

MNPEF
Mestrado Nacional
Profissional em
Ensino de Física



TUTORIAL PARA O DESENVOLVIMENTO DE UM CALORÍMETRO UTILIZANDO UMA PLATAFORMA DE PROTOTIPAGEM ELETRÔNICA DE HARDWARE LIVRE

Maringá - 2016

PREFÁCIO

Esse tutorial visa desenvolver um calorímetro utilizando uma placa eletrônica de hardware livre. O guia é estrategicamente dividido em oito partes para que o professor permita que seus alunos desenvolvam o produto, apenas orientando no processo de montagem.

Por meio do guia dos procedimentos didáticos o professor pode explorar vários conceitos da Física presentes nas etapas de construção do tutorial, despertando assim a curiosidade e interesse dos alunos na disciplina de Física.

Materiais

Os materiais utilizados para projeto e testes foram 7 kits de Arduino básico (*Robocore*[®]), além de alguns componentes específicos não contemplados nos kits. Os materiais contemplados nos kits estão descritos na tabela 1 e os materiais obtidos separadamente estão descritos na tabela 2.

Tabela - Kit Básico V7 Robocore

Componentes	Unidade
Arduino Uno Ver 3	1
Protoboard 840	1
Cabo USB	1
Sensor de Temperatura LM35	1
Sensor de Luminosidade (LDR 5mm)	1
Potenciômetro 10k	1
Barra Gráfica de LEDs	1
Display de 7 Segmentos	1
Circuito integrado 4511	1
LED RGB	1
Chave Momentânea (PushButton)	4
LEDs Amarelos	5
LEDs Verdes	5
LEDs Vermelhos	5
LED alto brilho	1
Resistores 300	15
Resistores 10k	5
Buzzer	1
Display de LCD 16x2 com backlight (com conector já soldado!)	1
fios Jumper Premium M/M de 20 cm	20
fios Jumper Premium M/M de 10 cm	10
Caixa Organizadora RC-Black	1

Componentes	Unidade
Módulo Relé	01
Sensor de Corrente	01
Sensor de Tensão (upgrade)	01
DS18B20	01
Transistor TIP 120	01
Diodo 1n4148	01
Motor 5V	01

Guia para procedimentos Didáticos

Este tutorial é uma ferramenta de ensino/aprendizagem que tem como objetivo orientar a execução do projeto, onde os alunos poderão trabalhar no seu desenvolvimento e o professor orientá-los.

Distribua o kit básico do Arduino para cada equipe e os componentes adicionais do projeto como constam nos tutoriais, orientando sobre os cuidados com o kit, é interessante também que o professor distribua o formulário de conceitos de eletricidade e eletrônica, que se apresenta logo a seguir.

O tutorial 01 (todas as equipes recebem) ensina a utilizar o ambiente de desenvolvimento do Arduino (IDE), a utilização de suas ferramentas e a compilação do projeto (*Skeetch*).

O tutorial 02 mostra como montar e programar um relé (interruptor eletromecânico) que tem como finalidade ligar a resistência do calorímetro. O professor poderá explorar alguns conceitos físicos como corrente elétrica, campos magnéticos e circuitos elétricos.

O tutorial 03 é a montagem do sensor de corrente. Tem o objetivo de verificar a corrente elétrica que passará no resistor do calorímetro. Uma sugestão para teste é a medição da potência de uma lâmpada, ou outro equipamento disponível, como mostra o esquema de ligação.

O tutorial 04 demonstra a utilização de um LCD (Mostrador de Cristal Líquido) para visualização das informações. Esse LCD tem um controlador que se adapta em qualquer plataforma de prototipagem eletrônica. O interesse dos alunos pode ser explorado durante essa etapa ao comparar a tecnologia LCD e LED (Diodo Emissor de Luz) envolvendo conceitos de ótica.

O tutorial 05 trata sobre o software de aquisição de dados PLX-DAQ em conjunto com Excel 2010. Para testes foi sugerido um esquema de ligação, utilizando o sensor de temperatura LM35. Para análise e configuração das tabelas e gráficos, o professor poderá explorar com seus alunos a interdisciplinaridade com a área da Matemática (construção de gráficos).

O tutorial 06 trata sobre configuração e montagem do sensor de temperatura que será utilizado no projeto (DS18B20). O professor poderá pedir para que os alunos verifiquem a temperatura da água fria ou quente e comparem

com um termômetro de mercúrio de laboratório (verificar a precisão). Além disso, existe a possibilidade de explorar o conceito de equilíbrio térmico.

O tutorial 07 refere-se a montagem de um *push button*, que tem a finalidade de ligar o relé do calorímetro. Nessa etapa, o professor pode explorar alguns conceitos de circuitos elétricos.

O tutorial 08 demonstra a montagem do motor 9V. Esse motor tem a finalidade de agitar o interior do calorímetro deixando a temperatura constante em todos os pontos. Nesse momento, o professor pode explorar o funcionamento de um motor CC (corrente contínua) e conceitos envolvidos.

Depois dos testes dos sensores e dos componentes, o professor deverá desafiar seus alunos a montarem o Arduino com o calorímetro e estimular a discussão com objetivo de solucionar o desafio. O ideal é que haja apenas um projeto composto por todos os grupos de alunos. É possível que o projeto final seja organizado em uma caixa patola ou no própria protoboard.

Formulário de Conceitos de Eletricidade e Eletrônica

Eletricidade: É a parte da Física que estuda fenômenos associados às cargas elétricas.

Eletrônica: É a ciência que estuda a forma de controlar a energia elétrica por meios elétricos nos quais os elétrons têm papel fundamental

Carga elétrica: O átomo é constituído por muitas partículas as três fundamentais são prótons, elétrons e nêutrons. Os prótons carga positiva e elétron carga negativa de mesmo módulo (valor), enquanto os nêutrons são eletricamente neutros. O valor da carga elétrica do próton e neutro é de: $1,602 \cdot 10^{-19}$ C (Coulomb)

Tensão: A tensão elétrica ou ddp (diferença de potencial) é uma indicação de quanta energia é envolvida na movimentação de cargas elétricas, por exemplo os elétrons, entre dois pontos. A unidade de medida é volt (V).

Corrente Elétrica: É o fluxo de partículas portadores de carga elétrica (por exemplo: prótons e nêutrons), ou seja é o deslocamento de cargas dentro de um condutor (exemplo: fio de cobre), quando existe uma *ddp* elétrico entre as extremidades. A unidade é o Ampère (A).

Campo Elétrico: É um campo de força provocado pela ação de cargas elétricas, esse campo está associado ao espaço onde o valor mensurável da sua intensidade se designa intensidade do campo dada por força (Newton) por unidade de carga (Coulomb) (N/C).

Campo Magnético: É um campo de força provocado pela ação do movimento de cargas elétricas ou pelos materiais magnéticos (ímãs). Unidade é o Testa (T).

Resistência Elétrica: É a capacidade de um corpo qualquer se opor a passagem de corrente elétrica mesmo quando existe um diferença de potencial aplicada. A unidade é ohms (Ω).

Energia Elétrica: É uma forma de energia baseada na geração de diferença de potencial entre dois pontos, que permite estabelecer uma corrente elétrica. A unidade é Joule (J).

Potência Elétrica: Em sistemas elétricos é o produto da diferença de potencial (ddp) entre os terminais e a corrente que passa através de um dispositivo. Sua unidade é Watts (W).

Potenciômetro: É um componente eletrônico que possui resistência elétrica ajustável.

Semicondutor: É um sólido geralmente cristalino de condutividade intermediária entre condutores e isolantes.

Diodo: É um componente eletrônico composto geralmente por um cristal semicondutor de dois terminais, que conduz corrente elétrica preferivelmente em um só sentido, bloqueando a sua passagem no sentido oposto.

LED (Light Emitting Diode): É um diodo emissor de Luz, é utilizado em locais e instrumentos onde se torna mais conveniente em lugar da lâmpada incandescente ou fluorescente.

LCD (Liquid Crystal Display): É um painel utilizado para mostrar ou exibir informações por via eletrônica, como texto ou imagens.

Transistor: É um componente eletrônico muito utilizado como interruptores, amplificadores de sinais elétricos o termo provém (Transfer Resistor) – Resistência de Transferência.

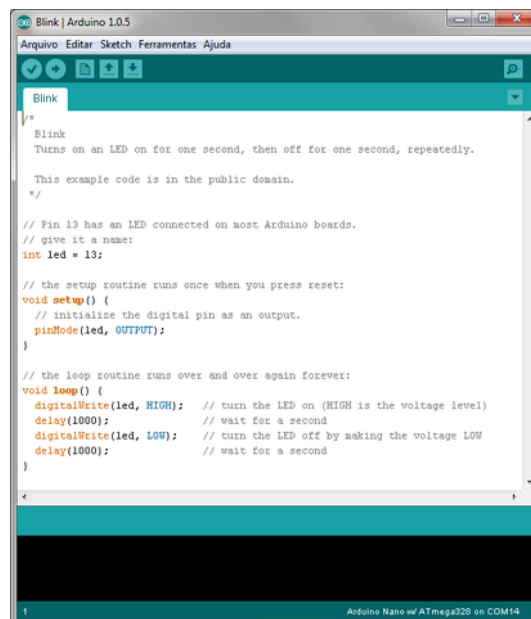
Tutorial 01 - Arduino

Nesse tutorial, precisaremos utilizar a placa Arduino e o cabo USB, que contém no kit básico. Caso ainda não tenha o software do ambiente de desenvolvimento, obtenha a última versão na página de download (<https://www.arduino.cc/en/Main/Software>).



Figura: Arduino e Cabo USB

Quando o download terminar, descompacte o arquivo baixado, garantindo que a estrutura das pastas seja mantida. Dê um duplo clique para abri-la, procure o arquivo executável “Arduino.exe”.

The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0.5". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The main text area contains the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

The status bar at the bottom indicates "Arduino Nano w/ ATmega328 on COM14".

Figura: Ambiente de desenvolvimento

É importante ressaltar que o Arduino obtém energia automaticamente seja da conexão USB ou de uma fonte de alimentação externa. Para verificar se está funcionando, conecte o Arduino no seu computador. O LED verde de energia (marcado como PWR) deve acender.



Figura: Indicativo do Arduino ligado corretamente

Selecione o dispositivo serial correspondente à placa do Arduino no menu **Ferramentas > Porta Serial**. É provável que ele seja COM3 (COM1 e COM2 são normalmente reservados para portas seriais nativas, e não por USB).

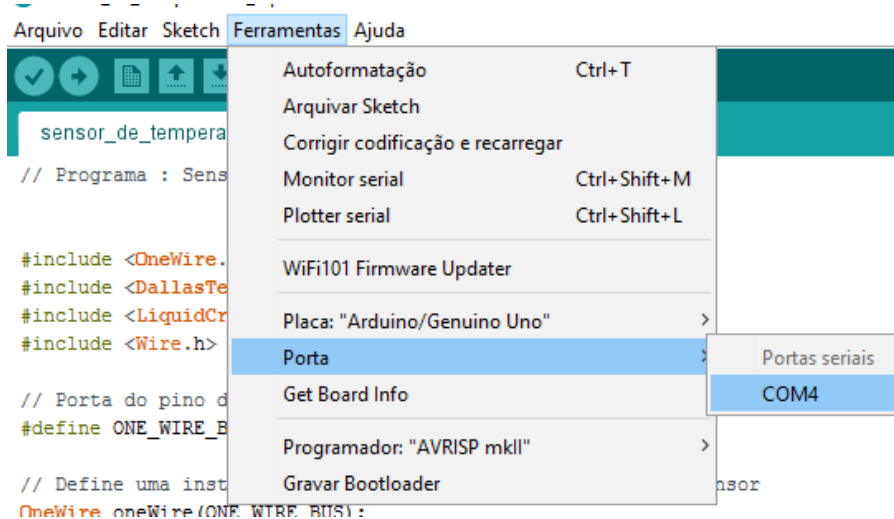
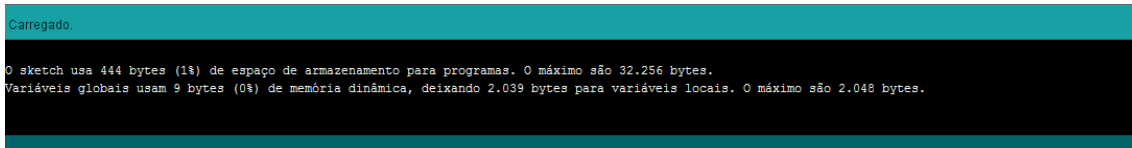


Figura: Selecionando a porta serial

Para fazer o upload do código na placa do Arduino, “Carregar” clique na opção upload ambiente de desenvolvimento. Espere alguns segundos. Os LED’s indicadores RX e TX devem estar piscando. Se o upload for bem-sucedido a mensagem “Carregado” irá aparecer no status do programa.



```
Carregado.  
O sketch usa 444 bytes (1%) de espaço de armazenamento para programas. O máximo são 32.256 bytes.  
Variáveis globais usam 9 bytes (0%) de memória dinâmica, deixando 2.039 bytes para variáveis locais. O máximo são 2.048 bytes.
```

Figura: Status do programa

Tutorial 02 – Relé

Como funciona um Relé?

O relé é um interruptor acionado eletricamente. Muitos relés usam um eletroímã para operar mecanicamente um interruptor, mas outros princípios operacionais são utilizados.

Esse componente simples consiste de uma bobina de fio enrolado em torno de um núcleo de ferro, onde a bobina é o principal componente do relé. É em torno dela que é gerado um campo eletromagnético quando energizado.

Este campo gera uma força capaz de movimentar um conjunto mecânico (armadura fixa) com contatos móveis alterando assim seu estado de normalmente aberto para fechado.

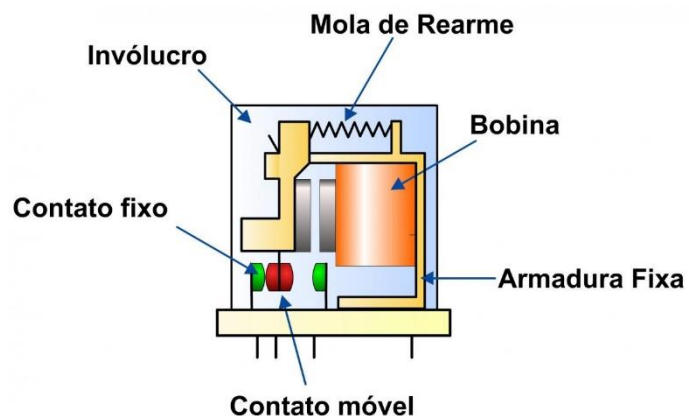


Figura: Parte do Relé eletromecânico
Fonte: (FINDERNET, 2016)

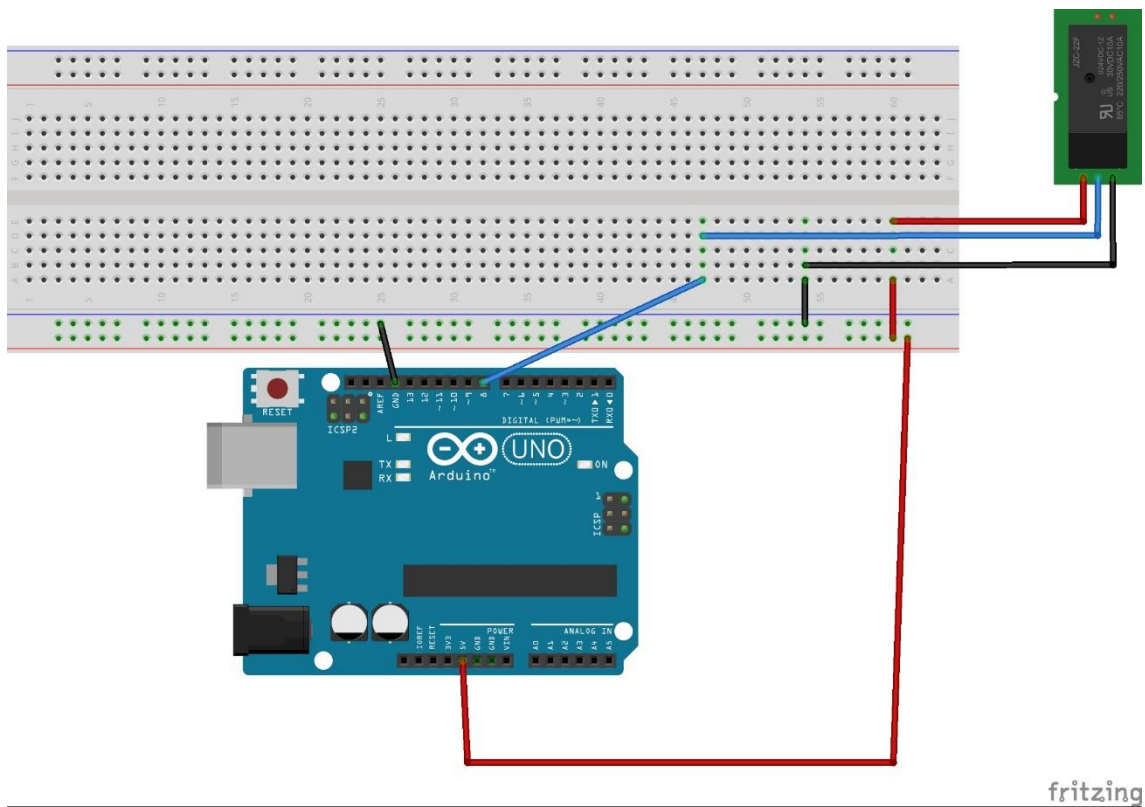
OBJETIVO:

- Testar a funcionalidade do Relé.
- Ligar o resistor do calorímetro.

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimentos
Módulo Relé (disponibilizado pelo professor)	01	Ligar em C (Comum) e NA (Normalmente Aberto)
Fios de Cobre	1,5 m	Utilizar um alicate de corte
Jumpers	03	Ligar na porta digital 13 no GND e VCC
Protoboard	01	-
Arduino UNO Ver 03	01	-
Bateria	12 V	Verificar o positivo e o negativo da bateria, antes de conectar os fios.
Multímetro	01	Para teste utilize a escala de tensão contínua em 20V
Notebook ou Computador	01	Deve estar instalado o ambiente de desenvolvimento do Arduino

Esquemas de ligação



Código:

```
int rele = 13; // relé conectado na porta digital 13
void setup() { // definições do programa

    Serial.begin(9600); // ativa o monitor serial na tela do computador, 9600 é a
    velocidade de transferência

    pinMode(rele, OUTPUT); // Define a porta de saída, representando a porta digital
    13
}

void loop() // Loop do programa, execuções de tarefas.
{
    char tecla; // utilização de caracteres do teclado
    tecla = Serial.read(); // função da leitura das teclas

    if(tecla == '1') // condicional (se somente se digitar 1)
    {
        digitalWrite(rele, HIGH); // nível lógica ativado
    }

    if(tecla == 'A')
    {
        digitalWrite(rele, LOW); // nível lógico desativado
    }

    delay(1000); // tempo para execução do loop
}
```

Tutorial 02 – Sensor de Corrente

Como funciona um sensor de corrente elétrica

Os sensores de **campos magnéticos** são dispositivos muito utilizados e presentes em nosso dia-a-dia e que possuem diversas aplicações, desde aparelhos de consumo até máquinas industriais. Uma das maneiras mais elementares de se fazer o sensoriamento de campo magnético é por meio de uma **bobina**. Existem, contudo, dispositivos semicondutores próprios para a detecção de campos magnéticos, como é o caso do **Sensor Hall**.

O **Efeito Hall** é o que ocorre quando uma corrente num condutor tem sua trajetória desviada pela ação de um campo magnético. Com um formato apropriado, este desvio faz com que a **Tensão de Hall** seja gerada, e esta pode ser aproveitada por um circuito externo, e é o que sensores Hall fazem. A tensão de Hall pode ser medida por um circuito externo ou ainda ser utilizada para efeitos de sensoriamento, pois ela é proporcional à intensidade do campo que cria.

Esse efeito é observado em todos os materiais. Contudo, sua aplicação é eficaz somente em materiais onde a mobilidade eletrônica seja relativamente alta. Desse modo, as aplicações práticas do Efeito Hall só se tornam possíveis com o desenvolvimento de tecnologias em materiais semicondutores. (W. FOUNDATION)



Figura: Sensor de corrente ACS712
Fonte: (DIGITAL, 2016)

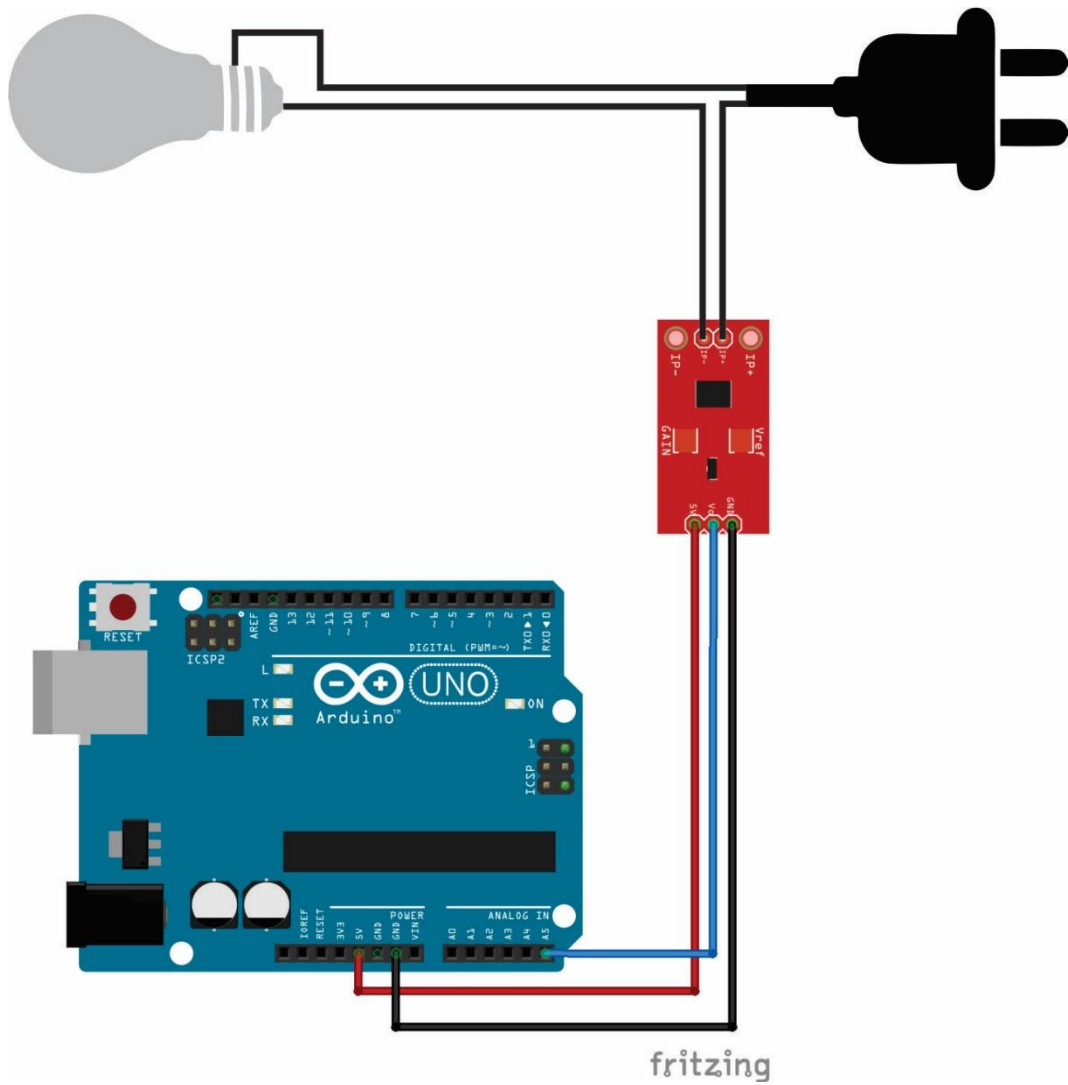
OBJETIVO:

- Testar o sensor de corrente em uma lâmpada incandescente.
- Comparar o valor da corrente com o multímetro.
- Ligar o sensor de corrente em série com o resistor do calorímetro.
- Verificar a corrente do resistor do calorímetro para calcular a potência dissipada.

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimentos
Arduino Uno Ver 3	01	
Sensor de Corrente	01	A contagem dos pinos é atrás do display
Jumpers	03	
Lâmpada	01	
Notebook ou computador	01	IDE instalado
Cabo USB	01	

Esquemas de Ligação



Código

```
float sensorValue = 0;
float currentValue = 0;
float voltsporUnidade = 0.004887586; // 5 /1023 é a razão
float corrente = 0;
int ACS712 = 5;

void setup(void)
{

for(int i=100; i>0; i--){
    sensorValue_aux = (analogRead(ACS712) -511); // le o sensor na pino
    analogico A0 e ajusta o valor lido ja que a saída do sensor é (1023)vcc/2 para
    corrente =0
    sensorValue += pow(sensorValue_aux,2); // somam os quadrados das
    leituras.
}

    sensorValue = (sqrt(sensorValue/ 100)) * voltsporUnidade; // finaliza o cálculo
    da méida quadratica e ajusta o valor lido para volts
    currentValue = (sensorValue/0.100);
    corrente = currentValue
    Serial.print ("Corrente:");
    Serial.println (corrente);
}
```

Tutorial 03 – LCD

Como funciona o LCD

Os cristais líquidos são substâncias que têm sua estrutura molecular alterada quando recebem corrente elétrica. Em seu estado normal, estas substâncias são transparentes, mas ao receberem uma carga elétrica tornam-se opacas, impedindo a passagem da luz.

O LCD (Liquid-crystal Display) – Mostrador de Cristal Líquido, que é utilizado no projeto é monocromático, ou seja, apresenta apenas dois estados, transparente e opaco. Para formar a tela de um monitor, uma fina camada de cristal líquido é colocada entre duas camadas de vidro. Estas finas placas possuem pequenos sulcos, isolados entre si, cada um com um eletrodo ligado a um transistor. Cada um destes sulcos representa um dos pontos da imagem. Este sanduíche por sua vez é colocado entre duas camadas de um elemento polarizador. Atrás desta tela é instalada uma fonte de luz, geralmente composta de lâmpadas fluorescentes (usadas por gerarem pouco calor) ou então LEDs, responsáveis pela iluminação da tela (MORIMOTO, 2007).

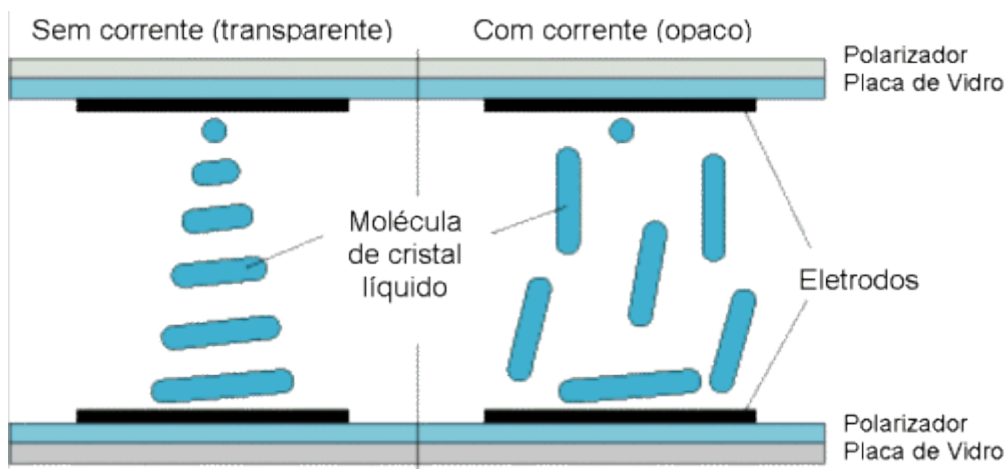


Figura: Mostrador de Cristal Líquido
Fonte: (HARDWARE, 2016)

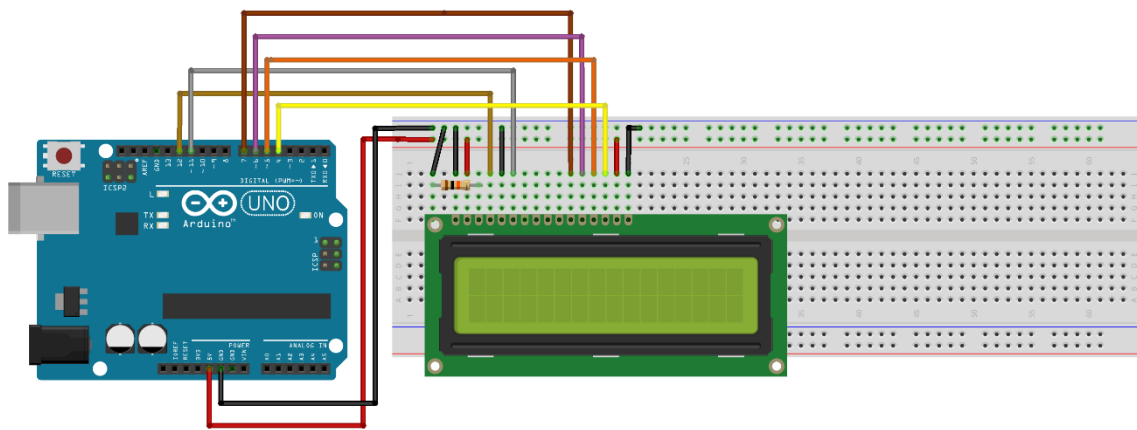
OBJETIVO:

- Verificar a funcionalidade do LCD.
- Desenvolver um mostrador digital da temperatura

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimentos
Arduino Uno Ver 3	01	
LCD 16x2	01	A contagem dos pinos é atrás do display
Jumpers	17	
Resistor 10k Ω	01	
Notebook ou computador	01	IDE instalado no computador ou notebook
Cabo USB	01	

Esquemas de Ligações



fritzing

Código

```
//Carrega a biblioteca LiquidCrystal
#include <LiquidCrystal.h>

//Define os pinos que serão utilizados para ligação ao display
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);

void setup()
{
  //Define o número de colunas e linhas do LCD
  lcd.begin(16, 2);
}

void loop()
{
  //Limpa a tela
  lcd.clear();
  //Posiciona o cursor na coluna 0, linha 0;
  lcd.setCursor(0, 0);
  //Envia o texto entre aspas para o LCD
  lcd.print("COLEGIO SESI");
  lcd.setCursor(0, 1);
  lcd.print(" 2 ANO");
  delay(1000);
}
```

Tutorial 4 – Sistema de Aquisição de Dados

Sistema de aquisição de dados

A aquisição de dados, pode ser definida como “medir informações de grandezas físicas em tempo real”. Para obter essas informações, precisa-se um software. O “PLX DAQ” em conjunto com o Excel 2010 pode fazer a aquisição de dados presentes na porta serial.

Deve-se efetuar o download no site <https://www.parallax.com/downloads/plx-daq>. Depois de descompactar o arquivo, instale.

Assim, irá se abrir a interface do PLX-DAQ, podendo configurar a porta serial e o status da comunicação.

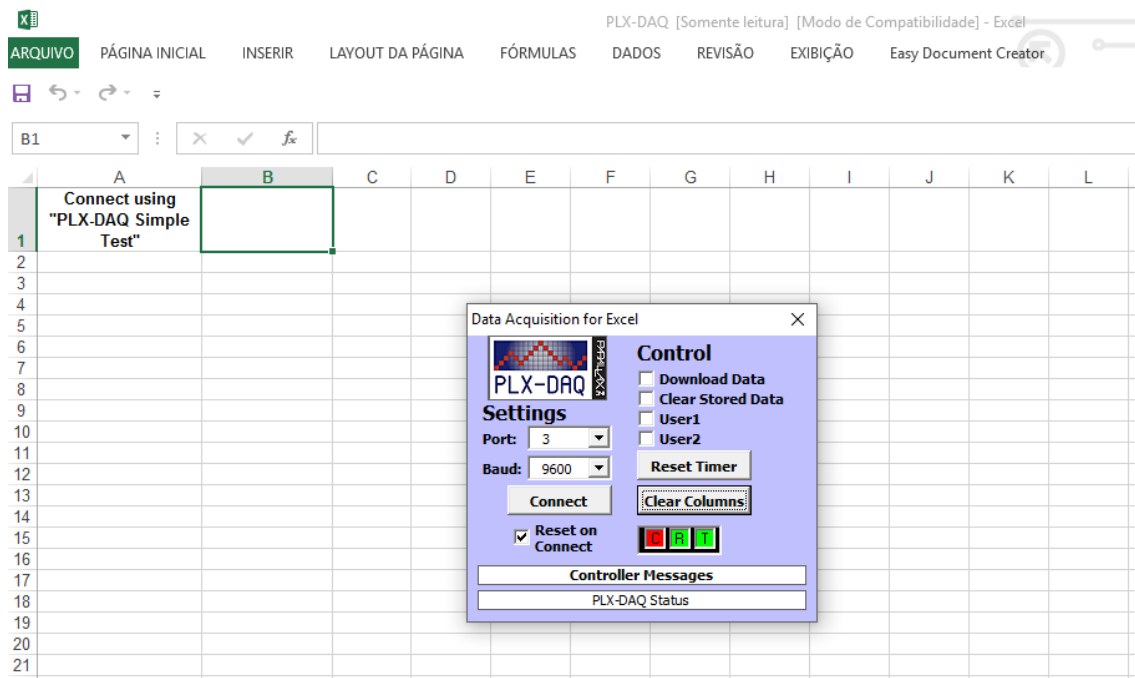


Figura: PLX DAQ o Excel 2010

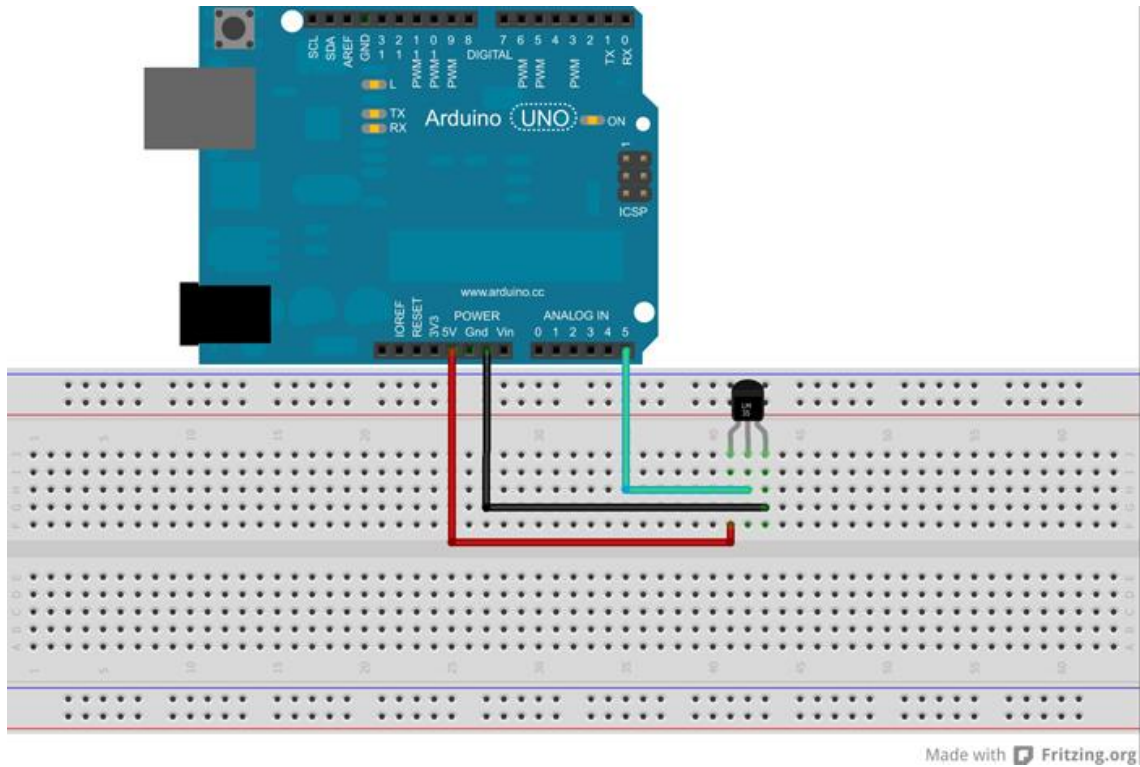
OBJETIVO:

- Aprender a usar o PLX-DAQ com o Excel.
- Configurar o gráfico para verificar a temperatura em função do tempo.

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimentos
Arduino Uno Ver 3	01	
LM35 (Sensor de Temperatura do kit)	01	Utilize o LM35 para gerar um gráfico de temperatura, pois ele vai ser substituído pelo DS18B20
Jumpers	03	Quando for colocar os jumpers no sensor, tome cuidado para não inverter a polaridade pois o sensor poderá esquentar bem rápido
Notebook	01	
Cabo USB	01	

Esquema de ligação



Código

```
const int LM35 = A0; // Define o pino que lera a saída do LM35
float temperatura; // Variável que armazenará a temperatura medida

//Função que será executada uma vez quando ligar ou resetar o Arduino
void setup() {
  Serial.begin(9600); // inicializa a comunicação serial
  Serial.println ("CLEARDATA");
  Serial.println ("LABEL, Hora, Temperatura);
}

//Função que será executada continuamente
void loop() {
  temperatura = (float(analogRead(LM35))*5/(1023))/0.01;
  linha ++;
  Serial.print ("DATA, TIME,");
  Serial.print(temperatura);
  delay(2000);
}
```

Tutorial 5 - Sensor de temperatura

Características do Sensor de Temperatura

O DS18B20 é um sensor de temperatura que opera na faixa de -55°C à 125°C e possui uma precisão de $\pm 0,5^{\circ}\text{C}$. Este sensor leva no máximo 750ms para realizar uma conversão de temperatura e salvar em sua memória.



Figura: Sensor DS18B20
Fonte: (EBURGO, 2016)

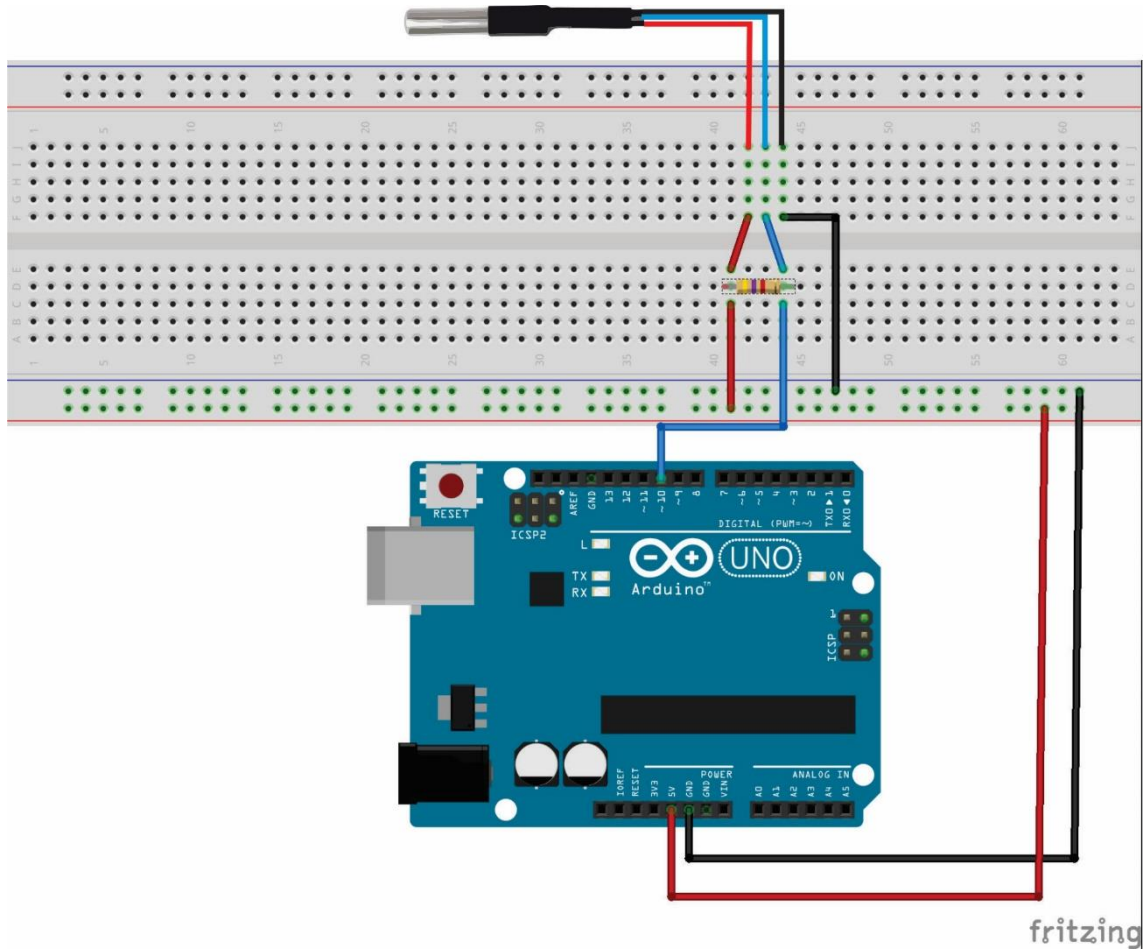
OBJETIVO:

- Comparar a temperatura do sensor com o termômetro de mercúrio
- Fazer medições em líquidos com temperaturas diferentes

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimentos
Arduino Uno Ver 3	01	
DS18B20 (será disponibilizado pelo professor)	01	O sensor necessita de uma biblioteca especial para o funcionamento. Sua equipe poderá fazer o download nesse site: http://playground.arduino.cc/Learning/OneWire
Resistor 4.7 k Ω (Amarelo, violeta e vermelho)	01	
Jumpers	04	
Notebook	01	
Cabo USB	01	

Esquema de ligação



Código

```
#include <OneWire.h> // Biblioteca para o funcionamento do sensor
#include <DallasTemperature.h> // Biblioteca para o funcionamento do sensor
#define ONE_WIRE_BUS 3 // Porta digital 3 definida como entrada do sensor
OneWire oneWire(ONE_WIRE_BUS);
DeviceAddress sensor1;

//Função que será executada uma vez quando ligar ou resetar o Arduino
void setup() {

Serial.begin(9600); // inicializa a comunicação serial
sensors.begin();// Inicializar o sensor
// Localiza e mostra endereços dos sensores
Serial.println("Localizando sensores DS18B20...");
Serial.print("Foram encontrados ");
Serial.print(sensors.getDeviceCount(), DEC);
Serial.println(" sensores.");
if (!sensors.getAddress(sensor1, 0))
    Serial.println("Sensores nao encontrados !");
// Mostra o endereco do sensor encontrado no barramento
Serial.print("Endereco sensor: ");
mostra_endereco_sensor(sensor1);

// Sub rotina para mostrar o sensor

void mostra_endereco_sensor(DeviceAddress deviceAddress)
{
for (uint8_t i = 0; i < 8; i++)
{
// Adiciona zeros se necessário
if (deviceAddress[i] < 16) Serial.print("0");
Serial.print(deviceAddress[i], HEX);
}
}

void loop()
{
// Lê a informação do sensor de temperatura
sensors.requestTemperatures();
float tempC = sensors.getTempC(sensor1);
}
```

Tutorial 7 – *PUSH BUTTON*

O que é um *Push Button*?

O botão de pressão (*Push Button*), tem o mesmo funcionamento que um interruptor elétrico, fechando ou abrindo um circuito elétrico; a principal diferença entre os dois é a força para acionar.

Muitas vezes o botão de pressão inclui um mecanismo para que volte ao estado de repouso quando a força deixa de atuar.

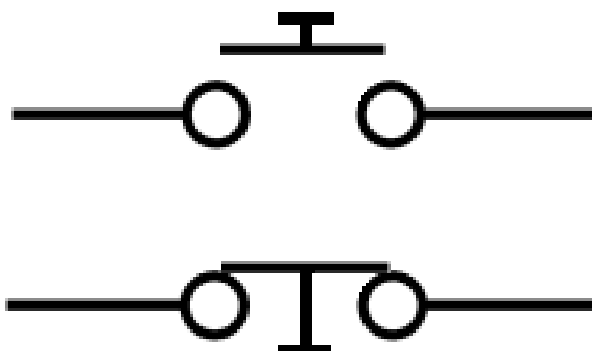


Figura: Símbolo definido do Push Button, aberto e fechado

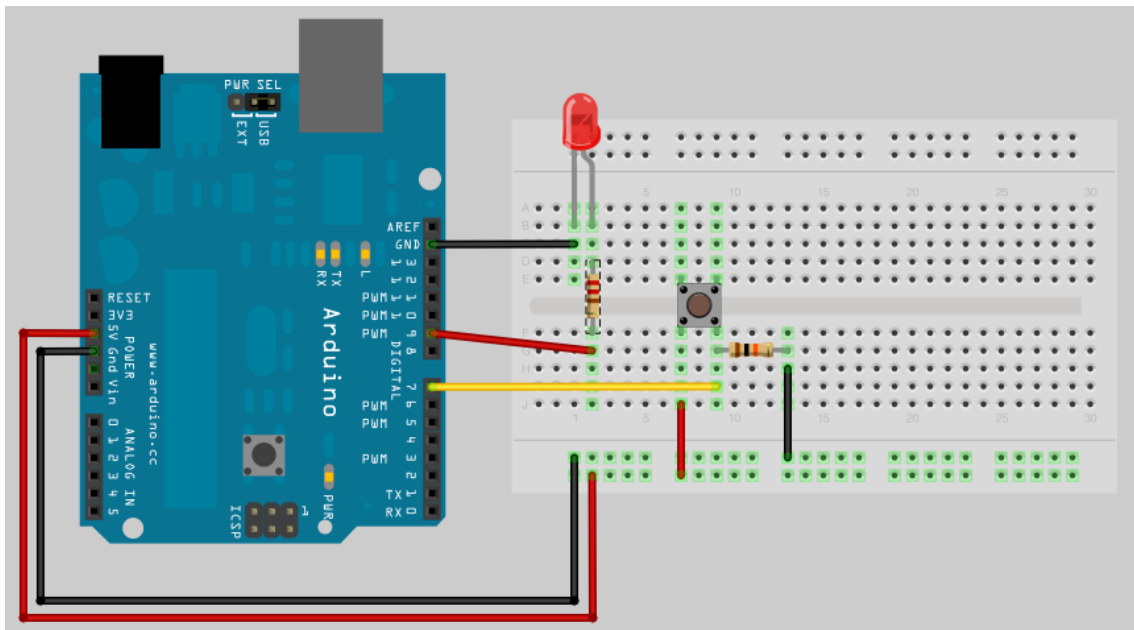
OBJETIVO:

- Testar o código para acender um LED
- Ligar o relé, utilizando o mesmo código para acender o LED

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimento
Arduino Uno Ver 3	01	
<i>Push Button</i>	01	
Resistor de 220Ω (vermelho, vermelho e preto)	01	
Resistor de 10kΩ (marrom, preto e laranja)	01	
Jumpers	07	
Notebook	01	
Cabo USB	01	

Esquemas de Ligação



Código

```
int ledPin = 13; // porta 13 em output para o LED

int inPin = 7; // porta 7 em input (para o push button)

int val = 0; // variável para ler o status do pino

void setup() {

  pinMode(ledPin, OUTPUT); // declare LED como output

  pinMode(inPin, INPUT); // declare pushbutton como input

}

void loop(){

  val = digitalRead(inPin); // ler a entrada de valor

  if (val == HIGH) { // verificar se a entrada é alta

    digitalWrite(ledPin, LOW); // LED OFF

  } else {

    digitalWrite(ledPin, HIGH); // LED ON

  }

}
```

Tutorial 8 – Motor 9V

Princípio do Funcionamento de um motor CC

Um motor de corrente contínua converte energia elétrica em energia mecânica, como qualquer motor, mas tem uma característica que o individualiza: deve ser alimentado com tensão contínua. Por exemplo, pilhas e baterias.



Figura: Motor CC 9V

Fonte: (MULTILÓGICA, 2016)

Os motores de corrente contínua (CC), são dispositivos que operam aproveitando as forças de atração e repulsão geradas por eletroímãs e ímãs permanentes.

O resultado disso será uma transformação da força de atração em repulsão, o que fará com que o rotor continue seu movimento, passando "direto" pela posição que seria de equilíbrio. De modo que os pólos do rotor se defrontassem com os de nome oposto do ímã fixa.

A velocidade de rotação deste tipo de motor não depende de nada a não ser da força que o rotor tenha de fazer para girar. Desta forma, os pequenos motores de corrente contínua têm uma velocidade muito maior quando giram livremente do que quando giram fazendo algum tipo de esforço (movimentando alguma coisa). (FRANCA, 2001)

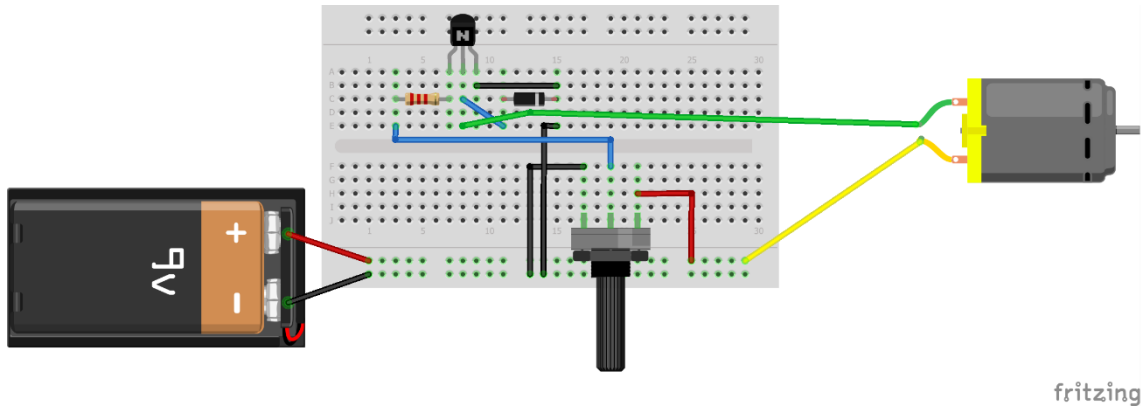
OBJETIVO:

- Testar o funcionamento do motor
- Acoplar o motor na tampa do calorímetro
- Controlar a velocidade do motor, utilizando um transistor.
- Montar um eixo e uma pá, com o intuito de agitar o líquido no interior do calorímetro

Material e Procedimentos

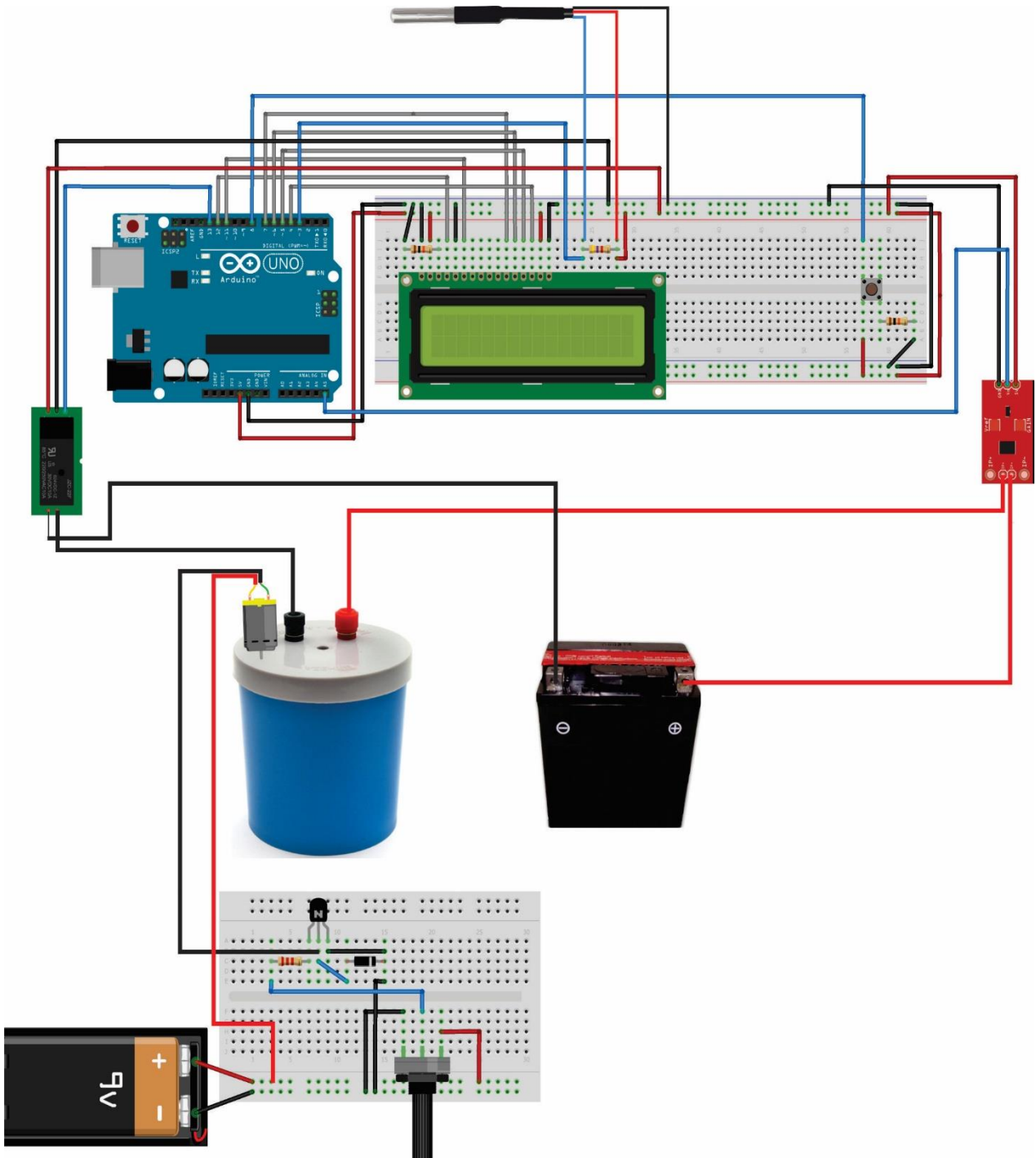
Equipamentos / Componentes	Qtde	Procedimento
Motor 9v	01	A equipe poderá alimentar o motor com pilhas ou baterias (Recomendado)
Resistor de 2.2kΩ (vermelho, vermelho e laranja)	01	
Transistor TIP 120	01	
Potenciômetro 10 kΩ	01	
Diodo 1n4148	01	
Jumpers	08	
Protoboard Half	01	

Esquemas de Ligação



PROJETO FINAL

O Intuito é deixar os alunos chegarem a esse projeto final. Mobilize as equipes para trabalhar em conjunto e reunir as partes.



Código

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>

// Porta do pino de sinal do DS18B20
#define ONE_WIRE_BUS 3

// Define uma instancia do oneWire para comunicacao com o sensor
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
DeviceAddress sensor1;
// Inicializa o LCD
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
// Variaveis inteiras e flutuantes
int linha = 0;
int LABEL = 1;
int rele = 13;
int botao=9;
int var=0;
int var2=0;
int estado=0;
int contador = 0;
int motor = 8;
int ACS712 = A5;
float potencia = 0;
int tensaoDc = A3;
float vout = 0.0;
float vin = 0.0;
float R1 = 30000.0; //
float R2 = 7500.0; //
int value = 0;
int sensorValue_aux = 0;
float sensorValue = 0;
float currentValue = 0;
float voltsporUnidade = 0.004887586;
float corrente = 0;
float tensao = 0
```



```

void setup(void)
{

  Serial.begin(9600);
  sensors.begin();
  Serial.println ("CLEARDATA");
  Serial.println ("LABEL,
Hora,tempo, Temperatura,Tensao,Corrente,Potencia,linha");
  // Localiza e mostra enderecos dos sensores
  Serial.println("Localizando sensores DS18B20...");
  Serial.print("Foram encontrados ");
  Serial.print(sensors.getDeviceCount(), DEC);
  Serial.println(" sensores.");
  if (!sensors.getAddress(sensor1, 0))
    Serial.println("Sensores nao encontrados !");
  // Mostra o endereco do sensor encontrado no barramento
  Serial.print("Endereco sensor: ");
  mostra_endereco_sensor(sensor1);
  Serial.println();
  Serial.println();
  lcd.begin(16, 2);
  pinMode (rele, OUTPUT);
  pinMode(botao,INPUT);

}

void mostra_endereco_sensor(DeviceAddress deviceAddress)
{
  for (uint8_t i = 0; i < 8; i++)
  {
    // Adiciona zeros se necessário
    if (deviceAddress[i] < 16) Serial.print("0");
    Serial.print(deviceAddress[i], HEX);
  }
}

```

```

void loop()
{
  // Le a informacao do sensor de temperatura
  sensors.requestTemperatures();
  float tempC = sensors.getTempC(sensor1);
  // Inserção manual da tensão da bateria
  tensao = VALOR;

  for(int i=100; i>0; i--){
    sensorValue_aux = (analogRead(ACS712) -511); // le o sensor na pino
    analogico A0 e ajusta o valor lido ja que a saída do sensor é (1023)vcc/2 para
    corrente =0
    sensorValue += pow(sensorValue_aux,2); // somam os quadrados das
    leituras.
  }
  sensorValue = (sqrt(sensorValue/ 100)) * voltsporUnidade; // finaliza o cálculo
  da méida quadratica e ajusta o valor lido para volts
  currentValue = (sensorValue/0.100);
  corrente = currentValue;
  //cálculo da potência
  potencia = tensao*corrente;

  //Váriaveis para inserção de dados no EXCEL
  linha ++;
  contador = contador +5;
  Serial.print ("DATA, TIME,");
  Serial.print(contador);
  Serial.print (" ");
  Serial.print (tempC);
  Serial.print (" ");
  Serial.print (tensao);
  Serial.print (" ");
  Serial.print (corrente);
  Serial.print (" ");
  Serial.print (potencia);
  Serial.print (" ");
  Serial.println (linha);
  if (linha>200)
  {
    linha = 0;
  }
}

```

```

Serial.println ("ROW, SET, 2");
}
lcd.setCursor(0,0);
lcd.print("Temp.:  ");
//Simbolo grau
lcd.write(223);
lcd.print("C");
lcd.setCursor(7,0);
lcd.print(tempC)
// botão
var=digitalRead(botao); // ler o valor enviado pelo botão: "HIGH" ou "LOW"
if ((var == HIGH) && (var2 == LOW)) {
    estado = 1 - estado;
    delay(20);
}
var2=var;
switch (var) {
case 1: if (estado == 1) {
    digitalWrite(rele, HIGH); // liga o relé
    lcd.setCursor(0,1);
    lcd.print (" ");
    lcd.print ("ligado");
    break;
}
}
if (estado == 0) {
    digitalWrite(rele, LOW); // desliga o relé
    lcd.setCursor(0,1);
    //lcd.clear();
    lcd.print ("desligado");
}

delay(4500);
}

```

Tutorial 9 – Sensor de Tensão

UPGRADE

Sensor de Tensão

O Sensor de Tensão é um módulo eletrônico desenvolvido capaz de detectar tensões DC (Corrente Contínua) por meio dos divisores de tensão. Para o uso em microcontroladores, informando aos sistemas microcontroladores em que estiver instalado, entre eles, o Arduino, se há ou não a existência de tensão.

Muito eficiente, o Sensor de Tensão também pode ser utilizado como voltímetro (medidor de tensão DC) em projetos eletrônicos, de forma a monitorar a tensão de entrada.



Figura: Sensor de tensão DC
Fonte: (ELETROGATE, 2016)

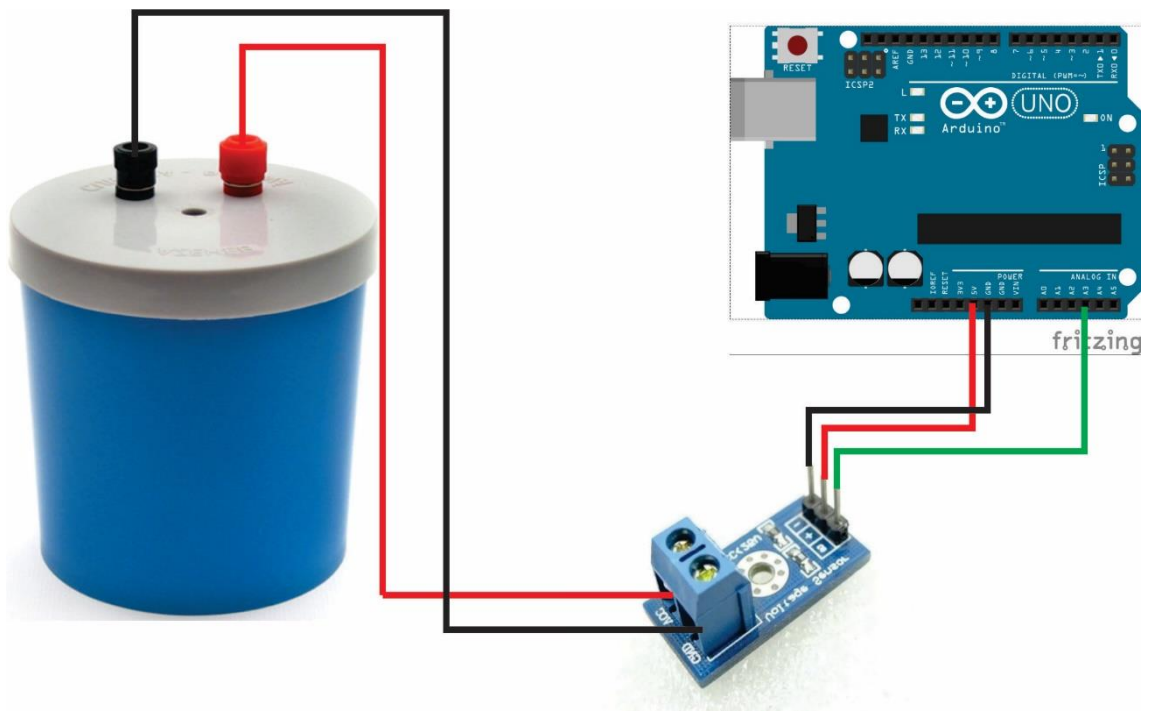
OBJETIVO:

- Monitorar a tensão da bateria em tempo real
- Calcular a potência do resistor do calorímetro

Material e Procedimentos

Equipamentos / Componentes	Qtde	Procedimento
<i>Sensor de tensão DC</i>	01	Adaptar o sensor de tensão no projeto final.

Esquemas de Ligação



Código

//INSIRA ESSE CÓDIGO NA ROTINA VOID LOOP ()

```
value = analogRead(tensaoDc);  
vout = (value * 5.0) / 1023.0; //  
vin = vout / (R2/(R1+R2));
```

```
tensao = vin
```

Considerações Finais

A maioria dos estudantes quando entra em contato com a plataforma de *hardware* livre não possui conhecimentos de programação e eletrônica básica. Com os tutoriais não há necessidade de pré-requisito, uma vez que o enfoque está no trabalho em equipe e na descoberta do conhecimento. O professor poderá dar uma breve introdução dos conceitos básicos de eletricidade e programação ao longo das atividades, fazendo com que os alunos procurem pesquisar e resolver os problemas encontrados, não individualmente, mas sim em equipe.

Com relação a programação aplicada, nos tutoriais há comentários sobre os códigos e a lógica envolvida na estrutura do programa. A inserção do algoritmo no ambiente é simples; o professor não tem necessidade em detalhar o ensino em linguagem de programação ou fluxogramas. O foco são os conceitos físicos no processo de aprendizagem.

Com as práticas e desenvolvimento das atividades os alunos podem aprender os conceitos envolvidos na construção do projeto e verificar como a Física está presente em todos os aspectos do desenvolvimento do calorímetro, despertando o interesse pela pesquisa e pela ciência.

REFERÊNCIAS

ARDUINO. Arduino UNO REV 3. Disponível em: <www.arduino.cc>.

DIGITAL, I. Sensor de Corrente ACS712 30 A. Disponível em: <<http://www.institutodigital.com.br/>>.

EBURGO. Sensor de temperatura DS18B20. Disponível em: <<http://www.eburgo.com.br/listing/sensor-de-temperatura-ds18b20-a-prova-dagua/53>>.

ELETROGATE. Sensor de tensão DC - 25V. Disponível em: <eletrogate.com>.

FINDERNET. Funcionamento do relé. Disponível em: <<http://www.findernet.com/en/node/47658>>.

FOUNDATION, W. D. PELA W. Sensores de efeito Hall. Disponível em: <https://pt.wikipedia.org/wiki/Sensor_de_efeito_Hall>.

FRANCA, A. **Máquinas de corrente contínua**. Hackaday, 2001.

HARDWARE. LCD - Cristal Líquido: Funcionamento. Disponível em: <www.hardware.com.br/livros/hardware-manual/como-funciona-lcd.html>.

MORIMOTO, C. **Hardware II: O guia definitivo**. Porto Alegre: Sul Editores, 2007.

MULTILÓGICA. Motor 9V. Disponível em: <<https://multilogica-shop.com/motor-cc-6v>>.