

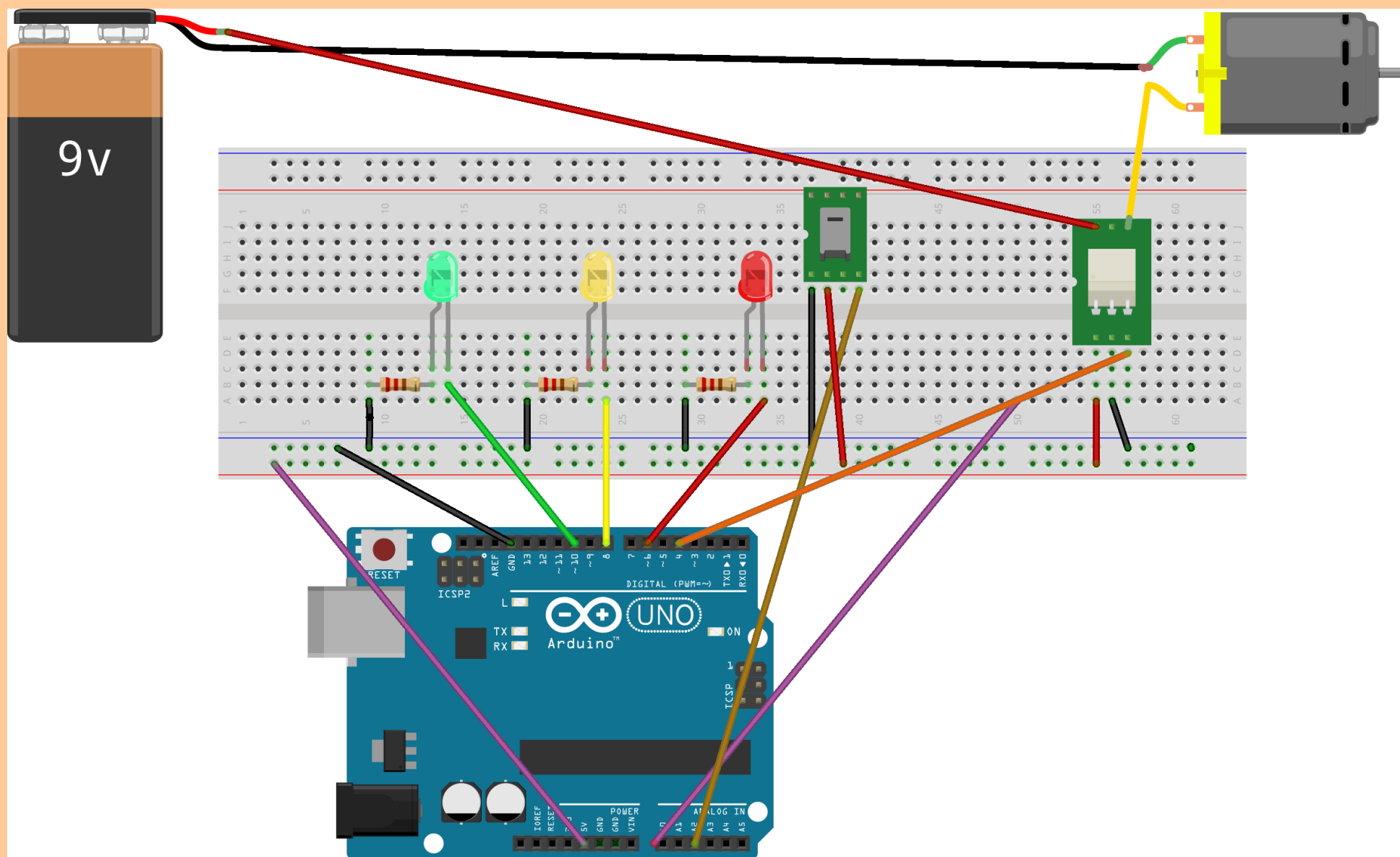
**INSTITUTO
FEDERAL**
Santa Catarina

Câmpus
Chapecó



EMI – Tópicos especiais – Arduino + Banco de dados

IFSC – Tópicos especiais



fritzing

Acionamento de dispositivos elétricos

Os dispositivos elétricos são acionados de forma manual e automatizada.

A forma manual mais simples é através de um interruptor de corrente(liga/desliga).

A forma automatizada pode ser realizada por diversos mecanismos elétricos, pneumáticos, sensores e controladores programáveis(PIC).

O Arduino está classificado na categoria de controladores programáveis, no entanto, é necessário receber o estímulo, decifrar, tomar a decisão e acionar o dispositivo.

Neste experimento será utilizado o sensor de temperatura DHT11 para medir a mesma de algum componente e em seguida o sketch realizará o acionamento de um dispositivo elétrico através do relê que tem a função de interromper/permitir a passagem da corrente elétrica.

DHT11 – DHT22

O **DHT11** é um **sensor de umidade e temperatura** integrado em um só módulo de baixo custo. Este sensor utiliza um **termistor** para medir a temperatura e um sensor **capacitivo** para medir a umidade do ambiente. Este sensor é uma excelente opção para monitoramento e controle climatização.

O **DHT11** possui um controlador de 8 bits que converte o sinal de **temperatura e umidade** dos sensores e um sinal serial e envia ao **Arduino** através do pino de dados (Data).

*O sensor **DHT11** pode medir temperaturas entre 0 a 50° Celsius com uma precisão de 2 graus, e umidade entre 20 a 90 % com uma precisão de 5%.*

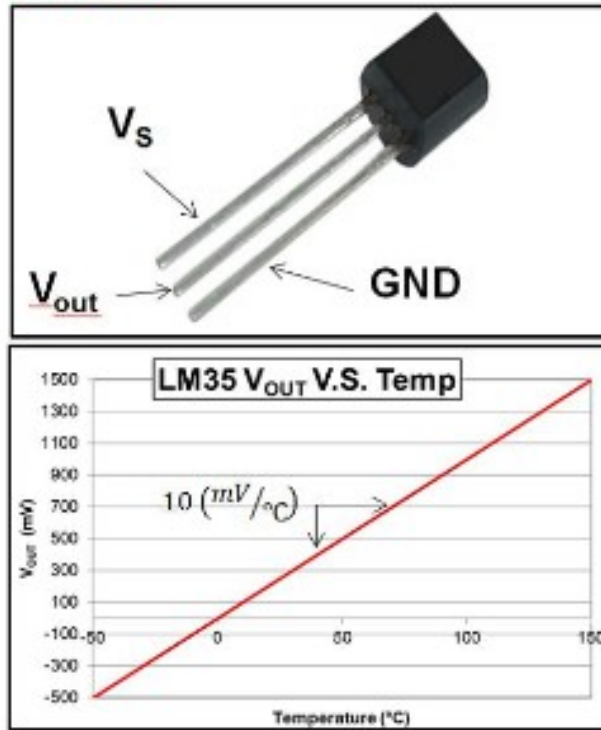


O LM 35 (utilizado em montagens eletrônicas)

Na sua linha de semicondutores, a National Semiconductor possui tipos indicados para a medida precisa de temperatura com saída linearmente proporcional a valores centígrados (Celsius). Estes pequenos componentes podem ser utilizados numa infinidade de aplicações, desde termômetros para a faixa de -55 a $+150^{\circ}\text{C}$, até controles, termostatos, detectores etc

A série LM35 consta de sensores de temperatura cuja tensão de saída é linearmente proporcional à temperatura, em graus centígrados ou Celsius. Uma característica importante destes sensores está no fato de que, ao contrário dos sensores ajustados para a escala Kelvin, não é necessário subtrair uma enorme tensão constante para se obter o valor em graus Celsius.

LM35



- Three-Pin
 - TO-92 Package
 - Easy to Use
 - 4V-20V Operating Range
 - 60 μ A Max Current Draw
- Analog Output
 - 0.5 $^{\circ}C$ Accuracy at 25 $^{\circ}C$
 - Easily read by Arduino
 - Highly Linear Transfer Function
 - 10 ($mV/^{\circ}C$) Slope

- Calibração diretamente em $^{\circ}C$ (centígrados ou Celsius)
- Precisão de 0,5 $^{\circ}C$ garantida a uma temperatura de 25 $^{\circ}C$
- Especificado para a faixa de -55 a 150 $^{\circ}C$
- Operação com tensões de 4 a 30 V
- Consumo de corrente menor que 60 μ A
- Baixo aquecimento próprio: 0,08 $^{\circ}C$ ao ar livre

Relê

Os relés são componentes eletromecânicos capazes de controlar circuitos externos de grandes correntes a partir de pequenas correntes ou tensões, ou seja, acionando um relé com uma pilha podemos controlar um motor que esteja ligado em 110 ou 220 volts, por exemplo.

O **Módulo Relé 5V 1 Canal** permite que a partir de uma plataforma microcontrolada seja possível controlar cargas AC (alternada) de forma simples e prática. Por ter apenas 1 canal, é possível controlar apenas uma carga AC de até 10A. Comumente é utilizado em projetos de automação residencial para controle de lâmpadas, ventiladores e outras saídas que possam ser acionadas através de relé.



Projetos com Arduino ou outras plataformas microcontroladas em que seja necessário fazer o controle de cargas AC. Caso o módulo seja utilizado junto a uma plataforma que esteja conectada à internet, torna-se possível controlar cargas AC através de uma página web, smartphone ou tablet.

Especificações:

Tensão de operação: 3,3V – 5VDC

Corrente de operação: 15 ~ 20mA

Capacidade do relé: 30VDC/10A e 250VAC/10A

1 canal

LED indicador para presença de tensão

LED indicador para acionamento do relé

Tempo de resposta: 5 ~ 10ms

Fonte: <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-rele-5v-1-canal/>

Incluindo a biblioteca DHT

Uma grande vantagem das placas Arduino é a grande diversidade de bibliotecas disponíveis que podem ser usadas em seu programa. Isso faz com o trabalho pesado em programação seja abstraído e resumido em simples comandos.

Com isso, o desenvolvedor não precisa de conhecimento muito aprofundado em programação, podendo gastar menos tempo nisso, resultando em mais tempo para trabalhar com empenho na estratégia de controle.

Para trabalhar de forma fácil com o DHT, podemos baixar uma biblioteca para ele no GitHub do Rob Tillaart (<https://github.com/RobTillaart/Arduino/tree/master/libraries/DHTlib>).

Link com instruções - <https://portal.vidadesilicio.com.br/dht11-dht22-sensor-de-umidade-e-temperatura/>

```
1 #include <DHT.h>
2 #define DHTPIN A5 // pino que estamos conectado
3 #define DHTTYPE DHT11 // DHT 11
```

```
5 int LedGreen=10;
6 int LedYellow=8;
7 int LedRed=6;
8 int Espera=1000;
9 int PinoRele=4;
```

```
11 String Texto;
12 DHT dht(DHTPIN, DHTTYPE, A5);
13 void setup() {
14   pinMode(LedGreen, OUTPUT);
15   pinMode(LedYellow, OUTPUT);
16   pinMode(LedRed, OUTPUT);
17   pinMode(PinoRele, OUTPUT);
18   Serial.begin(9600);
19 }
```

```
21 void loop() {
22     float  temperatura=dht.readTemperature ( ) ;
23     float umidade = dht.readHumidity();
24
25     Texto="Umidade: "+String(umidade)+" - Temperatura: "+String(temperatura);
26     Serial.println(Texto);
27
28     if (temperatura>=1 && temperatura<=20){
29         digitalWrite(LedGreen, HIGH);
30         digitalWrite(LedYellow, LOW);
31         digitalWrite(LedRed, LOW);
32         digitalWrite(PinoRele, LOW);
33     }
```

```
34 if (temperatura>20 && temperatura<=30) {
35   digitalWrite(LedGreen, LOW);
36   digitalWrite(LedYellow, HIGH);
37   digitalWrite(LedRed, LOW);
38   digitalWrite(PinoRele, LOW);
39 }
40 if (temperatura>30) {
41   digitalWrite(LedGreen, LOW);
42   digitalWrite(LedYellow, LOW);
43   digitalWrite(LedRed, HIGH);
44   digitalWrite(PinoRele, HIGH);
45 }
46 delay(Espera);
47 }
```