

COMENTÁRIOS ACIMA DO CÓDIGO

O código exporta duas bibliotecas, “math.h” para operações matemáticas e “LiquidCrystal.h” para código envolvendo o display.

```
#include <LiquidCrystal.h>
#include <math.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int CT = A0;
double corrente;
double medido;
double energia;
double fonte = 220;
double offsv = 2.5;
double sensi = 0.9995;
double ardvoltagem;
```

A quarta linha de código é a declaração do LCD, a quinta linha configura uma variável pra entrada A0, na qual corresponde ao terminal ligado à tensão de saída do sensor. São declaradas variáveis para:

- Corrente: Calcular o valor da corrente por meio de parâmetros do sensor.
- Medido: Receber o inteiro correspondente a tensão que chega no arduino.
- Energia; Calcular o valor da energia gasta a cada segundo.
- Fonte: Receber o valor da tensão de alimentação do circuito.
- Tensão Off-set: Valor de ajuste offset do sensor (especificado por dados de fabricação).
- Sensibilidade: Valor de ajuste da sensibilidade do sensor (especificado por dados de experimentais, variável de sensor pra sensor).
- Voltagem do arduino: Receber o valor da tensão de saída do terminal do sensor.

```

void setup() {
    lcd.begin(20, 4);
    constantes();
}

void loop() {
    calculaCorrente();
    lcd.setCursor(13,1);
    lcd.print(corrente);

    calculaEnergia();

    lcd.setCursor(7,3);
    lcd.print(energia);

    delay(1000);
}

```

Todas as funções não retornam nada (void) e não tem argumentos. Iniciando com a função setup, que consiste em comandos feitos antes do arduino começar qualquer procedimento, tem dois comandos, um para inicializar o LCD, de 20 colunas e 4 linhas; e outro para executar a função constantes, que consiste em valores que não se alteram no LCD conforme seu funcionamento, mais adiante explicada em detalhes.

A próxima função é a loop, que tem essa denominação por repetir por tempo indefinido, enquanto o arduino for alimentado, a aplicação se mantém em **loop**. Nela estão comandos de LCD, e chamadas de funções calculaCorrente e calculaEnergia. Os nomes das funções são subliminares e autoexplicativas, com respectivos detalhamentos adiante, os comandos LCD por outro lado, são bem simples:

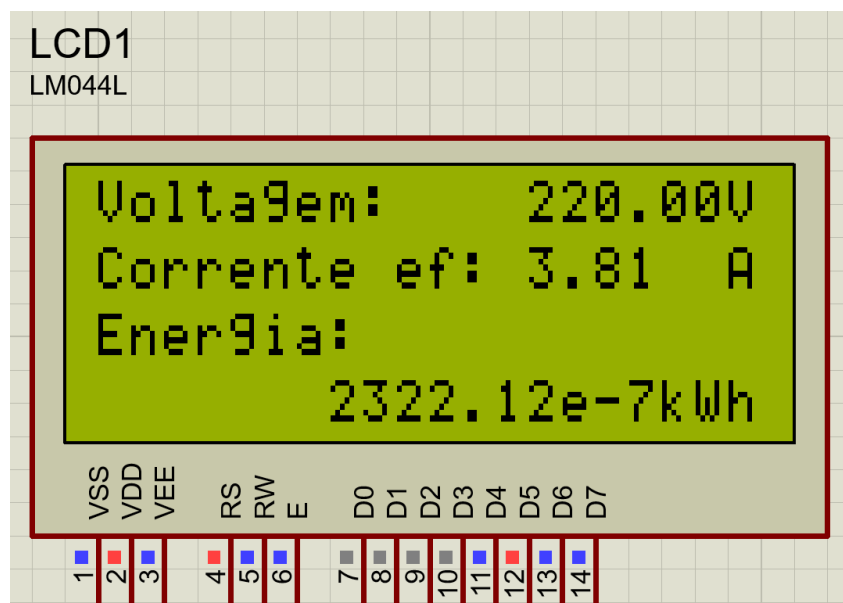
- lcd.setCursor(Coluna, Linha); é um comando que indica aonde “o cursor do LCD está localizado”, isto é, em qual região será “printada” valores de variáveis ou strings.
- Lcd.print(valor); é um comando semelhante a outros abordados em linguagem c ou Java (printf ou system.out.println); “printa” a informação que estiver entre os parênteses.

Para finalizar, utiliza-se um comando delay(); do arduino que serve para determinar tempos de espera majoritariamente para visualizar dados,

regularmente, informações tratadas no arduino são de velocidade de milissegundos, esses comandos de “atraso” servem para um observador ser capaz de observar e processar dados fornecidos ou calculados pelo arduino.

```
void constantes() {  
  
    lcd.setCursor(0,0);  
    lcd.print("Voltagem: ");  
    lcd.setCursor(13,0);  
    lcd.print(fonte);  
    lcd.setCursor(0,1);  
    lcd.print("Corrente ef: ");  
    lcd.setCursor(0,2);  
    lcd.print("Energia: ");  
  
    lcd.setCursor(19,0);  
    lcd.print("V");  
    lcd.setCursor(19,1);  
    lcd.print("A");  
    lcd.setCursor(14,3);  
    lcd.print("e-7kWh");  
}
```

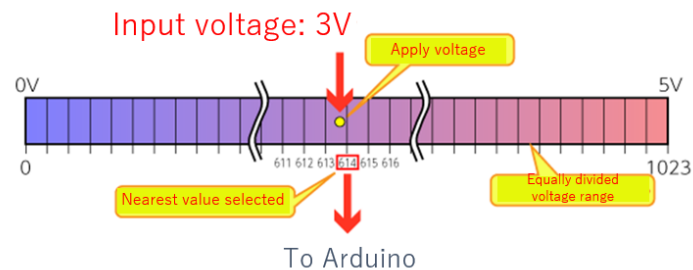
Uma vez explicadas as funções print e setCursor do LCD, e a proposta da função como elementos constantes disponibilizados no LCD, apenas um “print” da simulação é o suficiente para completo entendimento:



As últimas funções são as operações algébricas realizadas para disponibilizar valores de corrente e energia.

```
void calculaCorrente() {  
  
    medido = analogRead(CT);  
    ardvoltagem = (medido*5.0)/1024.0;  
  
    corrente = (10.0*(ardvoltagem - offsv)/sensi);  
}  
  
void calculaEnergia() {  
    energia = abs(fonte*corrente*2.77);  
}
```

A função calculaCorrente conta apenas com um comando excêntrico, conhecido como analogRead(); , esse comando lê o valor de tensão correspondente ao valor que recebe no terminal correspondente a CT (O nome dado ao terminal A0), porém esse dado é recebido em forma de número inteiro ao invés de número de ponto flutuante. A partir de uma escala relacional entre tensão e número inteiro, é possível determinar uma relação para determinar o valor da tensão que o terminal A0 recebe:



Integer values up to 1023 are used to process the program. However, the input value is not a voltage value. If you want to know the voltage value, you can calculate it as follows:

$$\text{Voltage} = \frac{5}{1023} \times \text{Input value}$$

Maximum voltage value

Number divided by AD converter

For example, if the value from the AD converter is "614", you can obtain about 3V by dividing by 1023 and then multiplying by 5.

A imagem acima explica a relação entre 5V corresponde ao número inteiro 1023. A última linha de comando da função que mede corrente se trata puramente de expressões orientadas pelo fabricante do sensor.

Enquanto a função calcula Energia é bem simples, consistindo na noção física de potência elétrica:

$$P = V \cdot I$$

P sendo Potência Elétrica, dado em Watts (W), produto entre Tensão e Corrente.

V sendo Tensão Elétrica, dada em Volts (V), geralmente usaremos 127V ou 220V.

I sendo Corrente Elétrica, dada em Ampères (A), medida pelo sensor.

Assim, a relação entre Potência Elétrica e Energia Elétrica

$$E = P \cdot \Delta t \quad \text{ou} \quad E = V \cdot I \cdot \Delta t$$

E sendo Energia Elétrica, dado em Joules (J), produto entre Potência e Tempo.

Δt sendo tempo, dado em segundos(s), por se tratar do kWh, vamos considerar de 1h (3600s)

Conversão J para kWh

Relação Joule com Watt e Segundo: $1J = 1W \cdot 1s$

Relação de Kilowatt com Watt: $1kW = 10^3W \therefore 1W = 10^{-3}kW$

Relação da Hora com o Segundo: $1h = 3600s \therefore 1s = \frac{1}{3600}h$

Convertendo o 1s para hora e 1W para kilowatt na expressão com Joule:

$$1J = \frac{1}{1000}kW \cdot \frac{1}{3600}h$$

$$1J = \frac{1}{3600000} \text{ kWh}$$

$$1J = \frac{1}{36 \cdot 10^5} \text{ kWh}$$

$$1J = 2,7778 \cdot 10^{-7} \text{ kWh}$$

Na linha de código, então, a expressão consistiria no produto entre Tensão Elétrica, Corrente Elétrica, Tempo e a conversão de Joule para kWh. Apesar disso, apresenta-se ausente a potência de 10^{-7} da conversão para kWh e o Tempo, e suas razões são plausíveis.

- No próprio display, na função constantes, é disponibilizado no display além da grandeza kWh, a potência de 10^{-7} pela notação característico de calculadoras e -7 .
- Por se tratar de uma medição instantânea, foi convencionado por nós a medição a cada 1s que é uma medição lenta em termos de algoritmos, porém razoável para um observador aferir valores. Também esse 1s é justificativa do tempo do comando delay ser de 1000ms.

Assim, a expressão de energia fica: $E = 2,77 \cdot VI$