

Medidor de Consumo de Energia Elétrica Conectado à Nuvem com Arduino

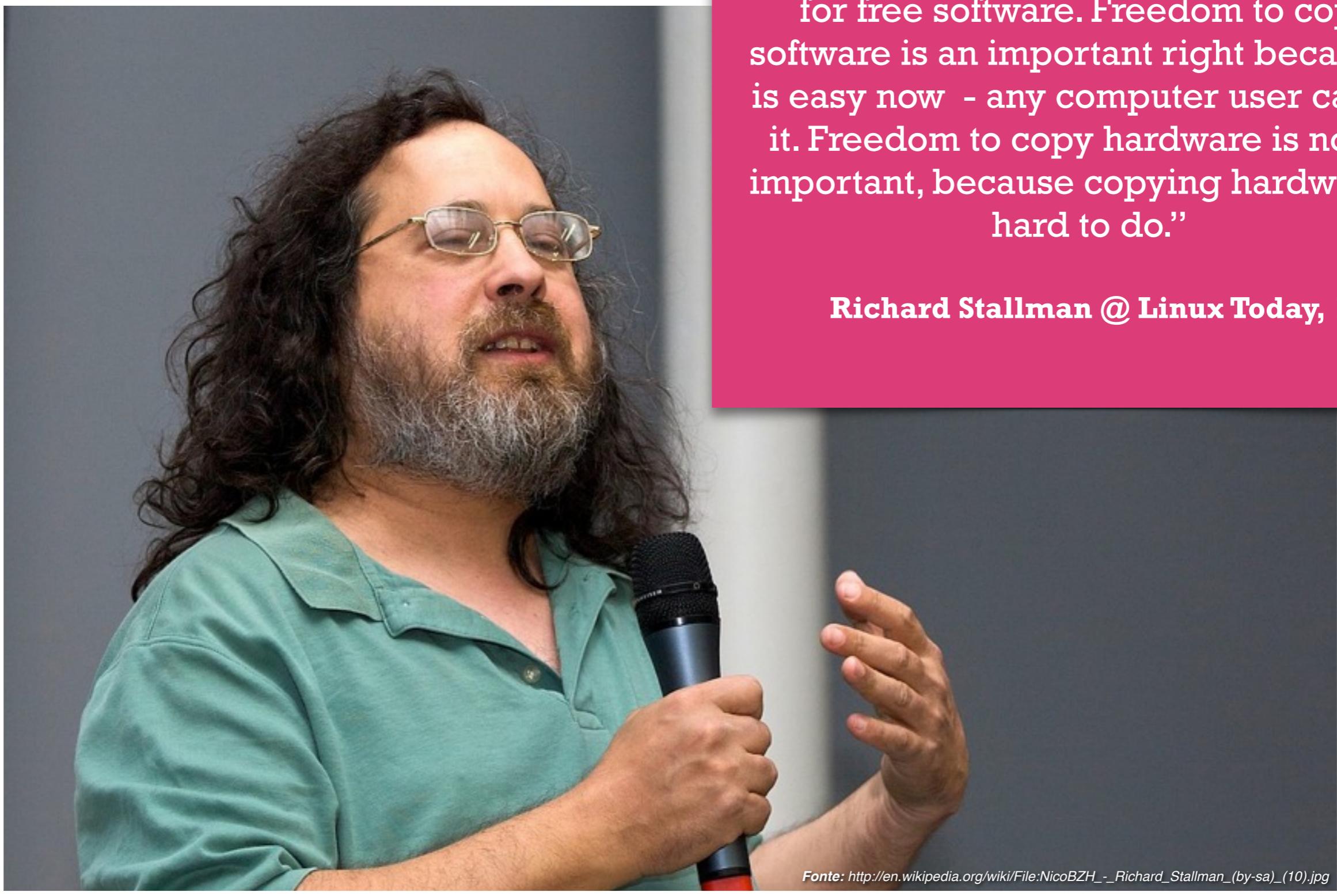
Manoel Lemos
manoel@lemos.net
[@mlemos](http://fazedores.com)
<http://fazedores.com>
<http://manoellemos.com>



Muito cuidado ao trabalhar com a rede elétrica!



Não arrisque sua vida!

A photograph of Richard Stallman, a man with long dark hair and a beard, wearing glasses and a green shirt, speaking into a microphone and gesturing with his hands.

"I see no social imperative for free hardware designs like the imperative for free software. Freedom to copy software is an important right because it is easy now - any computer user can do it. Freedom to copy hardware is not as important, because copying hardware is hard to do."

Richard Stallman @ Linux Today, 1999

Fonte: [http://en.wikipedia.org/wiki/File:NicoBZH_-_Richard_Stallman_\(by-sa\)_10.jpg](http://en.wikipedia.org/wiki/File:NicoBZH_-_Richard_Stallman_(by-sa)_10.jpg)

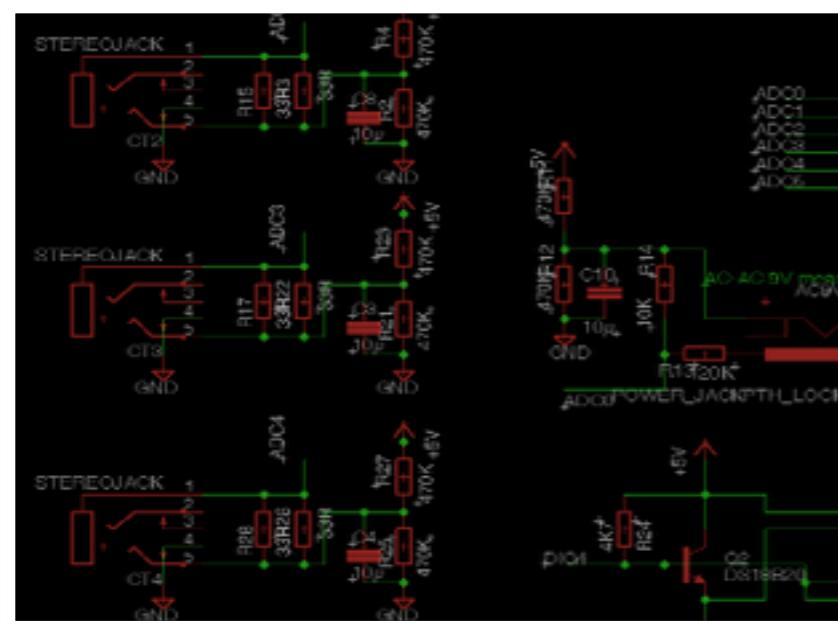
Hardware é Phodda!



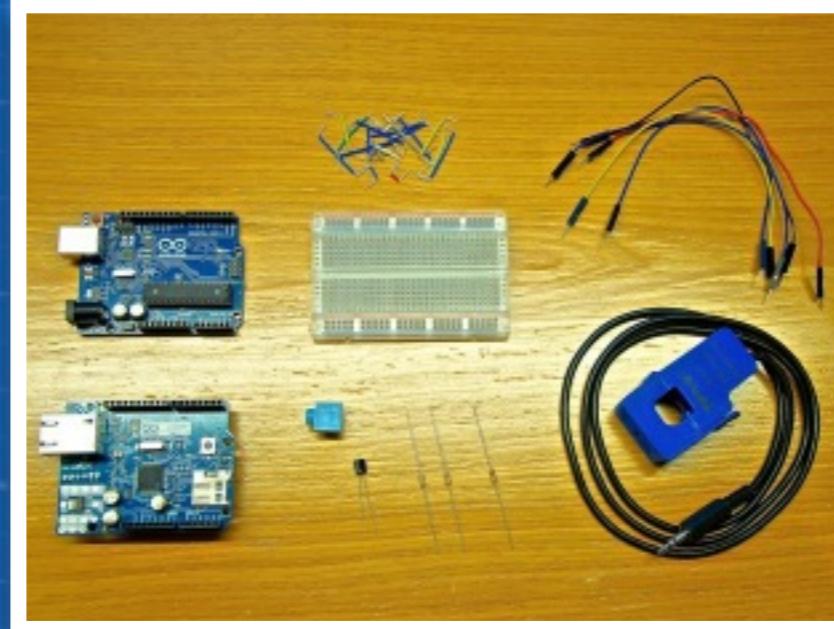
Fonte: [http://en.wikipedia.org/wiki/File:NicoBZH_-_Richard_Stallman_\(by-sa\)_\(10\).jpg](http://en.wikipedia.org/wiki/File:NicoBZH_-_Richard_Stallman_(by-sa)_(10).jpg)

Hardware é Phodda..

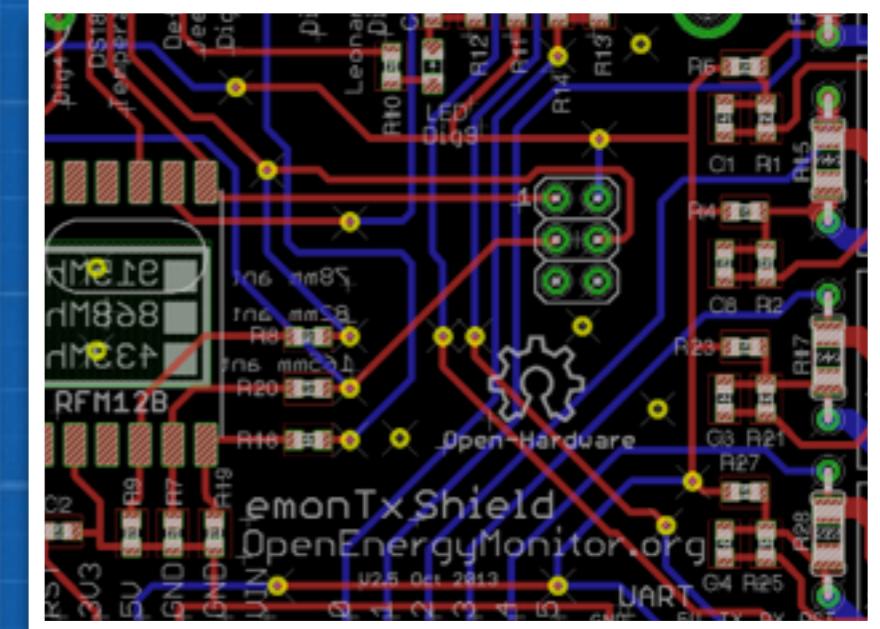
1. Diagrama Elétrico



2. Peças



3. Layout da Placa



4. Firmware

```
:10F0A0000C9492F80C9492F80C9492F80C9492F8B8
:10F0B0000C9492F80C9492F80C9492F80C9492F8A8
:10F0C0000C9492F80C9492F80C9492F80C9492F898
:10F0D0000C9492F80C9492F80C9492F80C9492F888
:10F0E0000C9492F811241FBECFEFD1E2DEBFCD8F4A
:10F0F00012E0A0E0B2E0EEEDFEEF01E00BBF02C0D7
:10F1000007900D92A833B107D9F71BBE13E0A8E30F
:10F11000B2E001C01D92A334B107E1F70E9412FAD8
:10F120000C946DFF0C9400F8982F9595959595F6
:10F130009595905D8F708A301CF1282F295A809107
:10F140003802813019F0823071F008958091C0004A
:10F1500085FFFCCF9093C6008091C00085FFFCCF57
:10F160002093C60008958091C80085FFFCCF90933E
:10F17000CE008091C80085FFFCCF2093CE0008957B
```

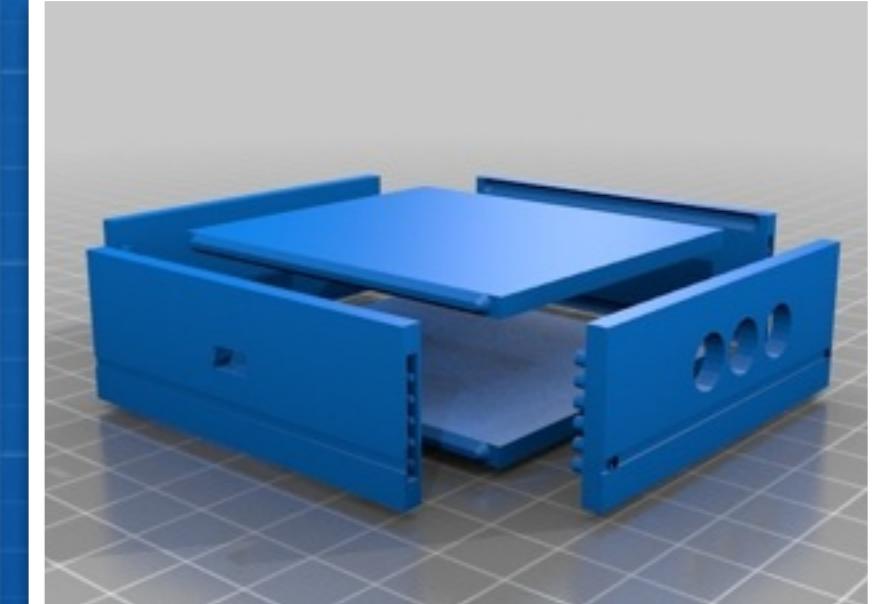
5. Software

```
void displayTotalPower() {
    // title bar
    display.fillRect(0,0,127,9,WHITE);
    display.setCursor(0,1);
    display.setTextColor(BLACK);
    display.print("Total Power");

    // body
    display.fillRect(0,10,127,45,BLACK);
    display.setCursor((128-countDigits(act_total_apow)
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.print(act_total_apow);
    display.print("W");

    // history
    float apow_max = 0;
    for (int i=0; i<24; i++) {
        float aux = hist_total_apow[i];
```

6. Diagrama Mecânico

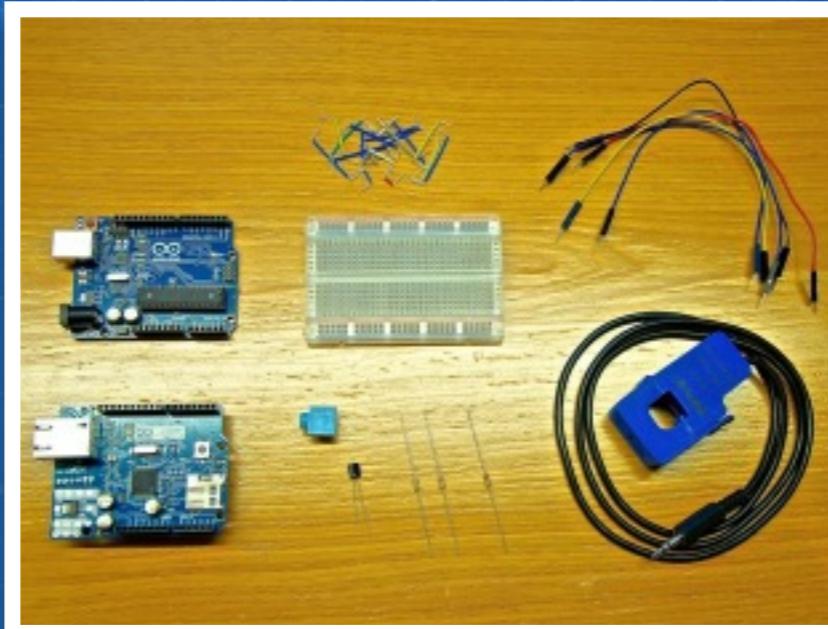


Hardware era Phodda...

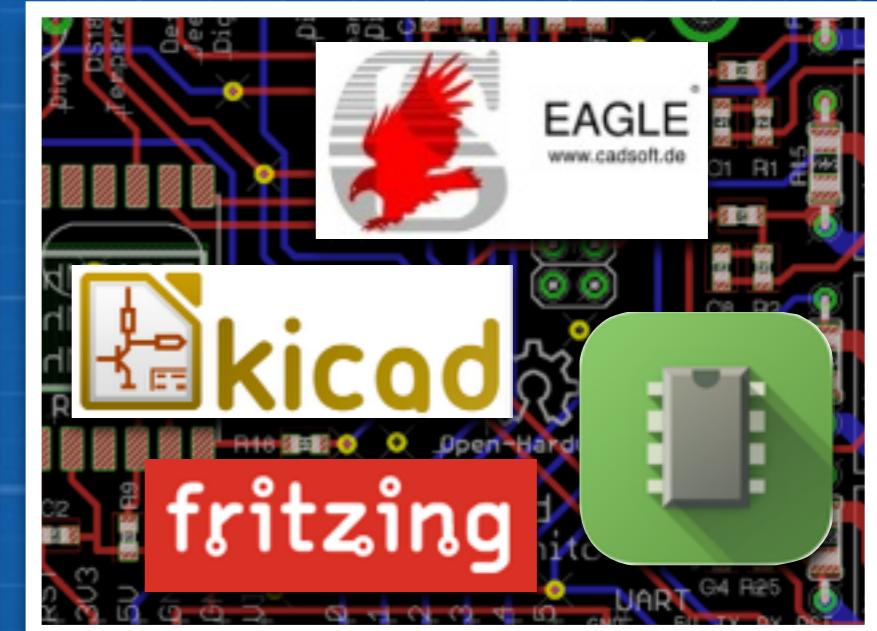
1. Diagrama Elétrico ✓



2. Peças ✓



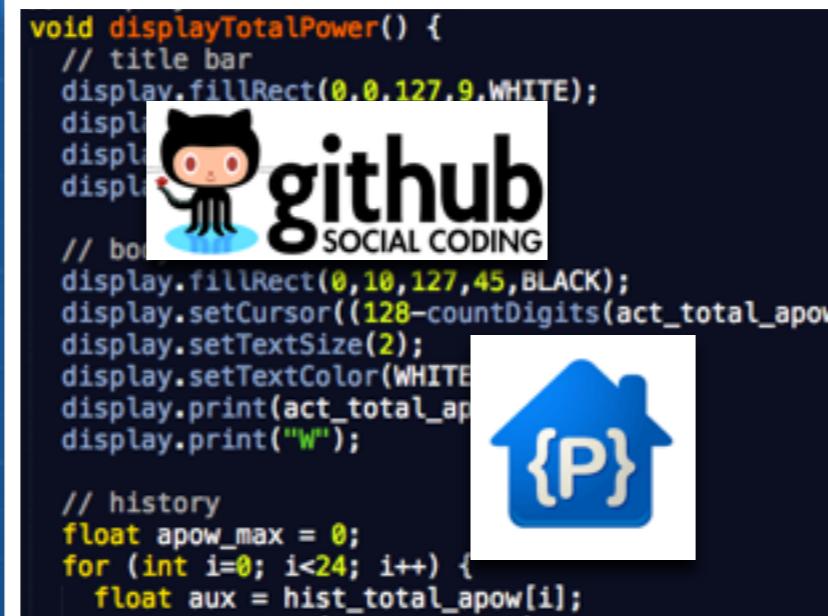
3. Layout da Placa ✓



4. Firmware ✓



5. Software ✓



6. Diagrama Mecânico ✓



Tudo Isto
+
Licenças Decentes &
Adequadas
=

Open Source Hardware

<http://www.inmojo.com/licenses/>

<http://www.shareable.net/blog/how-to-choose-an-open-source-hardware-license>

Open Source Hardware

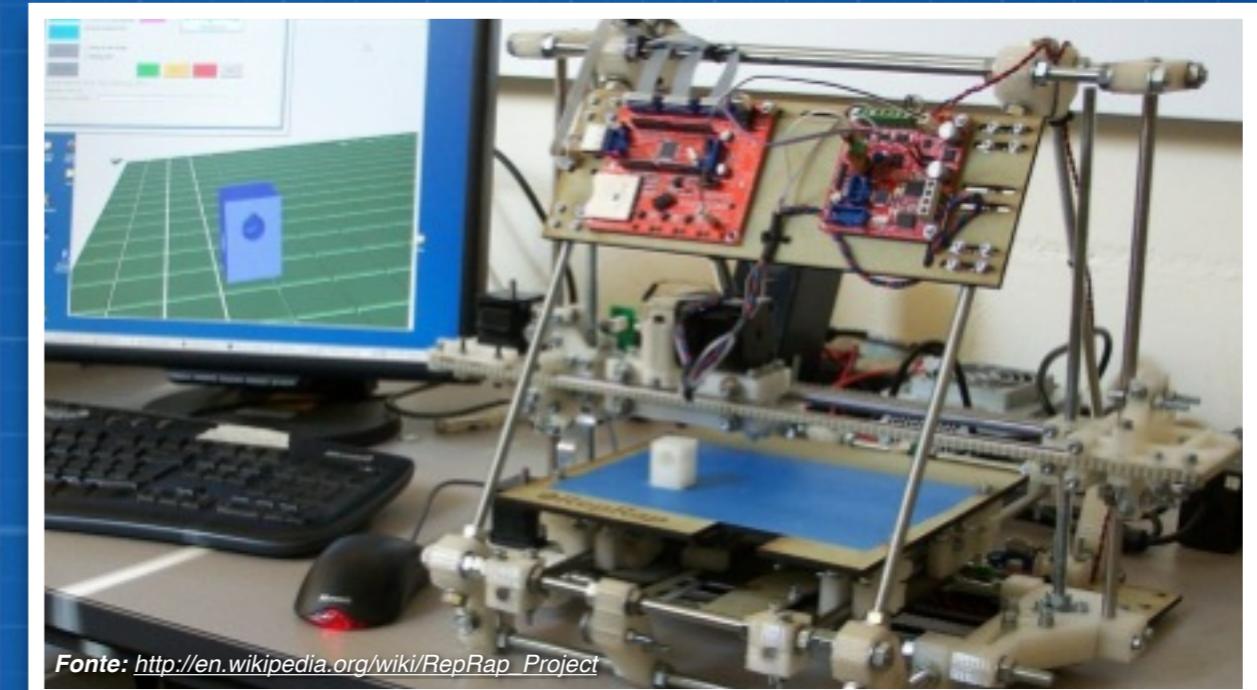
“Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. The hardware’s source, the design from which it is made, is available in the preferred format for making modifications to it.”

<http://www.oshwa.org/faq/>



open source
hardware

Alguns Projetos Open Source Hardware



Para saber mais...

- Wikipedia • http://pt.wikipedia.org/wiki/Hardware_livre
- Open Source Hardware Association • <http://oshwa.org>
- Lady Ada • O que é Open Hardware • <http://www.ladyada.net/library/openhardware/whatisit.html>
- Arduino Team • <http://www.slideshare.net/arduinoteam/open-source-hardware-summit-speech-2011>
- InterCon 2013 • Open Hardware • <http://blog.fazedores.com/open-hardware-no-intercon-2013/>

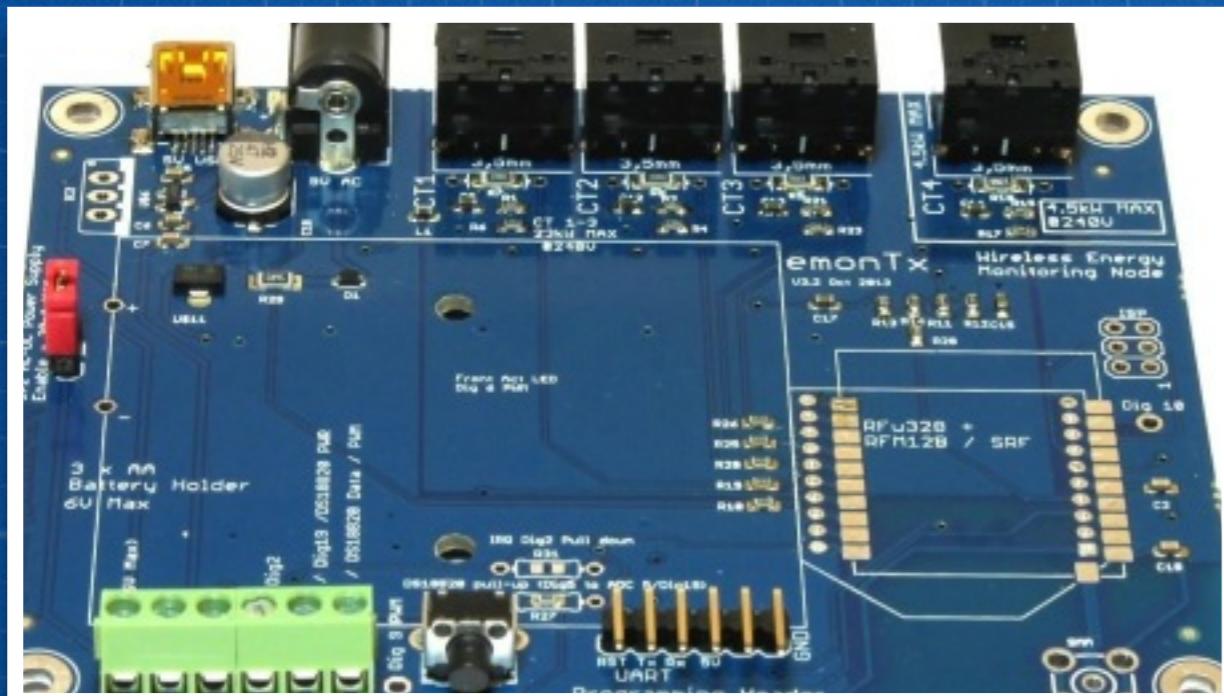
Mas por que
um medidor de
consumo de
energia?

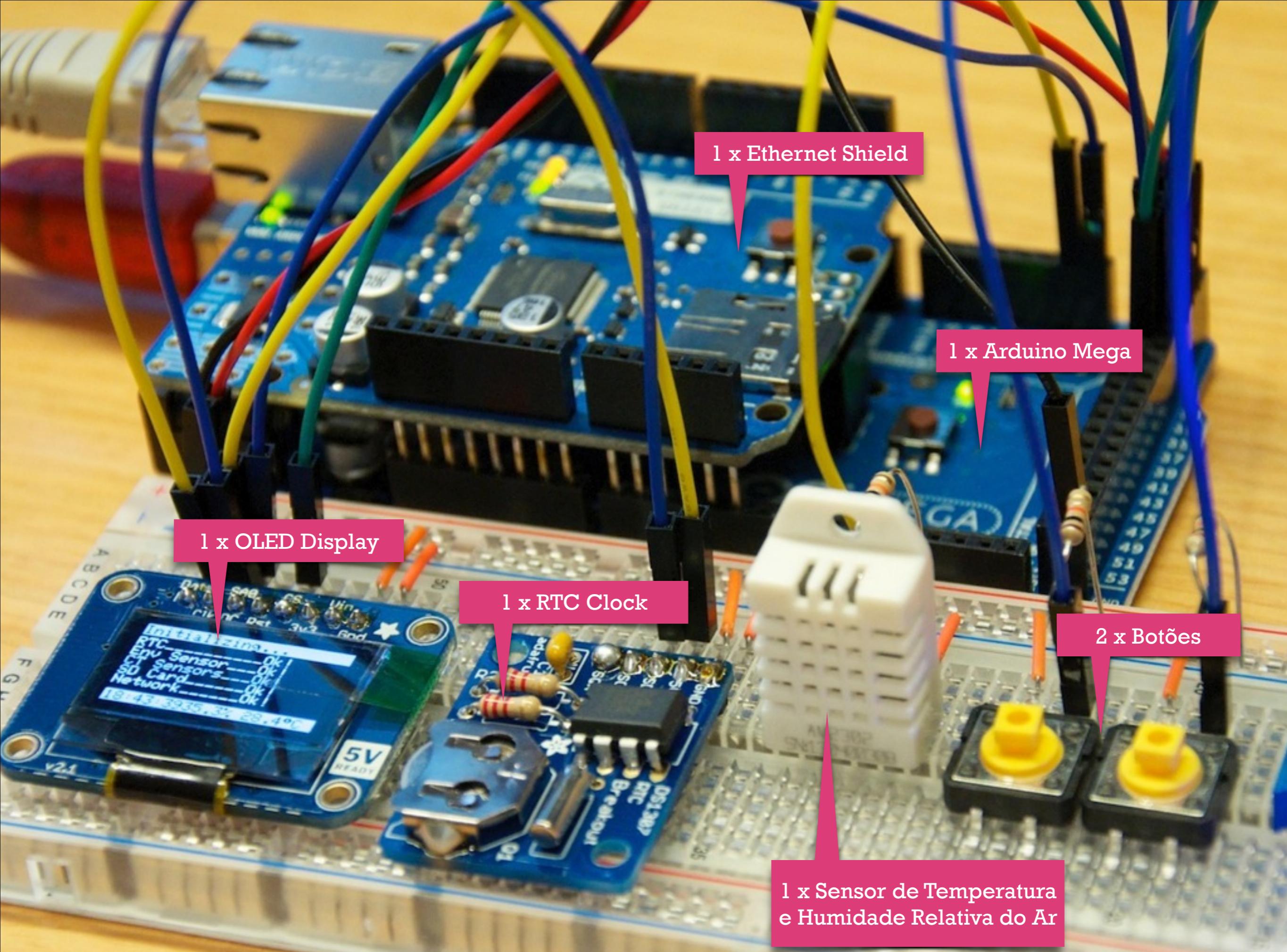


Porque
é
Possível
e
Divertido!

openEnergyMonitor

<http://openenergymonitor.org/>





Vamos ao nosso projeto!

- Baseado no OpenEnergyMonitor
- Protótipo de uma versão simplificada
- Apenas um sensor de corrente
- Servidor exemplo para receber dados na nuvem
- Potência aparente e aproximada (não estamos medindo a tensão da rede e nem a fase)

Um Pouco de Teoria

$$P = U \times I$$

Potência
Watts
(W)

Tensão
Volts
(V)

Corrente
Ampères
(A)

Um Pouco de Teoria

$$P = U \times I$$

Potência
Watts
(W)

Tensão
Volts
(V)

Corrente
Ampères
(A)

(joule/seg)

Quantidade de energia que está sendo consumida ou produzida por unidade de tempo!

Consumo (kWh)
959

(watt-hora)

Quantidade de energia necessária para alimentar uma carga de **1 watt** pelo período de **1 hora**.

(kWh ou Quilowatt-hora)
É por "energia" que pagamos na conta de luz.
3600000 joules

Como:

1 hora = 3600 segundos

Temos:

1 watt-hora

=

1 watt x 3600 segundos

=

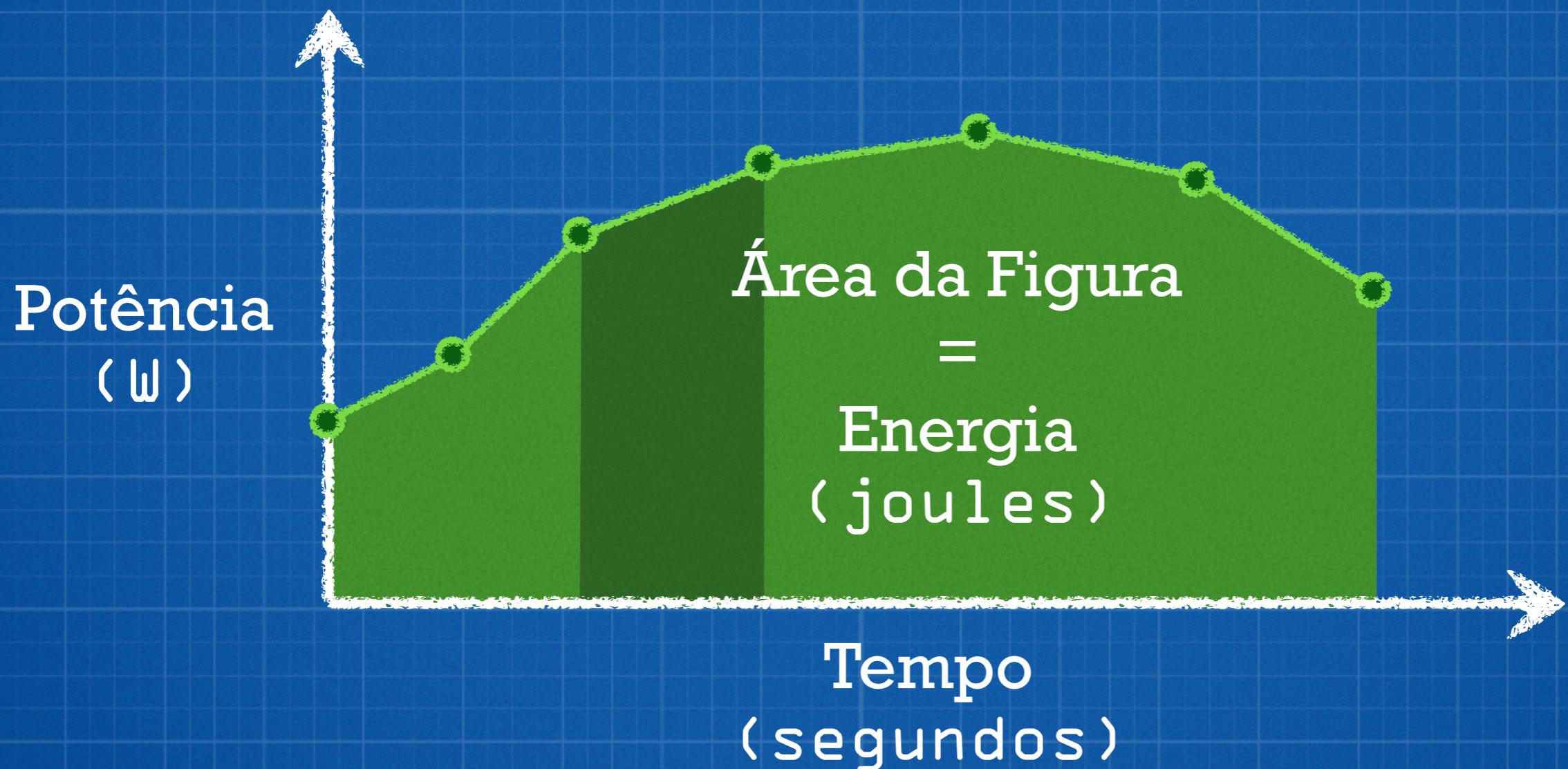
1 (joule / segundos) x

3600 segundos

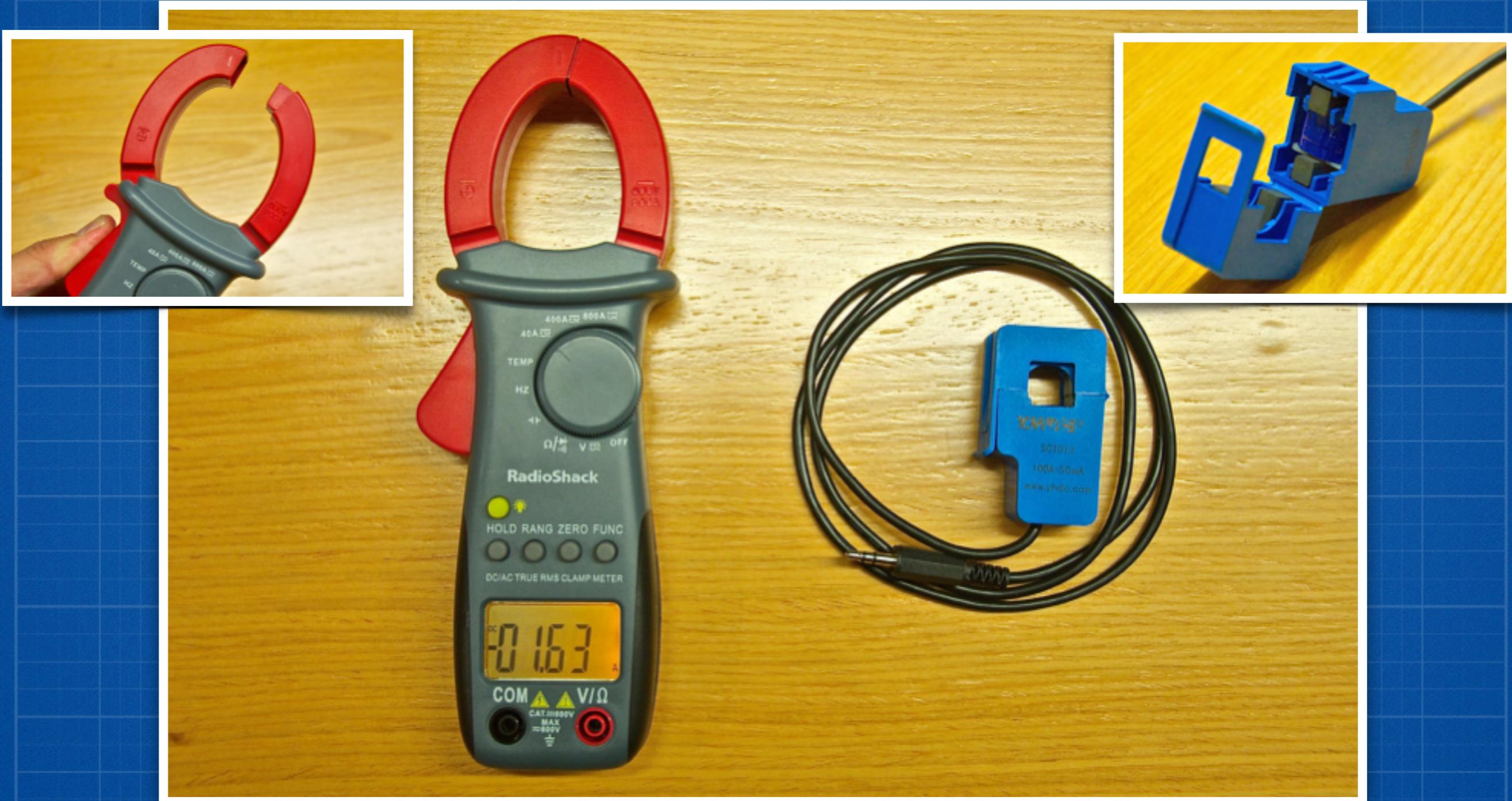
=

3600 joules

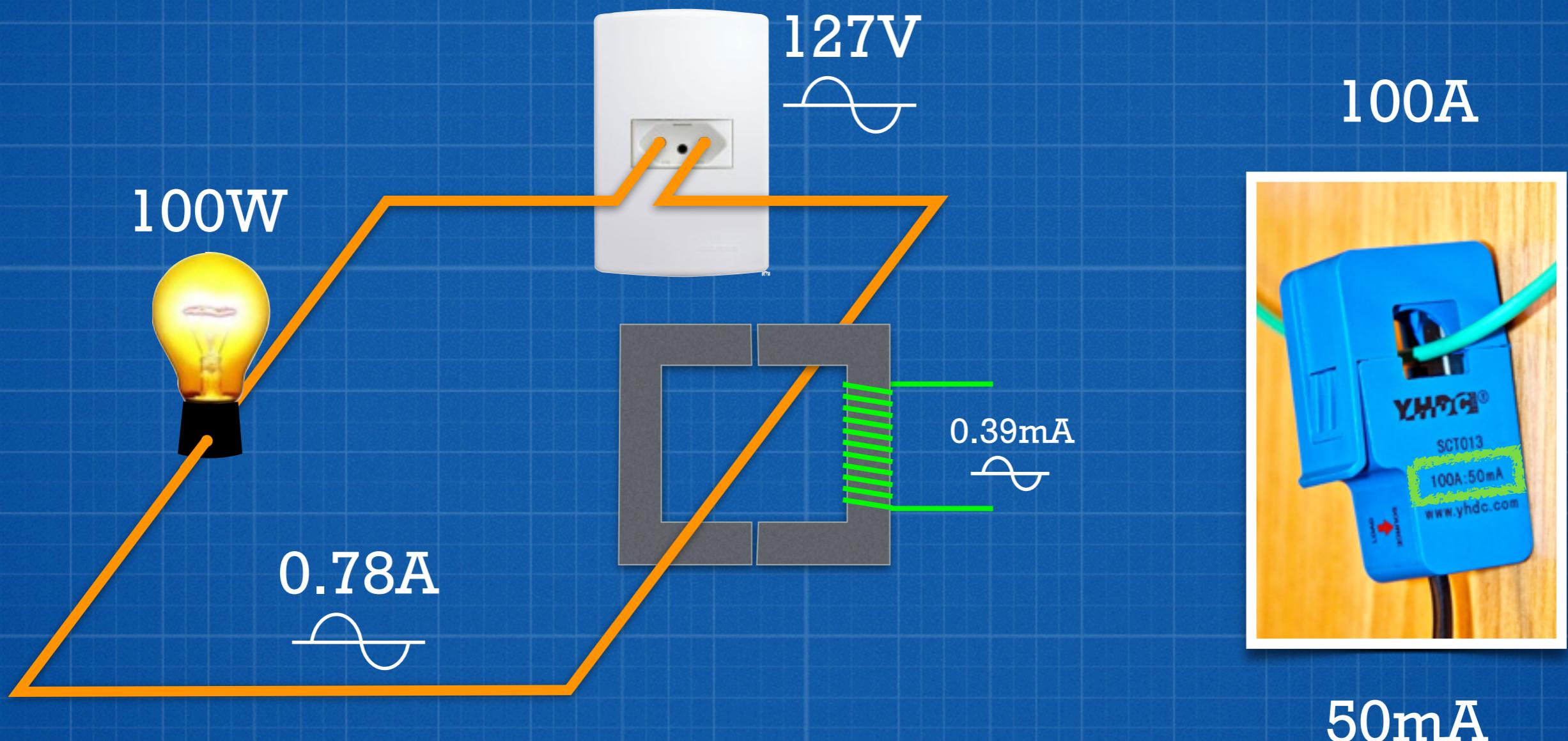
Calculando a Energia a partir de medidas da Potência



Um Sensor para Corrente Elétrica (AC)



Um Sensor para Corrente Elétrica (AC)



Interfaceando com o Arduino

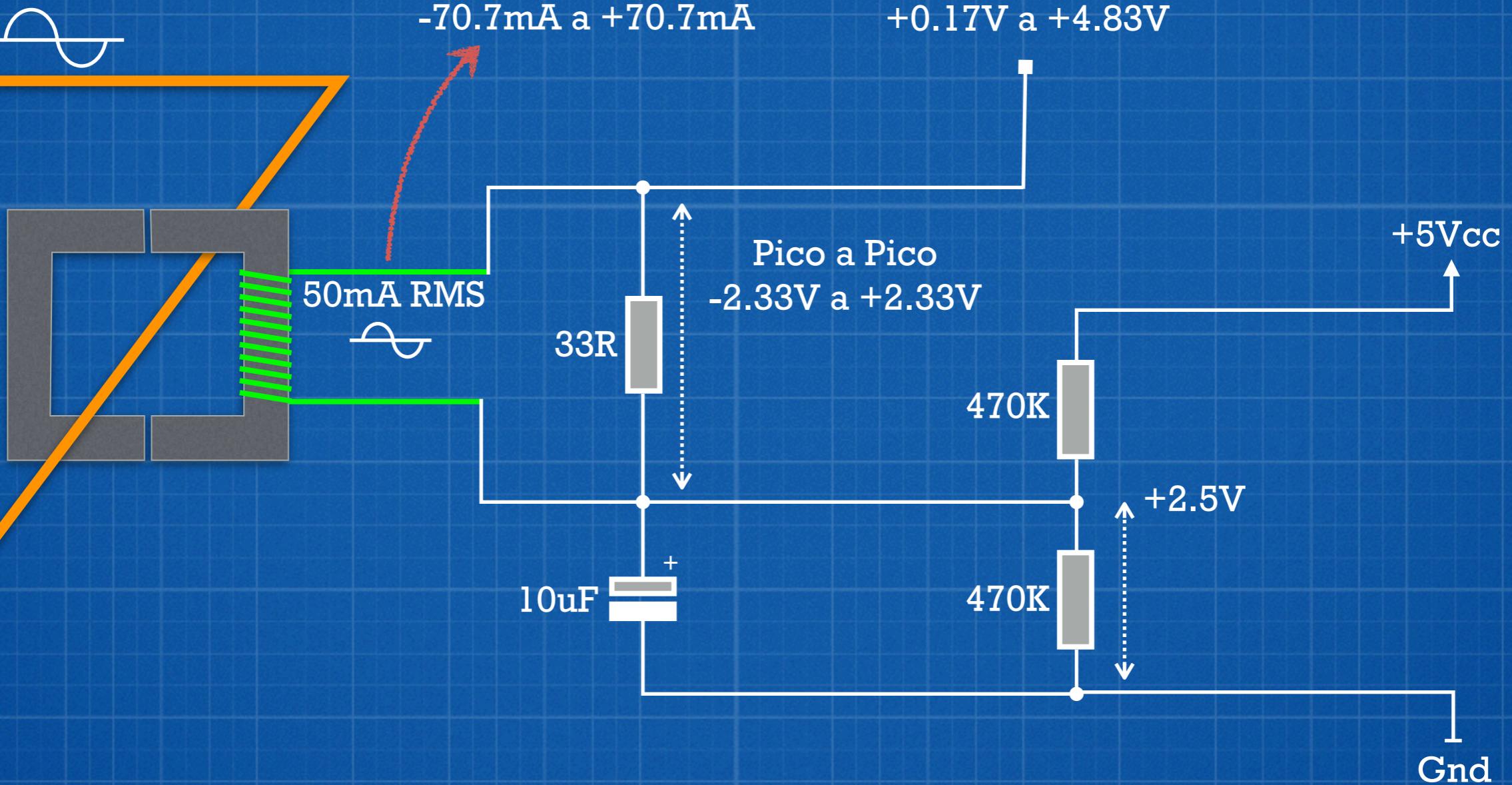
100A RMS

Pico a Pico
-70.7mA a +70.7mA

Pico a Pico
+0.17V a +4.83V

Pico a Pico
-2.33V a +2.33V

+5Vcc



Interfaceando com o Arduino

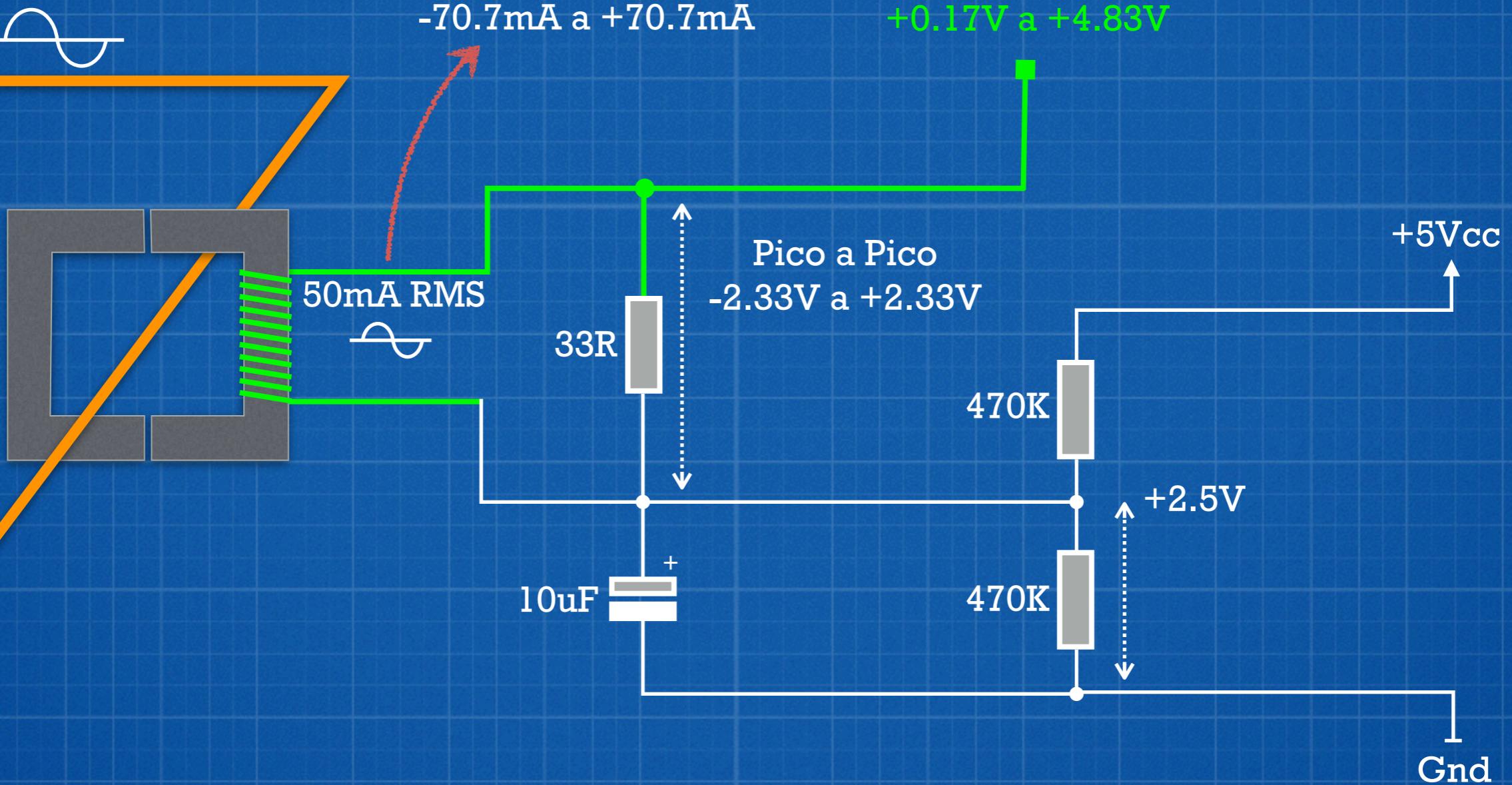
100A RMS

Pico a Pico
-70.7mA a +70.7mA

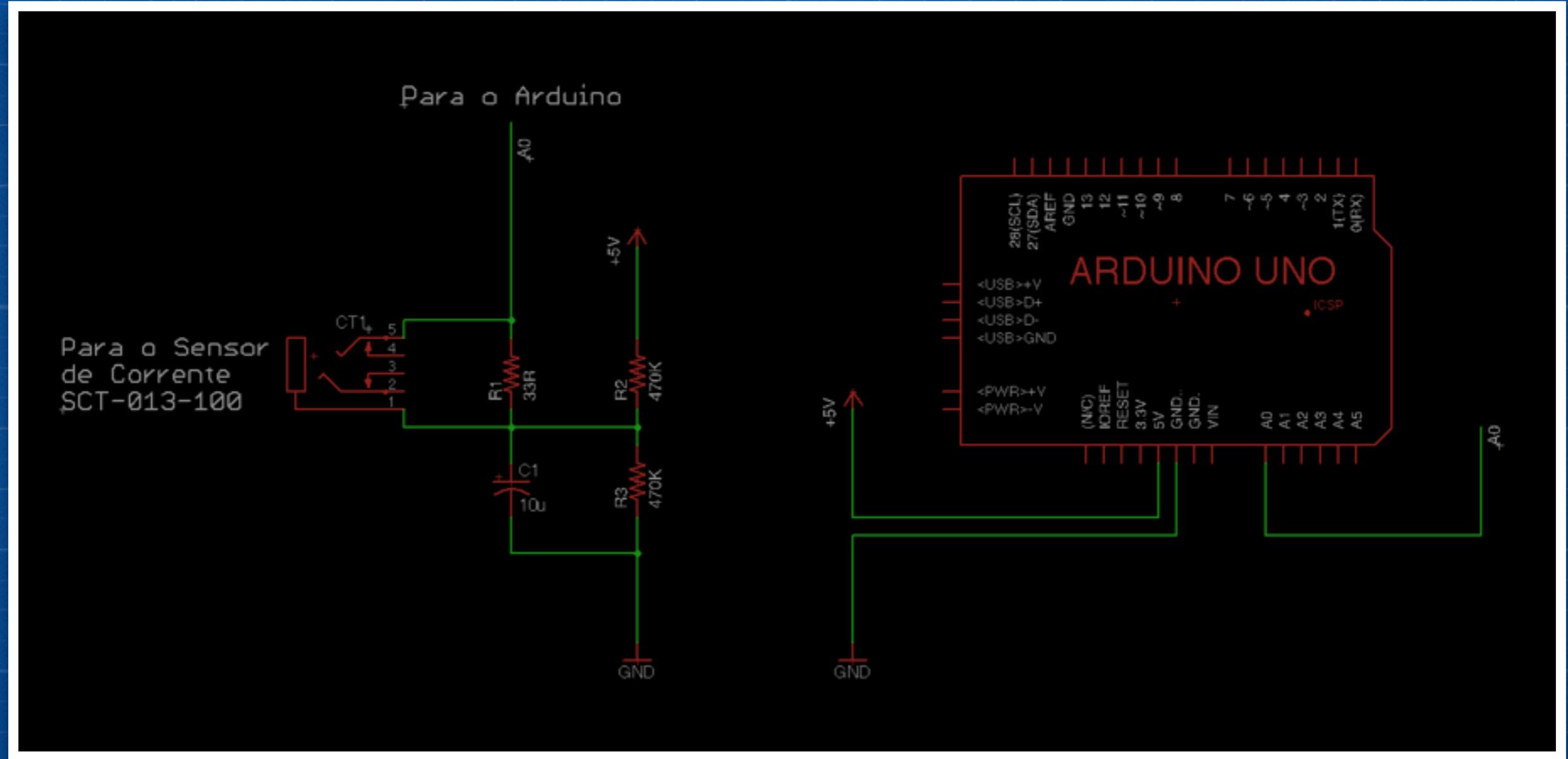
Pico a Pico
+0.17V a +4.83V

Pico a Pico
-2.33V a +2.33V

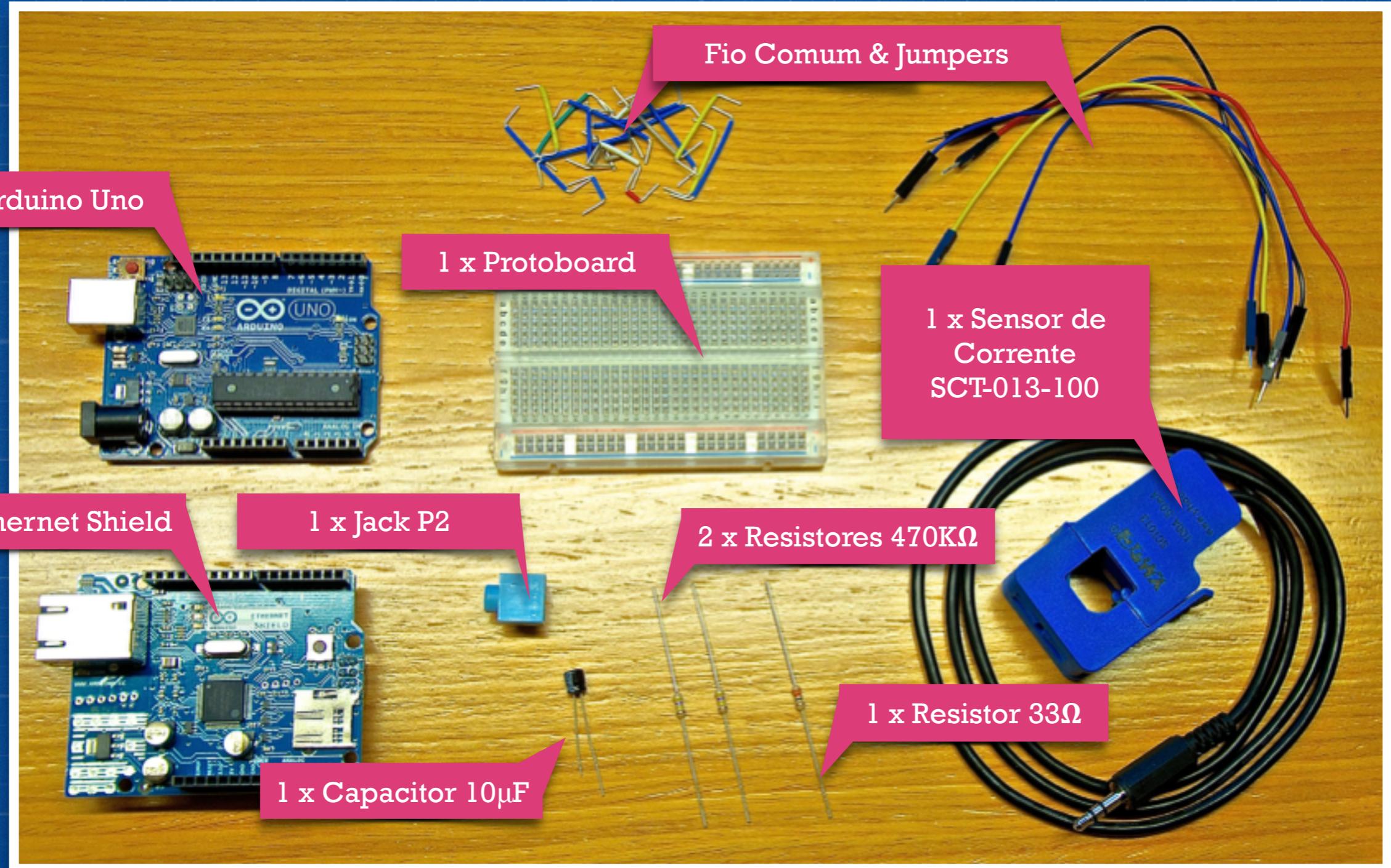
+5Vcc



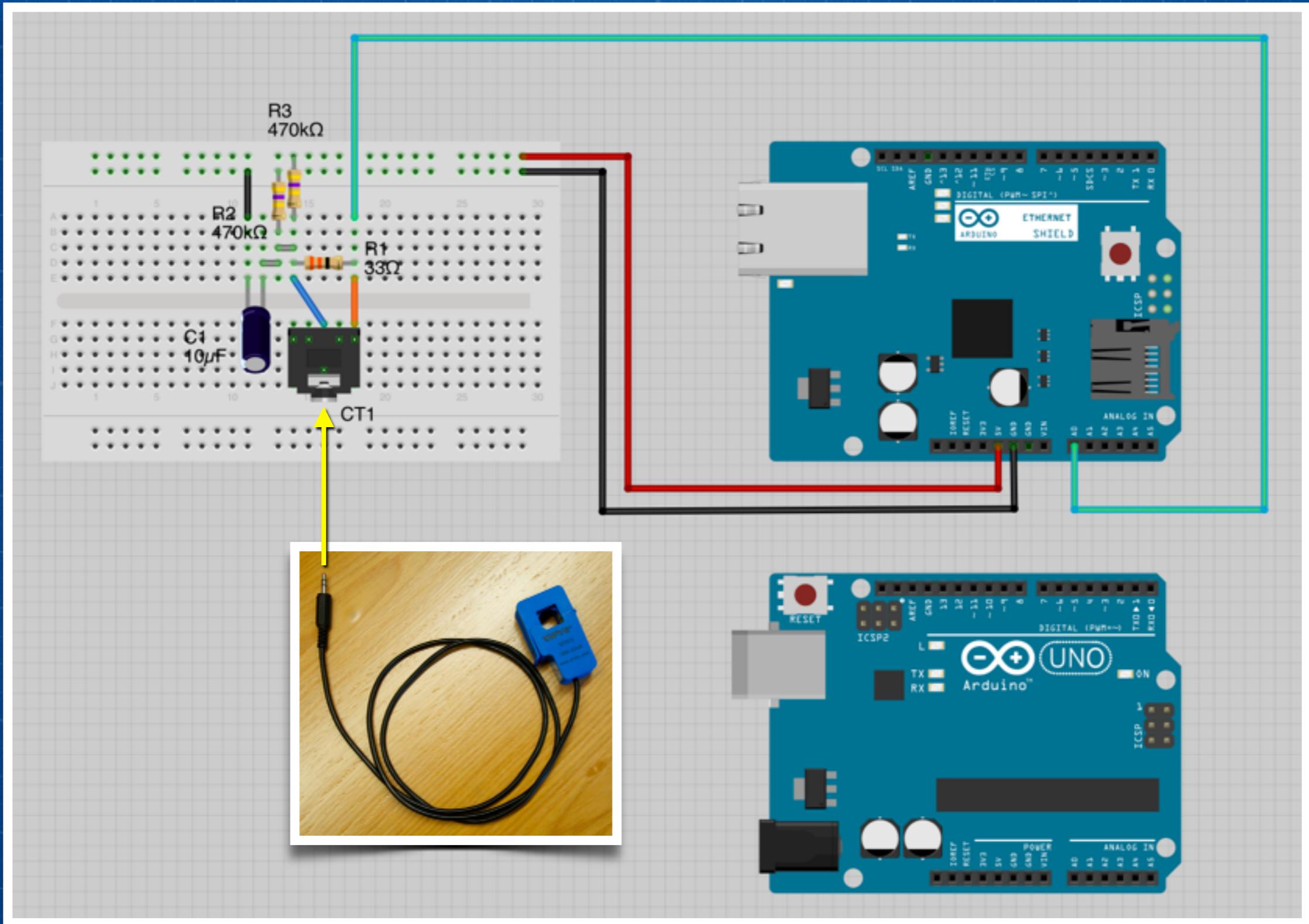
1. Diagrama Elétrico



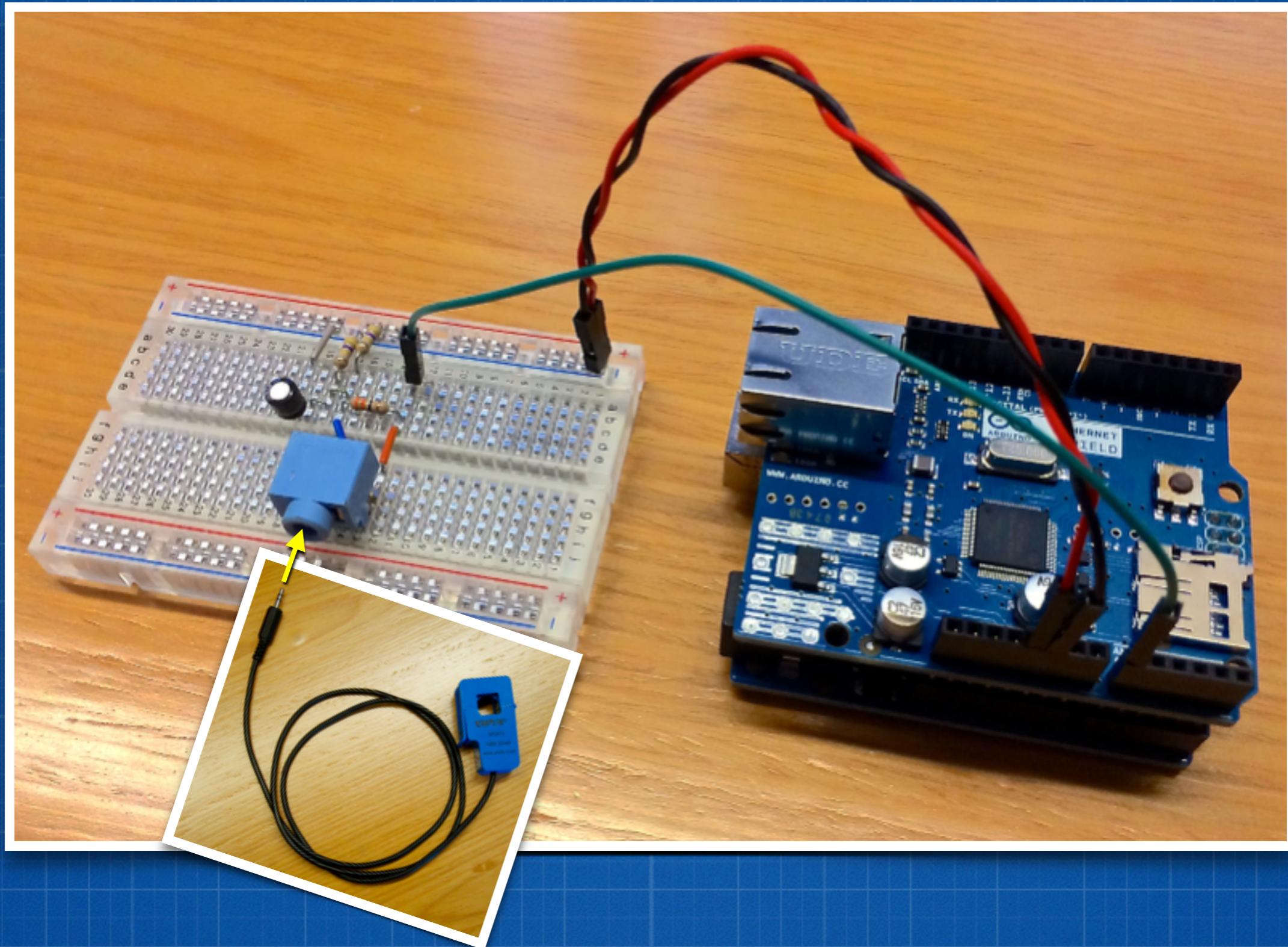
2. Peças



3. Layout



Quase lá!



4. Firmware

- Nada de especial por aqui.
- ;-)

5. Software

- **Arduino**
 - Realização das medidas elétricas (emonLib)
 - Comunicação com o servidor (Ethernet Lib)
- **Servidor**
 - Recebimento e armazenamento das medidas
 - Cálculo da energia consumida
 - Apresentação dos resultados

Arduino : Anatomia de um Sketch

```
// bibliotecas & dependências  
// variáveis globais  
// configurações  
// funções  
  
void setup()  
{  
    // inicialização  
    // executado antes de tudo e apenas uma vez  
}  
  
void loop()  
{  
    // laço principal  
    // executado indefinidamente  
}
```

Arduino : Bibliotecas & Configurações

```
// bibliotecas necessárias
#include <EmonLib.h> // Emon lib - monitores de energia
#include <SPI.h>      // SPI lib - para shield ethernet
#include <Ethernet.h> // Ethernet lib - para comunicação

// variáveis globais
EnergyMonitor emon1; // instância de um monitor de energia
const int CT_PIN = 1; // pino conectado ao CT (sensor)

byte mac[] ={0xAA,0xBB,0xCC,0xDD,0xEE,0xFF}; // endereço MAC
EthernetClient client; // instância de um cliente TCP
```

Arduino : setup()

```
void setup()
{
    // inicializa porta serial para usarmos em depuração
    Serial.begin(9600);
    // inicializa o monitor de corrente
    emon1.current(CT_PIN, 64);
    // inicializa o shield ethernet
    if (Ethernet.begin(mac) == 0) {
        Serial.println("Shield ethernet não inicializado.");
        while(1); // abortar (não temos rede)
    } else {
        Serial.println("Shield ethernet inicializado com
sucesso!");
    }
    delay(1000);
    Serial.print("Endereço IP: ");
    Serial.println(Ethernet.localIP()); // imprime endereço IP
}
```

Arduino : loop()

```
void loop()
{
    double Irms = emon1.calcIrms(1480); // mede a corrente
    double Potencia = Irms * 127.0; // calcula a potência

    Serial.print("Irms: ");
    Serial.print(Irms); // imprime a corrente na serial
    Serial.print(" A \t");
    Serial.print("Potencia: ");
    Serial.print(Potencia); // imprime a potência na serial
    Serial.print(" W");
    Serial.println();

    sendData(Irms, Potencia); // envia dados pro servidor

    delay(15000); // aguarde 15 segundos
}
```

Arduino : sendData()

```
void sendData(double Irms, double Potencia) {  
    client.connect("10.0.1.108",4000); // conecta ao servidor  
    delay(500);  
    if (client.connected()) {  
        // realiza o GET enviando dados como parâmetros  
        client.print("GET /medida?irms="); client.print(Irms);  
        client.print("&potencia="); client.print(Potencia);  
        client.println(" HTTP/1.1");  
        client.println("Host: 10.0.1.108");  
        client.println();  
        delay(500);  
        // lê a resposta (e a ignora)  
        while (client.available()) { char c = client.read(); }  
        delay(500);  
        client.stop(); // fecha a conexão  
        Serial.println("Dados enviados!");  
    } else {  
        Serial.println("Falha na conexão.");  
    }  
}
```

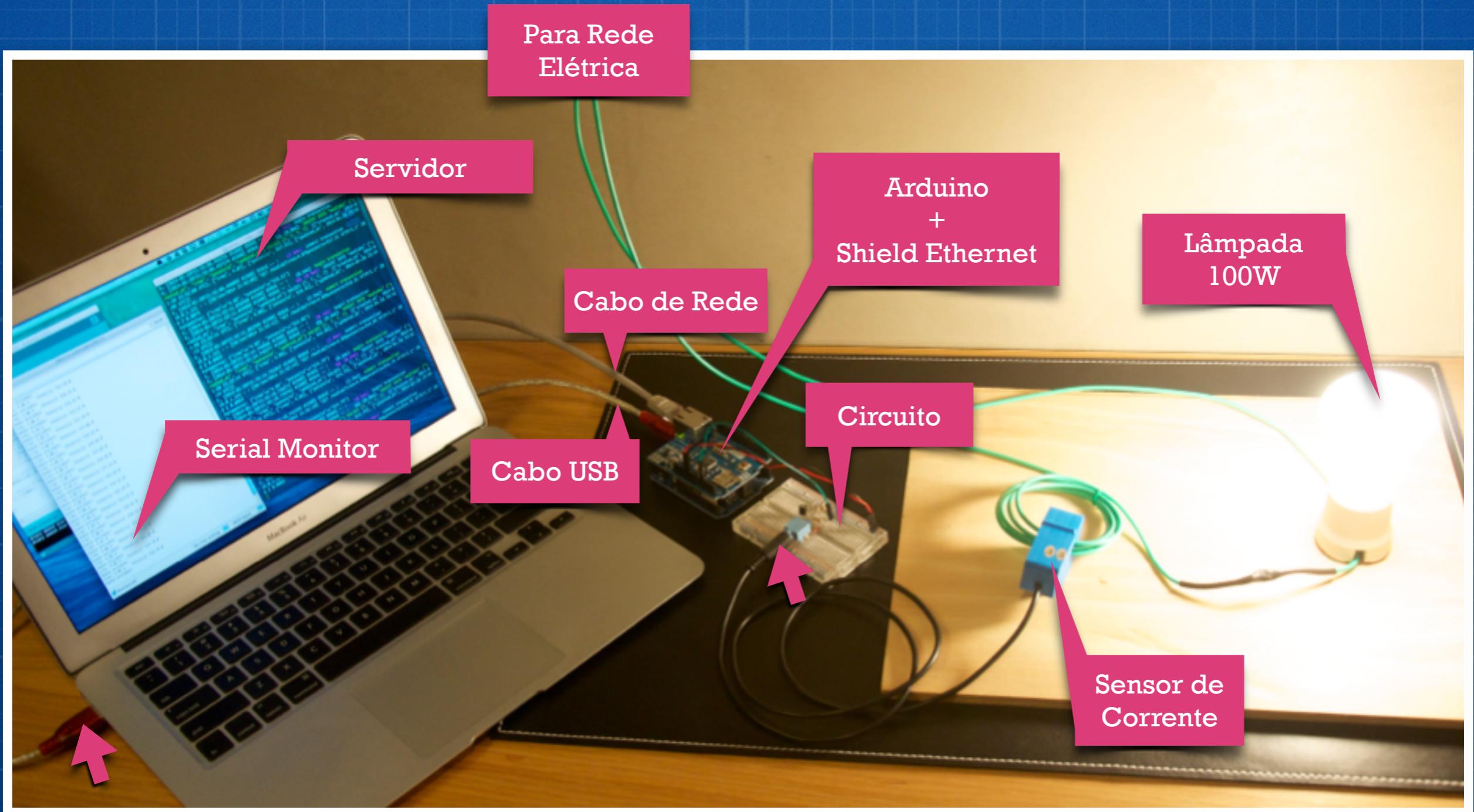
Servidor : GET /medida

```
# recebe dados de uma nova medida
# e armazena no banco de dados
get "/medida" do
  # cria objeto com parâmetros da requisição
  @medida = Medida.new(
    :irms => @params['irms'],
    :potencia => @params['potencia']
  )
  # salva no banco
  if @medida.save
    return 'ok'
  else
    return 'nok'
  end
end
```

Servidor : GET /dashboard

```
# calcula energia total consumida no dia
get "/dashboard" do
  # pega todas as medidas de hoje (não faça isto em prod!)
  @medidas = Medida.where("DATE(created_at) = DATE(?)" ,
                           Time.now)
  @consumo = 0 # acumulador da energia total
  anterior = nil # medida anterior
  @medidas.each do |atual|
    if anterior
      # calcula a energia consumida entre duas medidas
      tempo = atual.created_at - anterior.created_at
      energia = (atual.potencia + anterior.potencia) *
                 tempo / 2
      @consumo = @consumo + energia # acumula
    end
    anterior = atual # atualiza medida anterior
  end
  @consumo = @consumo / 3600000 # joules -> kWh
end
```

Juntando Tudo



Juntando Tudo



Serial Monitor - /dev/cu.usbmodem40111

UNREGISTERED

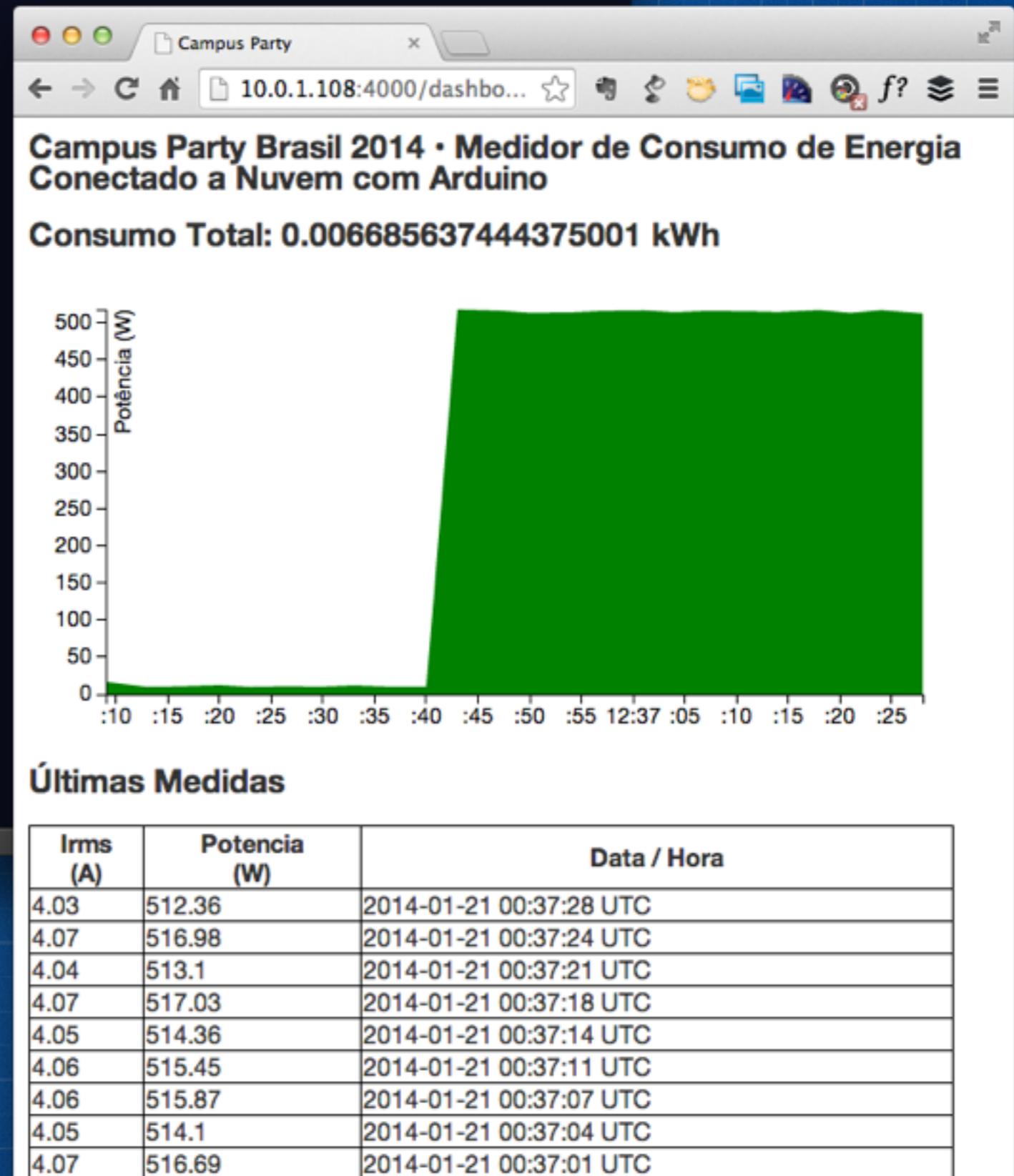
Serial Monitor - /dev/cu.usbmodem40111

```

1 Vamos lá Campus Party Brasil 2014!!
2 Inicializando Ethernet com DHCP...feito!
3 Endereço IP: 10.0.1.121
4 Iniciando medições:
5
6 Irms: 0.13 A Potencia: 16.74 W
7 Dados enviados!
8 Irms: 0.08 A Potencia: 10.06 W
9 Dados enviados!
10 Irms: 0.08 A Potencia: 10.75 W
11 Dados enviados!
12 Irms: 0.08 A Potencia: 10.10 W
13 Dados enviados!
14 Irms: 0.08 A Potencia: 10.59 W
15 Dados enviados!
16 Irms: 4.05 A Potencia: 513.87 W
17 Dados enviados!
18 Irms: 4.06 A Potencia: 515.89 W
19 Dados enviados!
20 Irms: 4.07 A Potencia: 516.69 W
21 Dados enviados!
22 Irms: 4.05 A Potencia: 514.10 W
23 Dados enviados!
24 Irms: 4.06 A Potencia: 515.87 W
25 Dados enviados!
26 Irms: 4.06 A Potencia: 515.45 W
27 Dados enviados!
28 Irms: 4.05 A Potencia: 514.36 W
29 Dados enviados!
30

```

Line 2, Column 1



Incrementando...

- Medição em 2 ou 3 fases
- Medição da tensão da rede
- Displays
- Sensores de temperatura
- Você manda...

Crie, Construa e Compartilhe



<https://github.com/mlemos/energy-monitor-cpbr7>



FAZEDORES.com

CRIE • CONSTRUA • COMPARTILHE



blog.FAΖEDORES.com
CRIE • CONSTRUA • COMPARTILHE



forum.FAZEDORES.com
CRIE • CONSTRUA • COMPARTILHE

Fórum Fazedores

forum.fazedores.com

FAZEDORES

Diego Gomes 1

Todas As Categorias Populares Novos (5) Não lido Favoritos Melhores Categorias + Criar Tópico

Tópico	Categoria	Usuários	Postagens	Curtidas	Visualizações	Atividade
★ 🔒 Como funcionam as categorias?			3	3 ❤️	40	4d 4d
★ 🔒 Bem Vindo ao Fórum do Fazedores			2	1 ❤️	37	8d 4d
★ Fazendo PCBs caseiras nova			1		6	4h 4h
★ Que ferramentas um Fazedor precisa ter obrigatoriamente no seu laboratório? nova			2	2 ❤️	29	11h 5h
★ Onde comprar Arduino no Brasil? nova			11		139	1d 7h
★ Como escolher uma impressora 3D? nova			4		20	13h 7h
★ Como foi feito o fórum fazedores? nova			5	12 ❤️	42	1d 11h
★ Onde e como fabricar PCBs (placas de circuito impresso)?			11	5 ❤️	209	3d 12h

Não há mais tópicos recentes. Que tal começar um assunto?

valeu!