Universidade Presbiteriana Mackenzie

Microprocessadores
Medidor de energia Elétrica

Victor Mitsuo Asato TIA: 41311574
Matheus Modesto Oliveira TIA: 31458841
Rafael Rocha da Silva TIA: 41133986
Henrique Kenzo Kuroki TIA: 31346707
Tasso Nayade Santos TIA: 41183703

Professor(a): Ivair Reis Neves Abreu
Turma: Turma 6V

São Paulo
2018

Projeto

Medidor de energia elétrica com arduino

É usado para medir o consumo de energia elétrica em tempo real de uma casa, apartamento, empresa ou qualquer equipamento. O consumo é exibido em Watts e a corrente em Amperes, no monitor serial e em uma página na internet.

Arduino UNO R3

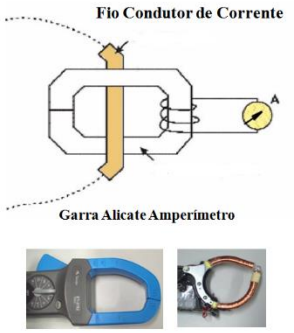
O Arduino Uno é uma placa de microcontrolador baseado no ATmega328. Ele tem 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP e um botão de reset. Ele contém todos os componentes necessários para suportar o microcontrolador, simplesmente conecte a um computador pela porta USB ou alimentar com uma fonte ou com uma bateria e tudo pronto para começar.

Especificações:

Microcontrolador	ATmega328
Tensão de operação	5V
Tensão de alimentação (recomendada)	7-12V
Tensão de alimentação (limite)	6-20V
Entradas e saídas digitais	14 das quais 6 podem ser PWM
Entradas analógicas	6
Corrente contínua por pino de I/O	40 mA
Corrente contínua para o pino 3.3V	50 mA
Memória Flash	32 KB (ATmega328)
Memória SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade do Clock	16 MHz
Dimensões	68,58mm x 53,34mm
Peso	150g

Sensor de corrente

O princípio de funcionamento do sensor é similar ao do alicate amperímetro que é baseado na interação eletromagnética, similar ao transformador, que possui dois enrolamentos (primário e secundário) e através do número de espiras e da relação entre os números de espiras, transforma uma corrente (elevando-a ou diminuindo-a). Fazendo uma analogia entre o transformador e o alicate amperímetro, podemos dizer que um condutor que possui certa corrente em circulação, seria o primário, e o alicate (que envolve o condutor) seria o secundário. Portanto, a corrente no condutor terá uma corrente muito menor no alicate, e desta forma, através da corrente induzida, o alicate converterá essa corrente e informará qual a corrente que percorre o condutor.



O sensor de corrente SCT-013-020 faz parte de uma família de sensores da Yhdc, e suporta correntes de até 20A. Dois fios saem do sensor e estão ligados à um plug P2, que fornece um sinal entre 0 e 1V na saída para o microcontrolador. O SCT-013-020 é um tipo de sensor considerado “não invasivo”. Isso quer dizer que, para medir a corrente, não precisamos efetuar nenhuma alteração no circuito que estamos medindo. Basta abrir o sensor, envolver o fio e realizar a medição.

Especificações:

– Modelo: SCT-013-020
– Corrente de entrada: 0-20A
– Sinal de saída: Tensão/1V
– Material do Core: Ferrite
– Dielétrico: 6000V AC/1min
– Taxa anti-chama: UL94-V0
– Plug de saída: 3,5mm
– Dimensão abertura: 13 x 13mm
– Temperatura de trabalho: -25 a +70°C
– Comprimento do cabo: 150cm

Shield ETHERNET

A interface no browser é feita com o shield Ethernet W5100, este shield possibilita a comunicação do Arduino com sua rede local e pode ser conectado a internet permitindo o monitoramento de qualquer lugar do mundo. O shield fornece um endereço IP baseado nos protocolos TCP e UDP.

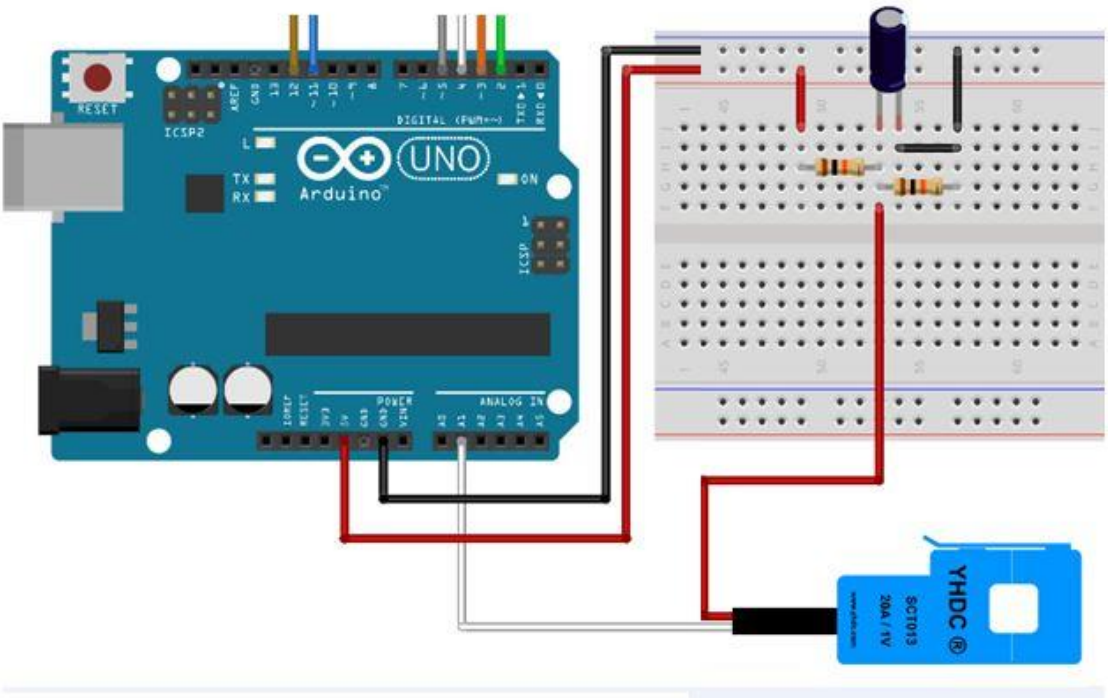
Especificações:

- Controlador Ethernet: W5500
  - Múltiplos protocolos TCP/IP integrados (TCP, UDP, IPv4, entre outros)
  - Suporta até 8 soquetes independentes simultaneamente
  - Buffer interno de 32 KB para comunicação (Rx/Tx)
- Tensão de alimentação: 5 V
- Tensão de operação: 3,3 V
- Frequência de operação: 25 MHz
- Conector para cartão de memória Micro SD (com detecção de cartão via código)
- Jumpers de solda:
  - Detecção de cartão de memória inserido: pino 2 *(pull-up para IOREF)*
  - Interrupção INTn do W5500: pino 3 *(pull-up para 3,3 V)*
- LEDs indicativos:
  - ON: indica que a placa e o shield estão ligados
  - LINK: indica a presença de um link de rede e pisca quando o shield transmite ou recebe dados
  - FDX: indica que a conexão à rede é *full duplex*
  - 100M: indica a presença de uma rede 100 Mb/s
  - ACT: pisca quando existe atividade de Tx e Rx
- Pinos utilizados:
  - No Arduino UNO R3: pinos digitais 4, 10, 11, 12 e 13
  - No Arduino Mega 2560 R3: pinos digitais 4, 10, 50, 51 e 52
- Sobre a utilização do W5500 e Micro SD:
  - Pelo W5500 e o MicroSD compartilharem o bus SPI, só um pode ser ativado por vez. Se você está usando ambos os periféricos em seu programa, você deve tomar cuidado com isso ao usar suas bibliotecas. Se você não está usando um dos periféricos em seu programa, contudo, você vai precisar explicitamente deselecionar isto. Para fazer isto com o MicroSD, coloque o pino 4 como uma saída e escreva um HIGH nele. Para fazer isto com o W5500, coloque o pino digital 10 como saída HIGH.

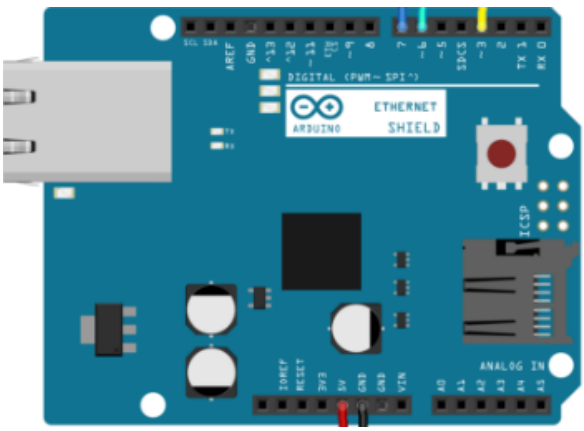
Material do projeto:

Placa Uno R3
Cabo USB
Fonte 9V 1A Plug P4
Sensor de Corrente Não Invasivo SCT-013 20A
Protoboard 830 pontos
Resistor 10K 1/4W
Capacitor 100µF 16V
Jumpers Macho-Macho
Fio Paralelo 1,0mm 1m
Plug Tomada Macho
Plug Tomada Fêmea
Ethernet Shield W5100 para Arduino
Conector de Áudio Jack 3,5mm

Ligação do esquema elétrico:



Mais a shield ethernet conectada em cima do arduino



Funcionamento do projeto:

Precisamos montar um circuito para calibrar o sensor de corrente, conectamos a ethernet shield em cima da placa arduino uno R3, colocamos o sensor em apenas uma das fases do equipamento que vamos usar para medição, usando um cabo de rede conectamos a ethernet na rede.

Após a montagem partimos para a programação, que começa fazendo a declaração das bibliotecas usadas, “EmonLib.h”, “SPI.h” e “Ethernet.h”, configuração do servidor usado e da pagina html que será exibido as informações do projeto. Vamos usar também para exibir os resultados a comunicação serial do arduino.

Para fazer a leitura do dado do sensor de corrente é necessário fazer uma conta de calibração descrito pelo fabricante 1800/62=29.apos a leitura da corrente é feito o calculo da potência através de cálculos matemáticos, P=U*I, onde o valor da tensão e uma constante que no caso é 110.

Será exibido os valores da corrente e da potência em uma pagina na internet em tempo real com o tempo de atualização de 5 segundos, e exibido também no monitor serial do arduino.

Código

```
//Programa : Medidor de energia elétrica com Arduino e SCT-013
//Baseado no programa exemplo da biblioteca EmonLib

//Carrega as bibliotecas
#include "EmonLib.h"
#include "SPI.h"
#include "Ethernet.h"

EnergyMonitor emon1;

EthernetClient client;
//Tensao da rede eletrica
int rede = 110.0;

//Pino do sensor SCT
int pino_sct = A1;
char server [] = "microhmrtv.ddns.net"; // meu servidor
String apikey = "c9ec04763e8973fa4c3ee2d754f89c26"; //API key
disponibilizada pelo emoncms
int node = 0;
byte mac[] = { 0xA4, 0x28, 0x72, 0xCA, 0x55, 0x2F }; //Endereço
MAC da placa ethernet na rede

void setup()
{
  Serial.begin(9600);
  pinMode(pino_sct,INPUT);
  if (Ethernet.begin(mac) == 0){
    Serial.println("Shield ethernet não inicializado.");
    while(1);// abortar (não temos rede)
  } else {
```

```
            Serial.println("Projeto Final de Microprocessadores -
Medidor de energia elétrica com arduino.");
            Serial.println("Henrique Kenzo Kuroki");
            Serial.println("Matheus Modesto Oliveira");
            Serial.println("Rafael Rocha da Silva");
            Serial.println("Tasso Nayade da Rosa Santos");
            Serial.println("Victor Mitsuo Asato");
            Serial.println();
            Serial.println("Shield ethernet inicializado com sucesso!");
        }
        delay(1000);
        Serial.print("Endereço IP: ");
        Serial.println(Ethernet.localIP()); // mostra o endereço IP


        //Pino, calibracao - Cur Const= Ratio/BurdenR. 1800/62 = 29.
        emon1.current(pino_sct, 29);

      }

      void loop()
      {
         client.connect (server,8082);
         if (client.available()) {
          char c = client.read();
          Serial.write(c);
        }
        client.stop();
        //Calcula a corrente
        double Irms = emon1.calcIrms(1480);
        double Potencia = Irms*rede;
        //Mostra o valor da corrente
        Serial.print("Corrente : ");
        Serial.print(Irms); // Irms
        Serial.print(" A ");


        //Calcula e mostra o valor da potencia
        Serial.print(" Potencia : ");
        Serial.print(Potencia);
        Serial.println(" W ");

        sendData (Irms, Potencia);

        delay (3000);
      }

      void sendData (double Irms, double Potencia){
        client.connect (server,8082); //conecta ao servidor
        delay (500);
        if (client.connected()){
          //Envia os dados como parametros através do metodo GET
          client.print("GET
/emoncms/input/post?node=medicoes&fulljson=");
          client.print("{\"Corrente\":");
          client.print(Irms);
          client.print(",\"Potencia\":");
          client.print(Potencia);client.print("}");
          client.print("&apikey=");client.print(apikey);
```

```
  client.println(" HTTP/1.1");
  client.println("Host: emoncms.org");
  client.println("User-Agent: Arduino-ethernet");
  client.println("Connection: close");
  client.println();

  Serial.println("Dados enviados!!");
  client.stop();
  Serial.println();
  }
    else{Serial.println("Não conectado!");client.stop();}
}
```

Descrição das bibliotecas

Biblioteca <SPI.h>: Biblioteca para a comunicação entre o Ethernet Shield e a placa do Arduino**.**

Biblioteca <Ethernet.h>: Biblioteca responsável por conectar o Arduino na internet, a biblioteca permite a placa tanto enviar quanto receber informações, ela suporte até 4 conexões (entradas, saídas ou combinações).

Nós usamos principalmente 3 comandos:

- Ethernet.begin(mac, ip, gateway, subnet)): para inicializar o dispositivo ethernet

- Client.connect(ip, port): especifica um endereço IP para se conectar, pode retornar um dos seguintes valores 1,-1,-2,-3,-4 referentes a respectivamente sucesso, time out, servidor invalido, truncado e resposta invalida.

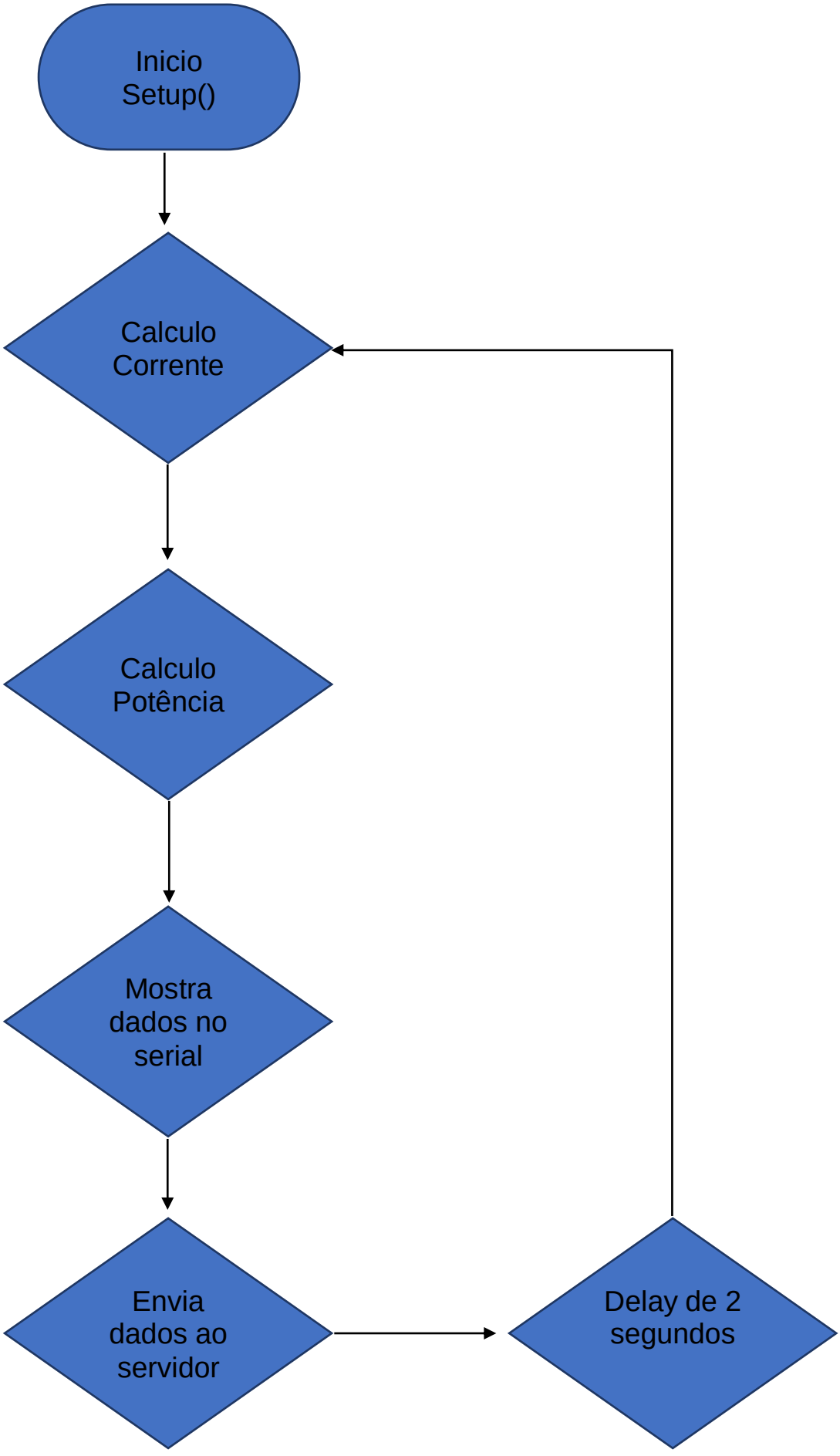
- Client.print(data, BASE): envia dados ao servidor conectado.

Biblioteca <Emonlib.h>: Biblioteca que faz toda a conversão ADC do nosso sinal, por se tratar de uma onda senoidal, retirar o valor de Vrms é feito de forma muito mais fácil com a biblioteca.

- EnergyMonitor.current(pino, calibração): Função que configura o pino onde receberemos o sinal de corrente e sua constante de calibração

- EnergyMonitor.calcIrms(amostras): Função que colhe um determinado número de amostras do sinal para fazer a conversão analógica digital e calcular a raiz do valor quadrático médio, o Irms, do nosso sinal.

Fluxograma


Inicio
Setup()
Calculo
Corrente
Calculo
Potência
Mostra
dados no
serial
Envia
dados ao
servidor
Delay de 2
segundos

Link para vídeo de demonstração do projeto
https://www.youtube.com/watch?v=agCpQ5fYdyc

Link para disponibilização do projeto e código
https://github.com/VictorAsato/Medidor-de-Energia-El-trica-com-Arduino