



## **ESCOLA TÉCNICA ESTADUAL**

**DEPUTADO ARY DE CAMARGO PEDROSO**

**CURSO TÉCNICO ELETRÔNICA**

Augusto De Campos Fernandes

Hillary Cauani Correa

Akira Pereira Vaz

Cleiton Araujo Pacheco

**Monitoramento Eficiente de Máquinas: Uma IHM para Coleta de Dados**

**Piracicaba**

**2024**

Augusto de Campos Fernandes

Hillary Cauani Correa

Akira Pereira Vaz

Cleiton Araujo Pacheco

**Monitoramento Eficiente de Máquinas: Uma IHM para Coleta de Dados**

Trabalho de Conclusão de Curso

apresentado ao Curso Técnico em

Eletrônica, da ETEC Deputado

Ary de Camargo Pedroso, orientado

pelo Profº Volcov como requisito parcial

para obtenção do título de Técnico em

Eletrônica.

**Piracicaba SP**

**2024**

## **AGRADECIMENTOS**

Agradeço ao acontecimento da Guerra das Correntes, que com seu desenvolvimento conseguiu proporcionar a toda humanidade mais um salto para o avanço da tecnologia que nos levaram até a eletrônica dos dias atuais.

Gostaríamos de agradecer a nós mesmos, porque conseguimos finalizar o curso e a AVT por proteger a linha do tempo sagrada.

Gostaríamos de agradecer também ao Baby Groot, por ser uma fonte de inspiração de sustentabilidade e crescimento.

Agradecimentos a John Wick por ensinar a como usar um lápis para passar seu conhecimento ao próximo e como valorizar os animais.

Agradecemos também ao Stitch, porque OHANA quer dizer FAMÍLIA, que foi o que nos tornamos durante o tempo em que fizemos este trabalho.

## RESUMO

O avanço da Indústria 4.0 e a crescente demanda por eficiência operacional têm impulsionado o desenvolvimento de sistemas de monitoramento em tempo real para máquinas industriais. A utilização de dispositivos de Interface Homem-Máquina (IHM) permite a coleta e análise de dados essenciais para a tomada de decisões estratégicas, aumentando a produtividade e reduzindo custos com manutenção e consumo energético. Este trabalho propõe o desenvolvimento de uma IHM baseada no microcontrolador ESP32, capaz de monitorar parâmetros como corrente elétrica, temperatura e vibração, indicadores críticos para avaliar a condição e desempenho de equipamentos industriais.

**Palavras-Chave:** ESP32, IHM's, monitoramento, informações e análise.

## ABSTRACT

The advancement of Industry 4.0 and the increasing demand for operational efficiency have driven the development of real-time monitoring systems for industrial machinery. The use of Human Machine Interface (HMI) devices allows the collection and analysis of essential data for strategic decision-making, increasing productivity and reducing maintenance and energy consumption costs. This work proposes the development of an HMI based on the ESP32 microcontroller, capable of monitoring parameters such as electric current, temperature and vibration, critical indicators to evaluate the condition and performance of industrial equipment.

**Key-Words:** ESP32, HMI's, monitoring, information and analysis.

## **SUMÁRIO**

### **1. INTRODUÇÃO**

1.1 O papel dos IHMs na melhoria do controle e monitoramento de processos industriais.

1.2 Benefícios no uso de IHMs para otimização de eficiência

### **2. Desenvolvimento Teórico**

2.1 Escolha do ESP32 como processador principal.

2.1.1 Programação do ESP32 para coleta de dados e envio de informações para um display ou interface gráfica

2.2 Gestão de Paradas de Manutenção

### **3. Principais Funções**

3.1 Acompanhar a vida útil do equipamento

### **4. Utilização da OEE**

### **5. Possíveis ganhas produtivos e de Qualidade**

5.1 Ponto de vista de qualidade

5.2 Ponto de vista Produtivo

5.3 redução de custo

### **6. Projeto**

6.1 Componentes

6.2 Linguagem de programação e lógica

6.2.1 Código

6.3 lei de ohms e cálculo de tempo

## **REFERÊNCIAS BIBLIOGRÁFICAS**

## 1. INTRODUÇÃO

O monitoramento constante das condições operacionais das máquinas tornou-se uma prática essencial para aprimorar a eficiência e a segurança no ambiente industrial. Com o surgimento da Indústria 4.0, a coleta e análise de dados em tempo real têm transformado a forma como são realizados o controle e a manutenção de máquinas. Esses dispositivos não apenas otimizam o uso de recursos, mas também ajudam a reduzir desperdícios e prolongar a vida útil dos equipamentos.

Neste trabalho, trouxemos o desenvolvimento de um dispositivo de Interface Homem-Máquina (IHM), baseado no microcontrolador ESP32, que combina sensores de corrente, temperatura e vibração para monitorar, de forma prática, o consumo de energia e o desempenho de máquinas. A IHM permite a visualização e o acompanhamento remoto dos dados históricos de operação dos equipamentos, fornecendo informações cruciais para o planejamento de manutenções preventivas e a redução de paradas inesperadas, promovendo maior segurança e reduzindo custos com reparos emergenciais.

Com isso, o dispositivo proposto busca não apenas oferecer uma solução de baixo custo para a gestão energética industrial, mas também contribuir para a automação e o monitoramento inteligente de máquinas. Esse dispositivo se alinha aos princípios da Indústria 4.0, proporcionando uma gestão mais eficaz dos recursos energéticos e promovendo práticas sustentáveis na indústria moderna.

## **1.2 O papel dos IHMs na melhoria do controle e monitoramento de processos industriais.**

As Interfaces Homem-Máquina (IHMs) têm se tornado elementos essenciais no monitoramento e controle de processos industriais, proporcionando aos operadores uma forma prática e eficiente de interagir com sistemas automatizados. Com o avanço tecnológico, as IHMs evoluíram significativamente, incorporando telas sensíveis ao toque e gráficos que facilitam a visualização dos dados, o que transforma a experiência do operador ao permitir que ele acompanhe, em tempo real, parâmetros críticos do processo. Esse monitoramento em tempo real ajuda a identificar falhas com antecedência, evitando problemas graves e dispendiosos, como paradas inesperadas na produção ou danos permanentes aos equipamentos.

A implementação de IHMs também contribui para a segurança industrial, pois elas são capazes de emitir alertas e, em algumas situações, até tomar ações corretivas automaticamente para evitar falhas graves. Com a coleta contínua de dados, as IHMs permitem uma manutenção preditiva mais eficaz, já que os dados históricos e em tempo real ajudam a identificar padrões de operação que indicam falhas iminentes, o que permite realizar manutenções preventivas e prolongar a vida útil dos equipamentos.

Com o avanço da Internet das Coisas (IoT) e da Inteligência Artificial (IA), as IHMs estão se tornando ainda mais sofisticadas, possibilitando a análise precisa de dados e a otimização dos processos em tempo real.

## **1.3 Benefícios no uso de IHMs para otimização de eficiência**

As IHMs são essenciais para aumentar a eficiência nos processos industriais, oferecendo uma interface intuitiva que facilita a interação entre operadores e sistemas complexos. Com gráficos claros e informações em tempo real, os operadores conseguem monitorar e ajustar parâmetros para evitar falhas, reduzir desperdícios e melhorar a confiabilidade dos processos. Assim, as IHMs são uma ferramenta fundamental para aprimorar a eficiência, segurança e sustentabilidade das operações industriais, e ajudam as empresas a se tornarem mais competitivas em um mercado cada vez mais globalizado.

Além disso, as IHMs possibilitam a automação do controle de aspectos do processo, como ajustes de temperatura e velocidade, sem a necessidade de intervenção manual. Isso minimiza erros, assegura que os processos operem sempre em condições ideais, e contribui para uma operação mais econômica e sustentável, reduzindo o desperdício de recursos. As IHMs também facilitam a manutenção preditiva, permitindo que os dados operacionais sejam coletados continuamente para identificar falhas potenciais.

A incorporação de novas tecnologias, como a IoT e a IA, torna as IHMs ainda mais potentes. A conectividade com sensores permite ajustes em tempo real, enquanto a IA analisa grandes volumes de dados, otimizando o processo com mais precisão. Em resumo, as IHMs são ferramentas valiosas para tornar a produção mais inteligente, sustentável e competitiva.

## 2. Desenvolvimento Teórico

### 2.1 Escolha do ESP32 como processador principal.

O ESP32 é uma excelente opção para ser o processador principal em projetos de automação e controle, como no caso de um dispositivo de monitoramento de gasto de energia para máquinas industriais. Desenvolvido pela Espressif, o ESP32 oferece recursos poderosos e versáteis que o tornam ideal para aplicações industriais, especialmente em sistemas que exigem conectividade, alto desempenho e integração com a Internet das Coisas (IoT).

#### Principais Vantagens:

- **Conectividade sem fio (Wi-Fi e Bluetooth):** O ESP32 permite comunicação sem fio, essencial para enviar dados sobre o consumo de energia e a temperatura das máquinas, seja localmente ou na nuvem.
- **Alto desempenho de processamento:** Com um processador dual-core de 240 MHz, o ESP32 é capaz de realizar multitarefas, como monitorar sensores e processar dados em tempo real sem comprometer o desempenho.
- **Eficiência energética:** Embora seja um microcontrolador de alto desempenho, o ESP32 possui modos de baixo consumo, permitindo uma operação eficiente por longos períodos, ideal para sistemas industriais contínuos.
- **Flexibilidade de GPIO:** O ESP32 oferece muitos pinos de entrada/saída, permitindo fácil integração com diversos sensores e dispositivos necessários para o monitoramento de energia e temperatura.
- **Facilidade de programação:** Sua compatibilidade com plataformas como o Arduino IDE torna o desenvolvimento mais acessível, com diversos tutoriais e bibliotecas de código aberto disponíveis.

- **Custo acessível:** Apesar de seu desempenho avançado, o ESP32 é uma opção de baixo custo, tornando-o uma escolha vantajosa para projetos industriais.
- **Suporte a IoT e nuvem:** O ESP32 facilita a integração com plataformas de nuvem, permitindo o acesso remoto e a análise de dados em tempo real, o que é fundamental para a manutenção preditiva e o controle remoto das máquinas.

### **2.1.1 Programação do ESP32 para coleta de dados e envio de informações para um display ou interface gráfica**

A programação do ESP32 para coleta de dados e exibição de informações é uma parte fundamental no desenvolvimento de sistemas de monitoramento e controle, como o seu projeto de controle de consumo de energia em máquinas industriais. O ESP32 se destaca por sua versatilidade, oferecendo suporte para diversos tipos de sensores e displays, além de permitir a comunicação sem fio com outros dispositivos ou sistemas, facilitando a integração de dados e o controle remoto.

#### **Coleta de Dados com Sensores**

Para monitorar o consumo de energia e as condições das máquinas, o ESP32 pode ser configurado para coletar dados de diversos tipos de sensores. Sensores de corrente, por exemplo, podem ser conectados a pinos GPIO do ESP32, utilizando o conversor analógico-digital (ADC) integrado para fazer a leitura das variáveis de corrente de forma precisa. O ESP32 possui múltiplos pinos ADC, o que permite a conexão de diversos sensores analógicos ao mesmo tempo, sem sobrecarregar a capacidade de entrada do microcontrolador.

#### **Processamento e Armazenamento de Dados**

O ESP32 é uma ferramenta poderosa para sistemas de monitoramento em tempo real, especialmente por seu processador dual-core de 240 MHz. Essa característica permite que ele realize múltiplas tarefas simultaneamente sem comprometer o desempenho. Em um sistema de controle de consumo energético, por exemplo, um dos núcleos pode ser dedicado à coleta de dados dos sensores, enquanto o outro realiza o processamento, como cálculos de consumo de energia e temperatura.

### **Armazenamento e Envio de Dados**

Os dados coletados podem ser armazenados na memória flash do próprio ESP32 ou enviados diretamente para a nuvem. Armazenar localmente facilita a consulta e auditoria futura dos registros, o que é útil para uma análise a longo prazo. Já o envio para a nuvem oferece a vantagem do monitoramento em tempo real, permitindo que os dados sejam acessados de qualquer lugar.

### **Exibição de Dados**

Após a coleta dos dados, exibi-los de maneira clara é fundamental para a tomada de decisões rápidas. O ESP32 permite o envio de informações para displays locais, como LCDs e OLEDs, ou para uma interface gráfica em uma plataforma remota. Em displays mais simples, como o LCD 16x2, é possível usar a biblioteca LiquidCrystal para apresentar dados como consumo de energia e temperatura de forma direta. Já em displays mais avançados, como OLEDs, bibliotecas como a U8g2 ou Adafruit\_SSD1306 possibilitam exibir gráficos detalhados e múltiplos parâmetros, otimizando a visualização dos dados.

### **Integração com Interfaces Gráficas Remotas**

Uma das maiores vantagens do ESP32 é sua conectividade sem fio, que permite enviar dados para uma plataforma remota, como uma aplicação web ou mobile, via Wi-Fi ou Bluetooth. Com isso, o ESP32 possibilita o monitoramento remoto de máquinas a partir de interfaces gráficas em tempo real. Usando plataformas como ThingSpeak, Blynk, ou mesmo soluções personalizadas, é possível criar um painel de controle acessível a partir de qualquer dispositivo conectado, centralizando as informações e tornando o monitoramento mais conveniente e ágil.

## **Flexibilidade e Expansão**

O ESP32 é extremamente flexível e pode ser escalado de acordo com as necessidades do sistema. Caso seja necessário monitorar mais máquinas ou adicionar novos sensores, o ESP32 permite essa adaptação sem grandes dificuldades, seja através dos pinos GPIO para novos sensores, ou integrando módulos e dispositivos adicionais. Isso garante que o sistema de monitoramento possa crescer conforme a planta industrial se expande, otimizando custos e assegurando a viabilidade a longo prazo.

## **2.2 Gestão de Paradas de Manutenção**

Tradicionalmente, muitas indústrias tratavam as paradas de manutenção de maneira reativa, ou seja, as máquinas só eram paradas quando apresentavam falhas. Esse modelo gerava altos custos de reparo e longos períodos de inatividade, afetando negativamente a produção. Com o tempo, as empresas passaram a adotar uma abordagem mais proativa, onde as manutenções são planejadas antes que os problemas aconteçam. Nesse contexto, a manutenção preventiva se tornou um padrão, sendo realizada com base no tempo de operação das máquinas ou em indicadores de desempenho.

A gestão eficiente das paradas de manutenção é essencial para garantir que as máquinas e equipamentos de uma indústria operem de forma contínua, com alta performance e sem interrupções. Isso envolve planejar e controlar os períodos de inatividade das máquinas, seja para manutenção preventiva, corretiva ou preditiva. O objetivo principal é minimizar o tempo de parada das máquinas, o que contribui para reduzir custos, melhorar a produtividade e aumentar a segurança operacional.

Além da manutenção preventiva, a manutenção corretiva também desempenha um papel importante na gestão de paradas. Ela é necessária quando ocorre uma falha inesperada, exigindo uma intervenção imediata para restaurar a operação da máquina. Embora a manutenção corretiva seja inevitável, ela tende a ser mais cara e disruptiva, pois interrompe a produção de forma inesperada.

Nos últimos anos, a manutenção preditiva tem ganhado destaque como uma solução mais moderna e eficiente. Ela utiliza dados coletados por sensores, que monitoram continuamente o estado das máquinas, como temperatura, vibração e consumo de energia. Com essas informações, é possível prever falhas antes que se tornem críticas, permitindo realizar manutenções de maneira mais precisa e no momento exato.

O uso de sistemas de gestão de manutenção, como softwares especializados, tem facilitado enormemente a administração desse processo. Esses sistemas permitem que as empresas programem as manutenções, controlem os custos e mantenham um histórico detalhado dos equipamentos. Dessa forma, as equipes podem planejar as paradas de forma mais estratégica, garantindo que as peças de reposição estejam disponíveis e que o tempo de inatividade seja minimizado.

A gestão de paradas de manutenção bem executadas traz diversos benefícios, como, a redução do tempo em que as máquinas ficam paradas resultando em uma menor perda de produção e menos custo, contribui também para aumentar a vida útil dos equipamentos, prevenindo falhas graves que poderiam levar a paradas prolongadas e melhorando a segurança, já que falhas inesperadas podem causar acidentes.

### **3. Principais Funções**

O dispositivo descrito no código é um sistema inteligente projetado para monitorar o consumo de energia de um equipamento ou máquina. Ele funciona com base em leituras de corrente e tensão, permitindo calcular o consumo de

energia e gerar relatórios completos sobre o desempenho da operação.

A operação do dispositivo começa quando a corrente medida atinge um valor

superior a 1,30 A, indicando que a máquina está em funcionamento. Nesse momento, o sistema exibe uma mensagem no LCD informando que a máquina

está operando, além de enviar uma notificação via Serial e Bluetooth com a mensagem "Início de operação". A partir desse ponto, o dispositivo começa a

monitorar e registrar os dados de consumo de energia.

Durante o funcionamento da máquina, o dispositivo continua a fazer medições,

armazenando as três maiores leituras de corrente para referência. Isso permite

que, ao final da operação, o sistema envie um relatório completo, incluindo o

tempo total de operação, as três maiores leituras de corrente e a energia total

consumida. Esse cálculo de energia é feito com base na fórmula de energia elétrica ( $Energia = V \times I \times t$ ), onde V representa a tensão (127V), I é a corrente

medida, e t é o tempo da operação, em horas.

No fim da operação, o dispositivo envia esse relatório detalhado via Serial e Bluetooth para facilitar o acesso às informações. Essas características são extremamente úteis para o monitoramento e análise do desempenho das máquinas, permitindo otimizar o uso de energia e tomar decisões informadas

sobre a manutenção e eficiência operacional.

Além disso, a interface gráfica, exibida no LCD, fornece apenas as informações

mais relevantes, como o status da operação, evitando sobrecarregar o usuário

com dados excessivos. O envio de relatórios completos após a operação

também garante que o operador tenha acesso a dados precisos, facilitando a

análise posterior e a tomada de decisões relacionadas ao consumo de energia

e manutenção preventiva.

Em termos gerais, o dispositivo não só monitora o consumo de energia em tempo real, mas também oferece uma visão detalhada da performance da máquina após o término da operação. Esse tipo de dispositivo é valioso para indústrias ou outros ambientes onde o controle de consumo energético e a manutenção preventiva são cruciais para garantir a eficiência e reduzir custos

operacionais.

### 3.1 Acompanhar a vida útil do equipamento

Acompanhar a vida útil de um equipamento é uma estratégia fundamental para maximizar a eficiência e a rentabilidade de processos industriais, e no contexto do dispositivo descrito no código que você forneceu, isso se torna ainda mais relevante. Através da coleta de dados em tempo real, como corrente, tensão e tempo de operação, o dispositivo permite monitorar o desempenho da máquina de forma contínua. Esse tipo de monitoramento é essencial para prever quando um equipamento pode estar próximo de atingir o fim de sua vida útil, ajudando a prevenir falhas inesperadas que poderiam causar paradas indesejadas ou danos maiores.

No caso do dispositivo, o cálculo da energia consumida ao longo da operação e a coleta das três maiores leituras de corrente durante o uso fornecem uma visão detalhada do desempenho do equipamento. Esses dados podem ser analisados para identificar padrões e anomalias que indicam desgaste ou falhas iminentes. Por exemplo, se o consumo de energia está subindo de maneira constante ou se as leituras de corrente estão fora do padrão esperado, isso pode ser um sinal de que algum componente está se desgastando e precisa de manutenção. Ao acompanhar esses indicadores, é possível agir de forma proativa, realizando manutenções preventivas ou trocando peças antes que o problema se torne crítico.

A utilização de sistemas como o descrito também se alinha diretamente ao conceito de manutenção preditiva. Com os dados coletados, é possível identificar tendências que indicam o desgaste do equipamento, permitindo realizar manutenções de forma planejada e no momento adequado. Isso não só evita falhas inesperadas, mas também prolonga a vida útil do equipamento, reduzindo custos com reparos emergenciais e maximizando o tempo de operação.

Além disso, o envio do relatório completo no final de cada operação, com informações detalhadas sobre o tempo de operação, as leituras de corrente e a energia consumida, pode ser usado como uma ferramenta importante para os gestores. Esses relatórios não apenas oferecem uma visão clara do desempenho da máquina em um determinado período, mas também fornecem dados que podem ser usados para analisar o ciclo de vida do equipamento e planejar futuras ações de manutenção.

Em suma, o dispositivo descrito não só contribui para o monitoramento do consumo de energia e o acompanhamento da operação da máquina, mas também desempenha um papel importante na gestão da vida útil do equipamento. Ao fornecer dados essenciais para a manutenção preditiva e permitir o planejamento de manutenções com base em informações precisas, ele ajuda a evitar falhas inesperadas, otimizar o tempo de operação e prolongar a vida útil dos equipamentos, tudo isso contribuindo para a eficiência e a redução de custos no processo industrial.

#### 4. Utilização da OEE na IHM

Com os dados em tempo real coletados pela IHM, o cálculo do OEE pode ser feito automaticamente, com o dispositivo consolidando e apresentando os índices de disponibilidade, desempenho e qualidade.

A aplicação do OEE envolve monitorar esses parâmetros (corrente, temperatura e vibração) para avaliar e otimizar a eficiência do equipamento. A OEE é composta por três principais indicadores: Disponibilidade, Desempenho e Qualidade. Em uma IHM com ESP32, que coleta dados em tempo real, cada um desses aspectos pode ser calculado e aprimorado da seguinte forma:

##### 1. Disponibilidade

- **Monitoramento da Operação:** O sistema utiliza os dados de corrente para detectar quando o equipamento está em funcionamento ou parado. Com base na corrente medida, a IHM pode registrar os tempos de operação e de inatividade, calculando o percentual de tempo em que o equipamento esteve disponível.
- **Detecção de falhas:** Variabilidades anormais em temperatura e vibração podem indicar falhas iminentes. Com isso, a IHM pode ajudar a prevenir paradas não planejadas, aumentando o tempo de disponibilidade.

##### 2. Desempenho

- **Cálculo de Taxas Operacionais:** O desempenho é avaliado comparando a velocidade real de operação com a ideal. A leitura de corrente e vibração ajuda a identificar desvios de desempenho, como consumo excessivo de energia ou vibração fora do padrão, que podem indicar sobrecarga ou funcionamento abaixo da capacidade nominal.
- **Identificação de Inconsistências:** Se a vibração aumenta ou a temperatura varia além dos limites esperados, isso pode indicar que o equipamento está operando com baixa eficiência. A IHM pode então ajustar o desempenho ao detectar essas condições, maximizando a eficiência energética e operacional.

##### 3. Qualidade

- **Monitoramento de Condições para a Produção de Qualidade:** A qualidade é afetada quando o equipamento não opera sob condições ideais. Com a IHM, é possível monitorar continuamente a temperatura e a vibração para garantir que o equipamento esteja dentro dos limites de operação que garantem uma produção de qualidade.
- **Detecção de Condições que Impactam a Produção:** Anomalias na corrente, temperatura ou vibração podem causar falhas na produção ou afetar a qualidade final. A IHM, ao registrar essas leituras, fornece dados que permitem identificar e corrigir problemas, evitando desperdícios e aumentando a conformidade da produção.

## 5. Possíveis ganhos produtivos e de Qualidade

### 1. Precisão e Consistência nos Dados

- **Cálculo de Corrente RMS:** A medição RMS fornece uma leitura precisa da corrente, melhorando a confiabilidade dos dados, especialmente útil em ambientes com variação de carga.
- **Registro de Dados Históricos:** O armazenamento de picos de corrente e energia consumida possibilita análise detalhada do comportamento do equipamento ao longo do tempo.

### 2. Facilidade de Monitoramento e Interatividade

- **Display LCD para Feedback Imediato:** A exibição em tempo real de corrente, estado e tempo de operação facilita a consulta dos dados sem necessidade de equipamentos adicionais.
- **Comunicação Bluetooth:** O envio de dados e ajuste de data/hora via Bluetooth permite o monitoramento remoto, facilitando o controle sem contato físico direto com o dispositivo.

### 3. Melhoria na Eficiência Energética

- **Monitoramento de Consumo em Tempo Real:** Acompanhar a corrente elétrica ajuda a detectar consumo elevado ou anômalo, proporcionando a oportunidade de ajustar o uso de energia e identificar ineficiências.

#### 4. Manutenção Preventiva e Redução de Custos

- **Relatório de Operação e Detecção de Sobrecarga:** A detecção automática com base em limiares de corrente previne sobrecargas e problemas elétricos, minimizando paradas não planejadas e reduzindo custos de manutenção.
- **Resumo Operacional:** Dados de corrente máxima e mínima permitem identificar padrões de uso e ajustar equipamentos para evitar desgaste desnecessário.

#### 5. Flexibilidade e Potencial de Expansão

- **Integração com Sistemas de Automação:** A possibilidade de integrar o sistema com outras plataformas facilita a criação de um ambiente de monitoramento completo.
- **Compatibilidade com Sistemas Educacionais e Prototipagem:** A modularidade do sistema permite sua aplicação em educação, possibilitando um aprendizado prático e experimental de eletrônica e monitoramento.

#### 6. Aplicações Versáteis

- **Monitoramento de Equipamentos Elétricos:** Em ambientes domésticos ou industriais, o sistema pode atuar no controle de carga e monitoramento do desempenho de equipamentos.
- **Alarmes e Alertas para Segurança:** Em sistemas de alarme, pode atuar em conjunto com sensores de corrente para detecção de sobrecarga ou falhas elétricas, promovendo segurança.

##### 5.1 Ponto de vista de qualidade

Do ponto de vista da qualidade, o projeto destaca-se em vários aspectos que contribuem para sua precisão e aplicabilidade. Aqui estão alguns critérios:

###### 1. Precisão e Confiabilidade na Medição

- A utilização do sensor ACS712 e o cálculo da corrente RMS com a biblioteca EmonLib garantem medições mais estáveis e precisas. Isso reduz erros de leitura e aumenta a confiabilidade dos dados, essenciais para aplicações que exigem precisão em diagnósticos e relatórios.

## **2. Interatividade e Feedback Visual**

- A inclusão de um display LCD permite uma interação direta e informativa com o sistema, apresentando dados em tempo real (corrente, status de operação e duração) de maneira clara e acessível. Isso melhora a usabilidade, especialmente para usuários que monitoram o equipamento localmente.

## **3. Conectividade e Acessibilidade dos Dados**

- A comunicação Bluetooth amplia o alcance do monitoramento, facilitando o acesso aos dados e permitindo ajustes no sistema (como data e hora). Isso é especialmente útil em ambientes onde o monitoramento à distância é essencial, além de facilitar a integração com aplicativos móveis ou sistemas de automação.

## **4. Capacidade de Armazenamento e Análise de Dados**

- O projeto registra os maiores e menores picos de corrente e o tempo de operação, o que permite uma análise posterior do comportamento do equipamento e consumo energético. Esse recurso é valioso para manutenção preventiva, pois facilita a identificação de padrões e anomalias no uso de corrente.

## **5. Escalabilidade e Integração**

- A estrutura do projeto permite futuras expansões, como a integração com sistemas de automação e o uso em aplicações mais complexas. Com pequenas adaptações, pode-se conectar a uma rede IoT ou a um sistema de gerenciamento de energia maior, aumentando sua aplicabilidade em escala industrial ou residencial.

## 6. Versatilidade para Diferentes Aplicações

- Com funcionalidades para monitoramento e controle, o projeto atende a diversas necessidades, desde aplicações educacionais e protótipos até monitoramento industrial. Essa versatilidade amplia o valor agregado, tornando-o uma excelente base para personalizações em diferentes contextos.

### 5.2 Ponto de vista Produtivo

Do ponto de vista produtivo, o projeto apresenta uma série de vantagens que contribuem para otimizar o desempenho e aumentar a eficiência. Ele é projetado para fornecer dados em tempo real. Abaixo, analiso os aspectos produtivos do projeto:

#### 1. Monitoramento Contínuo e Proativo

- Com a medição de corrente em tempo real e a detecção de limiares de corrente, o sistema possibilita o monitoramento contínuo do desempenho dos equipamentos. Isso permite uma abordagem proativa para o diagnóstico de problemas, ajudando a identificar falhas antes que elas causem paradas inesperadas, o que reduz significativamente o tempo de inatividade (downtime) e aumenta a produtividade.

#### 2. Redução de Custos com Manutenção Preventiva

- O projeto registra e analisa picos e variações na corrente, além de incluir um sensor de vibração para detecção de possíveis problemas mecânicos.

Essa análise permite prever manutenções necessárias com base em dados históricos, evitando que o equipamento chegue ao ponto de falha crítica. Com manutenções melhor planejadas, os custos operacionais diminuem, e a vida útil dos equipamentos aumenta.

### **3. Controle do Consumo Energético e Redução de Desperdícios**

- Com o cálculo da energia consumida, o sistema oferece uma visão detalhada do consumo elétrico ao longo do tempo. Esse controle permite identificar possíveis desperdícios energéticos e ajustar o uso de máquinas para horários de menor custo energético, reduzindo gastos e tornando a operação mais sustentável e econômica.

### **4. Automação e Controle Remoto**

- A conectividade Bluetooth permite não só o monitoramento remoto, mas também o ajuste de parâmetros de forma prática e rápida, economizando tempo e aumentando a flexibilidade no gerenciamento do sistema. Essa funcionalidade é especialmente valiosa em instalações com múltiplos equipamentos, permitindo ajustes e monitoramento sem deslocamento físico até cada máquina.

### **5. Flexibilidade e Escalabilidade**

- O projeto foi desenvolvido com uma arquitetura que permite escalabilidade, facilitando a adaptação para outras máquinas ou para aplicações mais amplas, como sistemas de monitoramento de fábricas inteiras. Essa flexibilidade garante que o investimento inicial seja aproveitado em longo prazo, pois o sistema pode ser ampliado e ajustado para novas demandas.

### **6. Potencial para Melhorias Contínuas**

- A coleta e análise dos dados de corrente e vibração fornecem uma base sólida para melhorias contínuas nos processos de produção. Esses dados podem ser usados para otimizar os ciclos de operação e até para treinar sistemas de aprendizado de máquina que prevejam o comportamento do equipamento sob diferentes condições de operação.

### **5.3 redução de custo**

Este projeto tem intenção potencial para gerar redução de custos para empresas, especialmente ao monitorar o consumo de energia de equipamentos industriais. Através do acompanhamento em tempo real da corrente elétrica consumida pelos dispositivos, é possível identificar anomalias de consumo, otimizar a operação e realizar manutenção preventiva. O sistema permite que a empresa automatize processos de desligamento de equipamentos ou ajuste de operação com base no consumo, evitando desperdícios de energia e custos extras com falhas não planejadas. Além disso, a coleta de dados detalhados de consumo e operação possibilita a análise de eficiência, permitindo ajustes estratégicos que resultam em economia de recursos e redução de custos operacionais a longo prazo.

#### **. Eficiência Energética:**

O monitoramento contínuo do consumo de energia permite que a empresa identifique padrões de consumo de energia, detectando tanto períodos de pico quanto de ociosidade. Isso possibilita a identificação de ineficiências no uso de energia, seja por falhas nos equipamentos, operação excessiva ou configurações inadequadas. Ao detectar essas ineficiências, a empresa pode implementar medidas corretivas para otimizar o consumo de energia, o que reduz diretamente os custos com a conta de eletricidade.

- **Manutenção Preventiva:**

Com a coleta constante de dados sobre o consumo e as condições operacionais das máquinas, é possível detectar anomalias como sobrecarga elétrica ou consumo elevado que podem indicar desgaste de componentes ou falhas iminentes. Isso proporciona uma manutenção preditiva mais eficaz, permitindo que a empresa realize reparos antes que os problemas se tornem falhas catastróficas, evitando custos inesperados com reparos emergenciais e reduzindo o tempo de inatividade das máquinas.

- **Automação de Processos e Redução de Mão de Obra:**

Integrar o sistema de monitoramento a uma rede de automação pode permitir o controle automático dos equipamentos com base no consumo de energia. Por exemplo, o sistema pode desligar máquinas que estão consumindo energia de forma desnecessária, ou ajustar a operação de acordo com a carga de trabalho, sem a necessidade de intervenção manual. Isso reduz a dependência de supervisão constante e diminui custos com mão de obra.

- **Planejamento e Programação de Uso:**

Com essas informações, a empresa pode reprogramar a utilização dos equipamentos para otimizar os períodos de operação, evitando sobrecarga em máquinas ou ociosidade. O planejamento adequado da utilização de equipamentos pode resultar em economia significativa de energia e tempo.

Este projeto não apenas proporciona redução de custos imediata, porém oferece uma abordagem estratégica e proativa para a gestão de recursos energéticos, manutenção de equipamentos e otimização dos processos de produção. Ao adotar esse tipo de tecnologia, as empresas podem não só melhorar sua eficiência operacional, mas também garantir uma redução contínua dos custos ao longo do tempo, o que representa um valor significativo para qualquer negócio que busque maior competitividade e sustentabilidade financeira.

## **6. Projeto**

Esse projeto que utiliza o ESP32 é um sistema que desenvolvemos para monitorar corrente elétrica em tempo real. Ele combina sensores, um RTC (relógio em tempo real), um display LCD e comunicação via Bluetooth. Com isso,

podemos medir, monitorar e registrar o consumo de corrente em um equipamento, calcular estatísticas como tempo em uso, picos de corrente e menores valores registrados, além de enviar todas essas informações para um dispositivo Bluetooth em tempo real.

### **Principais funcionalidades:**

**Medição de Corrente:** Utiliza o sensor ACS712 para captar a corrente elétrica em tempo real. Com a biblioteca EmonLib, consegue calcular a corrente RMS, o que permite definir se o equipamento está em operação ou não, com base em um limite de corrente (threshold) pré-definido.

**Monitoramento de Operação:** verifica se a corrente excede o threshold. Quando isso acontece, considera que o equipamento está em operação. Com isso, consegue monitorar o tempo total em uso e calcular a energia consumida, tanto com base na corrente média quanto na máxima.

**Registro das Maiores e Menores Leituras:** Durante a operação, armazena as três maiores e três menores leituras de corrente, junto com os horários de cada pico, para ajudar a entender o desempenho e a estabilidade do equipamento.

## **6.1 Componentes**

Um dos componentes principais deste projeto é o ESP32, um microcontrolador amplamente utilizado em sistemas de automação, dispositivos de interface homem-máquina (IHM) e em aplicações de IoT. Este microcontrolador oferece diversas funcionalidades, o que o torna ideal para aplicações que exigem comunicação sem fio, processamento em tempo real e interação com sensores e atuadores.

Graças à sua capacidade de conectar displays, sensores de toque e suportar comunicação em tempo real, o ESP32 permite a criação de interfaces interativas e responsivas. Além disso, seu suporte a modos de baixo consumo de energia

o torna indicado para dispositivos portáteis alimentados por bateria, prolongando a duração da carga.

O ESP32 coleta dados dos sensores, processa essas informações e pode exibi-las ao usuário ou transmitir para monitoramento e controle remoto. Dessa forma, ele permite a criação de sistemas de monitoramento eficientes e versáteis, adequados para ambientes industriais, domésticos e automotivos.

## Principais Funções e Utilidades do ESP32

### Conectividade:

- *Wi-Fi 802.11 b/g/n*: Permite a conexão do ESP32 a redes Wi-Fi para comunicação com a internet ou dispositivos locais.
- *Bluetooth v4.2*: Suporta Bluetooth clássico e BLE (Bluetooth Low Energy), possibilitando comunicação com dispositivos sem fio, como smartphones.

### Processamento:

- *Processador Dual-Core Tensilica Xtensa LX6*: Com dois núcleos de processamento a até 240 MHz, o ESP32 é capaz de executar tarefas em paralelo.
- *Unidade de Ponto Flutuante (FPU)*: Permite cálculos complexos de ponto flutuante, úteis em operações matemáticas avançadas.
- *Memória*: Possui cerca de 520 KB de SRAM e suporta memória flash adicional para armazenamento de código e dados.

### Entradas e Saídas (GPIO):

- *GPIOs*: Oferece de 30 a 36 pinos, configuráveis como entradas ou saídas digitais.
- *PWM*: Permite modulação por largura de pulso, usada para controlar intensidade de LEDs e velocidade de motores.
- *ADC e DAC*: Com até 18 canais ADC de 12 bits para sinais analógicos e dois canais DAC de 8 bits para geração de sinais analógicos.

### Comunicação Serial:

- *UART, I2C e SPI*: Vários protocolos para comunicação com sensores e outros módulos, incluindo displays LCD e sensores de alta velocidade.
- *CAN*: Suporta rede CAN, amplamente usada em sistemas automotivos.

### **Sensores e Periféricos Integrados:**

- *Sensor de Temperatura Interno*: Mais indicado para monitoramento interno do chip.
- *Touch Capacitivo*: Até 10 pinos com detecção de toque, permitindo criação de interfaces sensíveis ao toque sem hardware adicional.

### **Modos de Energia e Segurança:**

- *Modos de Baixo Consumo*: Light Sleep, Deep Sleep e Hibernation, adequados para dispositivos alimentados por bateria.
- *Criptografia e Boot Seguro*: Suporte a criptografia AES, SHA-2 e RSA, e boot seguro, assegurando comunicações e firmware autênticos.

### **Programação e Desenvolvimento:**

- Compatível com a IDE Arduino, Espressif IDF e MicroPython, proporcionando flexibilidade no desenvolvimento.

### **Expansor de Portas**

O módulo expansor de portas amplia a quantidade de pinos de entrada e saída (GPIOs) disponíveis para o ESP32, comunicando-se com o microcontrolador via I2C ou SPI. Esse módulo facilita o controle de múltiplos dispositivos, simplificando o cabeamento e ampliando as capacidades do projeto.

### **Sensor de Corrente**

O módulo sensor de corrente mede a quantidade de corrente elétrica em um circuito, útil para monitorar consumo de energia e detectar sobrecargas. Ele converte a corrente em um sinal de tensão proporcional, lido pelo ESP32 para avaliar o desempenho de dispositivos conectados.

### **Sensor de Temperatura**

No ESP32, sensores de temperatura são valiosos para monitorar ambientes e controlar processos industriais. Integrando-se ao microcontrolador, eles

permitem sistemas de resposta automática a mudanças de temperatura, como ajustes de climatização e alertas, promovendo eficiência energética.

### **Display LCD 16x2**

O display LCD 16x2 exibe até 32 caracteres em duas linhas de 16 colunas. Utiliza cristais líquidos para formar caracteres, e o fundo azul com letras brancas oferece boa visibilidade. Com o ESP32, ele permite a exibição de informações em tempo real, como leituras de sensores e alertas, tornando o sistema mais interativo.

### **RTC (Relógio de Tempo Real)**

O módulo RTC mantém a contagem precisa do tempo, armazenando horas, minutos, segundos, dias, meses e anos. Operando de forma independente do ESP32 e com uma bateria própria, ele mantém o tempo mesmo sem energia no sistema. No ESP32, o RTC é usado em sistemas que exigem monitoramento contínuo, registro de dados com marcação temporal e execução de tarefas em intervalos específicos, essenciais para automação e controle remoto.

## **6.2 Linguagem de programação e lógica**

Para esse projeto, a programação foi feita em C++, com foco na plataforma ESP32. A lógica principal envolve monitorar variáveis de corrente, vibração e outros parâmetros em tempo real, usando sensores para medir o comportamento do equipamento e registrar eventos que indicam operação, falhas ou necessidade de manutenção.

### **Lógica para cálculo do tempo de uso e consumo de energia**

- **Monitoramento de Corrente:** A corrente é medida continuamente pelo sensor ACS712, calculada como corrente RMS com a biblioteca EmonLib. Sempre que a corrente excede o limite (threshold) predefinido, consideramos que o equipamento está "em operação".
- **Tempo em Operação:** No início da operação (corrente acima do threshold), é registrado o horário de início com o RTC (DS3231). Quando o equipamento desliga (corrente abaixo do threshold), o tempo de

operação é calculado como a diferença entre o horário de início e o fim. O tempo é armazenado em milissegundos, mas é convertido em segundos ou horas, dependendo da necessidade.

- **Cálculo de Energia Consumida:** Com o valor de corrente e a tensão do sistema, é calculada a energia em watt-hora (Wh) multiplicando corrente média, tensão e tempo de operação. É feita uma distinção entre o consumo pela corrente média e o consumo pelo pico de corrente, possibilitando análises mais detalhadas.

### **Lógica para calcular possíveis manutenções**

- **Sensor de Vibração:** Monitora níveis de vibração durante a operação. Quando a vibração ultrapassa um limite pré-estabelecido, o sistema gera uma notificação via Bluetooth, indicando que a vibração está anormal.
- **Temperatura e Aquecimento:** Caso o projeto inclua um sensor de temperatura, este pode registrar a temperatura do equipamento. Ao ultrapassar um limite crítico, o sistema envia um alerta de possível sobrecarga.
- **Corrente Fora do Normal:** Se a corrente medida estiver muito acima ou abaixo do esperado, isso pode indicar uma falha elétrica ou problema mecânico. O sistema registra e envia informações sobre essas leituras fora do normal.

Esse conjunto de verificações, com variáveis para corrente, temperatura e vibração, ajuda a identificar possíveis condições de manutenção. A integração desses parâmetros permite um monitoramento contínuo e um aviso imediato, dando suporte a ações preventivas antes que ocorram danos mais graves.

#### **6.2.1 Código**

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <EmonLib.h> // Biblioteca EmonLib para leitura de corrente

#include <BluetoothSerial.h>
```

```
#include <RTClib.h>      // Biblioteca para RTC (DS3231)
```

```
// Definições do LCD I2C
```

```
#define I2C_ADDR 0x27
```

```
LiquidCrystal_I2C lcd(I2C_ADDR, 16, 2);
```

```
// Inicialização do Bluetooth
```

```
BluetoothSerial SerialBT;
```

```
// Inicialização do RTC
```

```
RTC_DS3231 rtc;
```

```
// Pino do sensor de corrente ACS712
```

```
const int currentSensorPin = 34;
```

```
// Parâmetros do sensor
```

```
EnergyMonitor emon;
```

```
float projectVoltage = 127.0; // Tensão de trabalho do sistema
```

```
float threshold = 1.30; // Limite para detectar operação
```

```
// Variáveis de controle e monitoramento
```

```
unsigned long tempoInicio = 0;
```

```
unsigned long tempoOperacao = 0;
```

```
float energiaTotalMediana = 0.0;
```

```
float energiaTotalMaior = 0.0;
```

```

float maioresLeituras[3] = {0.0, 0.0, 0.0};
float menoresLeituras[3] = {1000.0, 1000.0, 1000.0};
DateTime horarioMaiores[3], horarioMenores[3]; // Armazena horários
dos picos

bool emOperacao = false;
DateTime inicioOperacao, fimOperacao;

// Controle de atualização do LCD e Serial/Bluetooth
unsigned long lastLCDUpdate = 0;
unsigned long lastSerialUpdate = 0;
const unsigned long lcdInterval = 3000;
const unsigned long serialIntervalOperacao = 5000;
const unsigned long serialIntervalForaOperacao = 3000;

void inicializarI2C() {
    // Configura a comunicação I2C nos pinos GPIO 25 (SDA) e 26 (SCL)
    Wire.begin(25, 26);

    // Inicializa o RTC
    if (!rtc.begin()) {
        Serial.println("RTC não encontrado!");
        SerialBT.println("RTC não encontrado!");
        while (1);
    }
    if (rtc.lostPower()) {
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

```

```
}

// Inicializa o LCD

lcd.begin(16, 2);
lcd.backlight();

}

void setup() {
    Serial.begin(115200);
    SerialBT.begin("ESP32_Device");

    inicializarI2C();
    delay(2000);
    lcd.clear();
    lcd.print("Iniciando... ");

    emon.current(currentSensorPin, 21.1); // Configura sensor ACS712
    lcd.clear();
}

void atualizarLCD(float corrente, unsigned long tempoAtual) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Corrente: ");
    lcd.print(corrente, 2);
```

```
lcd.print(" A");

if (emOperacao) {
    lcd.setCursor(0, 1);
    lcd.print("Em uso");
    lcd.setCursor(0, 2);
    lcd.print("Duracao: ");
    lcd.print(tempoAtual);
    lcd.print(" s");
    lcd.setCursor(0, 3);
    lcd.print("Inicio: ");
    lcd.print(inicioOperacao.day(), DEC);
    lcd.print("/");
    lcd.print(inicioOperacao.month(), DEC);
    lcd.print("/");
    lcd.print(inicioOperacao.year(), DEC);
} else {
    lcd.setCursor(0, 1);
    lcd.print("Parado");
}

void verificarOperacao(float corrente) {
    if (corrente > threshold) {
        if (!emOperacao) {
```

```

    emOperacao = true;

    tempoInicio = millis();

    inicioOperacao = rtc.now();

    Serial.printf("Inicio: %02d/%02d/%04d %02d:%02d:%02d\n",
                  inicioOperacao.day(),               inicioOperacao.month(),
                  inicioOperacao.year(),              inicioOperacao.minute(),
                  inicioOperacao.hour(),             inicioOperacao.second());

    SerialBT.printf("Inicio      de      operacao:      %02d/%02d/%04d
%02d:%02d:%02d\n",
                  inicioOperacao.day(),               inicioOperacao.month(),
                  inicioOperacao.year(),              inicioOperacao.minute(),
                  inicioOperacao.hour(),             inicioOperacao.second());

    resetarLeituras();

}

atualizarMaioresMenoresLeituras(corrente);

} else if (emOperacao) {

    emOperacao = false;

    tempoOperacao = millis() - tempoInicio;

    fimOperacao = rtc.now();

    enviarResumo();

}

}

```

```
void resetarLeituras() {  
    for (int i = 0; i < 3; i++) {  
        maioresLeituras[i] = 0.0;  
        menoresLeituras[i] = 1000.0;  
    }  
    energiaTotalMediana = 0.0;  
    energiaTotalMaior = 0.0;  
}  
  
void atualizarMaioresMenoresLeituras(float corrente) {  
    DateTime agora = rtc.now();  
  
    if (corrente > maioresLeituras[0]) {  
        maioresLeituras[2] = maioresLeituras[1];  
        maioresLeituras[1] = maioresLeituras[0];  
        maioresLeituras[0] = corrente;  
        horarioMaiores[0] = agora;  
    }  
  
    if (corrente < menoresLeituras[0]) {  
        menoresLeituras[2] = menoresLeituras[1];  
        menoresLeituras[1] = menoresLeituras[0];  
        menoresLeituras[0] = corrente;  
        horarioMenores[0] = agora;  
    }  
}
```

```

}

}

float calcularMediana() {
    return (maioresLeituras[0] + menoresLeituras[0]) / 2;
}

void enviarResumo() {
    float tempoHoras = tempoOperacao / 3600000.0;
    float mediana = calcularMediana();
    float maiorValor = maioresLeituras[0];

    float energiaMediana = projectVoltage * mediana * tempoHoras;
    float energiaMaior = projectVoltage * maiorValor * tempoHoras;

    energiaTotalMediana += energiaMediana;
    energiaTotalMaior += energiaMaior;

    // Envia informações de resumo
    Serial.printf("Inicio da operacao: %02d/%02d/%04d
    %02d:%02d:%02d\n",
        inicioOperacao.day(), inicioOperacao.month(),
        inicioOperacao.year(),
        inicioOperacao.hour(), inicioOperacao.minute(),
        inicioOperacao.second());
}

```

```

Serial.printf("Fim      da      operacao:      %02d/%02d/%04d
%02d:%02d:%02d\n",
               fimOperacao.day(),
               fimOperacao.month(),
               fimOperacao.year(),
               fimOperacao.hour(),
               fimOperacao.minute(),
               fimOperacao.second());

```

```

Serial.printf("Tempo total: %.2f s\n3 Maiores: %.2f A, %.2f A, %.2f A\n"
               "3 Menores: %.2f A, %.2f A, %.2f A\n"
               "Consumo Mediana: %.2f Wh, Consumo Maior: %.2f Wh\n",
               tempoOperacao / 1000.0,
               maioresLeituras[0], maioresLeituras[1], maioresLeituras[2],
               menoresLeituras[0],               menoresLeituras[1],
               menoresLeituras[2],
               energiaMediana, energiaMaior
);

```

```

SerialBT.printf("Inicio      da      operacao:      %02d/%02d/%04d
%02d:%02d:%02d\n",
               inicioOperacao.day(),
               inicioOperacao.month(),
               inicioOperacao.year(),
               inicioOperacao.hour(),
               inicioOperacao.minute(),
               inicioOperacao.second());

```

```

SerialBT.printf("Fim      da      operacao:      %02d/%02d/%04d
%02d:%02d:%02d\n",
               fimOperacao.day(),
               fimOperacao.month(),
               fimOperacao.year());

```

```

    fimOperacao.hour(),           fimOperacao.minute(),
    fimOperacao.second());

SerialBT.printf("Tempo total: %.2f s\n3 Maiores: %.2f A, %.2f A, %.2f
A\n"

"3 Menores: %.2f A, %.2f A, %.2f A\n"

"Consumo Mediana: %.2f Wh, Consumo Maior: %.2f Wh\n",
tempoOperacao / 1000.0,
maioresLeituras[0], maioresLeituras[1], maioresLeituras[2],
menoresLeituras[0], menoresLeituras[1],
menoresLeituras[2],
energiaMediana, energiaMaior

);
}

void ajustarDataHora(String comando) {
    int dia, mes, ano, hora, minuto, segundo;
    sscanf(comando.c_str(), "%d/%d/%d %d:%d:%d", &dia, &mes, &ano,
&hora, &minuto, &segundo);

    // Ajusta o RTC
    rtc.adjust(DateTime(ano, mes, dia, hora, minuto, segundo));
    Serial.println("Data e hora ajustadas com sucesso!");
    SerialBT.println("Data e hora ajustadas com sucesso!");
}

```

```

void loop() {

    float Irms = emon.calcIrms(1480); // Cálculo da corrente RMS

    verificarOperacao(Irms); // Verifica se está em operação

    unsigned long tempoAtual = millis() - tempoInicio;

    // Atualiza o LCD a cada 3 segundos

    if (millis() - lastLCDUpdate > lcdInterval) {

        atualizarLCD(Irms, tempoAtual);

        lastLCDUpdate = millis();

    }

    // Envia dados via Bluetooth

    if (emOperacao && (millis() - lastSerialUpdate > serialIntervalOperacao)) {

        SerialBT.printf("Corrente: %.2f A\n", Irms);

        lastSerialUpdate = millis();

    } else if (!emOperacao && (millis() - lastSerialUpdate > serialIntervalForaOperacao)) {

        SerialBT.printf("Parado, Corrente: %.2f A\n", Irms);

        lastSerialUpdate = millis();

    }

    // Verifica se há comandos via Bluetooth

    if (SerialBT.available()) {

        String comando = SerialBT.readStringUntil('\n');

```

```

if (comando.startsWith("SET_TIME ")) {
    ajustarDataHora(comando.substring(9));
}

}
}

```

### 6.3 Lei de ohms e cálculo de tempo

A **Lei de Ohm** é fundamental para entender e calcular a relação entre tensão (voltagem), corrente e resistência em circuitos elétricos. A lei é expressa pela fórmula:

$$V=I \times R \quad V=I \times R$$

onde:

- $V$  é a tensão (em volts),
- $I$  é a corrente (em amperes),
- $R$  é a resistência (em ohms).

Com essa relação, podemos calcular qualquer uma dessas grandezas desde que conheçamos as outras duas. A Lei de Ohm é bastante usada para dimensionar circuitos, verificar valores seguros de operação e entender o comportamento de componentes elétricos.

### Cálculo de Potência e Consumo de Energia

A potência elétrica consumida em um circuito pode ser calculada pela fórmula:

$$P=V \times I \quad P=V \times I$$

onde:

- $P$  é a potência (em watts),

- $V$  é a tensão,
- $I$  é a corrente.

Para o cálculo do **consumo de energia**, usamos o valor da potência multiplicado pelo **tempo de operação**:

$$E = P \times t \quad E = P \times t$$

onde:

- $E$  é a energia consumida (em watt-hora, Wh),
- $P$  é a potência (em watts),
- $t$  é o tempo de operação (em horas).

Assim, se conhecemos a corrente e a tensão no circuito, podemos calcular a potência e, com o tempo de operação, o consumo total de energia.

### Exemplo de Cálculo de Tempo e Consumo de Energia

Se um equipamento opera com **tensão de 127 V** e consome **2 A de corrente** durante **3 horas**, temos:

#### 1. Potência:

$$P = V \times I = 127 \text{ V} \times 2 \text{ A} = 254 \text{ W}$$

$$P = V \times I = 127 \text{ V} \times 2 \text{ A} = 254 \text{ W}$$

#### 2. Energia:

$$E = P \times t = 254 \text{ W} \times 3 \text{ h} = 762 \text{ Wh}$$

$$E = P \times t = 254 \text{ W} \times 3 \text{ h} = 762 \text{ Wh}$$

Portanto, o equipamento consumiria **762 Wh** após 3 horas de operação contínua.

Esse cálculo é essencial para avaliar o consumo energético de um dispositivo, dimensionar fontes de alimentação, verificar limites de operação de componentes e, em um projeto como o de monitoramento de corrente com ESP32, fornecer dados precisos para análise de consumo e manutenção preventiva.

## REFERÊNCIAS BIBLIOGRÁFICAS

**Souza, R. de C.** (2018). *Indústria 4.0: Conceitos e fundamentos*. São Paulo: Blucher.

**Gimenez, Salvador Pinillos** Microcontroladores 8051: Teoria e Pratica

[Como a Indústria 4.0 está impulsionando a tecnologia IH](#)

[Placa ESP32: O que é, para que serve e uso! | Victor Vision](#)

[Conhecendo o ESP32 | Portal Vida de Silício \(vidadesilicio.com.br\)](#)

[O que é ESP32? Pra que serve? Quando usar? \(lobodarobotica.com\)](#)

[ESP32-DevKitC V4 - - — esp-dev-kits latest documentation \(espressif.com\)](#)

[ESP32: Detalhes internos e pinagem - Fernando K Tecnologia](#)