



A atuação do sensor utilizando o arduíno, funciona da mesma forma vista acima com o multímetro, porém com os cálculos efetuados de forma automatizada, e com uma maior precisão. O valor do pino 2 (Vout) é recebido pela porta analógica A0 do arduíno, como a resolução da entrada analógica é 10 bits, e varia os valores entre 0 a 1023, 0 representará 0V (0=0V), e 5V representará 1023 (5=1023); em seguida esse valor recebido é convertido em tensão elétrica por meio da fórmula  $(\text{valor\_analog\_lm35} * 5) / 1023$ , em que o valor\_analog é referente ao valor convertido no range de 0 à 5V; o 5 que está multiplicando esse valor, é referente ao maior valor convertido no range para que seja possível a utilização de todo o intervalo, e o 1023 é referente ao conversor AD de 10 bits que existe no Arduino que comporta valores de 0 a 1023. Após a conversão do valor analógico em tensão elétrica, dividimos esse valor por 10mV, pois cada °C = 10mV, Então 10mV/°C, e assim o valor da divisão é referente ao valor real em graus Celsius, resultando na temperatura.

### Software do funcionamento do sensor de Temperatura LM35DZ com arduino, e exibição no monitor serial

```

/*****
-----Eletródex Eletrônica-----
----- Teste Sensor de Temperatura LM35DZ com arduino, e exibição no monitor serial -----
*****/

#define pin_lm35 A0 //Pino o qual o sensor está conectado

void setup() {
  Serial.begin(9600); // Inicia a porta serial configurando a taxa de dados para 9600
  pinMode(pin_lm35, INPUT); // Define o sensor como uma entrada de sinal
}

void loop() {
  float valor_analog_lm35 = float(analogRead(pin_lm35)); // Obtém o valor analógico que varia de 0 a 1023 (0 = 0V, e 5V =1023)
  float tensao = (valor_analog_lm35 * 5) / 1023; // Vamos converter esse valor para tensão elétrica
  float temperatura = tensao / 0.010; // dividimos a tensão por 0.010 que representa os 10 mV pois cada °C = 10mV, Então 10mV/°C

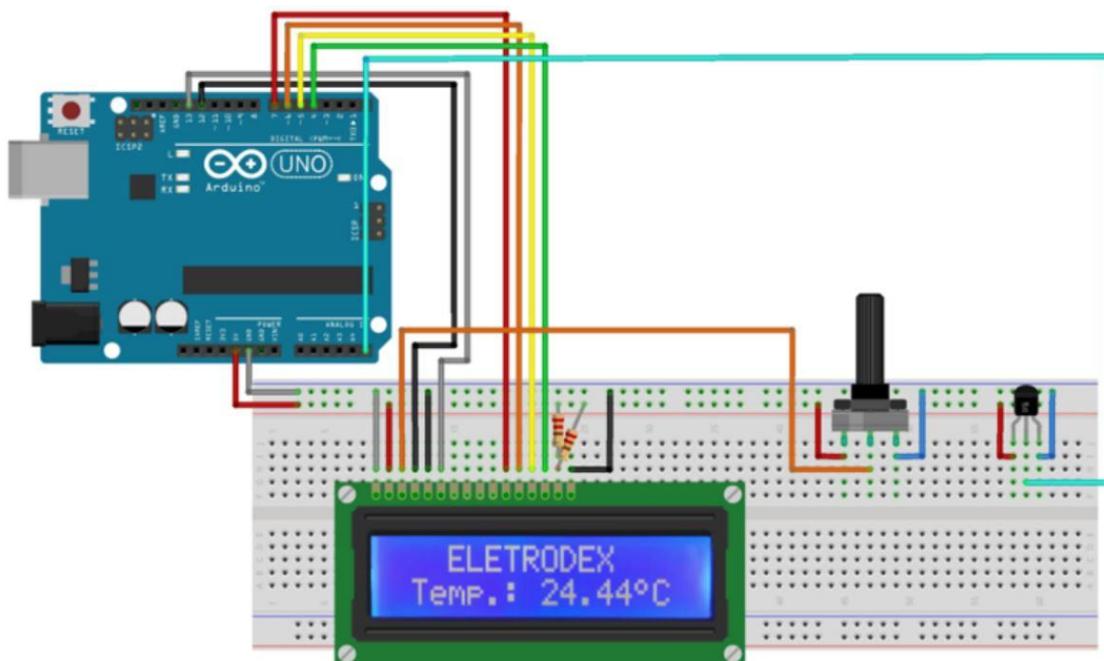
  Serial.print("Temperatura atual = ");
  Serial.print(temperatura); // Mostra na serial a temperatura do sensor
  Serial.print("°C");
  Serial.println();

  delay(1000); // aguarda 1 segundo
}

```

### Diagrama eletrônico do funcionamento do sensor de Temperatura LM35DZ com arduino, e display LCD

O diagrama a seguir expõe a forma indicada de montagem do Sensor de temperatura LM35DZ com arduíno e display LDC, para a montagem são necessários o Sensor de temperatura LM35DZ, uma protoboard, um arduíno UNO, Fios Jumperes, dois resistores de 220 ohms conectado no anodo do display para iluminação da luz de fundo e um potenciômetro para ajuste de contraste, o utilizado nesse teste foi de 20 Kohms.



A atuação do sensor utilizando o arduíno e display, funciona da mesma forma vista acima com exibição no monitor serial, em que o valor do pino 2 (Vout) é recebido pela porta analógica A0 do arduíno, como a resolução da entrada analógica é 10 bits, e varia os valores entre 0 a 1023, 0 representará 0V (0=0V), e 5V representará 1023 (5=1023); em seguida esse valor recebido é convertido em tensão elétrica por meio da fórmula  $(\text{valor\_analog\_lm35} * 5) / 1023$ , em que o valor\_analog é referente ao valor convertido no range de 0 à 5V; o 5 que está multiplicando esse valor, é referente ao maior valor convertido no range para que seja possível a

utilização de todo o intervalo, e o 1023 é referente ao conversor AD de 10 bits que existe no Arduino que comporta valores de 0 a 1023. Após a conversão do valor analógico em tensão elétrica, dividimos esse valor por 10mV, pois cada  $^{\circ}\text{C} = 10\text{mV}$ , Então  $10\text{mV}/^{\circ}\text{C}$ , e assim o valor da divisão é referente ao valor real em graus Celsius, resultando na temperatura. A diferença entre os códigos é a parte da exibição da temperatura no display, junto à criação do caractere de grau  $^{\circ}$ , que normalmente não é reconhecido pelo display, tornando necessário assim, a criação do mesmo, por meio de um array.

## Software do funcionamento do sensor de Temperatura LM35DZ com arduino, e exibição no Display LCD 16X2

```
/******  
-----Eletrodex Eletrônica-----  
-----_----- Teste de funcionamento do sensor de Temperatura LM35DZ com arduino, e display LCD -----  
*****/  
  
// include the library code for LCD display:  
#include <LiquidCrystal.h>  
  
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 13, 7, 6, 5, 4);  
  
//Array simbolo grau  
byte grau[8] = { B00001100,  
                B00010010,  
                B00010010,  
                B00001100,  
                B00000000,  
                B00000000,  
                B00000000,  
                B00000000,};  
  
#define pin_lm35 A5 //Pino o qual o sensor está conectado  
  
void setup() {  
  Serial.begin(9600); // Inicia a porta serial configurando a taxa de dados para 9600  
  pinMode(pin_lm35, INPUT); // Define o sensor como uma entrada de sinal  
  lcd.begin(16, 2); // Inicia o lcd de 16x2  
  lcd.clear();  
  lcd.createChar(0, grau);  
  
}  
float pre_temp = 0;  
void loop() {  
  float valor_analog_lm35 = float(analogRead(pin_lm35)); // obtém o valor analógico que varia de 0 a 1023 (0 = 0V, e 5V =1023)  
  float tensao = (valor_analog_lm35 * 5) / 1023; // vamos converter esse valor para tensão elétrica  
  float temperatura = tensao / 0.010; // dividimos a tensão por 0.010 que representa os 10 mV pois cada  $^{\circ}\text{C} = 10\text{mV}$ , Então  $10\text{mV}/^{\circ}\text{C}$   
  
  Serial.print("Temperatura atual = ");  
  Serial.print(temperatura); // Mostra na serial a temperatura do sensor  
  Serial.print("C");  
  Serial.println();  
  
  lcd.clear();  
  lcd.setCursor(3, 0); // 2 = 2 colunas para a direita. 1 = Segunda linha  
  lcd.print("ELETRODEX");  
  lcd.setCursor(0, 1); // 2 = 2 colunas para a direita. 1 = Segunda linha  
  lcd.print("Temp.: ");  
  lcd.setCursor(8, 1);  
  lcd.print(temperatura);  
  lcd.setCursor(13, 1);  
  lcd.write((byte)0); //Mostra o símbolo do grau formado pelo array  
  lcd.print("C");  
  delay(1000);  
  
  delay(1000); // aguarda 1 segundo  
}
```