

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIA
CURSO DE ENGENHARIA ELÉTRICA
ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

MONITORAMENTO DE CONSUMO ELÉTRICO E ARMAZENAMENTO EM
CARTÃO SD

ALECSANDER FELIPE DOS SANTOS DE LIMA
PROFESSORA LUCIA GUIMARÃES

CAMPINAS - SP
NOVEMBRO 2017

1. Objetivo

O objetivo do projeto é monitorar o consumo de aparelhos elétricos domésticos, coletar dados de uso e com isso descobrir quais horários há picos de consumo e elaborar manobras para que seja feito um melhor uso da rede, assim evitando desperdícios e barateando os custos com energia elétrica.

Fomos inspirados pelo projeto OpenEnergyMonitor, o qual possui vários módulos que podem medir consumo elétrico, temperatura e umidade. Os módulos são constituídos de Arduino e Raspberry e uma vasta gama de sensores e atuadores. O uso do Raspberry possibilita o uso de redes *wireless* e uma infinidade de dispositivos conectados ao conjunto e não é necessário o uso de um computador ligado enquanto realiza as leituras. O projeto também usa uma interface mobile, para Android e iOS possibilitando o usuário monitorar em tempo real seu consumo e integridade dos dispositivos conectados ao sistema.



FIGURA 1 - PROJETO QUE NOS INSPIROU

2. Introdução

Usaremos um Arduino Uno e um amperímetro não invasivo (SCT013 000) e alguns resistores e capacitores.



FIGURA 2 - SCT013 000

O SCT013 000 tem como saída corrente. Ou seja, a cada 100A lido, ele entrega 50mA.



FIGURA 3 - ARDUINO UNO

Para criar a interface *hardware software*, usaremos o Arduino e linguagem C. Usaremos as bibliotecas SdFat.h, EmonLib.h, Wire.h e rgb_lcd.h para podermos armazenar os dados lidos no cartão de memória e mostrar no visor do LCD.

3. Projeto

3.1. Interface

O projeto consiste em ler valores com o sensor, mostrar no visor LCD e armazenar em um cartão de memória para depois construir gráficos e verificar picos de consumo.

Para isso, usamos um módulo de cartão SD, um Grove - LCD 16 por 2 e uma placa para conexão entre os periféricos que foi construída pelo grupo.

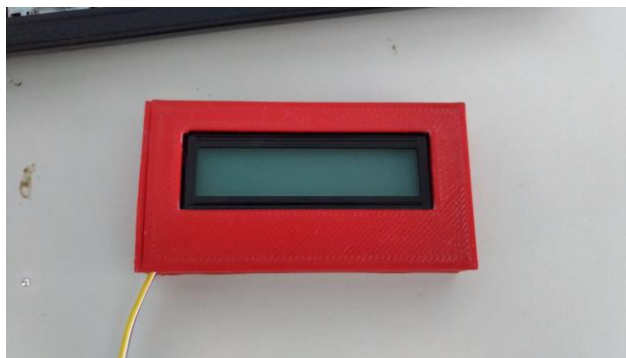


FIGURA 4 - GROVE LCD COM CASE IMPRESSO EM 3D

O Grove LCD trabalha com interface I2C, porém de uma forma bem mais simples do que os displays encontrados no mercado. Ele possui um conector de 4 vias com alimentação e os pinos de dados, facilitando assim a conexão e evitando mal contato.



FIGURA 5 - MÓDULO SD

Optamos por usar o módulo de cartão SD ao invés do microSD pelo preço. Este custava R\$10,00 (Cinestec, Campinas) contra R\$21,50 (Cinestec, Campinas) e possui a mesma funcionalidade. Para utilizar esse módulo, precisamos criar um circuito divisor de tensão, pois o mesmo trabalha com 3,3V e as portas lógicas do Arduino fornecem 5V. Utilizamos alguns resistores de 11k Ω e uma placa de fenolite para a construção do circuito.

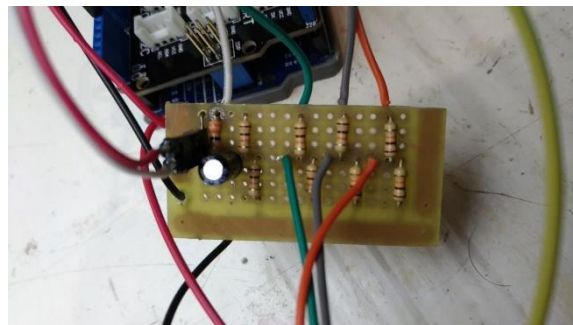


FIGURA 6 - PLACA COM CIRCUITO DIVISOR DE TENSÃO

Aproveitamos o espaço e criamos o circuito com o resistor de carga (Burden Resistor) e o capacitor. Esse circuito é necessário para podermos converter a corrente fornecida pelo sensor em dados que o Arduino compreenda. Optamos por cortar o conector p2 do SCT013 para facilitar na conexão com a placa confeccionada por nós, pois não tínhamos em mãos um conector p2 fêmea.

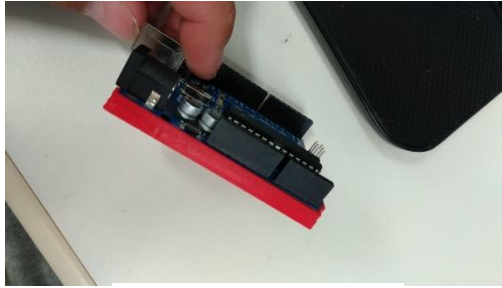


FIGURA 7 - BASE ARDUINO

Utilizamos uma impressora 3D para fabricar alguns acessórios: case do LCD e a base para o Arduino para que evitasse problemas com curto circuito com objetos metálicos que pudessem estar presentes na bancada de testes.

1. Código

Vamos ser sucintos ao explicar os códigos, mostrando apenas o necessário do código. As bibliotecas SdFat, EmonLib e do LCD necessitam que criemos instancias antes do setup e das variáveis globais:

```
EnergyMonitor emon1;  
SdFat sdCard;  
SdFile consumo;  
rgb_lcd lcd;  
  
int rede = 127; //Tensao na rede eletrica  
double sensor = A0; //Sensor esta; ligado a porta A0  
double potencia; //Variavel da potencia  
const int cartao_sd = 4; //Pino em que sera ligado o módulo SD card  
int i = 1; //Contador
```

Parindo disso, vamos para o loop, onde fica a lógica em si do projeto.

O código pode ser melhor entendido através do fluxograma a seguir:

```
emon1.current(sensor, 60); //variável e calibração

double Irms = emon1.calcIrms(1480); //1480 e o valor para Irms

potencia = Irms * rede; //Calculo de potencia

Serial.println(potencia); //Imprime no serial print
consumo.println(potencia); //Imprime no cartao SD
lcd.print(i); //Imprime o contador no LCD
lcd.print(" - ");
lcd.print(potencia ); //Imprime a potencia no LCD
lcd.print("W");

delay (1000); //Intervalo de leituras de 1 segundo

i++;

lcd.clear(); //Limpa o LCD

if (i == leituras)
{
    Serial.println("Processo de gravacao interrompido. Retire o
SD!");
    lcd.print("Gravado, retire SD");
    consumo.close();
}
```

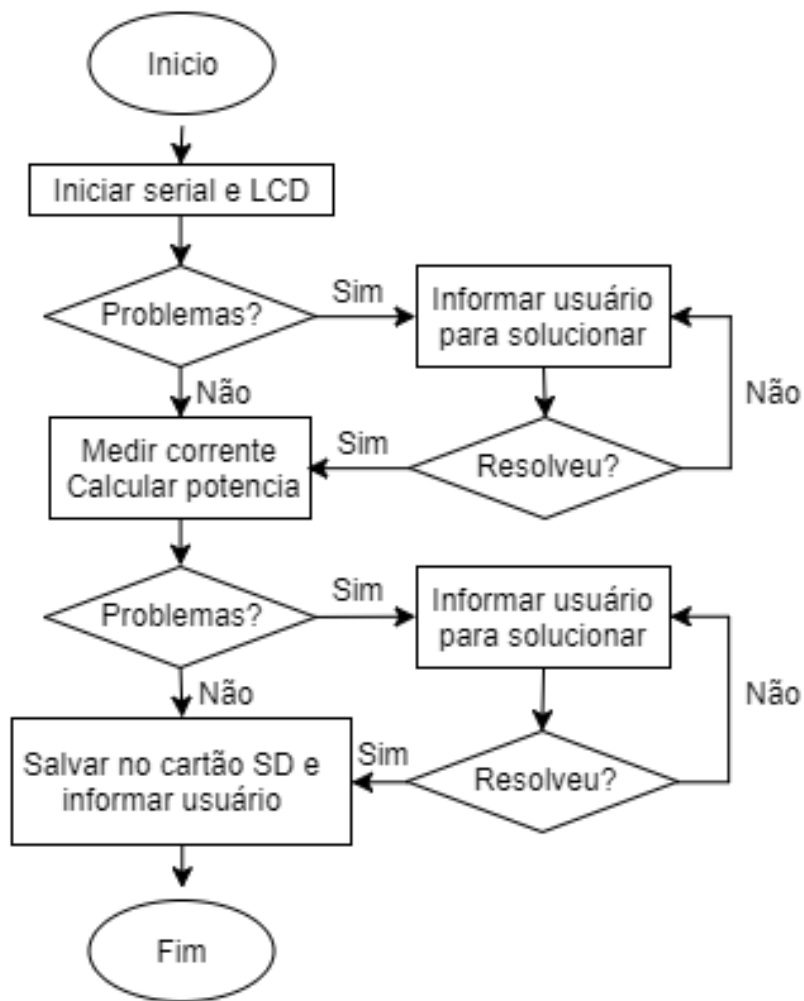


FIGURA 8 - FLUXOGRAMA DA LÓGICA

3.2. Teste

Realizamos alguns testes com nosso sistema. Medimos o consumo de uma impressora 3D, modelo AIP A3 da Sethi3D. No site do fabricante consta que o consumo máximo da impressora é de 350W. Para confirmar isso, instalamos o sensor em um de seus cabos de alimentação e realizamos leituras com intervalo de 1 segundo por 10 minutos. No total tivemos 600 leituras e usamos esses dados para construir o gráfico a seguir:

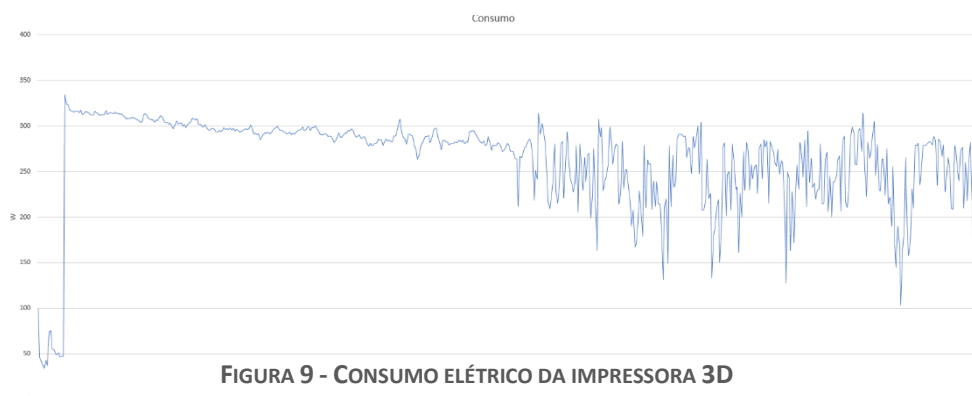


FIGURA 9 - CONSUMO ELÉTRICO DA IMPRESSORA 3D

O gráfico condiz com a realidade. A impressora ao ligar, aciona todos os motores e dispositivos aquecedores, levando a um pico de consumo (entorno de 100 W). Porém, a curva de maior consumo foi quando ligamos sua mesa aquecida, dispositivo que mais consome corrente do equipamento. Tendo um pico de quase 350 W que foi caindo até a mesa alcançar 110 °C. A partir do centro do gráfico podemos notar que o consumo perdeu sua linearidade e passou a ter picos. Isso se deve pelo motivo da impressora trabalhar com PID para manter a mesa aquecida, ou seja, ela modula a tensão e consequentemente a corrente com que alimenta a mesa, fazendo ter picos altos e baixos em um intervalo de tempo.

4. Orçamento

O grupo teve problemas para adquirir alguns produtos por conta da disponibilidade no varejo e também por falta de recursos financeiros. Compramos na Cinestec (Campinas) o módulo SD, o SCT013-000 e alguns jumpers. O Grove – LCD e o Grove – Adapter é da Universidade, e foi concedido para somente uso interno, o que dificultou o desenvolvimento do projeto.

Item	Preço
SCT013	R\$ 55,00
Módulo SD	R\$ 10,00
Jumpers	R\$ 4,80
Total	R\$ 69,80

5. Bibliografia

CT sensors - An Introduction. (s.d.). Fonte: OpenEnergyMonitor:

<https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/introduction>

CT Sensors - Interfacing with an Arduino. (s.d.). Fonte: OpenEnergyMonitor:

<https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino>

EmonLib - OpenEnergyMonitor. (18 de outubro de 2016). Acesso em 18 de setembro de 2017, disponível em GitHub: <https://github.com/openenergymonitor/EmonLib>

Home Energy. (s.d.). Acesso em 18 de setembro de 2017, disponível em OpenEnergyMonitor:

<https://guide.openenergymonitor.org/applications/home-energy/>

Install. (s.d.). Acesso em 18 de setembro de 2017, disponível em OpenEnergyMonitor:

<https://guide.openenergymonitor.org/setup/install/>

Library SD. (s.d.). Fonte: Arduino: <https://www.arduino.cc/en/Reference/SD>

System Overview. (s.d.). Acesso em 18 de setembro de 2017, disponível em OpenEnergyMonitor:
<https://guide.openenergymonitor.org/setup/>

Thomsen, A. (11 de fevereiro de 2015). *Como fazer um medidor de energia elétrica com arduino.*
Acesso em 18 de setembro de 2017, disponível em FilipeFlop:
<https://www.filipeflop.com/blog/medidor-de-energia-eletrica-com-arduino/>

Thomsen, A. (13 de julho de 2015). *Como gravar dados no Cartão SD com Arduino.* Fonte:
Filipeflop: <https://www.filipeflop.com/blog/cartao-sd-com-arduino/>

Using the SD library to log data. (s.d.). Fonte: Arduino:
<https://www.arduino.cc/en/Tutorial/Datalogger>

Valgolio, R. (s.d.). *Arduino Excel - Roberto Valgolio.* Acesso em 18 de setembro de 2017, disponível
em <http://www.robertovalgolio.com/sistemi-programmi/arduino-excel>