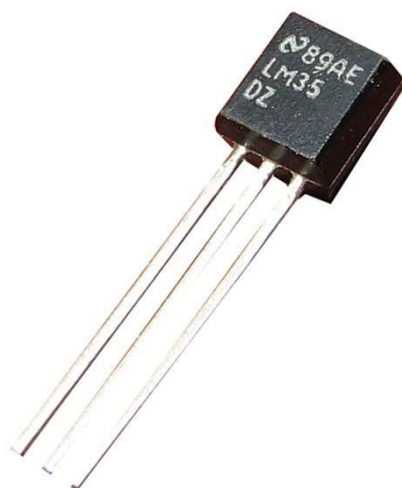


Experiência 7 – Sensor de temperatura

7.1. Sensor de temperatura - LM35

O sensor LM35 é um sensor de precisão que apresenta uma saída de tensão linear proporcional à temperatura em que ele se encontra no momento, tendo em sua saída um sinal de 10mV para cada Grau Célsius de temperatura.

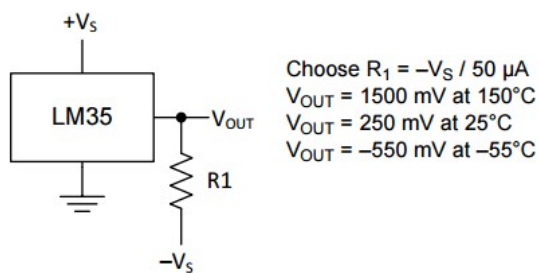


LM35

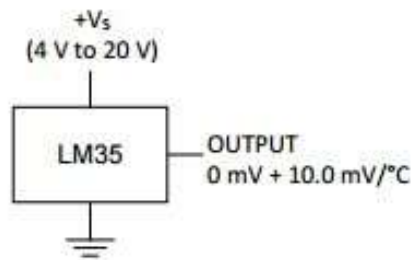
Esse sensor não necessita de qualquer calibração externa para fornecer com exatidão, valores temperatura com variações de $\frac{1}{4}^{\circ}\text{C}$ ou até mesmo $\frac{3}{4}^{\circ}\text{C}$ dentro da faixa de temperatura entre -55°C e 150°C .

Ele pode ser usado de duas formas, com alimentação simples ou simétrica, dependendo do que se desejar como sinal de saída, mas independentemente disso, a saída continuará sendo de $10\text{mV}/^{\circ}\text{C}$.

Em cada uma dessas duas formas de alimentação, o range de temperatura, ou seja, a temperatura máxima e mínima medida com exatidão, é diferente.



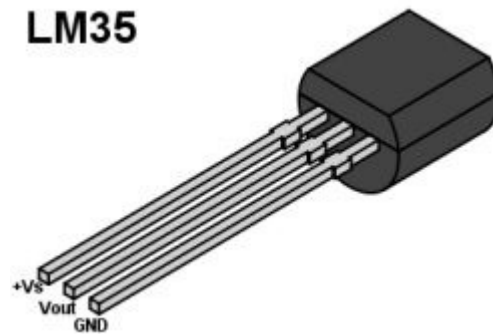
Modo escala completa – (-55°C a 150°C)



Modo Básico – (-2°C a 150°C)

Uma vantagem é o fato desse sensor drenar apenas 60µA para estas alimentações. Dessa forma, seu auto aquecimento é de aproximadamente 0.1°C ao ar livre.

O sensor LM35 é apresentado com vários tipos de encapsulamentos, sendo o mais comum o TO-92, que mais se parece com um transistor, e oferece ótima relação custo-benefício, por ser o encapsulamento mais barato sem diferenças em seu uso ou exatidão.

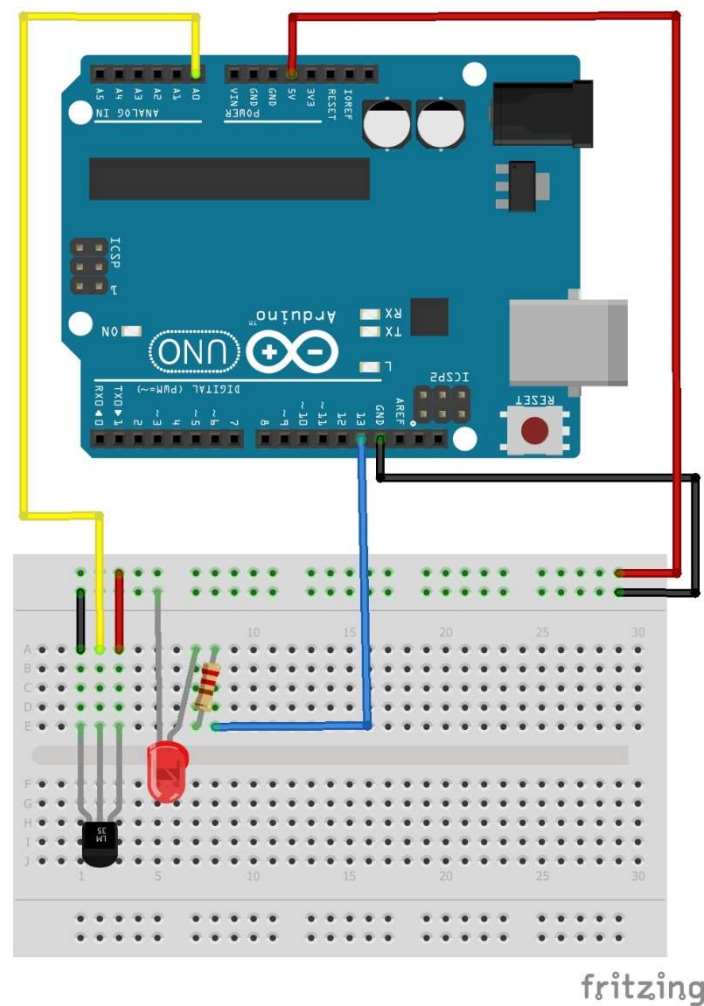


7.2. Ingredientes:

- Fios Jumper's
- Protoboard
- Arduino Uno Rev3
- LM35
- 1 LED
- 1 Resistor 330 Ohm

7.3. Misturando os ingredientes

Agora vamos conectar os componentes do projeto. Para isso, desligue o cabo USB de seu Arduino e monte seu circuito conforme a figura a seguir.



7.4 Levando ao forno

Conecte seu Arduino ao computador e abra a IDE Arduino.

Antes de carregar um programa, você precisa selecionar qual porta você deseja usar para fazer carregar o programa no Arduino (upload). Dentro do Arduino IDE, clique no menu Ferramentas (tools) e abra o submenu Porta (Port). Clique na porta que seu Arduino está conectado, tal como COM3 ou COM4. Geralmente aparece o nome da placa Arduino: “COM3 (Arduino Uno)”.

Você também precisa garantir que o tipo de placa apropriado está selecionado em Ferramentas (Tools) no submenu Placa (Board).

7.5. Preparando a cobertura

Crie um programa (sketch) e salve com o nome de “programa_sensor_de_temperatura”.

Com o seu programa salvo, escreva nele o código conforme escrito abaixo.

```
//Sensor de temperatura usando o LM35

int LM35 = A0; // Define o pino que lê a saída do LM35
float temperatura; // Variável que armazena a temperatura medida
int ledPin = 13; //Led no pino 13

//Função que é executada uma vez quando ligar ou resetar o Arduino
void setup() {
    pinMode(ledPin,OUTPUT); //define a pino 13 como saída
    Serial.begin(9600); // inicializa a comunicação serial
}

//Função que é executada continuamente
void loop() {
    temperatura = (float((((analogRead(LM35)*5.0)/1023)*1000)-500)/10);
    Serial.print("Temperatura: ");
    Serial.println(temperatura);

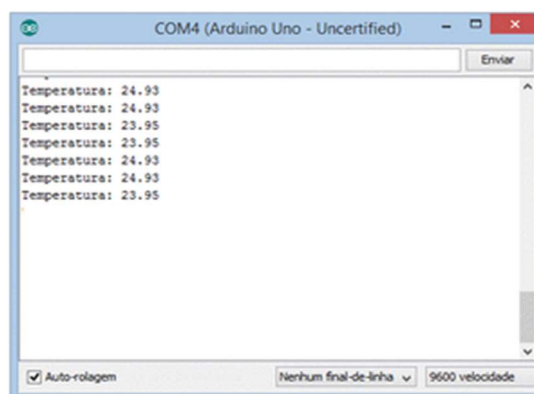
    //se o valor lido for maior que 27, liga o led
    if (temperatura >= 27)
        digitalWrite(ledPin,HIGH);
    // senão, apaga o led
    else
        digitalWrite(ledPin,LOW);

    delay(3000);
}
```

Após escrever o código, clique em Upload para que o programa seja transferido para seu Arduino.

7.6. Experimentando o prato

Caso tenha ocorrido tudo conforme esperado, poderemos fazer a leitura da temperatura através do monitor serial. Abra o monitor serial para verificar o que está sendo lido na entrada A0.



7.7. Entendendo o Software

Nosso programa lerá qual é o valor do sinal em A0, que varia de 0 a 1023, onde 0 corresponde a 0Volts e 1023 corresponde a 5Volts. Como sabemos, 1°C é igual a 10mV. Sendo assim, temos:

$$\text{Tensão em A0} = (\text{Valor lido em A0}) * (5/1023)$$

$$\text{Temperatura} = (\text{Tensão em A0} * 1000) - 500 / 10\text{mV}$$

Logo:

$$\text{Temperatura} = [(\text{Valor lido em A0}) * (5/1023) * 1000] - 500 / 10\text{mV}$$

Em linguagem de programação, ficará:

```
temperatura = (float((((analogRead(LM35) * 5.0) / 1023) * 1000) - 500) / 10);
```

Perceba que colocamos o comando de leitura do valor analógico, `analogRead`, dentro de `float()`. Você saberia me dizer o motivo?

Quando o Arduino faz uma leitura analógica, ele converte o valor lido, que pode ser um valor de tensão entre 0V e 5V, em um número entre 0 e 1023. Ou seja, o Arduino divide 5Volts, que é o maior valor que ele é capaz de ler, em 1024 partes iguais e lhe informa quantas partes tem o valor que ele está medindo.

O número que o Arduino lhe informa é do tipo inteiro, contudo, o valor de temperatura é um número racional, que pode assumir valores decimais. Por conta disso, no nosso programa, declaramos a temperatura como uma `float`.

Em programação, quando multiplicamos uma variável inteira por uma variável racional, o programa considera que o resultado deve ser inteiro, eliminando a parte decimal da variável. Dessa forma, para que tenhamos um resultado racional, devemos transformar o número inteiro em um número racional usando o operador “cast”.

Em virtude disso, em nosso código foi necessário colocar o comando de leitura do valor analógico, `analogRead`, dentro de `float()`.

```
numeroracional = float(numero);  
temperatura = (float((((analogRead(LM35) * 5.0) / 1023) * 1000) - 500) / 10);
```

Sempre que for necessário fazer um cálculo com o valor analógico, precisamos convertê-lo para uma variável do tipo `float`.

Em alguns casos, precisamos transformar uma variável qualquer para o tipo inteiro. O procedimento é o mesmo, ou seja, basta colocar o valor ou variável dentro dos parênteses de `int()`;

```
numerointeiro= int(numero);  
temperatura=int(float((((analogRead(LM35) * 5.0) / 1023) * 1000) - 500) / 10);
```

Exercícios

- 1 – Acrescente um buzzer a montagem. Modifique o programa para o led acender e o buzzer apitar.
- 2 – Acrescente mais 2 leds a montagem. Conforme a temperatura aumenta os leds vão acendendo progressivamente. O buzzer deve apitar quando o último led acender.
- 3- Transforme a leitura do sensor de temperatura em uma função que retorna o valor lido.
- 4 – Transforme o controle de cada um dos led em uma função que recebe o valor lido.
- 5 – Transforme o controle do buzzer em uma função que recebe o valor lido.