

Nome: Un Hee Schiefelbein

Curso: Sistemas para Internet - Universidade Federal de Santa Maria – RS

TUTORIAL DE COMO FAZER UM MEDIDOR DE ENERGIA ELÉTRICA, USANDO UM SENSOR DE CORRENTE NÃO INVASIVO SCT013 LIGADO AO ARDUINO

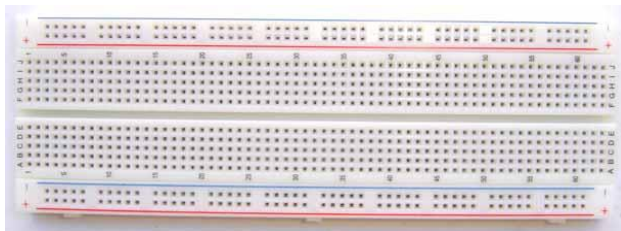
Materiais necessários:

1. Arduino Uno
2. Protoboard
3. Sensor de corrente não invasivo SCT013 - 20A
4. Relê 5v
5. 2 resistores de 10 K
6. 1 resistor de 33k
7. 1 capacitor de 100 uF
8. 9 fios jumpers

1 - Arduino Uno



2 - Protoboard



3 - Sensor de Corrente não invasivo SCT013 - 20 A



4 - Relê 5V



5 - Resistores 10K (Marrom, Preto, Laranja, Dourado)



6 - Resistor 33 k (Laranja, Laranja, Laranja, Dourado)



7 - Capacitor de 100 uF



8 - Fios jumpers

EXECUÇÃO DO CÓDIGO

Ferramentas necessários:

1 - IDE Arduino.

2 - Biblioteca Emonlib ([Link Biblioteca](#))

3 - Biblioteca RTXCom (passo a passo <http://www.devmedia.com.br/utilizando-a-api-rxtx-para-manipulacao-da-serial-parte-i/6722>)

Código no IDE Arduino

Algoritmo para coleta de dados do Sensor SCT- 013-020 com Arduino Uno

```
#define limiarMin 0.1

#include "EmonLib.h" // Adição da biblioteca

EnergyMonitor emon1; // Criar instância

int rede = 210.0; //Tensao da rede elétrica valor fixo

int pino_sct = 5; //Pino do sensor SCT

int sinalparaorele = 7; // pino do rele

int byteEntrada = 0;

//variaveis para tempo

unsigned long tempolnicio = 0;

unsigned long tempoFim = 0;

unsigned long tempoPercorrido = 0;

unsigned long tempoAtual = 0;
```

```

void setup(){

    Serial.begin(9600);

    emon1.current(pino_sct, 26); // Definição de pino de entrada e parâmetro de
    calibração do sensor

    pinMode(sinalparaorele, OUTPUT); //defini o pino como saída

    delay(10); }

void ligaRele() {

    digitalWrite(sinalparaorele, HIGH); //Aciona o rel

    tempoInicio = millis();

    tempoFim = 0; //quando tiver 0, é pq ainda nao acabou }

void desligaRele() {

    digitalWrite(sinalparaorele, LOW); //desliga o rele

    tempoFim = millis(); }

double potencia = 0;

void loop(){

    if (Serial.available() > 0) { //espera valor vir da serial para chamar as funções liga e
    desliga

        byteEntrada = Serial.read();

        if (byteEntrada == '1') {

            desligaRele();  }

        if (byteEntrada == '2') {

            ligaRele(); }  }

        //Calcula a corrente

        double Irms = 0;

```

```

int p = 0;

int AMOSTRAGEM = 20;

// Filtro de primeira ordem

for (p = 0; p < AMOSTRAGEM; p++) {

    Irms += emon1.calcIrms(1480);

    delay(10); }

Irms /= AMOSTRAGEM;

if (Irms < limiarMin) { //se limiar

    Irms = 0; }

double potenciaAtual = Irms * rede; //cálculo da potencia

if (potencia == 0) {    potencia = potenciaAtual; }

else {    potencia = potenciaAtual; }

if (tempoInicio == 0) {    tempoPercorrido = 0; }

else if (tempoFim == 0 ) {    tempoPercorrido = (millis() - tempoInicio) / 1000;
//transforma em s }

else {    tempoPercorrido = (tempoFim - tempoInicio) / 1000; //transforma em s }

double reais = (potencia / 1000) * ((float) tempoPercorrido / 3600) * 0.55;

// Imprime em CSV

Serial.print(potenciaAtual);//0

Serial.print(",");

Serial.println(tempoInicio);//1

Serial.print(",");

Serial.println(tempoFim);//2

Serial.print(",");

```

```
Serial.println(millis()); //3  
delay(300); }
```