

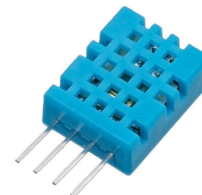


# Programação & Arduino

Desenvolvimento de Sistemas

## Medidor de Temperatura com Arduino<sup>1</sup>

Neste projeto veremos a aplicação do [Display LCD](#) (Liquid Crystal Display, em inglês) usado para representar um medidor de temperatura que será dado pelo [LM35](#). Para deixar o projeto mais visual possível, utilizaremos também 2 [LED's](#) para indicar se a temperatura está acima ou abaixo da temperatura padrão. Este projeto te possibilita fazer medições em qualquer ambiente que estiver, do seu próprio corpo e de objetos ao seu redor.



### Materiais necessários:

- 1 x Placa Uno R3 com cabo USB
- 1 x Protoboard 830 Pontos
- 1 x Display LCD 16x2
- 1 x Sensor de Temperatura LM35DZ
- 2 x LED Difuso 5mm Vermelho
- 20 x Jumpers Macho/Macho
- 1 x Potenciômetro Linear 10k $\Omega$
- 2 x Resistores

### Como Funciona o Medidor de Temperatura.

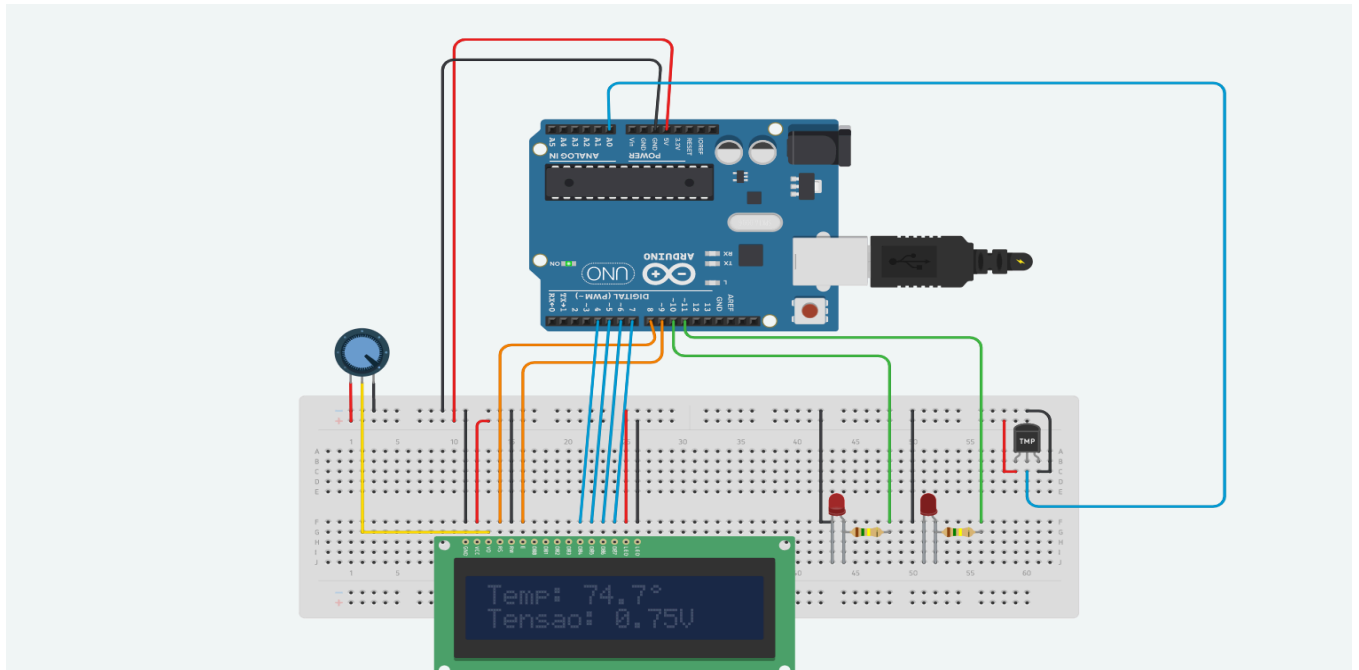
Utilizar o Sensor de Temperatura HW-481 em conjunto com o Arduino e medir a temperatura ambiente. O valor da temperatura será exibido no monitor serial do ambiente de programação do Arduino.



---

<sup>1</sup> Projeto disponível em: <https://github.com/hise066/Medidor-de-Temperatura-com-Arduino.git>  
Adaptações foram utilizadas para o uso em sala de aula.

## Pinagem



## Código

```

/*
    Projeto Medidor de Temperatura Eletrogate
    Tutorial programando um sensor de temperatura LM35 e um display LCD
    Criado por Flávio Babos (https://flaviobabos.com.br/arduino/)
*/
//Inclua a biblioteca do LCD
#include <LiquidCrystal.h>
//Defina as portas do Arduino para cada pino do LCD
const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
//Defina a temperaturaC ambiente do lugar em que estiver para que as LED's
acendam com base nisso
int tempPadrao = 24;
//Array que desenha o simbolo de grau
byte a[8]= {B00110,B01001,B00110,B00000,B00000,B00000,B00000,B00000,};
//declare a variável de leitura do sensor
const int sensorLM35 = A0;

int i;
float temperaturaC = 0;
void setup() {
    //Inicializa comunicação serial
    Serial.begin(9600);
    //Método de percorrer os pinos e ao mesmo tempo declara-los como saída
    for(i=8; i<12; i++){

```

```

    pinMode(i, OUTPUT);
}
//Inicializa o LCD e posteriormente exibe a temperaturaC e a tensão
lcd.begin(16, 2);
lcd.print("Temp: ");
lcd.setCursor(0, 2);
lcd.print("Tensao: ");
lcd.setCursor(12, 2);
lcd.print("V");
//Atribui a "1" o valor do array "A", que desenha o simbolo de grau
lcd.createChar(1, a);
//Coloca o cursor na coluna 7, linha 1
lcd.setCursor(10,0);
//Escreve o simbolo de grau
lcd.write(1);
}
void loop() {
    //Variável que faz a leitura do sensor
    int valorSensor = analogRead(sensorLM35);
    //Conversao da leitura do sensor em tensão V
    float tensao = (valorSensor/1024.0)*5.0;

    //Divide a variavel temperaturaC por 8, para obter precisão na medição
    temperaturaC = (tensao) * 100.0;
    //Conversão de Celsius para Fahrenheit
    float temperaturaF = (temperaturaC*9)/5 + 32;
    //mostra no monitor serial o valor lido
    Serial.print("Valor do sensor: ");
    Serial. print(valorSensor);
    //Mostra no monitor serial a tensão lida
    Serial.print(", Tensao (V): ");
    Serial.print(tensao);
    //Mostra no monitor serial os graus lidos
    Serial.print(", Celsius: ");
    Serial.print(temperaturaC);
    //Printa no serial monitor o valor convertido
    Serial.print(", Fahrenheit: ");
    Serial.println(temperaturaF);
    delay(100);
    lcd.setCursor(6, 0);
    //Escreve no display o valor da temperaturaC
    lcd.print(temperaturaC,1);
    lcd.setCursor(8, 1);
    //Escreve no display o valor da tensão
    lcd.print(tensao,2);

    if(temperaturaC < tempPadrao){ //se a temperaturaC lida for inferior a
27 °C
        digitalWrite(10, HIGH); //apaga led da porta 10
        digitalWrite(11, LOW); //apaga led da porta 11

    }else if(temperaturaC >= tempPadrao){ //se a temperaturaC lida estiver
entre 27 °C e 31 °C
        digitalWrite(10, HIGH); //acende led da porta 10

```

```
digitalWrite(11, HIGH);           //apaga led da porta 11  
  
delay(1000);  
}  
}
```