

Protótipo SmartPlug (A sua tomada inteligente!)

Pedro Aleph Gomes de Souza Vasconcelos

Departamento de Ciência da Computação da Universidade Federal de Roraima (UFRR)
– Boa Vista – RR – Brasil

pedro.aleph.allen@gmail.com

Resumo. *Um sistema de tomada para ajudar o usuário a gerenciar o consumo de energia na tomada. Utilizando principalmente arduino uno, resistor de 220 ohms e cabos diversão, todos posicionados estrategicamente para simular a corrente e tensão que estariam circulando na tomada. os teste feitos consistiam no uso da tomada em diferentes componentes, obtendo diferentes valores de corrente e tensão em cada teste.*

1. Introdução

Este projeto consiste em um sistema de tomada inteligente para ajudar no gerenciamento de consumo de energia. Para simular utilizou-se a plataforma TinkerCad para testar os componentes de tomada e o framework React JS, para desenvolvimento da interface de gerenciamento da tomada para o usuário.

2. Descrição do Projeto

O gerenciamento da tomada consiste na coleta de consumo constante da tomada, usando a corrente e tensão passadas no circuito da tomada, esses dados são salvos num servidor e então são plotados em gráficos de uma das variáveis de interesse(corrente e tensão, uma por vez) em função do tempo(dado um intervalo).

Principais componentes utilizados:

- 1x microcontrolador Arduino Uno R3
- 1x 220 ohms Resistor
- cabos diversos

Componentes de Auxílio:

- 1x amperagem Multímetro
- 1x tensão Multímetro
- 1x Placa de ensaio pequena

Componentes não utilizados por limitações da plataforma:

- 1x Esp8266

Componentes para teste:

Teste 1:

- 1x vermelho Led
- 1x 220 ohms Potenciômetro

Teste 2:

- 1x Lâmpada
- 1x Interruptor deslizante

Big Picture



Figura 1

Storyboard

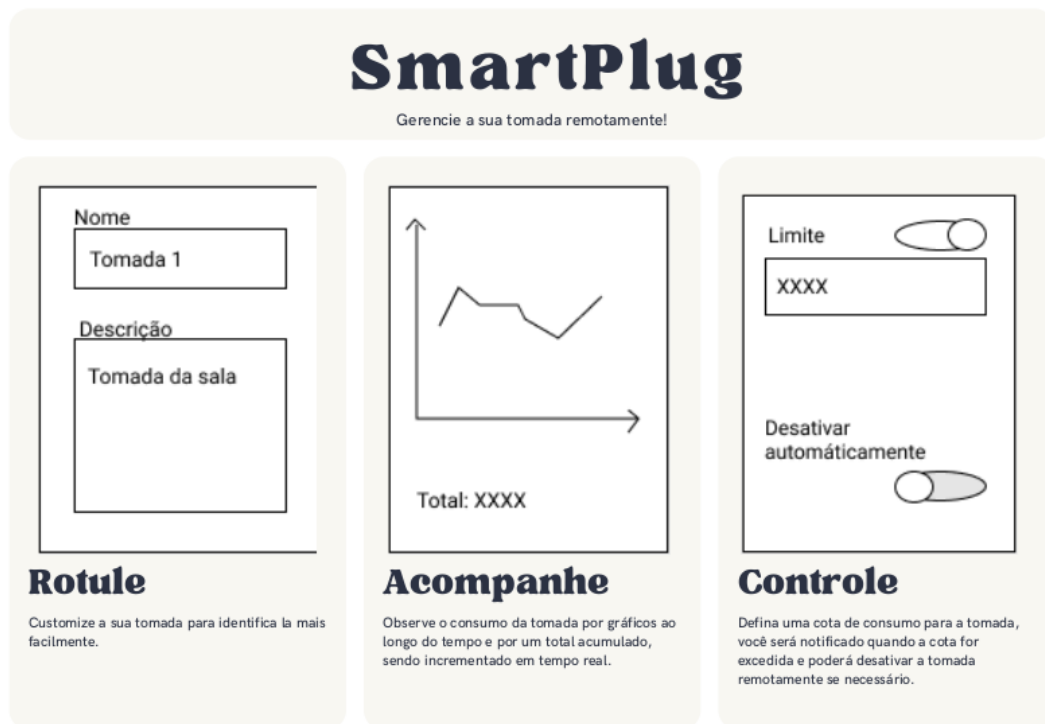


Figura 2

2.1. Modelagem do Sistema Proposto

A seguir apresenta se o modelo de sistema da tomada:

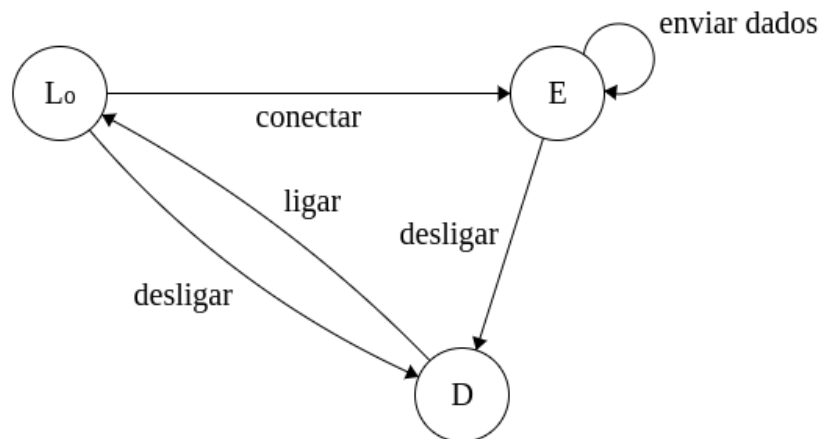


Figura 3

Onde:

- Lo, estado inicial
- E, estado da tomada ligada e enviado dados
- D, tomada desligada

2.2. Esquema de Conexões

Pinagem dos componentes principais:

GND: Pino GND do arduino(polo negativo)

VCC: Pino 5V do arduino(polo positivo)

- Resistor de 220 ohms encontra-se no polo negativo.
- Pino A0 utilizado para leitura da corrente, conectado no polo negativo logo a frente do resistor.
- Pino A1 utilizado para leitura da tensão, conectado no polo positivo antes dos componentes de teste.

Teste 1:

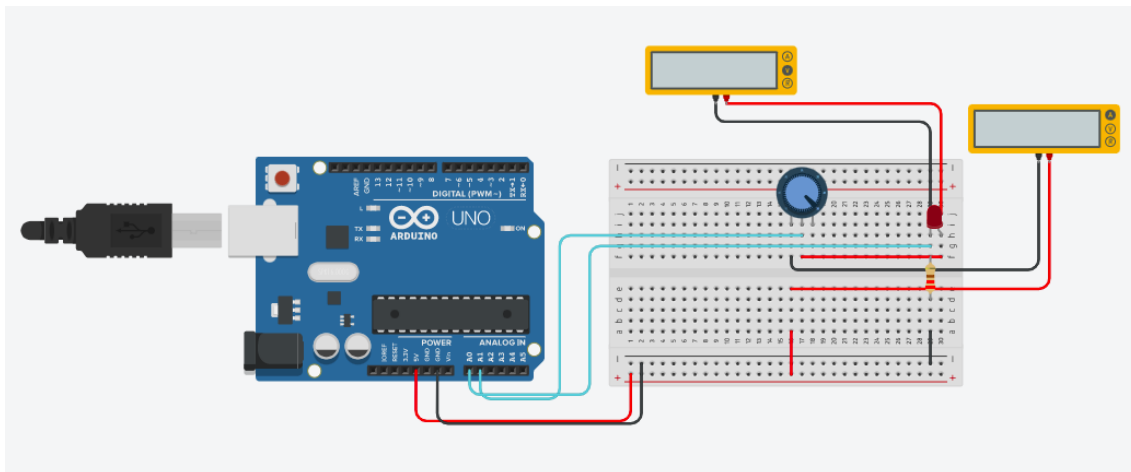


Figura 4

- Multímetro de amperagem conectado com seu lado positivo no polo positivo.
- Potenciômetro conectado com terminal 1 no lado negativo do multímetro de tensão.
- Limpador do potenciômetro conectado com o anodo do led.
- Catódica do led conectada no polo negativo antes do resistor de 220 ohms.
- Multímetro de tensão conectado em paralelo com o led, lado positivo no anodo e lado negativo no catódica, respectivamente.

OBS: Neste caso o potenciômetro é somente usado para obter uma variação na voltagem do circuito, então o pino para leitura da tensão(A1) é conectado no limpador do potenciômetro.

Teste 2:

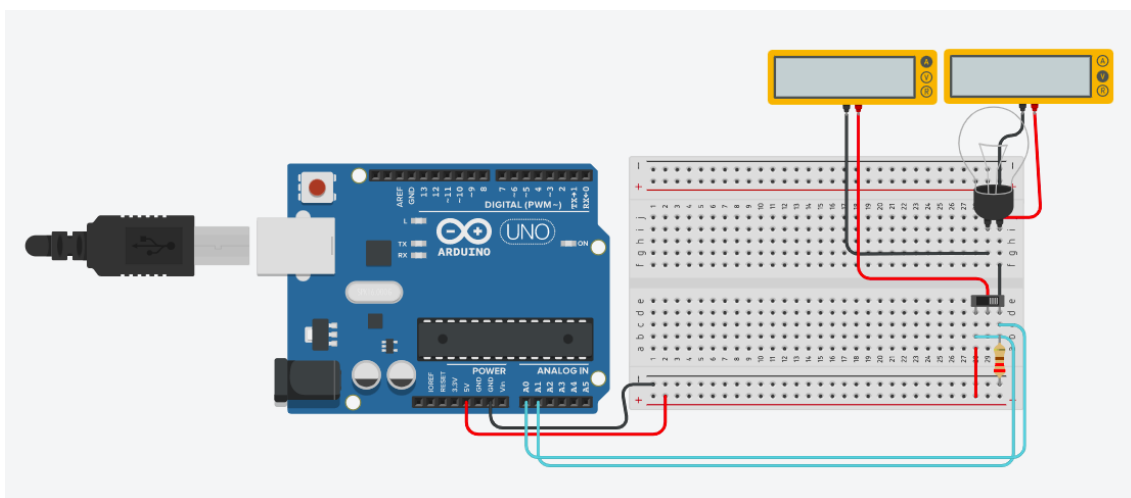


Figura 5

- Terminal 1 do interruptor deslizante conectado no polo positivo.
- Multímetro de amperagem, lado positivo conectado no comum do interruptor deslizante, lado negativo conectado no terminal 1 da lâmpada.
- Multímetro de tensão conectado em paralelo com a lâmpada, lado positivo no terminal 1 e lado negativo no terminal 2, respectivamente.
- Terminal 2 da lâmpada conectado ao terminal 2 do interruptor deslizante.
- Terminal 2 do interruptor deslizante conectado no polo negativo antes do resistor de 220 ohms.

3. Testes e Avaliação Experimental

Código no arduino:

```
const int corrente = A0;
const int tensao = A1;
const int resistencia = 220; // em ohms
float I = 0;
float V = 0;
int intervalo = 1000; // em milissegundos

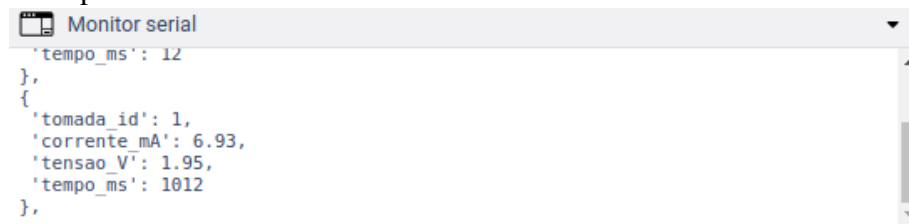
void setup() {
  Serial.begin(9600);
}

void loop() {
  I = analogRead(corrente);
  V = analogRead(tensao) - I;
  if(millis() % intervalo == 0)
    print_json();
}

void print_json() {
  Serial.println("{");
  Serial.println("  'tomada_id': 1,");
  Serial.print("  'corrente_mA': ");
  Serial.print(I * 5.0 * 1000 / (1023.0 * 220));
  Serial.println(",");
  Serial.print("  'tensao_V': ");
  Serial.print(V * 5.0 / 1023.0);
  Serial.println(",");
  Serial.print("  'tempo_ms': ");
  Serial.println(millis());
  Serial.println("},");
}
```

- Após a leitura da corrente e tensão nos pinos A0 e A1, é imprimido os dados em formato json no intervalo definido (a cada 1 segundo neste caso),.
- Além dos valores de corrente e potência, também são enviados, identificação da tomada e o tempo em milissegundos do instante em que estes dados são enviados.

Exemplo de saída:



```

{
  'tempo_ms': 12
},
{
  'tomada_id': 1,
  'corrente_mA': 6.93,
  'tensao_V': 1.95,
  'tempo_ms': 1012
},

```

Figura 6

Teste 1:

Usando o potenciômetro, pode-se observar a variação da corrente e tensão no led. Aumentado no potenciômetro, decrementa se os valores, e incrementam caso contrário, isso se deve pelo valores dividirem a carga total do circuito(5V dado pelo arduino) com o potenciômetro e sua resistência interna.

Teste 2:

Usando um interruptor deslizante, assim ligando e desligando a lâmpada, observa-se a mudança da corrente e tensão no circuito. Quando a lâmpada está ligada, a carga se distribui livremente no circuito(corrente e tensão maiores que 0); quando a lâmpada está desligada, o interruptor impede a carga de se distribuir, então esta fica toda concentrada na tensão (corrente é 0 e tensão é 5V). Os valores se mantêm constantes em cada caso.

4. Considerações Finais

O sistema propostos deve limitações na plataforma de teste, o sensor esp8266 seria utilizado para a conexão entre a tomada e a interface do usuário, essa limitação implica em:

- Como não foi possível se comunicar com o servidor e enviar os dados, foi colocado um simples print no monitor serial.
- Como não foi possível salvar os dados, usou-se dados estáticos para plotar os gráficos na interface para o usuário.
- A interface de usuário não apresenta a funcionalidade de desligar a tomada, por não ser possível de comunicar com a tomada.

A interface gráfica também limita-se em apenas plotar os dados em gráficos num intervalo de 60 minutos, e uma estimativa da potência elétrica com a soma das potências das tomadas. Essas limitações dependem diretamente das limitações da tomada, no ambiente de mundo real, essas limitações poderiam ser facilmente resolvidas, tendo em vista a possibilidade de usar componentes específicos para medição do consumo de energia e utilização de um sensor para tornar possível a comunicação com a interface.

Mesmo com essas limitações, foi bom desenvolver o protótipo para assim dar início ao que é necessário para que este sistema entre em prática, caso um dia tenha continuidade de desenvolvimento. E apesar da medição para obtenção dos dados se feita de forma simples, foi bem complicado entender, vale ressaltar que apesar das explicações de como fazer para funcionar minimamente, ainda foi explorado outras formas de medição para assim poder colocar mais casos de teste, mas essas explorações falharam e assim tive que deixar funcionando minimamente mesmo.

5. Referências

Plataforma de desenvolvimento da tomada: <https://www.tinkercad.com>

Criação de Big Picture e StoryBoard: <https://www.canva.com>

Materiais Complementares:

- medindo tensão e corrente simultaneamente: <https://youtu.be/Sze07xpcpoM>
- circuito simples com lâmpada, voltímetro e amperímetro: <https://youtu.be/0L-OKBZ08RA>

Cálculo da potência elétrica: <https://www.todamateria.com.br/potencia-eletrica/>

Teste 1:

- https://www.tinkercad.com/things/3fTpXb39wdR-medidordeconsumo/editel?sharecode=wRxLanaj81-V_NFehpDXIRJqDHU98arLACHwIU5bISA

Teste 2:

- <https://www.tinkercad.com/things/7XUv99MWTuh-daring-jaagub-crifft/editel?sharecode=4vu0RZaIKCYvJm4eZEK9iVSmNxbuTjPH7w7BAiuXgHw>

Interface de usuário: <https://ise-pedroaleph-2022.netlify.app>

Repositório de código fonte:

- https://github.com/pedroaleph/ISE_PedroAleph_UFRR_2022.git