

Monitoramento inteligente de geradores: Desenvolvimento de um protótipo para leitura e armazenamento de dados em plataforma Arduino.

Álvaro Dantas de Medeiros Filho
Departamento de Engenharia e
Tecnologia
Universidade Federal Rural do Semi
Árido
Mossoró, Brasil
alvaro.filho@alunos.ufersa.edu.br

Idalmir de Souza Queiroz Júnior
Departamento de Engenharia e
Tecnologia
Universidade Federal Rural do Semi
Árido
Mossoró, Brasil
idalmir@ufersa.edu.br

Resumo — O estudo de circuitos eletrônicos e digitais se faz importante para qualquer nível de automação, seja útil em tarefas rotineiras ou automação mais complexa. A engenharia elétrica, portanto, tem papel fundamental nessa automação, pois está diretamente relacionada a esse processo. Neste contexto estamos diante de algumas aplicações deste processo de automação utilizando Arduino. Comprovamos que essa plataforma é um meio eficiente de buscar essa automação, até mesmo para tarefas complexa em residências e indústrias. Dessa maneira, o presente trabalho propõe um sistema de automação de um protótipo que realiza a leitura de valores de tensão, corrente e potência gerados por um gerador, além de imprimir esses valores em uma tela e armazená-los em um banco de dados. O objetivo é fornecer ao dono do gerador um maior controle sobre o seu consumo de energia, permitindo que ele monitore e ajuste o seu uso. O protótipo utiliza um microcontrolador Arduino e um módulo SD para armazenar os dados coletados. A partir desses dados, é possível realizar análises e tomar decisões em relação ao uso do gerador, aumentando sua eficiência e reduzindo custos.

Palavras-chave— gerador, Arduino, monitoramento, eficiência

Abstract— The study of electronic and digital circuits is important for any level of automation, whether useful in routine tasks or more complex automation. Electrical engineering, therefore, plays a fundamental role in this automation, as it is directly related to this process. In this context we are facing some applications of this automation process using Arduino. We have proven that this platform is an efficient means of seeking this automation, even for complex tasks in homes and industries. In this way, this work proposes an automation system for creating a prototype that reads voltage, current and power values generated by a generator, in addition to printing these values on a screen and storing them in a database. The objective is to provide the generator owner with greater control over his energy consumption, allowing him to monitor and adjust his use. The prototype uses an Arduino microcontroller and an SD module to store the collected data. Based on this data, it is possible to carry out analyzes and make decisions regarding the use of the generator, increasing its efficiency and reducing costs.

Keywords— generator, Arduino, monitoring, efficiency

I. INTRODUÇÃO

A utilização de geradores se faz importante em diversas situações, tanto domésticas quanto industriais. Esses dispositivos fornecem energia em apagões, emergências, eventos ao ar livre e locais remotos onde a energia elétrica não está disponível.

No entanto, os proprietários desses geradores têm um controle preciso sobre a quantidade de energia produzida para garantir que a quantidade de energia produzida seja suficiente para atender às suas necessidades e monitorar o consumo de combustível do gerador.

Para facilitar esse controle, torna-se indispensável a utilização de um sistema impresso em tela. Esta tecnologia permite aos proprietários monitorar os valores de tensão, corrente e energia produzidos pelo sistema em tempo real de forma clara e intuitiva. Portanto, o sistema de monitoramento por tela e sistema de armazenamento de dados permite que os proprietários de geradores verifiquem se seus equipamentos estão funcionando corretamente, reduzam o consumo de combustível, economizem dinheiro e evitem quedas de energia.

Utilizar um Arduino para monitorar a quantidade de eletricidade produzida por um gerador a combustão é uma forma econômica e eficiente de monitorar a energia elétrica produzida pelo sistema.

Em resumo, usar um Arduino para monitorar a quantidade de eletricidade produzida por um gerador de combustão garante um fornecimento confiável e seguro de energia elétrica, identifica possíveis problemas e otimiza o consumo.

A seguir será apresentada a fundamentação teórica, a metodologia desenvolvida, os resultados e discussões e as conclusões, por fim é apresentado o código no Anexo I.

II. FUNDAMENTAÇÃO TEÓRICA

O Arduino é uma plataforma microcontrolada barata e fácil de programar que pode ser programado para diversas funções, inclusive pode-se ler esses sensores em forma de números que variam de acordo com o sinal e na conversão se faz a adaptação para cada tipo de dado e pode-se também

exibir informações em um display LCD ou enviar esses dados para um sistema de armazenamento.

Ao usar um Arduino com o conjunto de sensores corretos designados para obtenção de dados de maneira coesa pode-se monitorar a quantidade de eletricidade produzida por um gerador. Monitorando o consumo de energia elétrica que sai do gerador para energizar um aparelhamento conectado em sua saída, medindo o desempenho do gerador correlacionando o quanto foi gerado em KW/h e o quanto gerou de custo para produzir tal energia.

A. Circuitos eletrônicos

Segundo [1] os circuitos eletrônicos têm inúmeras aplicações, entre elas muitas estão correlacionadas com aplicações no cotidiano de pessoas e indústrias.

Existem vários tipos de circuitos que são classificados de acordo com o comportamento energético, mecanismos de fabricação e o tipo de tensão. Conceitos como a lei de Ohm é requerido nestes tipos de circuitos. A corrente elétrica pode ser relacionada de forma direta com a diferença de potencial, porém é relacionada de forma inversa a resistência daquele circuito ou sistema [2]. Essa lei pode ser interpretada pela seguinte equação:

$$V = R \times I \quad (1)$$

B. Circuitos digitais

Circuitos digitais são um tipo de circuito eletrônico em que o funcionamento é baseado na lógica binária, aonde toda a informação é armazenada e processada na forma de 0's e 1's. Esta forma de representação é conseguida através de dois níveis discretos de tensão elétrica. Segundo [3] a representação de comportamentos na forma digital tem algumas vantagens se comparadas a esse mesmo comportamento descrito na forma analógica, devido a forma digital ser mais fácil de representar em aplicações eletrônicas.

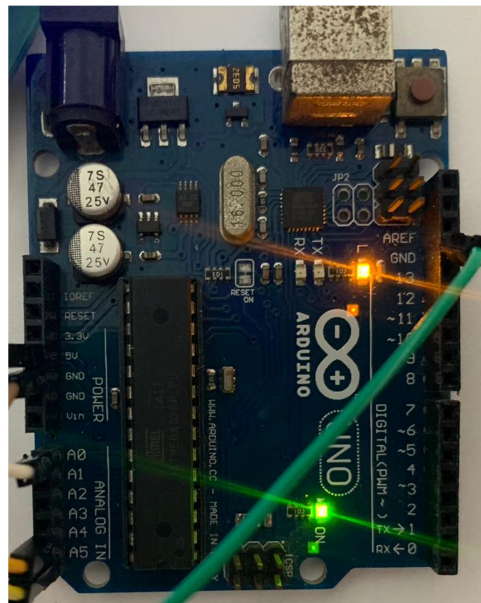
C. Arduino

O Arduino funciona como um computador pequeno, programável e que pode processar entradas e saídas entre componentes externos e o próprio dispositivo [4]. Por isso o Arduino é um Sistema que consegue interagir com o ambiente utilizando o seu hardware e software.

Além disso, esse Sistema pode ser ampliado com o auxílio de sensores, como um sensor de umidade, por exemplo, que possibilitaria um controle maior da plataforma sobre o sistema criado. O Arduino é composto por alguns componentes como um microprocessador, um cristal ou oscilador e também por um regulador linear com tensão de operação de 5 V [4].

A alimentação do Arduino de forma geral pode ser feita por meio de uma fonte de alimentação de 7 a 12 V, pelo conector de alimentação do Arduino ou direto pelas portas Vin e GND. Além disso, a alimentação também pode ser feita via cabo USB de 5 V. A figura 1, abaixo, mostra o Arduino utilizado.

Figura 01 – Arduino UNO



Fonte: Autoria Própria ,2023

A programação do Arduino usa software livre que permite escrever livremente o código do programa. Este IDE é um Ambiente de Desenvolvimento Integrado (IDE do inglês Integrated Development Environment), permite aos usuários criar um programa de computador que atua como uma série de inscrições e fazer o upload deste programa para uma placa Arduino [4].

Quando o programa é escrito na IDE ele precisará ser transferido para a placa Arduino através da porta USB, contudo, para as placas que não possuem esta entrada, podem ser programadas através dos pinos ICSP.

A placa Arduino é composta por vários componentes, um desses componentes é o microcontrolador que funciona como o cérebro da placa. Este componente é onde fica armazenado o programa e responsável também por receber os sinais de entrada (*inputs*), processar as informações e enviar de volta as informações tratadas. Abaixo a figura mostra o microcontrolador ATMEL ATMEGA328 do Arduino UNO.

Figura 02 – Microcontrolador



Fonte: Autoria Própria, 2023.

Outro componente é o botão reset que é encontrado em algumas placas. Esse botão é muito utilizado para quando existirem travamentos ou reações não esperadas no

funcionamento da própria placa do Arduino. O botão reset encontra-se ilustrado abaixo.

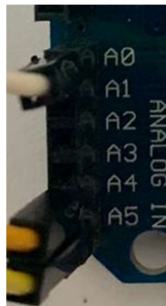
Figura 03 – Reset



Fonte: Autoria Própria, 2023.

A placa Arduino é composta por diversas portas ou pinagem. Essa pinagem pode ser dividida em pinos de alimentação, portas digitais e portas analógicas. As portas analógicas conseguem interpretar sinais analógicos e transforma-los em digitais, com isso o Arduino consegue trabalhar com esse tipo de informação. Para tanto as placas conseguem dividir em até 1024 “partes” a tensão da entrada, ou seja, utilizando uma placa com 5 V de entrada o Arduino consegue dividir e identificar uma leitura de 0,00488 Volts cada. As portas analógicas são ilustradas abaixo na placa de um Arduino UNO.

Figura 04 – Portas Analógicas



Fonte: Autoria Própria, 2023.

As portas digitais trabalham com dois tipos de sinais, ou dois níveis de tensão, um nível lógico alto (5 V) e um nível lógico baixo (0 V). Estas portas digitais podem ser configuradas como saída ou entrada, OUTPUT e INPUT respectivamente. Quando esta porta é definida como saída, ela fornece de 0 a 5 V, todavia a corrente normalmente é limitada a 30 mA. Quando utilizada como entrada, a porta receberá de um acessório ou periférico, 0 V ou 5 V. Abaixo as portas digitais são ilustradas em um Arduino UNO.

Figura 05 – Portas Digitais



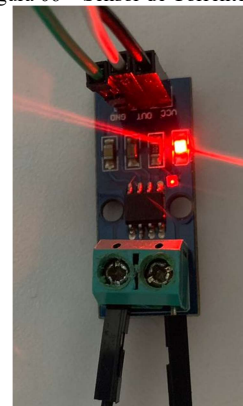
Fonte: Autoria Própria, 2023.

D. Sensores

A plataforma Arduino permite o uso de componentes como sensores para a criação dos circuitos. Segundo [5], os sensores funcionam basicamente como um dispositivo que tem como função, detectar e responder com eficiência a algum estímulo fornecido pelo ambiente.

Com isso com a junção do Arduino com os sensores, diversos projetos podem ser executados para diferentes objetivos. A escolha por algum sensor depende diretamente do objetivo da instalação, com isso avaliar as condições do ambiente é necessário para a escolha de um sensor mais adequado para aquela atividade proposta.

Figura 06 – Sensor de Corrente ACS712



Fonte: Autoria Própria, 2023.

E. Armazenamento de Dados

O Arduino permite o uso de componentes chamados de módulo cuja a função é integrar-se ao sistema no intuito de aumentar as possibilidades de programação, a integração de bancos de dados com módulos de armazenamento SD fornece uma solução de armazenamento de dados para o projeto eletrônico baseado no Arduino.

Através do uso de uma biblioteca específica o Arduino pode se comunicar com um banco de dados armazenado em um cartão SD e armazenar dados coletados em uma estrutura de banco de dados organizada. A figura abaixo explicita o módulo cartão SD.

Figura 07 – Módulo Cartão SD



F. Análise de Energia

A análise de energia é o ato de coletar e extrair dados e parâmetros elétricos, com o intuito de diagnosticar possíveis problemas e anomalias existentes na rede elétrica para que ao reparar aumente a eficiência energética, que nada mais é do que o estudo que se constitui de uma somatória de ações que buscam melhorar o uso da energia e sistema.

G. Gerador

Em muitas áreas, os geradores são equipamentos essenciais para fornecer energia elétrica em áreas remotas que não são alimentadas pela rede pública ou inacessíveis à rede. Um exemplo de gerador amplamente utilizado é o gerador Toyama TC950S.

O Toyama TC950S Generator é um dispositivo compacto, projetado para fornecer uma fonte de alimentação confiável. Com uma potência nominal de 800 W e uma potência máxima de 950 W, atende a diversas necessidades, desde uso doméstico até uso comercial leve.

Uma das vantagens do gerador Toyama TC950S é sua portabilidade, porém o gerador não é equipado com nenhum mostrador de dados, seja ele analógico ou digital. Abaixo a imagem do gerador em questão.

Figura 08 – Gerador Toyama TC950S



Fonte: Autoria Própria, 2023.

III. METODOLOGIA

A elaboração do projeto foi dividida em etapas, cada etapa foi desenvolvida com a finalidade de alcançar o objetivo proposto que é de projetar um sistema que faça o análise da energia gerada e possa imprimir os valores em tela e armazena-los em banco de dados.

A. Desenvolvimento do código

Nesta etapa foi elaborada a programação que comanda o circuito do sistema automatizado do projetado. Esta programação foi feita tomando como base o circuito desenvolvido no simulador. Podemos sintetizar a

programação em blocos, sendo eles a configuração do sistema e o bloco de repetição.

B. Programação

À princípio o código constitui em obtenção de valores pelas portas analógicas A0 e A1, onde se obtém os valores dos sensores de tensão e corrente respectivamente (valores de tensão em paralelo com o sistema e valores de corrente em série com a fase). Na sequência se fez o tratamento dos dados para convertê-los em unidades de medidas, tensão, corrente, potência.

Figura 09 – Pinagem do sistema e determinação de variáveis

```
int sensorPin = A0; //Sensor de tensão conectado ao pino A0
int sensorPin = A1; //Sensor de corrente conectado ao pino A1
int sensorValue = 0; //Valor lido do sensor de tensão
float voltagem = 0; //Tensão em Volts
float corrente = 0; //Corrente em Amperes
float potencia = 0; //Potência em Watts
float kwh = 0; //Energia em kWh
```

Fonte: Autoria Própria, 2023

A potência se dá pela multiplicação da tensão multiplicada pela corrente. Após a obtenção do valor da potência consegue-se explicitar o valor da energia dissipada em KW/h, através da multiplicação da potência pelo intervalo de verificação de valor (1000), dividido pela hora (3600 segundos).

Figura 10 – Conversão de potência para KW/h

```
energia += potencia * (intervalo / 3600.0); //
intervalo em segundos, dividido por 3600 para converter
em horas
lcd.setCursor(10, 1);
lcd.print("KWh:");
lcd.print(energia, 2);
```

Fonte: Autoria Própria, 2023

Após a primeira verificação o programa gera um valor em KW/h e retorna para verificação de valores e disponibiliza de novos valores através do comando “void loop” dependendo das grandezas lidas pelos sensores.

Figura 11 – Comando void loop

```
void loop() {
    //Verifica se é hora de fazer a leitura
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        //Atualiza o tempo da última leitura
```

Fonte: Autoria Própria, 2023

Prosseguindo com a explicitação do programa, durante o período de verificação o sistema segue armazenando e atualizando valores de acordo com o time em um modulo contendo cartão SD.

Figura 12– Comandos envolvendo módulo cartão SD card

```
#include <SD.h>
unsigned long previousMillis = 0; //Armazena o tempo de
leitura anterior
const long interval = 1000; //Intervalo de tempo de
leitura em milissegundos
// Salvar valores no cartão SD
logFile = SD.open("datalog.txt", FILE_WRITE);
if (logFile) {
    logFile.print(tensao);
    logFile.print(",");
    logFile.print(corrente);
    logFile.print(",");
    logFile.print(potencia);
    logFile.print(",");
    logFile.print(energia, 2);
    logFile.println();
    logFile.close();
}
else {
    // Esperar intervalo de tempo
    delay(intervalo * 1000); // intervalo em segundos,
multiplicado por 1000 para converter em milissegundos
}

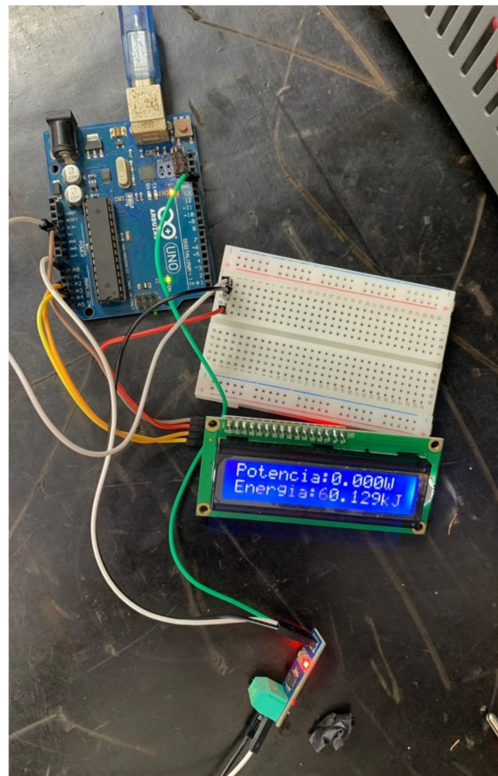
Serial.println("Erro ao gravar arquivo");
}
```

Fonte: Autoria Própria, 2023

C. Testagem do circuito

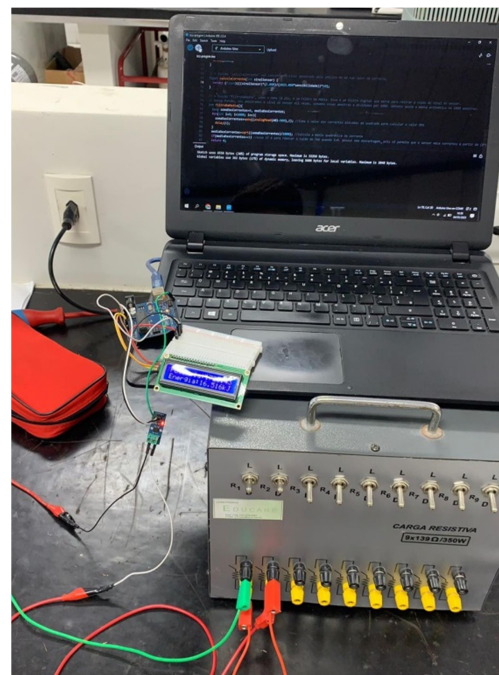
Para testar a funcionalidade do sistema foi efetuada a pinagem nos respectivos componentes e submetidos a teste laboratorial.

Figura 13 – Sistema em operação sem carga



Fonte: Autoria Própria, 2023.

Figura 14 – Sistema em operação com carga



Fonte: Autoria Própria, 2023.

Para acionamento do sistema foi utilizada uma carga resistiva de 278Ω a uma tensão de 220v onde o mesmo produziu uma corrente de aproximadamente 0,79A. Após o procedimento fazendo uso de uma chave amperímetro comprovou-se a exatidão no sensoriamento.

Figura 15 – Amperímetro Minipa ET-4030



Fonte: Autoria Própria, 2023.

IV. RESULTADOS E DISCUSSÕES

O projeto está funcionando da forma prevista, após a montagem e ao submeter o sistema a carga ele imprimiu valores em tela de forma já prevista.

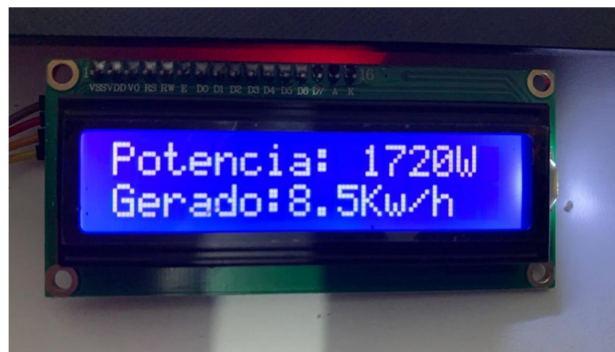
Figura 16 – Sistema imprimindo Tensão e Corrente.



Fonte: Autoria Própria, 2023.

O sistema imprime em tela de forma autônoma valores atualizados de quanto o sistema gerou até determinado momento.

Figura 17 – Sistema imprimindo Potência e valor gerado pelo tempo de uso (KW/h).



Fonte: Autoria Própria, 2023.

O sistema faz a conversão do valor de KW/h para valores em real, levando em consideração valores de taxas de transferência e uso de energia. No caso se fez uso de valores fictícios que nesse caso foi de R\$ 0,94 por KW/h gerados.

Figura 18 – Valor em reais (R\$) gerado.



Fonte: Autoria Própria, 2023.

V. CONCLUSÃO

Fazendo um análise superficial deste projeto, pode-se concluir que, a automação cada vez mais se torna acessível a sociedade, devido a flexibilização de diversos projetos para inúmeros problemas onde pode-se utilizá-la, pois até mesmo algo de tal complexidade pode ser sintetizado a um projeto relativamente barato e de fácil manuseio. Esse projeto alcançou o objetivo proposto, com o auxílio da plataforma Arduino um sistema o protótipo se mostrou uma ótima solução para pequenos sistemas de geração monofásico, sendo assim uma solução para todo e qualquer meio de geração de energia alternada que não possua sistema de amostragem de dados.

AGRADECIMENTOS

Gostaria de agradecer a Deus por ter me dado saúde e força para concluir o curso de Engenharia Elétrica. Agradeço também aos meus pais, minha irmã e toda a minha família pelo apoio incondicional em todos os momentos. Agradeço ao meu orientador pela orientação, ensinamentos e paciência durante o desenvolvimento do meu trabalho de conclusão de curso. Agradeço também a todo o corpo docente, que contribuiu para a minha formação acadêmica. Por fim, agradeço aos meus amigos e colegas de

turma pelo apoio e amizade durante toda a jornada acadêmica.

REFERÊNCIAS

- [1] MATTED, Henrique. Conceitos de eletricidade. **MUNDO DA ELÉTRICA**. Disponível em: <https://www.mundodaeletrica.com.br/tipos-e-aplicacoes-dos-circuitos-eletronicos/>. Acessado em: 8 de maio 2023.
- [2] HALLIDAY, David; RESNICK, Robert. WALKER, Jearl. Fundamentos Física. Tradução de Ronaldo Sérgio de Biasi. 9. Ed. Rio de Janeiro: LTC, 2012.
- [3] FLOYD, Thomas. Sistemas Digitais: Fundamentos e aplicações. 9ªed. São Paulo: Artmed, 2007.
- [4] MCROBERTS, Michael. Arduino básico, p. 22 – 26. Novatec Editora. 2ªedição. 2015.
- [5] Matted, Henrique. Conceitos de sensores. **MUNDO DA ELÉTRICA**. Disponível em: <https://www.mundodaeletrica.com.br/o-que-sao-sensores-e-quais-as-suas-aplicacoes/>. Acessado em: 08 de maio 2023.

ANEXO I

```
#include <FastIO.h>
#include <I2CIO.h>
#include <LCD.h>
#include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>
//Sensor de Corrente ACS712 medindo Corrente AC,
Potencia e Energia
#include <Wire.h> //Biblioteca para Comunicação I2C
#include <LiquidCrystal_I2C.h> //Biblioteca com as
funções do display
#include <SD.h>
LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7,3,
POSITIVE);// Inicializa o display no endereço 0x16

const int chipSelect = 4;
File dataFile;

int sensorPin = A0; //Sensor de tensão conectado ao
pino A0
int sensorPin = A1; //Sensor de corrente conectado ao
pino A1
int sensorValue = 0; //Valor lido do sensor de tensão
float voltagem = 0; //Tensão em Volts
float corrente = 0; //Corrente em Amperes
float potencia = 0; //Potência em Watts
float kWh = 0; //Energia em kWh
unsigned long previousMillis = 0; //Armazena o tempo de
leitura anterior
const long interval = 1000; //Intervalo de tempo de
leitura em milissegundos

void setup() {
    Serial.begin(9600);
```

```
    lcd.begin(16, 2);
    lcd.print("Leitura de");
    lcd.setCursor(0, 1);
    lcd.print("Tensao, Corr. e Pot.");

    pinMode(chipSelect, OUTPUT);

    if (!SD.begin(chipSelect)) {
        lcd.clear();
        lcd.print("Erro ao");
        lcd.setCursor(0, 1);
        lcd.print("iniciar SD");
        return;
    }

    lcd.clear();
    lcd.print("Cartao SD");
    lcd.setCursor(0, 1);
    lcd.print("iniciado");
    delay(2000);
    lcd.clear();
}

void loop() {
    //Verifica se é hora de fazer a leitura
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        //Atualiza o tempo da última leitura
        previousMillis = currentMillis;

        //Lê o valor do sensor de tensão
        sensorValue = analogRead(sensorPin);

        //Converte o valor lido para Volts
        voltagem = sensorValue * (5.0 / 1023.0);

        //Lê o valor do sensor de corrente
        corrente = analogRead(A1);

        //Converte o valor lido para Amperes
        corrente = (corrente - 512.0)

        // Exibir valores no display LCD
        lcd.setCursor(0, 0);
        lcd.print("Tensao: ");
        lcd.print(voltagem);
        lcd.print("V");
        lcd.setCursor(0, 1);
        lcd.print("Corrente: ");
        lcd.print(corrente);
        lcd.print("A");
```

```
// Calcular e exibir a potência
float potencia = tensao * corrente;
lcd.setCursor(10, 0);
lcd.print("P:");
lcd.print(potencia);
lcd.print("W");
// Calcular e exibir a energia gerada
energia += potencia * (intervalo / 3600.0); //
intervalo em segundos, dividido por 3600 para converter
em horas
lcd.setCursor(10, 1);
lcd.print("KWh:");
lcd.print(energia, 2);

// Salvar valores no cartão SD
logFile = SD.open("datalog.txt", FILE_WRITE);
if (logFile) {
    logFile.print(tensao);
    logFile.print(",");
    logFile.print(corrente);
    logFile.print(",");
    logFile.print(potencia);
    logFile.print(",");
    logFile.print(energia, 2);
    logFile.println();
    logFile.close();
}
else {
    Serial.println("Erro ao gravar arquivo");
}

// Esperar intervalo de tempo
delay(intervalo * 1000); // intervalo em segundos,
multiplicado por 1000 para converter em milissegundos
}
```