Open Hardware na prática:

Construindo um medidor de consumo de energia elétrica conectado à nuvem com Arduino

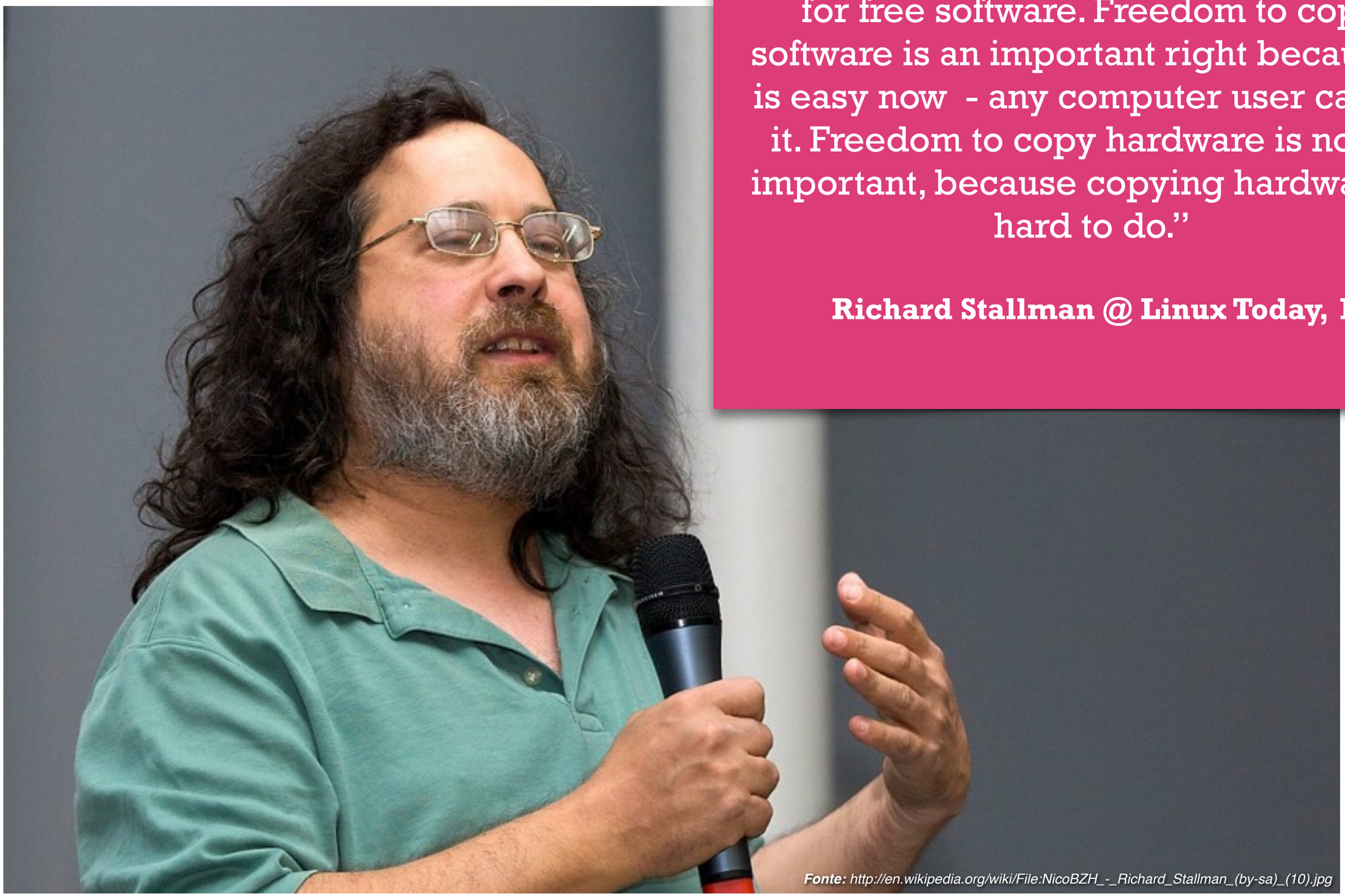
Manoel Lemos
manoel@lemos.net
@mlemos
http://fazedores.com
http://manoellemos.com

Muito cuidado ao trabalhar com a rede elétrica!

Não arrisque sua vida!

"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



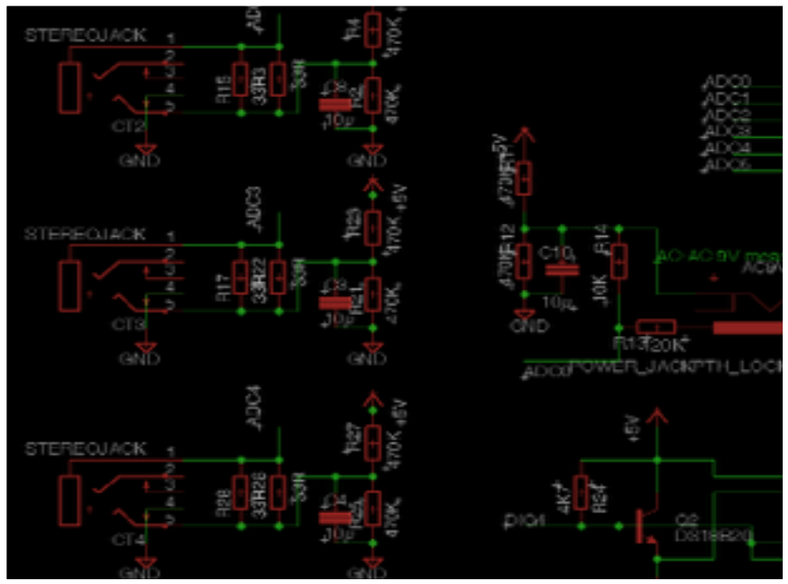
Fonte: *http://en.wikipedia.org/wiki/File:NicoBZH_-_Richard_Stallman_(by-sa)_(10).jpg*

Hardware é Phodda!

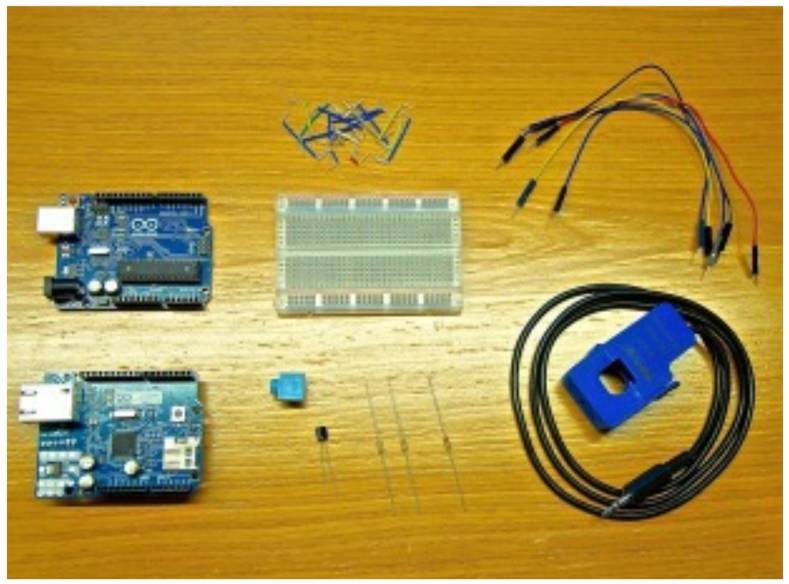
Fonte: http://en.wikipedia.org/wiki/File:NicoBZH_-_Richard_Stallman_(by-sa)_(10).jpg

Hardware é Phodda..

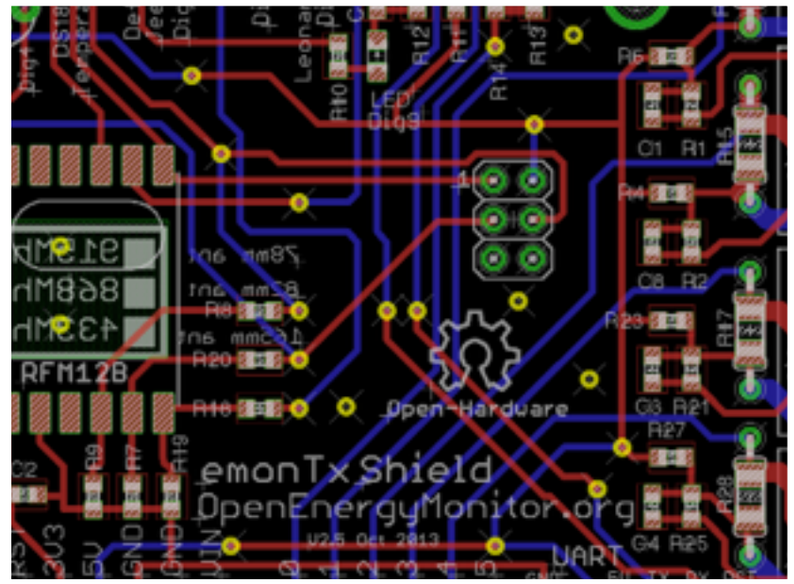
1. Diagrama Elétrico



2. Peças



3. Layout da Placa



4. Firmware

```
:10F0A0000C9492F80C9492F80C9492F80C9492F8B8
:10F0B0000C9492F80C9492F80C9492F80C9492F8A8
:10F0C0000C9492F80C9492F80C9492F80C9492F898
:10F0D0000C9492F80C9492F80C9492F80C9492F888
:10F0E0000C9492F811241FBECFEFD1E2DEBFCDBF4A
:10F0F00012E0A0E0B2E0EEEDFEEF01E00BBF02C0D7
:10F1000007900D92A833B107D9F71BBE13E0A8E30F
:10F11000B2E001C01D92A334B107E1F70E9412FAD8
:10F120000C946DFF0C9400F8982F9595959595F6
:10F130009595905D8F708A301CF1282F295A809107
:10F14003802813019F0823071F008958091C0004A
:10F1500085FFFCCF9093C6008091C00085FFFCCF57
:10F160002093C60008958091C80085FFFCCF90933E
:10F17000CE008091C80085FFFCCF2093CE0008957B
```

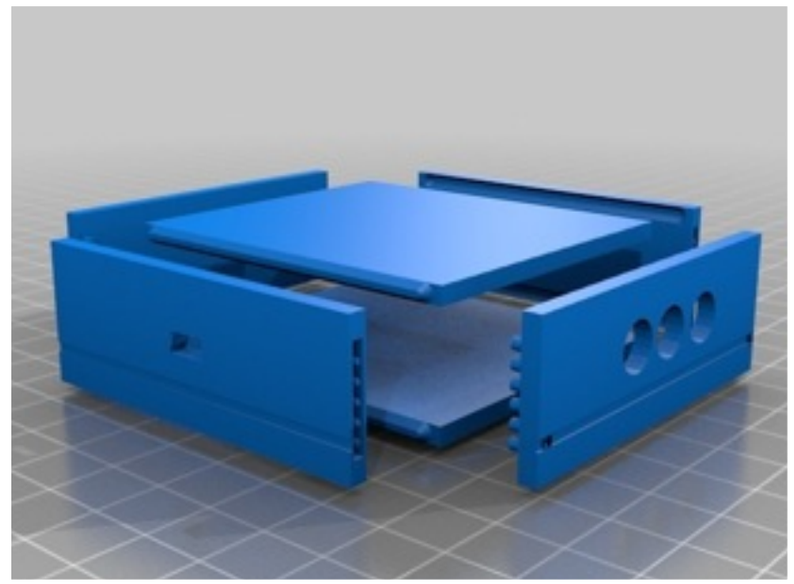
5. Software

```
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];
```

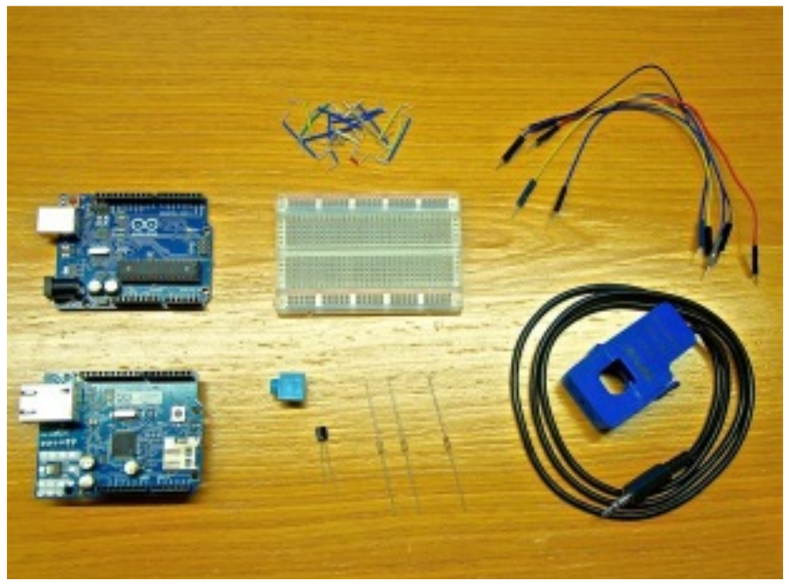
6. Diagrama Mecânico



Hardware era ≠ Phodda..

1. Diagrama Elétrico ✔

2. Peças ✔



3. Layout da Placa ✔

4. Firmware ✔

```
:10F0A0000C9492F80C9492F80C9492F80C9492F8B8
:10F0B0000C9492F80C9492F80C9492F80C9492F8A8
:10F0C0                  492F80C9492F898
:10F0D0                  492F80C9492F888
:10F0E0                  FD1E2DEBFCDBF4A
:10F0F0                  F01E00BBF02C0D7
:10F1000007900D92A833B107D9F71BBE13E0A8E30F
:10F11000B2E001C01D92A33                 412FAD8
:10F120000C946DFF0C9400F                 59595F6
:10F130009595905D8F708A3                 A809107
:10F140003802813019F0823                 1C0004A
:10F1500085FFFCCF9093C60                 FFCCF57
:10F160002093C6000895809                 F90933E
:10F17000CE008091C80085FFFCCF2093CE0008957B
```


5. Software ✔

```
void displayTotalPower() {
  // title bar
  display.fillRect(0,0,127,9,WHITE);
  displ
  displ
  displ

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

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


6. Diagrama Mecânico ✔

Tudo Isto

+

Licenças Decentes & Ádequadas

=

Open Source Hardware

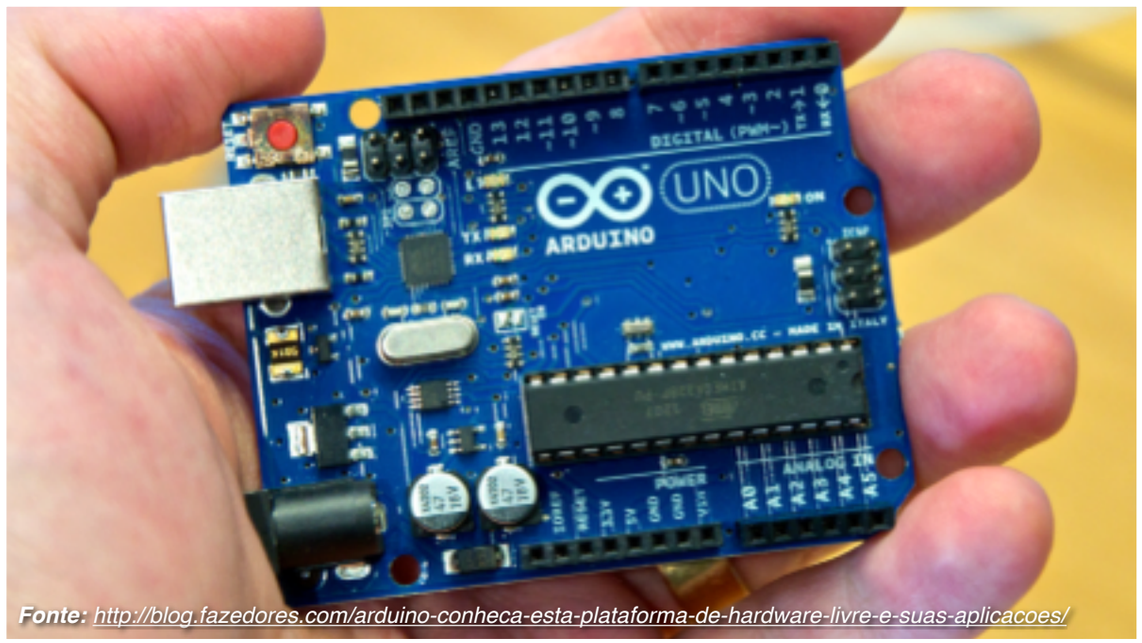
http://www.inmojo.com/licenses/
http://www.shareable.net/blog/how-to-choose-an-open-source-hardware-license

Open Source Hardware

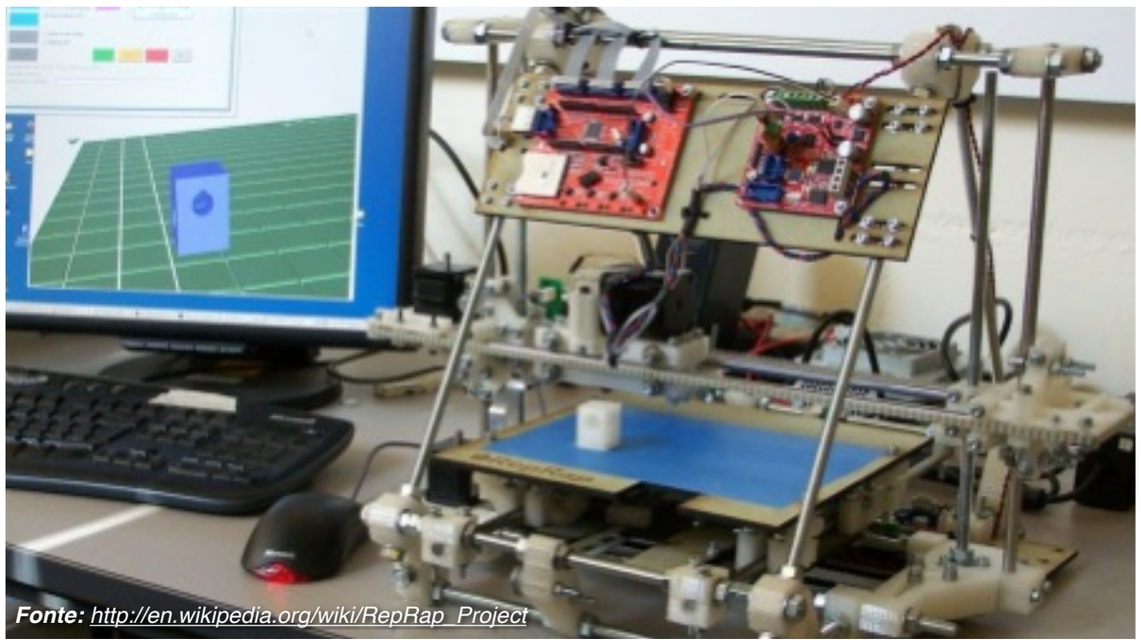
"Open source hardware is hardware whose design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design. The hardware's source, the design from which it is made, is available in the preferred format for making modifications to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

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

Alguns Projetos Open Source Hardware



Fonte: http://blog.fazedores.com/arduino-conheca-esta-plataforma-de-hardware-livre-e-suas-aplicacoes/



Fonte: http://en.wikipedia.org/wiki/RepRap_Project

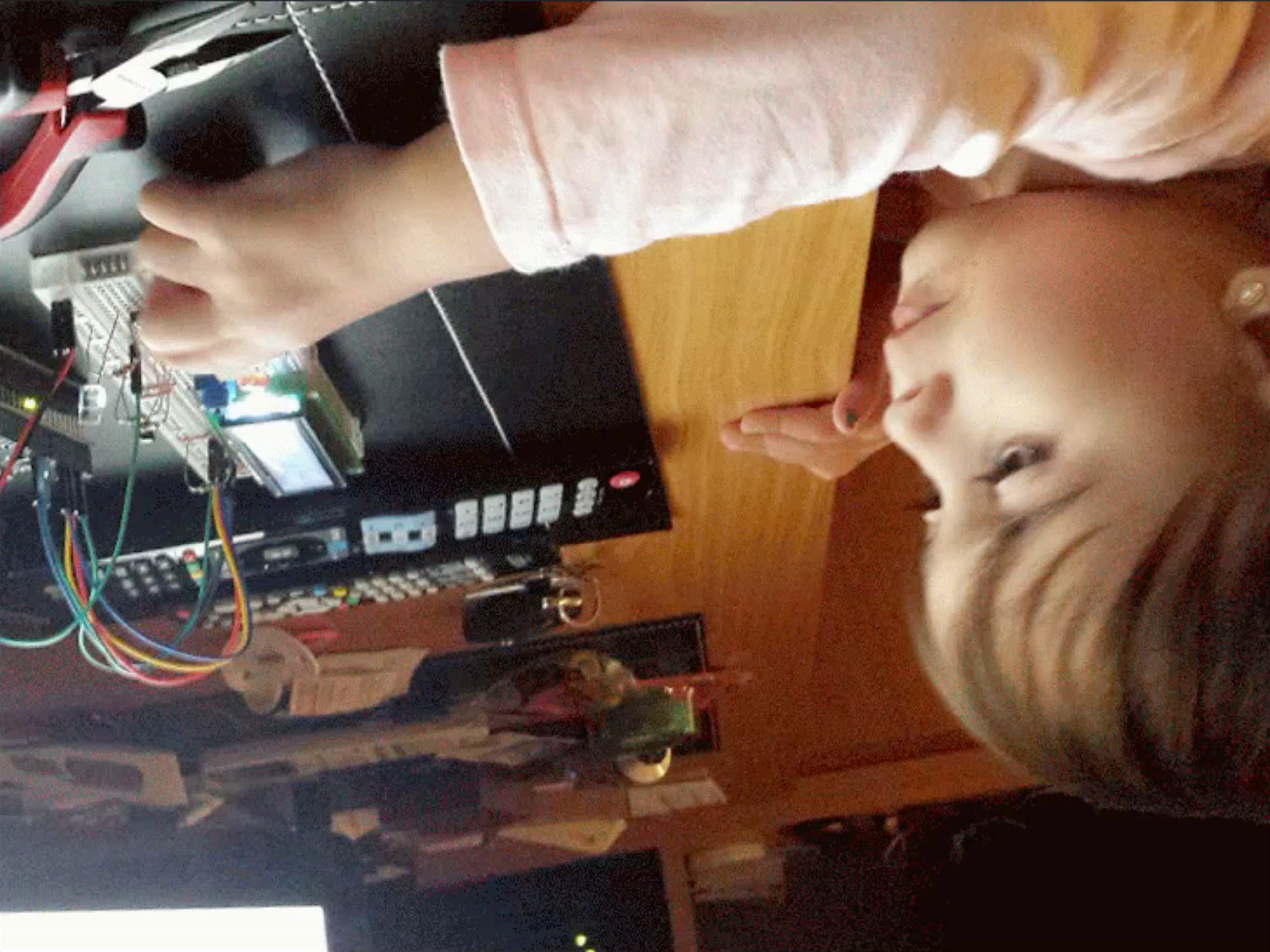
Fonte: http://www.indiegogo.com/projects/the-open-hand-project-a-low-cost-robotic-hand

Fonte: http://opensourceecology.org/

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?

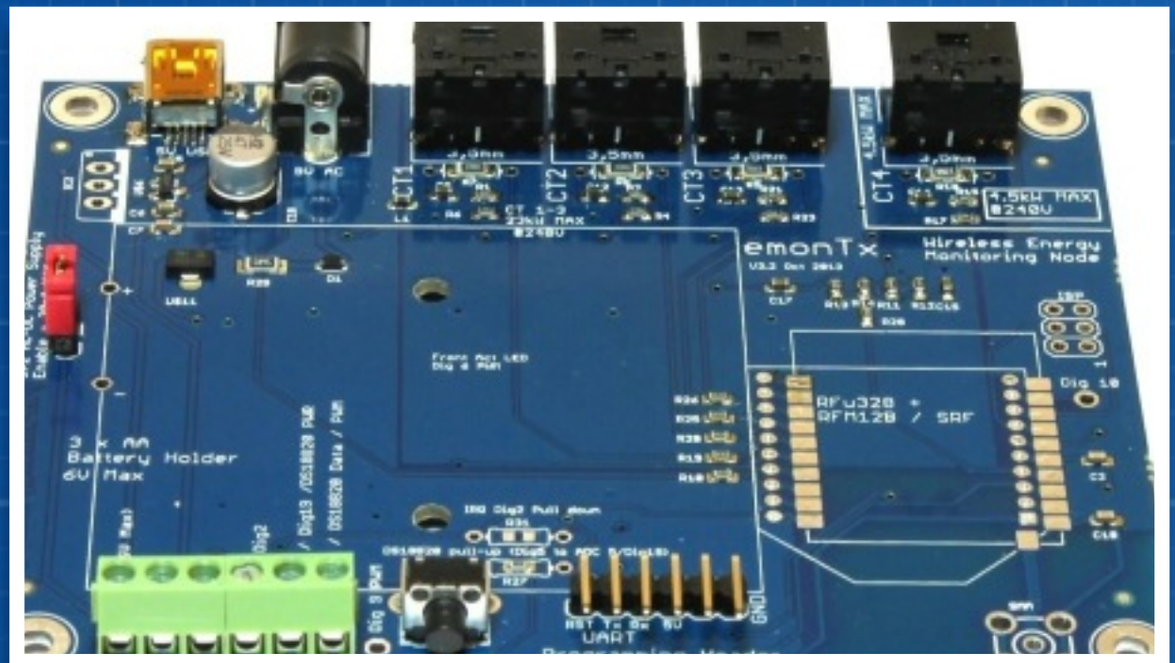


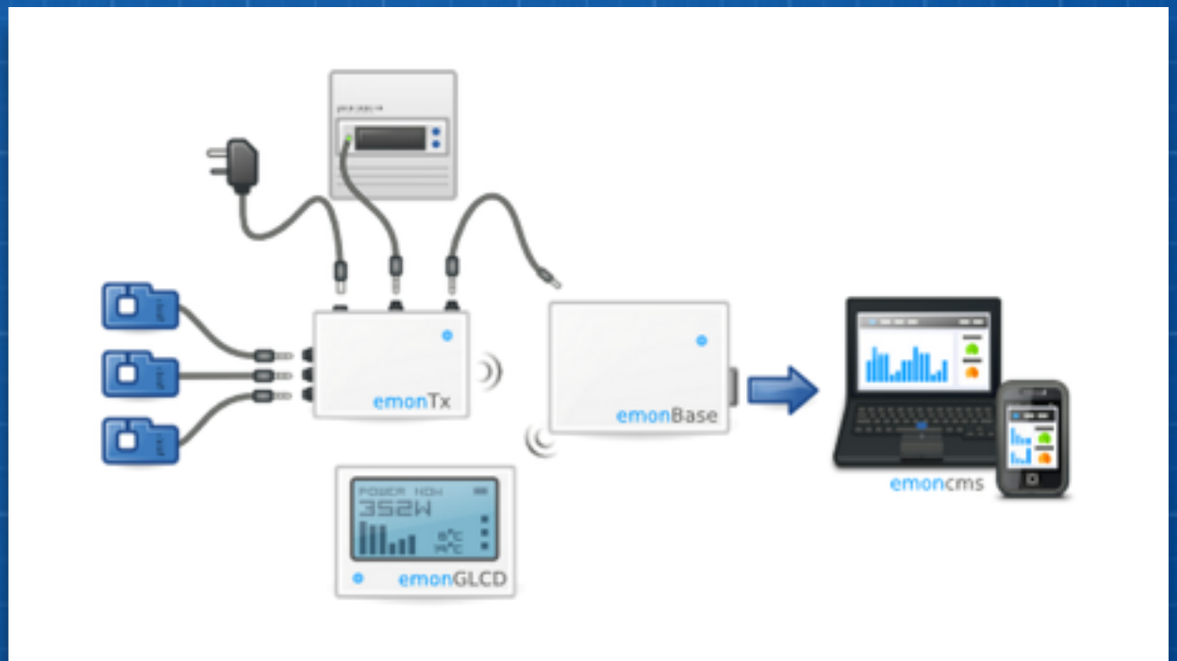
Porque é Possível e Divertido!

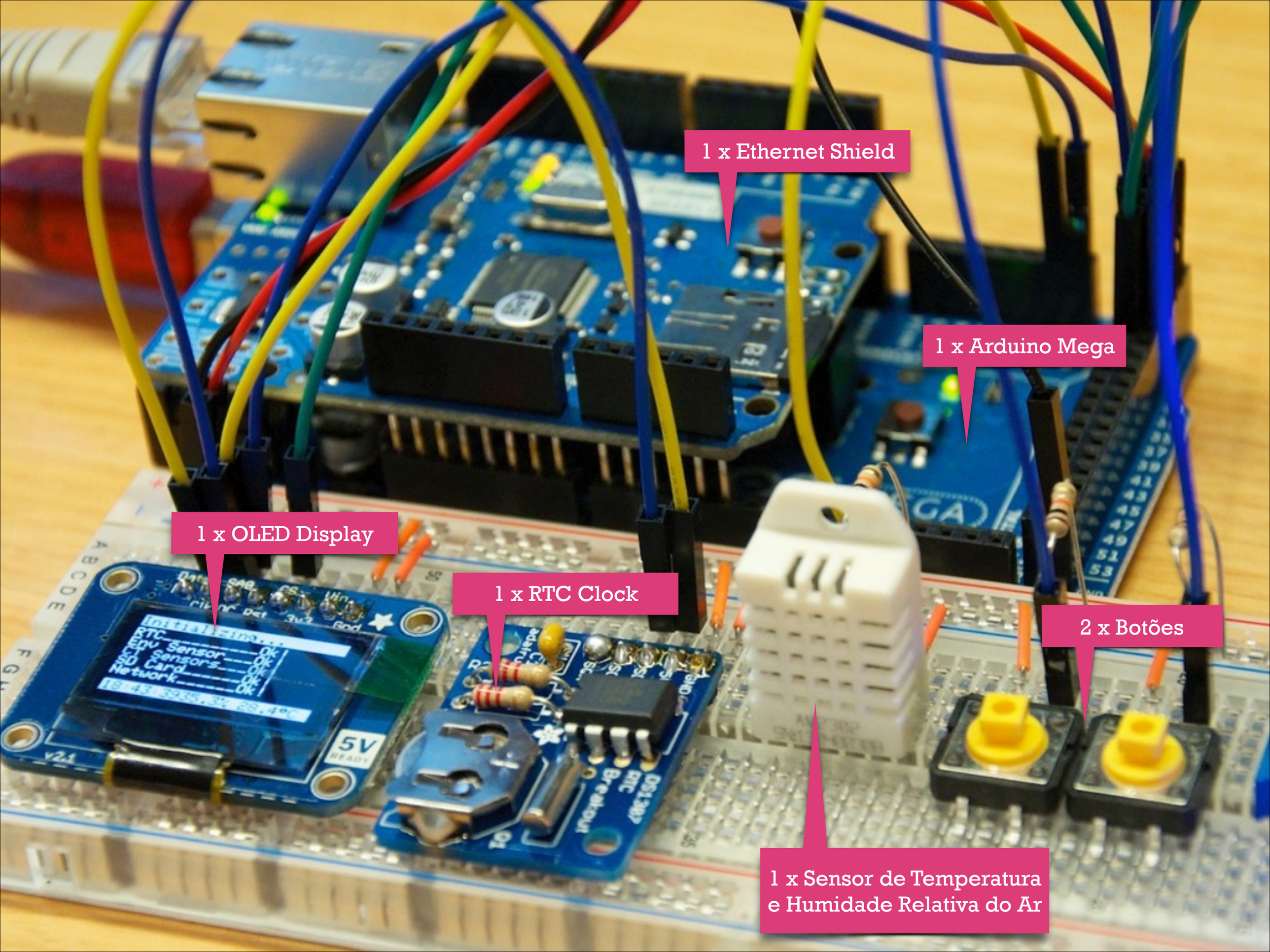
Campus Party 2014 • 30/01/2014 • São Paulo, Brasil • By Manoel Lemos • manoel@lemos.net • fazedores.com • manoellemos.com

OpenEnergyMonitor

http://openenergymonitor.org/






1 x Ethernet Shield
1 x Arduino Mega
1 x OLED Display
1 x RTC Clock
2 x Botões
1 x Sensor de Temperatura
e Humidade Relativa do Ar

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 x I

Potência
Watts
(W)

Tensão
Volts
(V)

Corrente
Ampères
(A)

Um Pouco de Teoria

P = U x 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!

(Watt-hora)

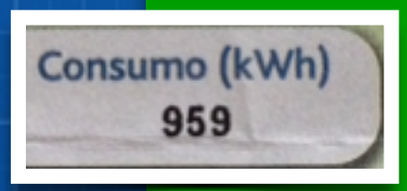
Quantidade de energia necessária para alimentar uma carga de **1 watt** pelo período de **1 hora**.

Como:

1 hora = 3600 segundos

Temos:

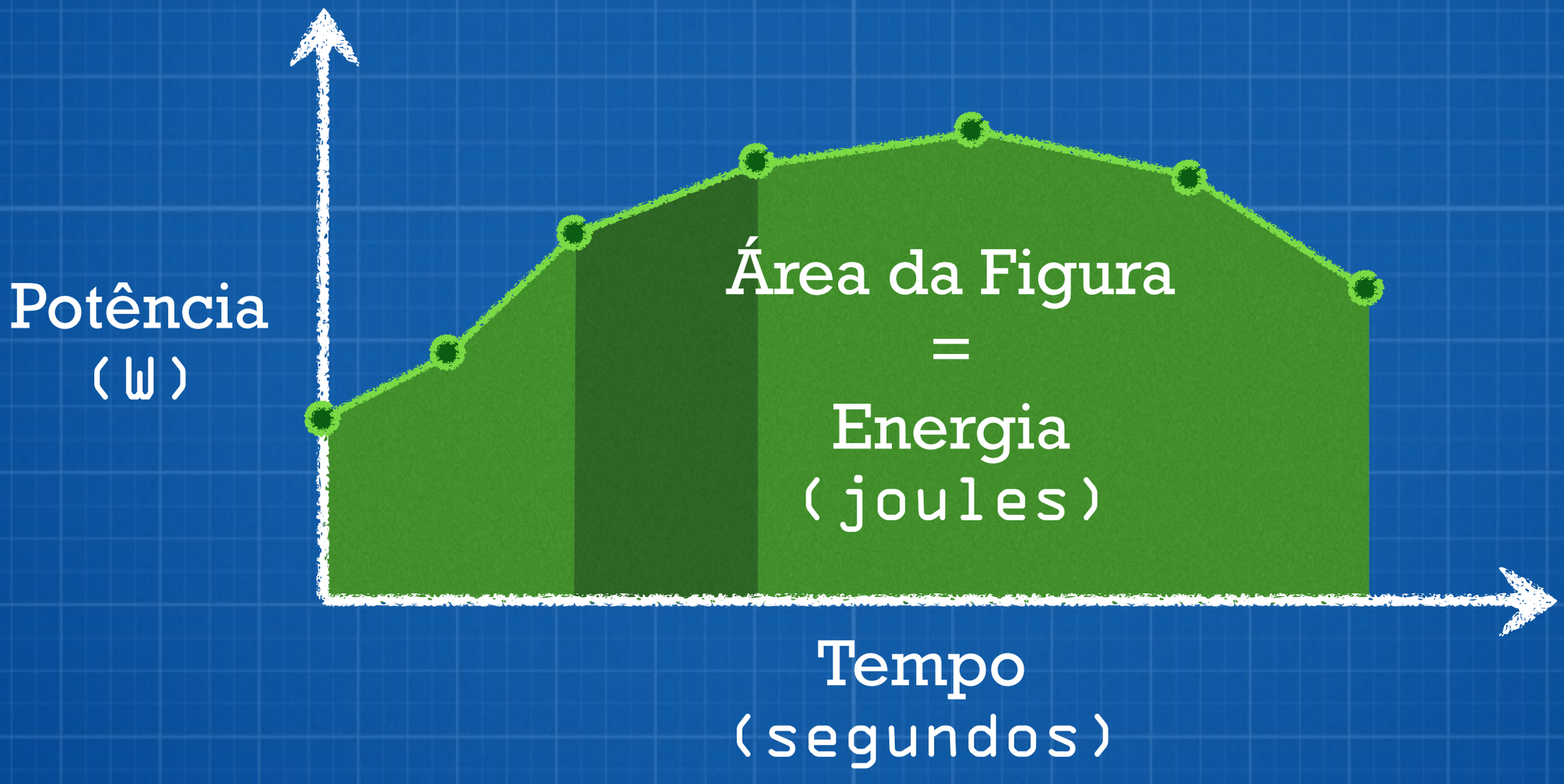
1 watt-hora
=
1 watt x 3600 segundos
=
1 (joule / segundos) x 3600 segundos
=
3600 joules



(kWh ou Quilowatt-hora)

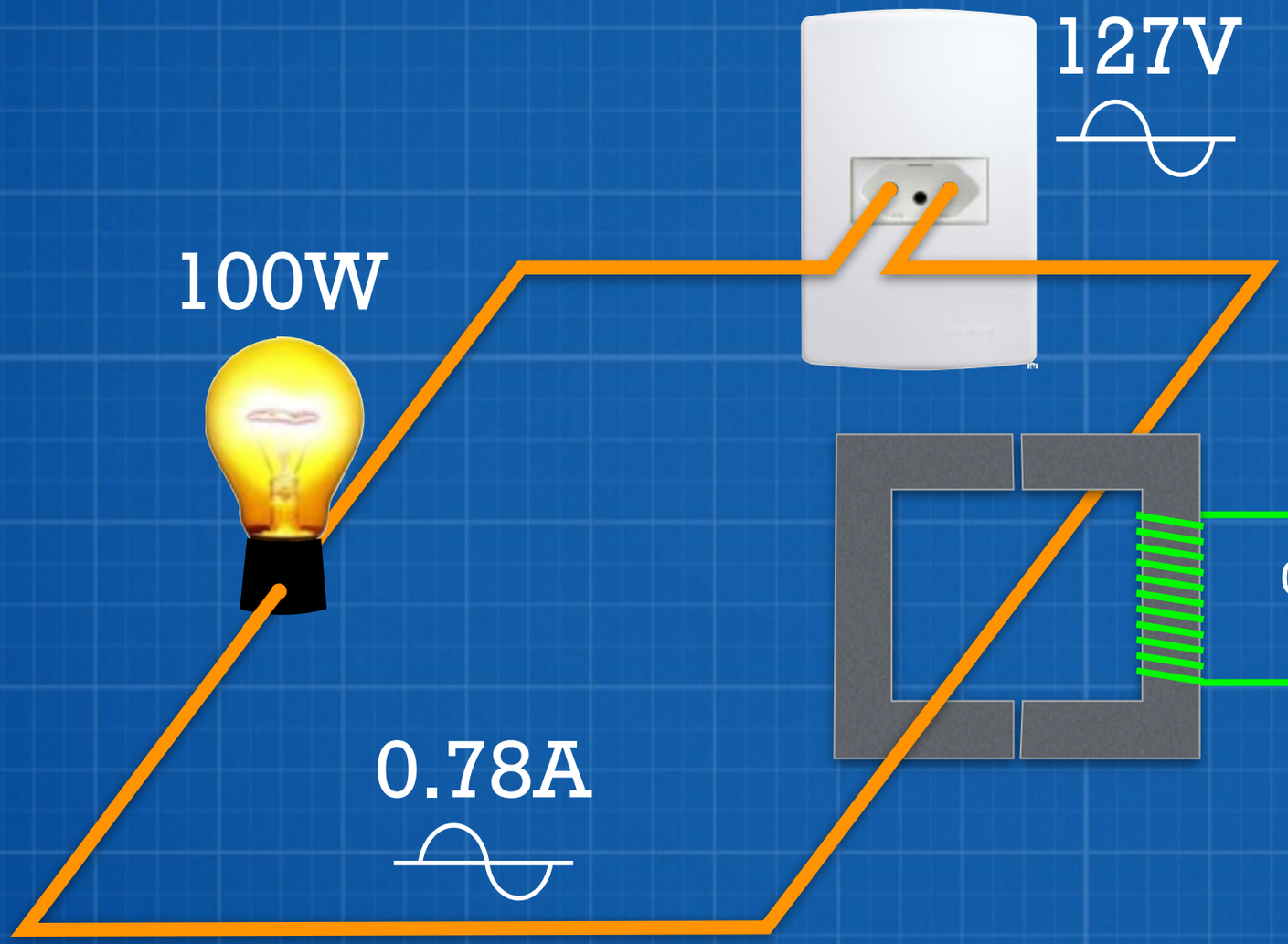
É por "**energia**" que pagamos na conta de luz.
3600000 joules

Calculando a Energia a partir de medidas da Potência

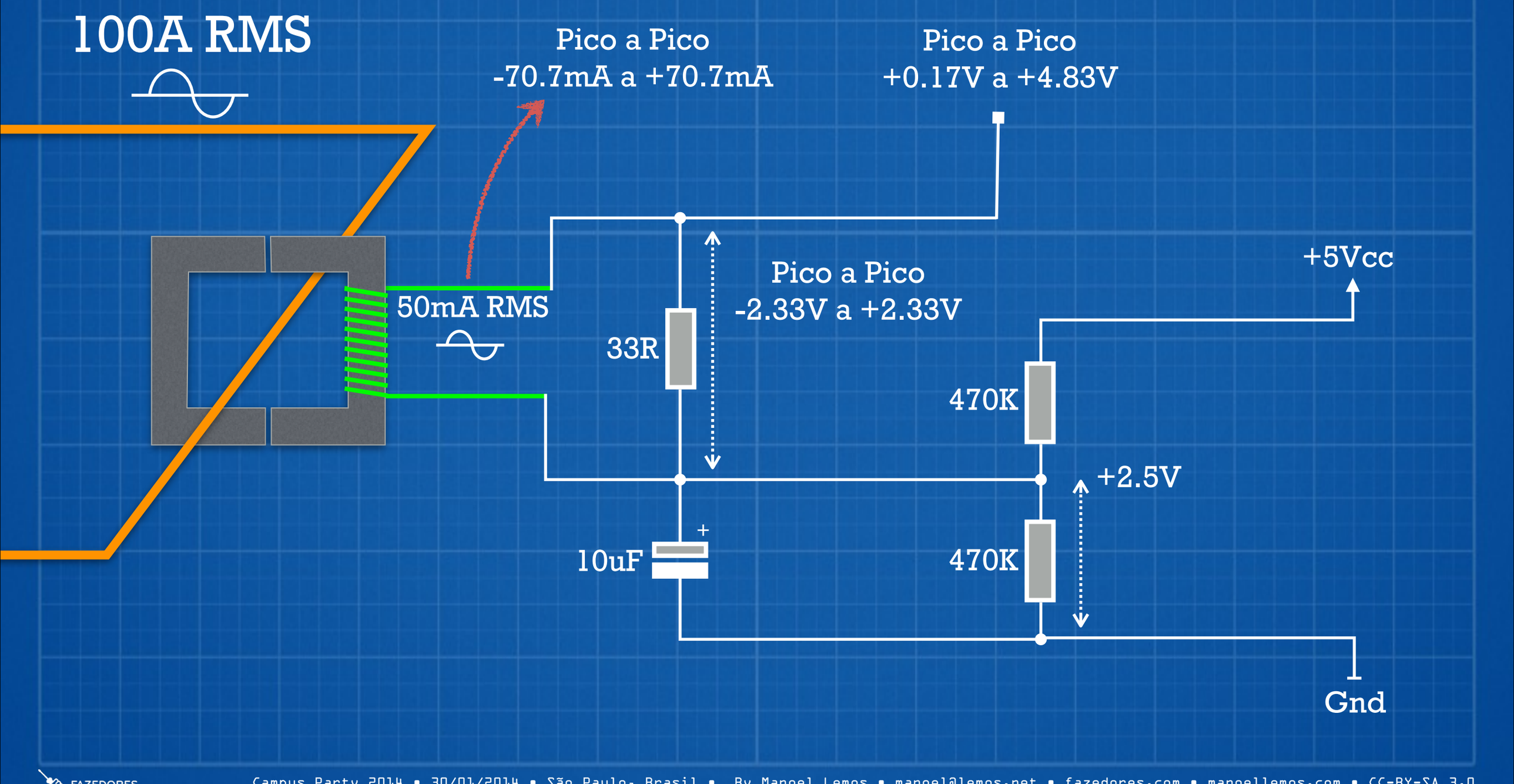


Um Sensor para Corrente Elétrica (AC)

Um Sensor para Corrente Elétrica (AC)

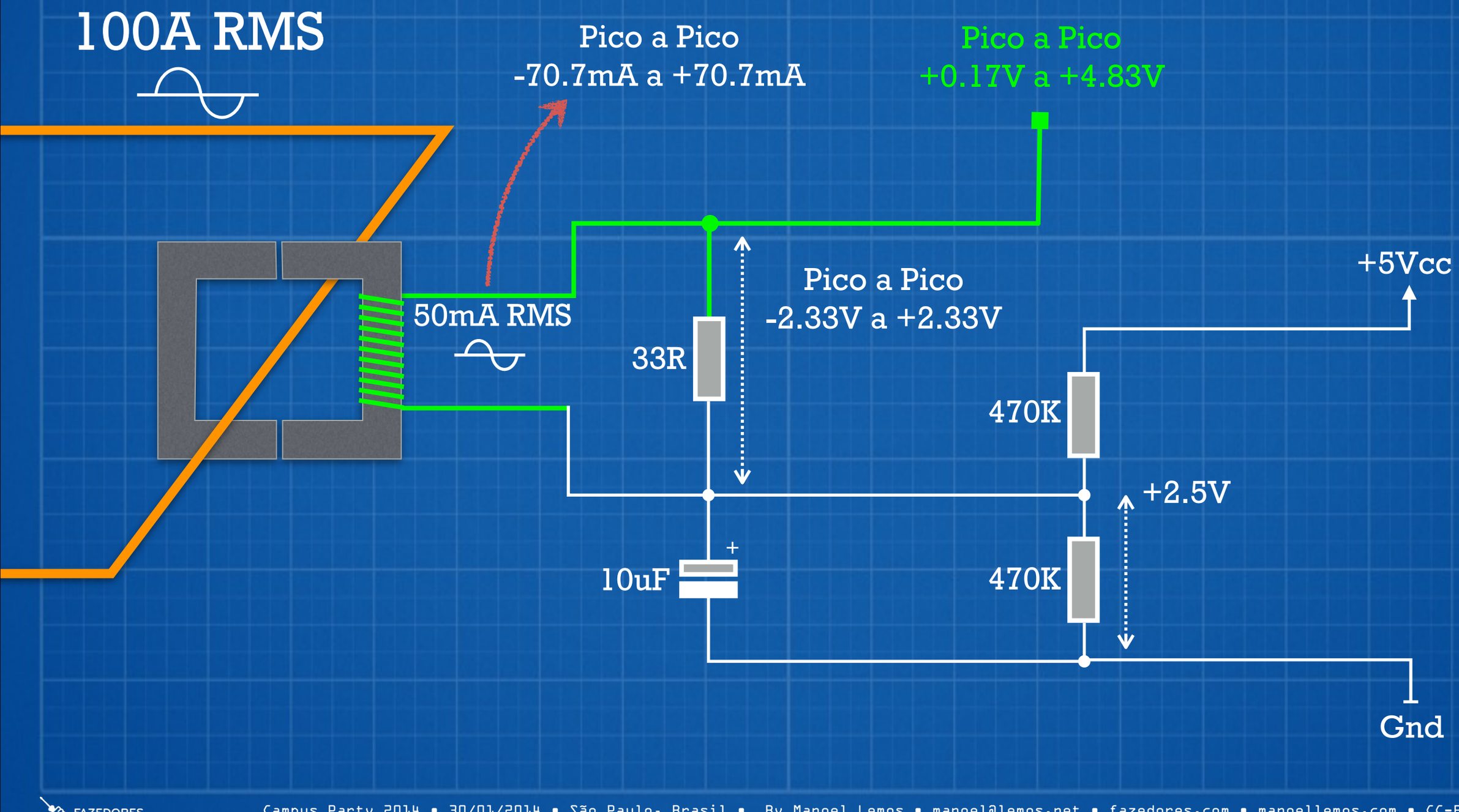


Interfaceando com o Arduino

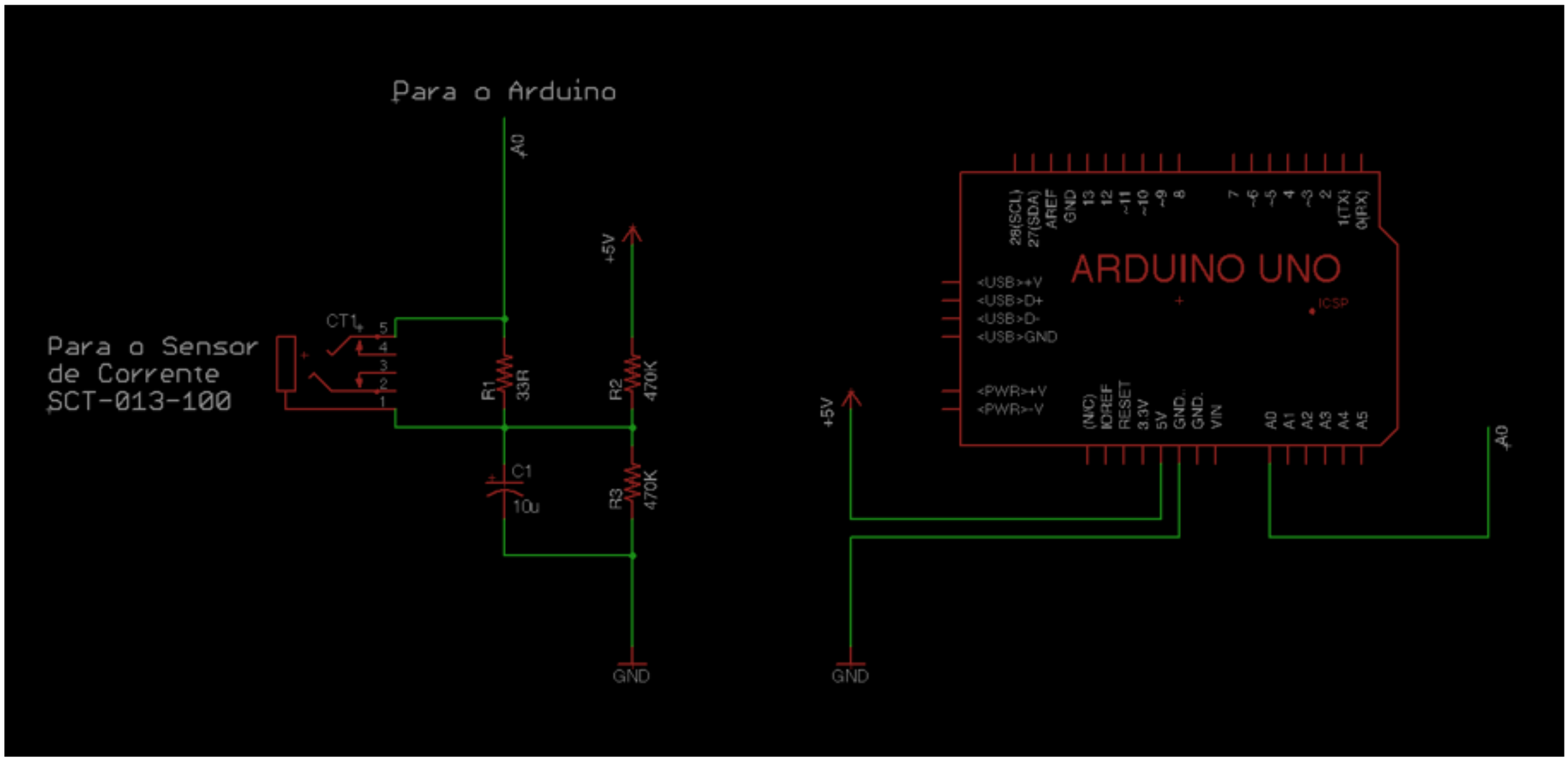


 By Manoel Lemos • manoel@lemos.net • fazedores.com • manoellemos.com

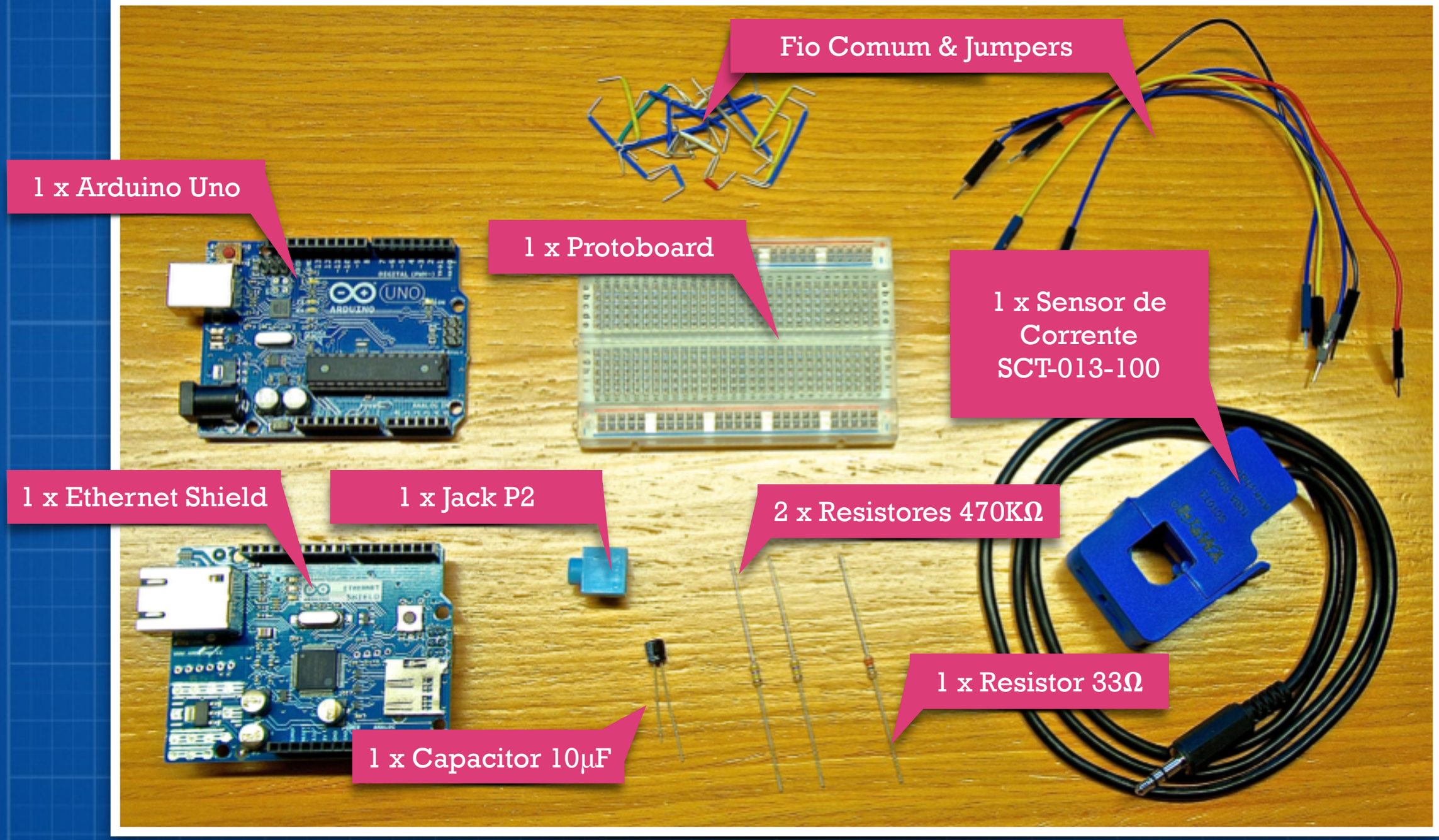
Interfaceando com o Arduino



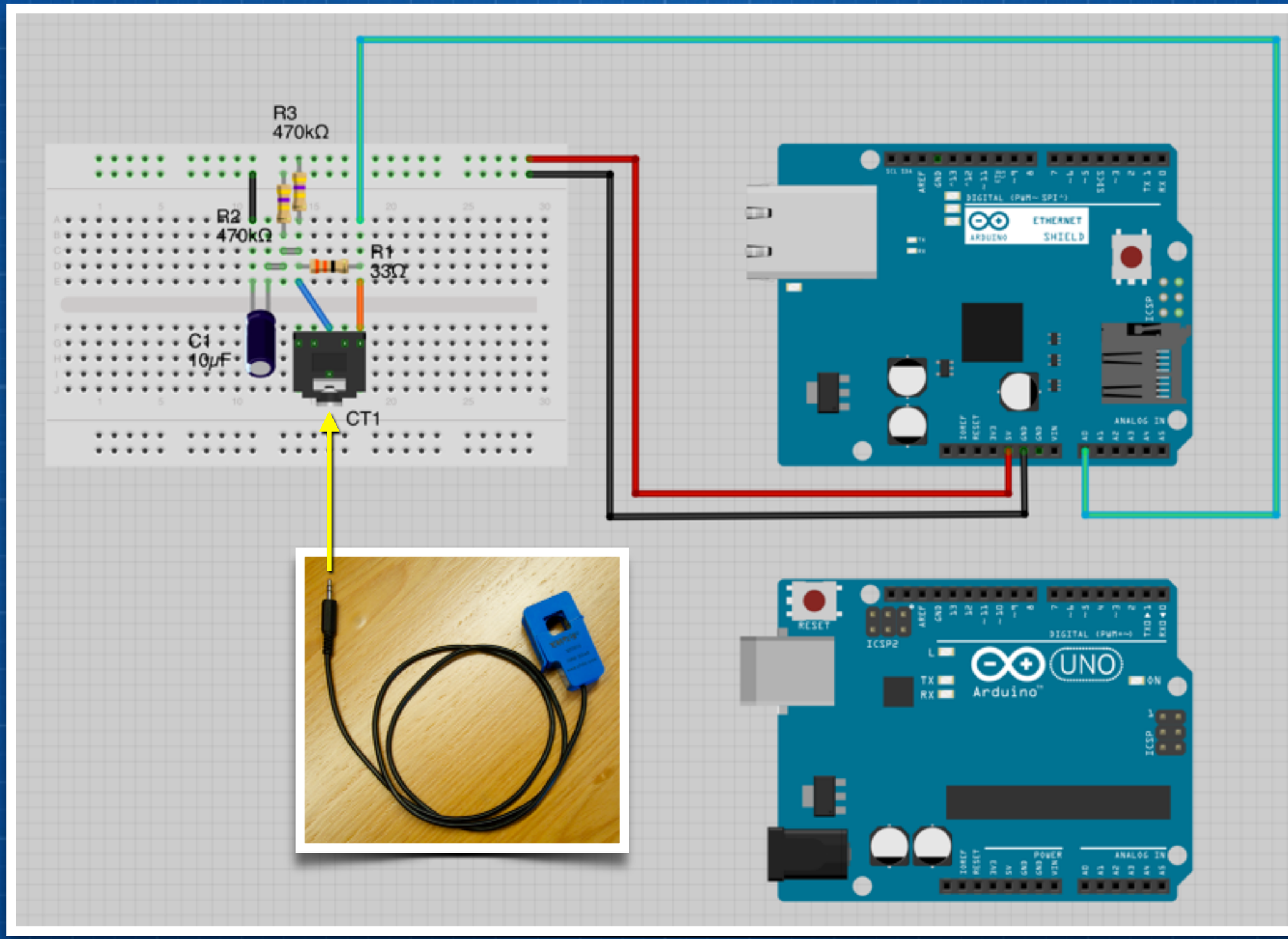
1. Diagrama Elétrico



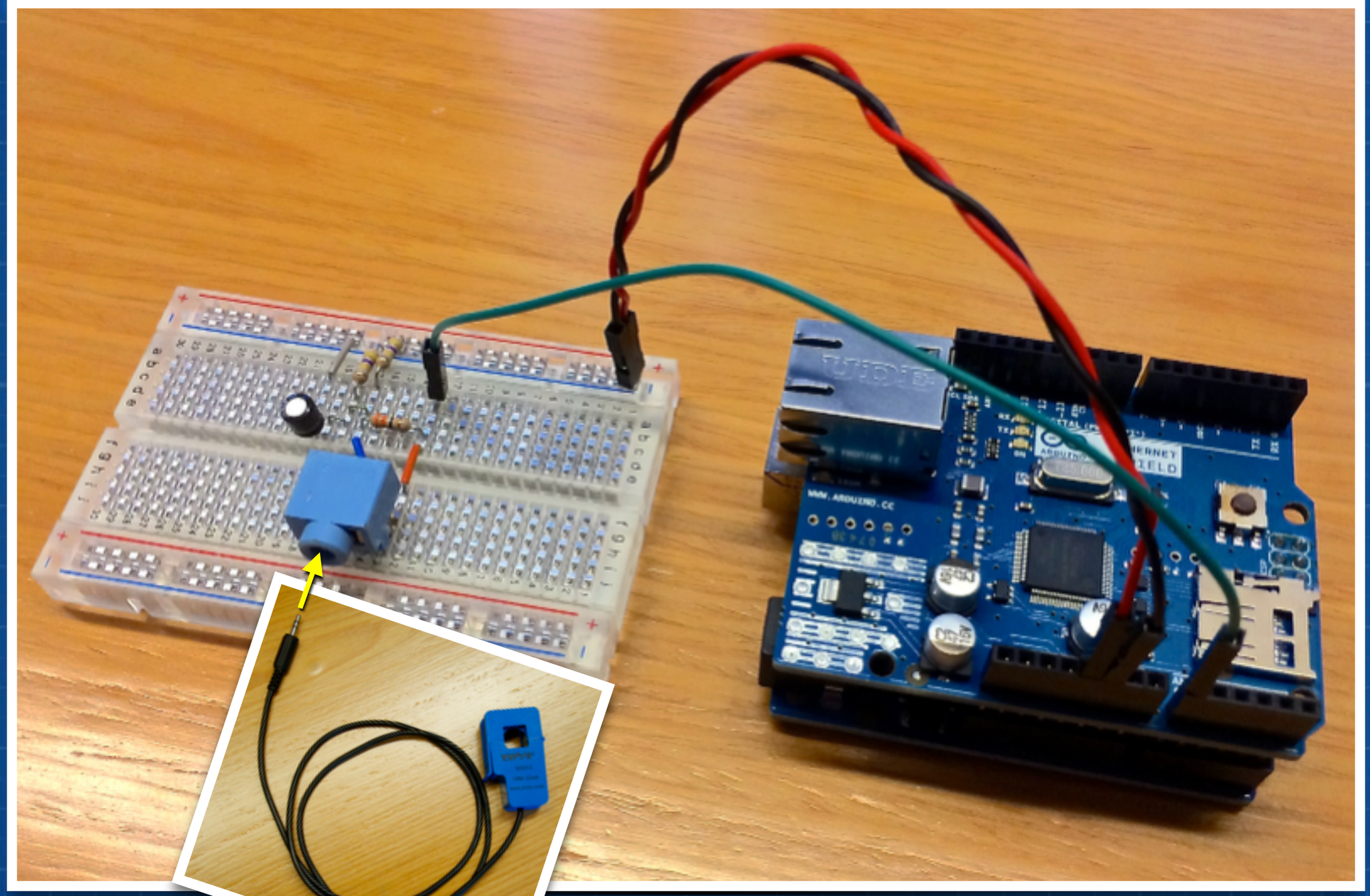
2. Peças


1 x Arduino Uno
Fio Comum & Jumpers
1 x Protoboard
1 x Sensor de Corrente SCT-013-100
1 x Ethernet Shield
1 x Jack P2
2 x Resistores 470KΩ
1 x Capacitor 10µF
1 x Resistor 33Ω

3. Layout


R3
470kΩ
R2
470kΩ
R1
33Ω
C1
10µF
CT1
ETHERNET
SHIELD
ARDUINO
UNO
Arduino
DIGITAL (PWM~ SPI)
ANALOG IN
POWER
RESET
ICSP
ICSP2

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
// funçoes

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 conexao
    Serial.println("Dados enviados!");
  } else {
    Serial.println("Falha na conexão.");
  }
}
```

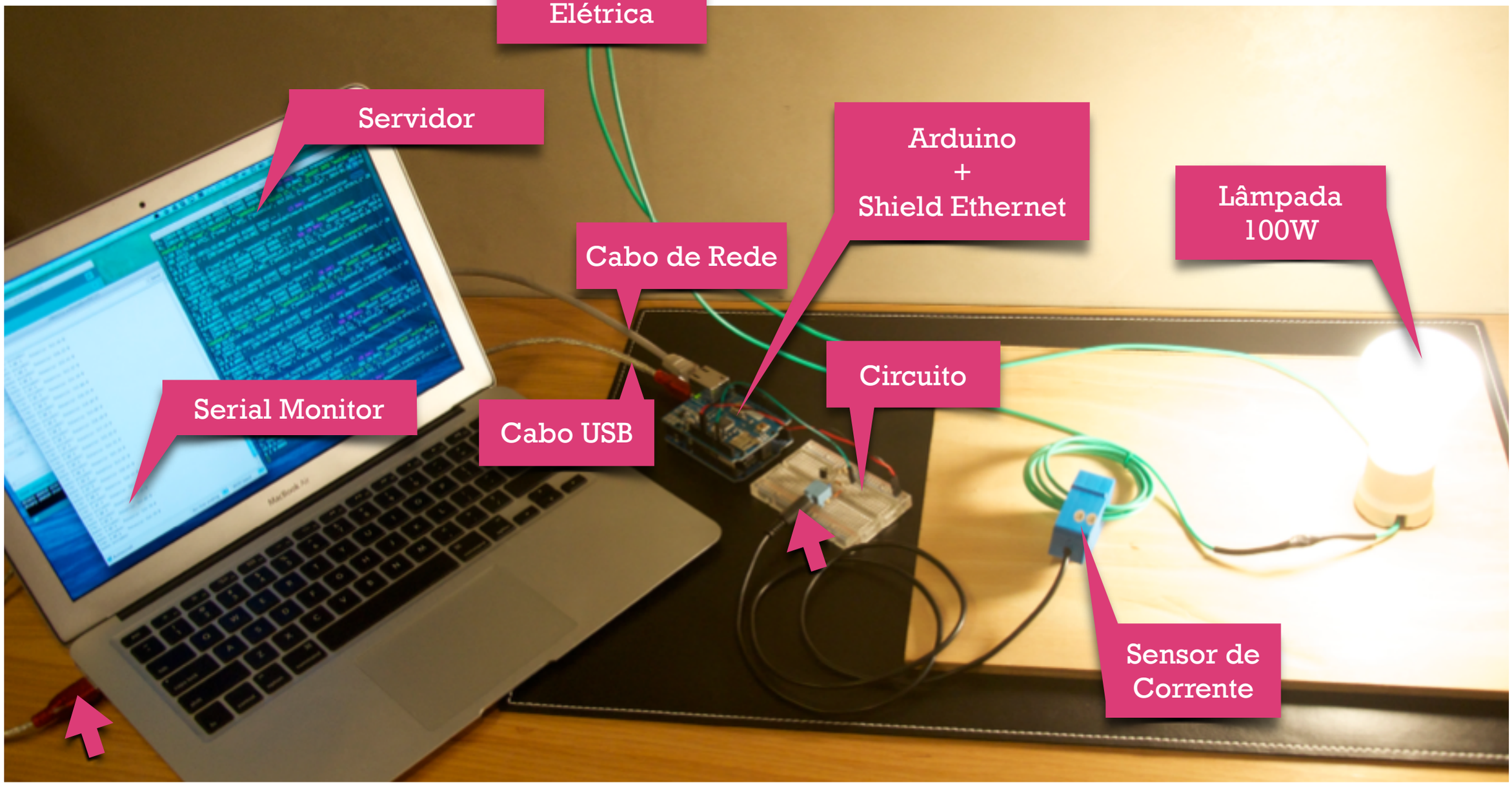
Servidor : GET /medida

```ruby
# 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
```

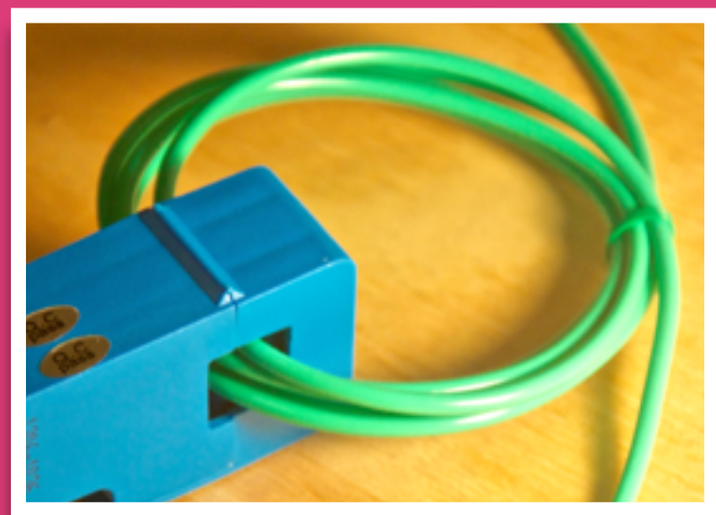
Servidor : GET /dashboard

```ruby
# 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



Macete

A corrente é multiplicada pelo número de voltas.

Serial Monitor – /dev/cu.usbmodem40111

UNREGISTERED

Serial Monitor – /dev/cu.usbmodem40111

```
Vamos lá Campus Party Brasil 2014!!
Inicializando Ethernet com DHCP...feito!
Endereçoo IP: 10.0.1.121
Iniciando medições:

Irms: 0.13 A  Potencia: 16.74 W
Dados enviados!
Irms: 0.08 A  Potencia: 10.06 W
Dados enviados!
Irms: 0.08 A  Potencia: 10.75 W
Dados enviados!
Irms: 0.08 A  Potencia: 10.10 W
Dados enviados!
Irms: 0.08 A  Potencia: 10.59 W
Dados enviados!
Irms: 4.05 A  Potencia: 513.87 W
Dados enviados!
Irms: 4.06 A  Potencia: 515.89 W
Dados enviados!
Irms: 4.07 A  Potencia: 516.69 W
Dados enviados!
Irms: 4.05 A  Potencia: 514.10 W
Dados enviados!
Irms: 4.06 A  Potencia: 515.87 W
Dados enviados!
Irms: 4.06 A  Potencia: 515.45 W
Dados enviados!
Irms: 4.05 A  Potencia: 514.36 W
Dados enviados!

```

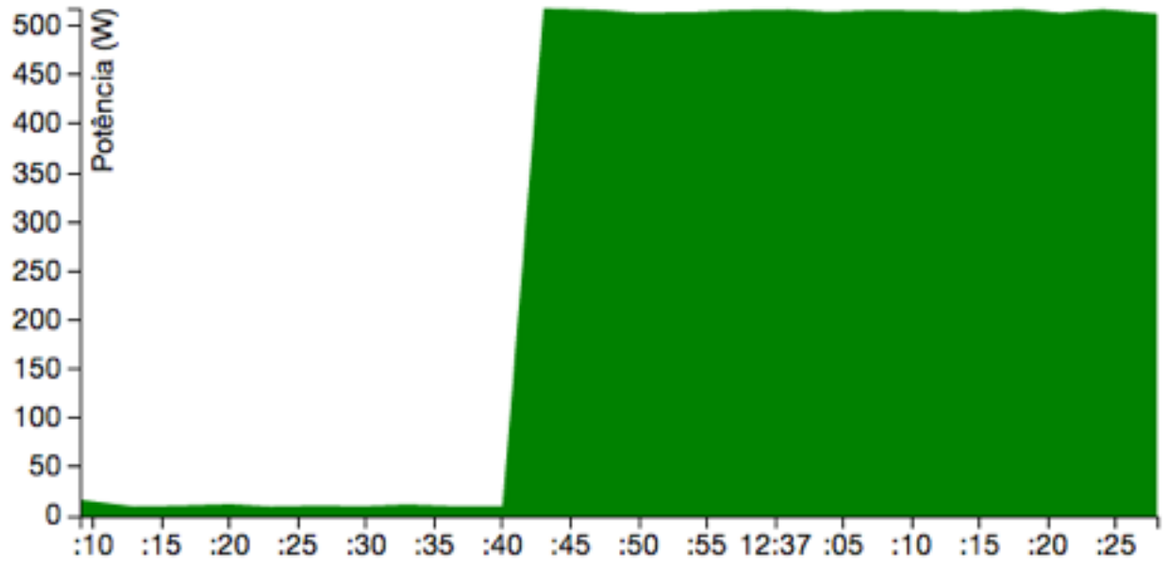
Line 2, Column 1

Campus Party

10.0.1.108:4000/dashbo...

Campus Party Brasil 2014 · Medidor de Consumo de Energia Conectado a Nuvem com Arduino

Consumo Total: 0.0066856374443750O1 kWh



Últimas Medidas

Irms (A)	Potencia (W)	Data / Hora
4.03	512.36	2014-01-21 00:37:28 UTC
4.07	516.98	2014-01-21 00:37:24 UTC
4.04	513.1	2014-01-21 00:37:21 UTC
4.07	517.03	2014-01-21 00:37:18 UTC
4.05	514.36	2014-01-21 00:37:14 UTC
4.06	515.45	2014-01-21 00:37:11 UTC
4.06	515.87	2014-01-21 00:37:07 UTC
4.05	514.1	2014-01-21 00:37:04 UTC
4.07	516.69	2014-01-21 00:37:01 UTC

Incrementando...

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

Crie, Construa e Compartilhe

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

FAZEDORES.com

CRIE • CONSTRUA • COMPARTILHE

Campus Party 2014 • 30/01/2014 • São Paulo, Brasil • By Manoel Lemos • manoel@lemos.net • fazedores.com • manoellemos.com •

Valeu!