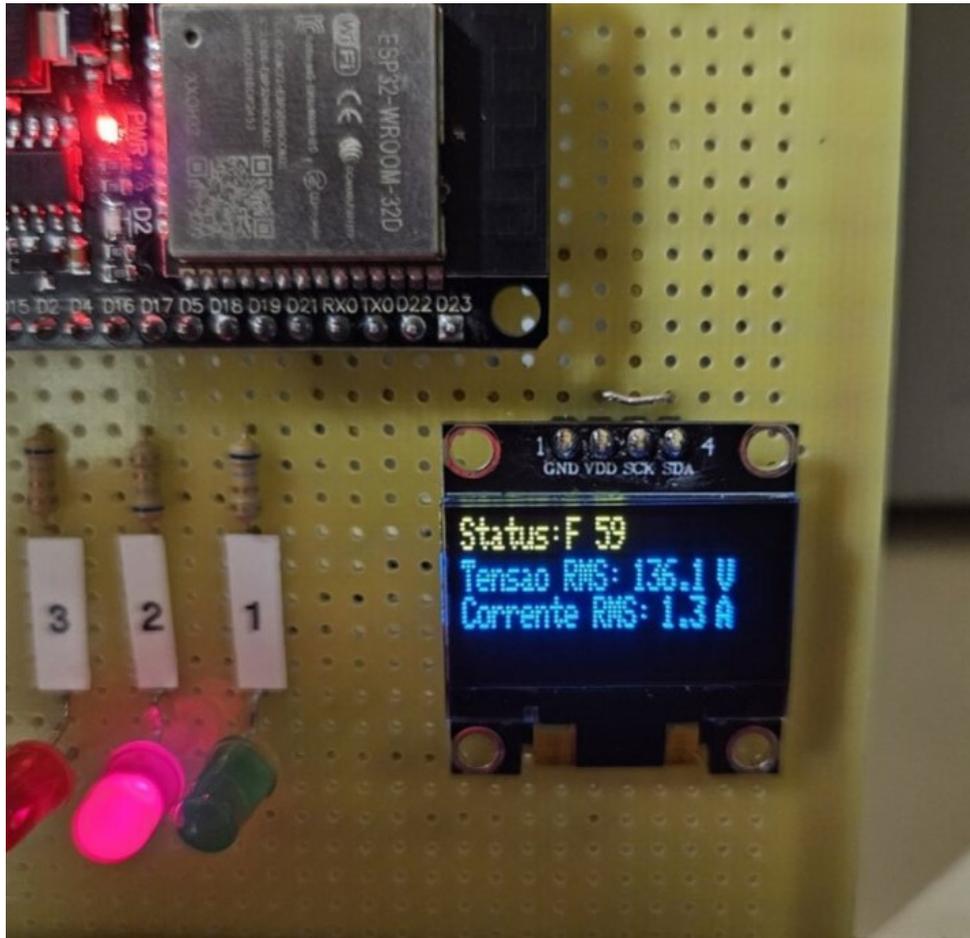
#### 5.4 Sobrecorrente Instantânea - Função 50

A atuação por sobrecorrente instantânea, função 50, deve ocorrer sempre que a corrente do sistema for superior a uma corrente máxima determinada e programada no relé. Para os testes realizados em laboratório, por questões de segurança, optou-se em definir a corrente de atuação da função 50 como 4 A.

Iniciou-se então o teste com o sistema estabilizado com corrente de 0,5 A e tensão de 115 V. O próximo passo foi diminuir a carga do sistema lentamente até que a corrente de 4 A fosse ultrapassada e o relé de proteção atuasse.

A Figura 19 apresenta o aumento da corrente imposto no sistema durante este teste. A forma de onda foi obtida através do próprio relé de proteção e apresentada no SerialPlot. Como nos casos anteriores, o eixo horizontal representa a quantidade de amostras obtidas

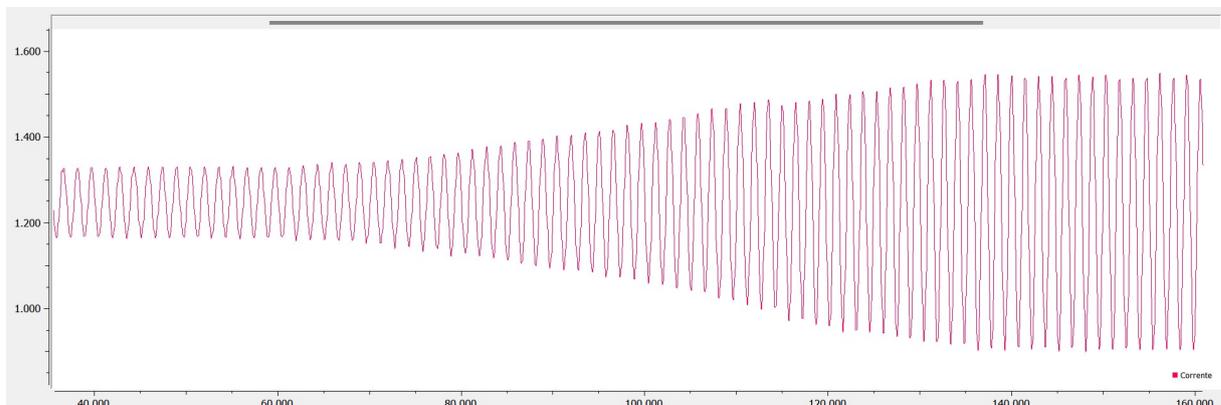
Figura 18 – Teste de sobretensão



Fonte: Os Autores, (2024).

pelos módulos sensor, enquanto o eixo vertical representa o valor da saída do ADC, que é representado por números de 0 a 4095.

Figura 19 – Sinal de Sobrecorrente Instantânea Imposta no Sistema



Fonte: Os Autores, (2024).

Para este teste os resultados obtidos ocorreram como esperado, com o relé de proteção apresentando a condição de atuação instantaneamente. A Figura 20 foi obtida

durante um dos testes de atuação da função 50 e mostra o display após a atuação do relé de proteção por corrente instantânea.

Figura 20 – Teste de sobrecorrente instantânea



Fonte: Os Autores, (2024).

Outro teste realizado para a função 50, foi a inicialização do relé com o sistema apresentando corrente de 4,5 A enquanto o ajuste para corrente instantânea era mantido em 4 A. Assim como no primeiro teste da função 50, os resultados obtidos foram atuações instantâneas em todas as tentativas.

### 5.5 Sobrecorrente Temporizada - Função 51

A função de sobrecorrente temporizada foi testada para Curva IEC Normal Inversa com múltiplos de tempo de valores 0,1 e 1. Abaixo é apresentada a equação inserida na programação do relé de proteção para atuação da função 51.

$$t = \frac{0,14}{M^{0,02} - 1} \cdot k$$

Onde t é o tempo de atuação para determinado valor de M, sendo M a razão entre a corrente do sistema e o valor de corrente do ajuste de pickup, k é o múltiplo de tempo da curva IEC.

Figura 21 – Teste de Pick-up



Fonte: Os Autores, (2024).

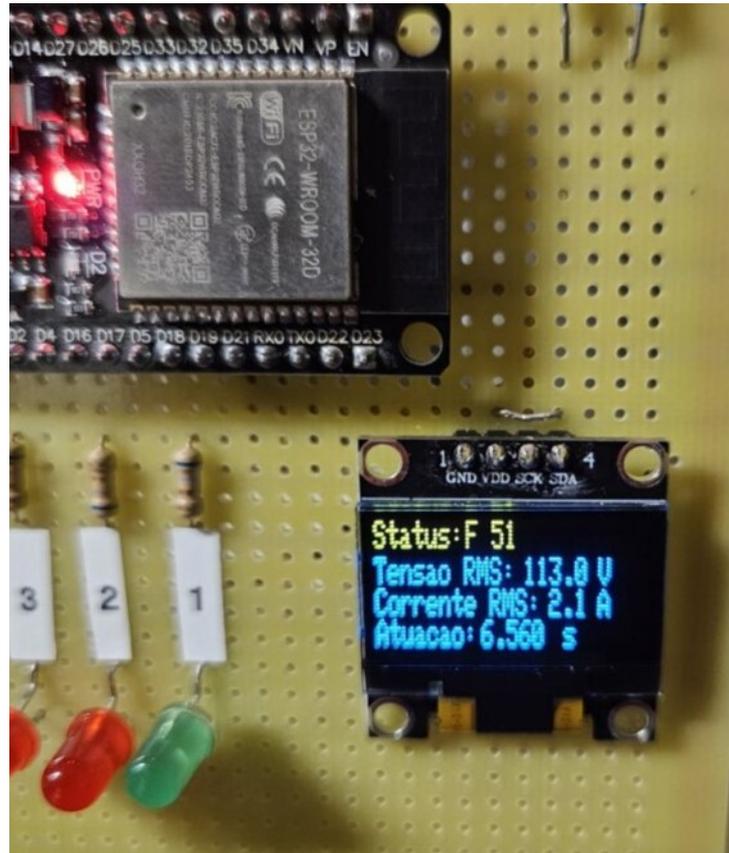
Para os testes realizados para função 51, foi definida uma corrente de ajuste de pickup de 1,5 A. As resistências variáveis, responsáveis por alterar a corrente do sistema foram ajustadas para que quando o relé de proteção fosse inicializado, a corrente do sistema fosse a desejada.

Em ambos os testes de tempo de atuação, para  $k = 1$  e  $k = 0,1$ , os valores de ajustados de corrente no instante de inicialização do relé foram os mesmos. A Figura 21 mostra o display quanto a corrente ultrapassa o valor de pick-up e a Figura 22 mostra o display no momento de atuação por corrente temporizada.

A tabela 4 apresenta os valores de corrente utilizados nos testes e também os tempos de atuação esperados e obtidos para cada valor do múltiplo  $k$ .

Como observado na tabela, os tempos obtidos na atuação do relé de proteção desenvolvido são extremamente próximos dos tempos definidos pela curva IEC Normal Inversa, diferenciando-se por no máximo 2 ms, este atraso acontece devido a taxa de amostragem que o sistema apresenta, na casa de 1000 amostras por segundo. A Figura 23 e Figura 24, apresentam os tempos de atuação do relé de proteção obtidos nos testes

Figura 22 – Teste de sobrecorrente temporizada



Fonte: Os Autores, (2024).

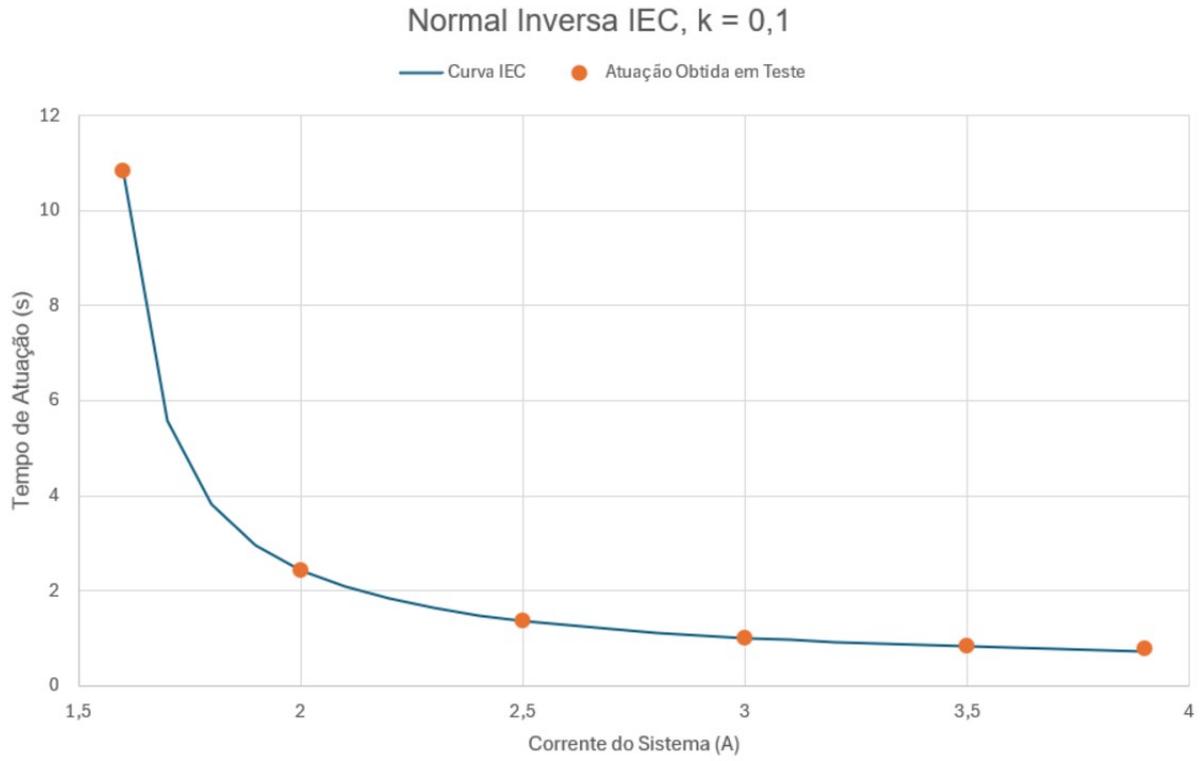
Tabela 4 – Comparação dos tempos de atuação (IEC vs. Obtido) para diferentes valores de corrente e múltiplos de tempo ( $k = 1$  e  $k = 0, 1$ ).

Corrente	k = 1		k = 0, 1	
	IEC	Obtido	IEC	Obtido
1,6	108,392	108,393	10,839	10,840
2,0	24,262	24,263	2,426	2,428
2,5	13,633	13,634	1,363	1,365
3,0	10,029	10,031	1,003	1,004
3,5	8,192	8,194	0,819	0,821
3,9	7,256	7,257	0,726	0,727

Fonte: Os Autores, (2024).

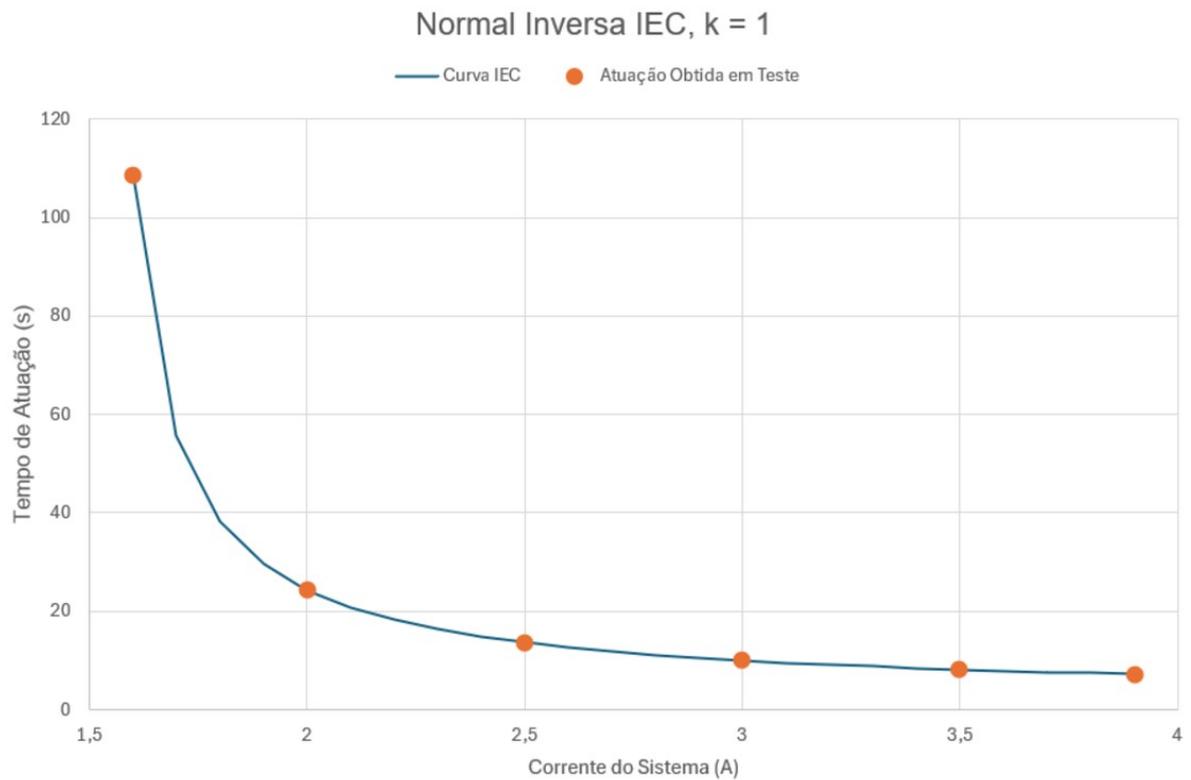
juntamente da curva IEC Normal Inversa para os dois valores de múltiplo de tempo utilizado.

Figura 23 – Teste F51 com  $K=0,1$



Fonte: Os Autores, (2024).

Figura 24 – Teste F51 com  $K=1$



Fonte: Os Autores, (2024).

## 6 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um protótipo de relé de proteção utilizando o microcontrolador ESP32, com sensores de tensão e corrente comerciais. O objetivo principal foi criar uma solução capaz de monitorar e proteger sistemas elétricos de potência (SEP) contra falhas, como sobrecorrente e sobretensão, atendendo aos padrões técnicos estabelecidos.

Os resultados obtidos indicam novas perspectivas para a proteção de sistemas elétricos de potência, ao evidenciar que soluções acessíveis e eficientes podem ser desenvolvidas com microcontroladores comerciais. A integração do ESP32 com os sensores ZMPT101B e ACS712-20A mostrou-se uma alternativa viável para aplicações que demandam precisão, confiabilidade e baixo custo.

O protótipo destacou-se pela combinação de economicidade e funcionalidade, apresentando desempenho consistente nas medições de tensão e corrente e atuação eficaz em cenários simulados de falhas elétricas. Baseado em componentes amplamente disponíveis e de fácil aquisição, o sistema demonstra grande potencial para aplicações educacionais e projetos de menor escala.

No entanto, o protótipo apresentou algumas limitações que devem ser consideradas. O processo de calibração dos sensores, realizado manualmente, pode introduzir erros humanos e demanda maior tempo na configuração inicial. Essa característica, somada ao projeto voltado para sistemas monofásicos, restringe sua aplicabilidade em redes trifásicas, comuns em ambientes industriais e sistemas de maior porte.

A ausência de integração com protocolos de comunicação, como Modbus ou MQTT, limita o monitoramento remoto e a conectividade com sistemas de supervisão (SCADA), reduzindo sua viabilidade em cenários onde a supervisão contínua é indispensável. Além disso, os sensores ZMPT101B e ACS712 apresentam desafios relacionados à necessidade de calibração frequente, que pode ser influenciada por fatores externos, como temperatura, ruídos eletromagnéticos e desgaste natural dos componentes.

Esse processo de calibração manual não apenas aumenta a complexidade operacional, como também compromete a confiabilidade das medições, especialmente em aplicações onde altos níveis de precisão são necessários. Embora adequados para protótipos e aplicações de

pequena escala, essas limitações tornam os sensores menos apropriados para sistemas de potência robustos e ambientes industriais, onde automação e consistência são essenciais.

Adicionalmente, o protótipo não foi submetido a testes em condições adversas, como variações extremas de temperatura ou presença de ruídos eletromagnéticos elevados, que poderiam impactar negativamente a precisão das medições e a confiabilidade das atuações. Esses fatores limitam a análise do desempenho do protótipo em cenários reais e adversos, comuns em sistemas elétricos de maior complexidade.

Para ampliar a aplicabilidade e eficiência do protótipo desenvolvido, diversas melhorias podem ser implementadas. A calibração manual dos sensores, atualmente uma etapa crítica, pode ser substituída por um sistema de autocalibração, permitindo ajustes automáticos em tempo real, reduzindo erros e simplificando a configuração inicial. Alternativamente, a substituição dos sensores ZMPT101B e ACS712 por modelos mais robustos e confiáveis pode apresentar-se como uma solução.

A integração com protocolos de comunicação, como Modbus ou MQTT, representa outro avanço importante. Essa funcionalidade permitiria ao protótipo realizar monitoramento remoto e integração com sistemas SCADA, ampliando sua aplicabilidade em cenários modernos de supervisão centralizada. Além disso, a expansão do sistema para redes trifásicas possibilitaria sua utilização em instalações industriais e comerciais de maior porte. Para isso, seria necessária a inclusão de sensores adicionais e o desenvolvimento de uma lógica de proteção adaptada a redes trifásicas.

Outra melhoria essencial seria a realização de testes em condições adversas, como variações extremas de temperatura, presença de ruídos eletromagnéticos elevados e transientes de energia. Esses ensaios garantiriam a robustez e a confiabilidade do protótipo em ambientes reais e desafiadores, assegurando que ele atenda aos padrões exigidos para aplicações industriais e de grande escala.

A substituição do display OLED por uma interface de visualização mais avançada, com possibilidade de fácil parametrização do relé de proteção, também pode agregar valor significativo ao protótipo. Essa mudança proporcionaria uma interação mais dinâmica e intuitiva com os dados do sistema, facilitando o uso e a interpretação das informações. Além disso, a integração de funcionalidades para armazenamento de dados históricos em memória interna ou sistemas externos tornaria o protótipo ainda mais completo. Essa

funcionalidade permitiria análises detalhadas sobre o desempenho do sistema e os eventos de proteção, sendo essencial para manutenção e avaliação contínua.

Essas propostas de melhorias tornam o protótipo mais adaptável às demandas modernas de proteção de sistemas elétricos de potência, viabilizando sua aplicação em cenários de maior complexidade e ampliando seu impacto em ambientes industriais e educacionais.

Assim, conclui-se que o protótipo desenvolvido atingiu seus objetivos e pode servir como base para futuras inovações na área de proteção de sistemas elétricos de potência.

## A Código Completo do Sistema de Proteção

Este apêndice apresenta o código-fonte completo utilizado no desenvolvimento do sistema de proteção baseado no ESP32.

```
1 #include "MultiMap.h"
2 #include <SPI.h>
3 #include <Wire.h>
4 #include <Adafruit_GFX.h>
5 #include <Adafruit_SSD1306.h>
6
7 #define OLED_RESET 32
8 Adafruit_SSD1306 display(OLED_RESET);
9
10 #define SENSOR_T 13 // Pino do sensor: 34 para tens o e 35 para
    corrente
11 #define SENSOR_C 34 // Pino do sensor: 34 para tens o e 35 para
    corrente
12
13 #define TENSAO_NOMINAL 115.0 // Tens o nominal do sistema em
    volts
14 #define TENSAO_LIMITE_SUPERIOR (TENSAO_NOMINAL * 1.1) // Limite
    superior de tens o para atua o de prote o por
    sobretens o
15 #define TENSAO_LIMITE_INFERIOR (TENSAO_NOMINAL * 0.9) // Limite
    inferior de tens o para atua o de prote o por subtens o
16 #define CORRENTE_INSTANTANEA_LIMITE 4.0 // Limite de corrente
    para atua o instant nea por sobrecorrente
17 #define CORRENTE_AJUSTE 1.5 // Corrente de ajuste para
    sobrecorrente temporizada
18 #define DIALTIME 0.3
19
20 #define PINO_LED_VERDE 17 // LED verde indica funcionamento
    normal do sistema
21 #define PINO_LED_SOBRETENSAO 16 // LED vermelho indica atua o
```

```

    por sobretens o
22 #define PINO_LED_SUBTENSÃO 4 // LED vermelho indica atua o por
    subtens o
23 #define PINO_LED_PICKUP 25 // LED amarelo indica que a corrente
    de ajuste foi ultrapassada
24 #define PINO_LED_SOBRECORRENTE_TEMPORIZADA 26 // LED vermelho
    indica atua o por sobrecorrente temporizada
25 #define PINO_LED_SOBRECORRENTE_INSTANTANEA 27 // LED vermelho
    indica atua o por sobrecorrente instantanea
26
27 unsigned long instanteInicioSobrecorrente = 0; // Armazena o
    instante de inicio da sobrecorrente para temporiza o
28 bool monitorandoSobrecorrente = false; // Indica se o sistema
    est monitorando sobrecorrente temporizada
29
30
31
32 //VALORES DE LEITURA:
33 float leitura_tensao =0; //Leitura do sensor
34 float tensao_RMS=0;
35 float corrente_RMS=0;
36
37 //VALORES AUXILIARES PARA TENS O RMS
38 float soma_quadrados_tensao=0;
39 float amostras_tensao=0;
40 float soma_quadrados_corrente_fase=0;
41 float amostras_corrente_fase=0;
42
43 //VALORES AUXILIARES PARA DETEC O DO PICO
44 float leitura_corrente=0;
45 float corrente_pico = 0; // Armazena o pico
    atual da corrente
46 float maior_corrente_pico = 0; // Armazena o maior
    pico de corrente detectado
47 float maior_corrente_pico_anterior=0;

```

```

48 float leitura_corrente_fase_anterior = 0; // Armazena a
    leitura anterior da corrente
49
50
51 unsigned long tempo=0; // vari vel auxiliar de tempo
52
53
54 float soma_quadrados=0;
55 float amostras=0;
56 float amostras_2 = 0;
57 float soma_quadrados_2=0 ;
58
59 float soma_quadrados_c=0;
60 float amostras_c=0;
61 float amostras_c2 = 0;
62 float soma_quadrados_c2=0 ;
63
64 double tempoAtuacao=0;
65 double tempoAtuacaoReal=0;
66 double instanteInicioSobretensao=0;
67 double instanteInicioSubtensao=0;
68 double instanteFimSobrecorrente=0;
69 unsigned long tcc=0; // vari vel auxiliar de tempo
70 unsigned long tt=0; // vari vel auxiliar de tempo
71 unsigned long trip=0; // vari vel auxiliar de tempo
72 //unsigned long tempo=0; // vari vel auxiliar de tempo do
    display
73 float soma_amostras=0;
74 float off_set_tensao=0;
75 float off_set_corrente=0;
76
77 float corrente_RMSP=0;
78 float corrente_PK=0;
79
80

```

```

81 float in[] = {90.9, 95.5, 100.3, 105.5, 109.8, 114.6, 119.5,
124.7, 129.9, 134.6, 139.7};
82 float out[] = { 90.2, 95, 100.1, 105, 110, 115.3, 120.5, 125.3,
130.5, 135.5, 140.6};
83
84 //CORRENTE
85 float in_c[] = {25.8, 51.8, 78.2, 104.6, 131.4, 156.6, 182.6,
209.8, 233.0};
86 float in_cp[] = {1147, 1192, 1223, 1266, 1295, 1336, 1374,
1408, 1439};
87
88 float out_c[] = {500, 1000, 1500, 2000, 2510, 3000, 3500, 4020,
4490};
89
90 float corrente;
91 float tensao;
92
93 const unsigned long intervaloDisplay = 1000; // Atualizar o
display a cada 100ms
94 unsigned long ultimoDisplay = 0;
95
96 int status=1;
97 bool executado=false;
98 void setup() {
99 // put your setup code here, to run once:
100 pinMode(PINO_LED_VERDE, OUTPUT);
101 pinMode(PINO_LED_SOBRETENSAO, OUTPUT);
102 pinMode(PINO_LED_SUBTENSAO, OUTPUT);
103 pinMode(PINO_LED_PICKUP, OUTPUT);
104 pinMode(PINO_LED_SOBRECORRENTE_TEMPORIZADA, OUTPUT);
105 pinMode(PINO_LED_SOBRECORRENTE_INSTANTANEA, OUTPUT);
106
107 Wire.begin();
108 display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
109

```

```

110 Serial.begin(256000);
111
112     //DISPLAY PARA TEMPO DE ATUA 0
113 display.display();
114 delay(1000);
115 }
116
117 void loop() {
118     // put your main code here, to run repeatedly:
119     status=1;
120
121     leitura_tensao = lerTensao(20);
122     leitura_corrente = lerCorrente(20);
123
124
125     //ENCONTRAR ZERO TENSAO
126     if(amostras<500){
127         soma_quadrados=((leitura_tensao)*(leitura_tensao))+
128             soma_quadrados;
129         //soma_quadrados=((leitura_tensao-midpoint)*(leitura_tensao-
130             midpoint))+soma_quadrados;
131         amostras++;
132     }
133     else{
134         off_set_tensao=sqrt(soma_quadrados/amostras);
135         soma_quadrados=0;
136         amostras=0;
137     }
138
139     //CALIBRA 0 TENSAO RMS
140     if(amostras_2<1000){
141         soma_quadrados_2=((leitura_tensao-off_set_tensao)*(
142             leitura_tensao-off_set_tensao))+soma_quadrados_2;
143         //soma_quadrados=((leitura_tensao-midpoint)*(leitura_tensao-

```

```

        midpoint))+soma_quadrados;
142     amostras_2++;
143 }
144 else{
145     tensao=(1*(sqrt(soma_quadrados_2/amostras_2)));
146     soma_quadrados_2=0;
147     amostras_2=0;
148 }
149 tensao_RMS= multiMap<float, float>(tensao, in, out, 11);
150
151 //ENCONTRAR ZERO CORRENTE
152 if(amostras_c<500){
153     soma_quadrados_c=((leitura_corrente)*(leitura_corrente))+
        soma_quadrados_c;
154     //soma_quadrados=((leitura_tensao-midpoint)*(leitura_tensao-
        midpoint))+soma_quadrados;
155     amostras_c++;
156 }
157 else{
158     off_set_corrente=sqrt(soma_quadrados_c/amostras_c);
159     soma_quadrados_c=0;
160     amostras_c=0;
161 }
162
163 //CALIBRA O CORRENTE RMS
164 if(amostras_c2<500){
165     soma_quadrados_c2=((leitura_corrente-off_set_corrente)*(
        leitura_corrente-off_set_corrente))+soma_quadrados_c2;
166     //soma_quadrados=((leitura_tensao-midpoint)*(leitura_tensao-
        midpoint))+soma_quadrados;
167     amostras_c2++;
168 }
169 else{
170     corrente=(1*(sqrt(soma_quadrados_c2/amostras_c2)))+0;
171     soma_quadrados_c2=0;

```

```

172     amostras_c2=0;
173 }
174 corrente_RMS= (multiMap<float, float>(corrente, in_c, out_c, 9)
175     )/1000;
176
177     //OBTEN O DO MAIOR VALOR DE PICO
178     if(leitura_corrente>off_set_corrente && leitura_corrente<
179         leitura_corrente_fase_anterior){
180         corrente_pico=leitura_corrente_fase_anterior; //DETECTOR DE
181             PICO
182         if(corrente_pico>maior_corrente_pico){
183             maior_corrente_pico=corrente_pico; //DETECTOR DO MAIOR
184                 VALOR DE PICO
185         }
186     }
187     if(leitura_corrente<off_set_corrente){
188         if(maior_corrente_pico!=0){
189             maior_corrente_pico_anterior=maior_corrente_pico;
190         }
191         maior_corrente_pico=0;
192     }
193     leitura_corrente_fase_anterior= leitura_corrente;
194
195     //ENCONTRANDO VALOR DA CORRENTE RMS PELO PICO:
196     corrente_RMSP= (multiMap<float, float>(
197         maior_corrente_pico_anterior, in_cp, out_c, 9)-1100);
198     corrente_RMSP=(corrente_RMSP/1000)-0.37;
199
200     // Prote o de sobrecorrente temporizada
201     if (corrente_RMS > CORRENTE_AJUSTE) {
202         status=1;
203
204         if (!monitorandoSobrecorrente) {

```

```

202     instanteInicioSobrecorrente = millis(); // Armazena o
        instante em que a sobrecorrente come ou
203     monitorandoSobrecorrente = true;
204     corrente_PK=corrente_RMS;
205 }
206
207     tempoAtuacao = (0.14 / (pow((corrente_RMS / CORRENTE_AJUSTE),
        0.02) - 1)) * DIALTIME;
208     tempoAtuacao *= 1000; // Convertendo para milissegundos
209
210     // Prote o de sobrecorrente instantanea
211     if (corrente_RMS >= CORRENTE_INSTANTANEA_LIMITE) {
212         atuarSobrecorrenteInstantanea(); // Atua instantaneamente
        por sobrecorrente
213         status=5;
214     }
215     else if (millis() - instanteInicioSobrecorrente >=
        tempoAtuacao) {
216         status=6;
217         atuarSobrecorrenteTemporizada(); // Atua por sobrecorrente
        temporizada
218     }
219     else {
220         digitalWrite(PINO_LED_PICKUP, HIGH); // Sinaliza que a
        corrente de ajuste foi ultrapassada
221         status=4;
222     }
223 }
224 else {
225     // Corrente voltou ao normal antes de atuar
226     monitorandoSobrecorrente = false;
227     digitalWrite(PINO_LED_PICKUP, LOW); // Desliga LED de pickup
228 }
229
230 if (tensao_RMS > TENSAO_LIMITE_SUPERIOR) {

```

```

231     instanteInicioSobretensao = millis();
232     atuarSobretensao(); // Atua por sobretens o
233     status=2;
234 }
235 else if (tensao_RMS < TENSAO_LIMITE_INFERIOR) {
236     instanteInicioSubtensao = millis();
237     atuarSubtensao(); // Atua por subtens o
238     status=3;
239 }
240
241 if((corrente_RMS<=CORRENTE_AJUSTE) && (tensao_RMS>=
    TENSAO_LIMITE_INFERIOR) && (tensao_RMS<=
    TENSAO_LIMITE_SUPERIOR)){
242     digitalWrite(PINO_LED_VERDE, HIGH);
243     digitalWrite(PINO_LED_SOBRETENSAO, LOW);
244     digitalWrite(PINO_LED_SUBTENSAO, LOW);
245     digitalWrite(PINO_LED_PICKUP, LOW);
246     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
247     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
248 }
249
250 if (millis() - ultimoDisplay >= intervaloDisplay) {
251     mostrarDisplay();
252     ultimoDisplay = millis();
253 }
254 }
255
256
257 double lerTensao(int numAmostras) {
258     // Fun o para ler a tens o do sensor e calcular o valor
    RMS
259     double soma = 0;
260     for (int i = 0; i < numAmostras; i++) {
261         int valorADC = analogRead(SENSOR_T) - off_set_tensao; //
    L o valor ADC e subtrai o offset

```

```

262     soma += valorADC;
263 }
264 return ((soma / numAmostras) + off_set_tensao); // Retorna o
    valor RMS da tens o
265 }
266
267 double lerCorrente(int numAmostras) {
268     // Fun o para ler a tens o do sensor e calcular o valor
    RMS
269     double soma = 0;
270     for (int i = 0; i < numAmostras; i++) {
271         int valorADC = analogRead(SENSOR_C) - off_set_corrente;
                // L o valor ADC e subtrai o offset
272         soma += valorADC;
273     }
274     return ((soma / numAmostras) + off_set_corrente); // Retorna
    o valor RMS da tens o
275 }
276
277 void atuarSobretensao() {
278     digitalWrite(PINO_LED_SOBRETENSAO, HIGH); // Acende LED de
    sobretens o
279     digitalWrite(PINO_LED_VERDE, LOW);
280     digitalWrite(PINO_LED_SUBTENSAO, LOW);
281     digitalWrite(PINO_LED_PICKUP, LOW);
282     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
283     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
284     while (true); // Sistema pausado at reset
285 }
286
287 void atuarSubtensao() {
288     digitalWrite(PINO_LED_SUBTENSAO, HIGH); // Acende LED de
    subtens o
289     digitalWrite(PINO_LED_VERDE, LOW);
290     digitalWrite(PINO_LED_SOBRETENSAO, LOW);

```

```

291     digitalWrite(PINO_LED_PICKUP, LOW);
292     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
293     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
294     while (true); // Sistema pausado at reset
295 }
296
297 void atuarSobrecorrenteInstantanea() {
298     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, HIGH); //
299         Acende LED de sobrecorrente instantanea
300     digitalWrite(PINO_LED_VERDE, LOW);
301     digitalWrite(PINO_LED_SOBRETENSAO, LOW);
302     digitalWrite(PINO_LED_SUBTENSAO, LOW);
303     digitalWrite(PINO_LED_PICKUP, LOW);
304     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, LOW);
305     while (true); // Sistema pausado at reset
306 }
307
308 void atuarSobrecorrenteTemporizada() {
309     digitalWrite(PINO_LED_SOBRECORRENTE_TEMPORIZADA, HIGH); //
310         Acende LED de sobrecorrente temporizada
311     digitalWrite(PINO_LED_VERDE, LOW);
312     digitalWrite(PINO_LED_SOBRETENSAO, LOW);
313     digitalWrite(PINO_LED_SUBTENSAO, LOW);
314     digitalWrite(PINO_LED_PICKUP, LOW);
315     digitalWrite(PINO_LED_SOBRECORRENTE_INSTANTANEA, LOW);
316     instanteFimSobrecorrente = millis(); // Armazena o instante
317         do fim
318     tempoAtuacaoReal = (instanteFimSobrecorrente -
319         instanteInicioSobrecorrente) / 1000.0; // Converte para
320         segundos
321     mostrarDisplay();
322     while (true); // Sistema pausado at reset
323 }

```

```

321 void mostrarDisplay() {
322
323     //PAR METROS DO DISLPAY:
324     display.clearDisplay();
325     display.setTextColor(WHITE);
326
327     //DISPLAY PARA TENSAO
328     display.setCursor(0,8);
329     display.print("Tensao RMS:");
330     display.setCursor(70,8);
331     display.print(tensao_RMS,1);
332     display.setCursor(105,8);
333     display.print("V");
334
335     //DISPLAY PARA CORRENTE
336     display.setCursor(0,16);
337     display.print("Corrente RMS:");
338     display.setCursor(82,16);
339     display.print(corrente_RMS,1);
340     display.setCursor(105,16);
341     display.print("A");
342
343     //DISPLAY PARA STATUS
344     display.setTextSize(1);
345     display.setCursor(0,0);
346     display.print("Status:");
347     display.setCursor(43,0);
348     if(status==1){
349         display.print("Normal");
350     }
351     else if (status==2) {
352         display.print("F 59");
353
354         //while (true);
355     }

```

```

356     else if (status==3) {
357         display.print("F 27");
358         //while (true);
359     }
360     else if (status==4) {
361         display.print("Pick-up");
362         //DISPLAY PARA CORRENTE DE PICK-UP
363         display.setCursor(0,24);
364         display.print("Corrente PKUP:");
365         display.setCursor(85,24);
366         display.print(corrente_PK,3);
367         display.setCursor(120,24);
368         display.print("A");
369     }
370     else if (status==5) {
371         display.print("F 50");
372     }
373     else if (status==6) {
374         display.print("F 51");
375         //DISPLAY PARA TEMPO DE ATUA 0
376         display.setCursor(0,24);
377         display.print("Atuacao:");
378         display.setCursor(50,24);
379         display.print(tempoAtuacaoReal,3);
380         display.setCursor(89,24);
381         display.print("s");
382         display.display();
383
384         while (true);
385     }
386
387
388 display.display();
389
390

```

391 }

Código A.1 – Código-fonte completo do sistema de proteção

## BIBLIOGRAFIA UTILIZADA

- [1] Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-Time Signal Processing*. Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 1997.
- [2] Bárbara Vieira Xavier and Renata Romano. Implementação de um relé de sobrecorrente temporizado utilizando a linguagem 'models' do alternative transients program, 2013.
- [3] Beatriz Alves de Souza Gomes and Alexandre Akira Kida. Concepção de um relé de proteção digital em arduino utilizando uma plataforma virtual. In *Instituto Federal de Educação, Ciência e Tecnologia da Bahia*, 2021. Trabalho Técnico.
- [4] Daniel da Silva Sarmento. Medição de potência elétrica utilizando o circuito integrado de conversão analógico-digital ade7753 e a plataforma arduino, 2018.
- [5] Edwin H. Hall. On a new action of the magnet on electric currents. *American Journal of Mathematics*, 1879.
- [6] ElectroSchematics. Current Sensor Module and Microcontrollers, 2024. Acesso em: 27 nov. 2024.
- [7] Espressif Systems. *ESP32 Technical Reference Manual*, 2024. Acesso em: 04 set. 2024.
- [8] Ewerton S. Farias, Cecília M. Costa, Emerson L. Soares, Ademar G. Costa Junior, and Gilvan V. Andrade Junior. Desenvolvimento de um sistema de proteção utilizando disjuntor residual monofásico controlado por arduino. In *Conferência de Automação (CBA)*, 2016.
- [9] Fábio Pereira. *Microcontroladores MSP430: Teoria e Prática*. Érica, São Paulo, 2005.
- [10] Fernanda Oliveira. Simulação de sensor tensão corrente para aplicações em iot. *ResearchGate*, 2021. Disponível em: <https://www.researchgate.net/publication/352955241>. Acesso em: 03 jul. 2021.
- [11] Franco Maloberti. *Data Converters*. Springer, Dordrecht, 2007.

- [12] G. Kindermann. *Proteção de Sistemas Elétricos de Potência*, volume 1, 2 e 3. UFSC–EEL–LabPlan, Florianópolis-SC, 2ª edition, 2005.
- [13] Jáder Fernando Dias Breda. Um modelo computacional para o relé digital de sobrecorrente empregado na proteção de sistemas elétricos de potência, 2009.
- [14] João Pedro Bezerra Sampaio. Modernização de sistemas de proteção com a utilização dos relés digitais. *Universidade Federal de Pernambuco*, 2014.
- [15] Maria Talissa Araujo Pereira. Simulação em malha fechada da função de proteção de sobrecorrente (50/51) no software hypersim, 2024. Orientadora: Dra. Priscila de Lima Vianna.
- [16] Microcontrollers Lab. AC Voltage Measurement with Arduino, 2024. Acesso em: 27 nov. 2024.
- [17] P. Rush. *Proteção e Automação de Redes: Conceitos e Aplicações*. Blusher, São Paulo, 2009.
- [18] Random Nerd Tutorials. ESP32 and SSD1306 OLED Display with Arduino IDE, 2024. Acesso em: 27 nov. 2024.
- [19] Vida de Silício. Conhecendo o ESP32, 2024. Acesso em: 27 nov. 2024.