



**UNIVERSIDADE FEDERAL DO PARÁ - UFPA**

**INSTITUTO DE TECNOLOGIA – ITEC**

**FACULDADE DE ENGENHARIA DA COMPUTAÇÃO**

**RELATÓRIO DO PROJETO DE MEDIDOR DE  
TEMPERATURA E LUMINOSIDADE AMBIENTES**

Bruno Santos de Sousa

Dilton Vilhena Brandão

Ivan Pedro Lima de Sousa

Izidio Sousa de Carvalho

Jamile Lima Leite

Marcelo Henrique de Paula dos Santos

**BELÉM - PARÁ**

**2016**

# Sumário

<b>1. Introdução</b>	<b>3</b>
<b>2. Esquemático do Circuito Eletrônico</b>	<b>4</b>
2.1. Ilustração da Placa Arduino	4
2.2. Ilustração do Circuito Eletrônico	4
<b>3. Memória de Cálculos</b>	<b>5</b>
3.1.1 Calculo da resistência em relação aos LEDs	5
3.1.2 Calculo da resistência em relação ao botão	5
3.1.3 Calculo da resistência em relação ao NTC	5
3.1.4 Calculo da resistência em relação ao LDR	6
<b>4. Descrição do Código Solução</b>	<b>7</b>
4.1. Explicação do Código Solução	7
4.2. Descrição Narrativa do Código Solução	7
<b>5. Conclusão</b>	<b>8</b>
<b>6. Anexo I: Código Solução</b>	<b>9</b>

## 1. Introdução

Este projeto consiste em um circuito que calcula a temperatura e a luminosidade do ambiente e as exibe em um LCD. A medição da temperatura é feita por um termistor NTC cujo valor é transformado em Celsius através de cálculos com o *Método do Fator Beta*, a medição da luminosidade é relativa, o que significa que seu valor é mostrado em percentual (de 0% a 100%). Essa medição é feita por um LDR cujos valores variam de 0 até 1023 e que através de cálculos são transformados em percentual, sendo 0 o valor mínimo (0%) e 1023 o valor máximo (100%).

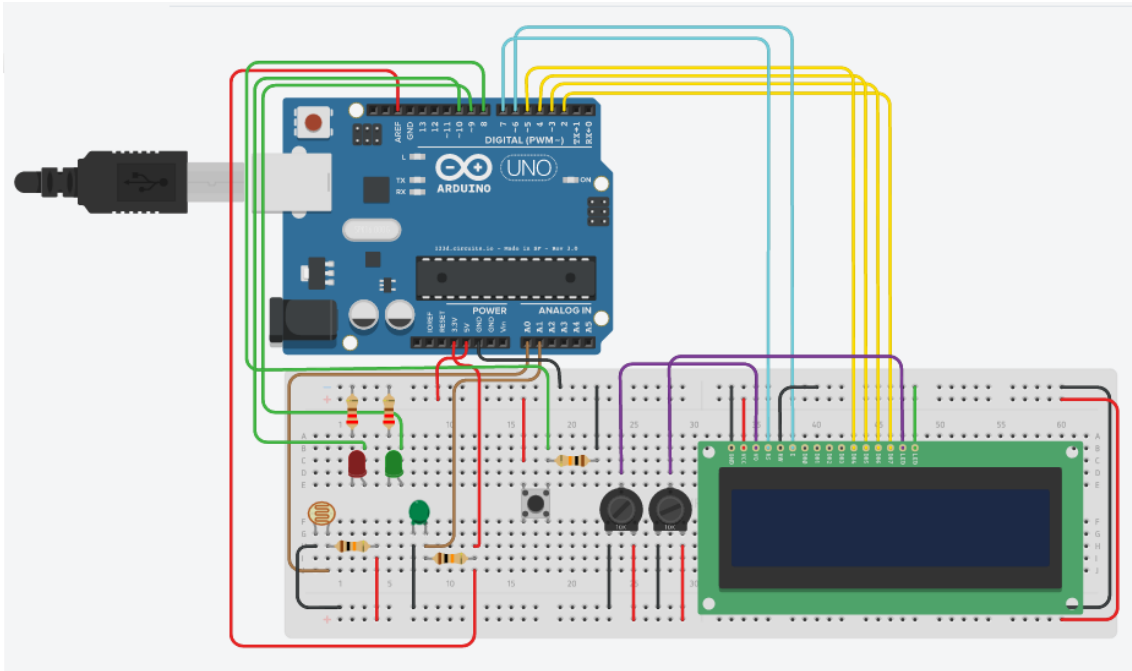
Há dois LEDs que acompanham a medição da luminosidade: um verde e um vermelho, o verde é acionado quando seu percentual for baixo (de 0% a 40%), e o vermelho quando seu percentual for alto (de 80% a 100%). Os dois serão acessos quando esse percentual for médio (quando estiver com o percentual entre o baixo e o alto).

O circuito também possui um botão que, enquanto pressionado, fixa o valor da temperatura e da luminosidade referentes ao momento em que foi inicialmente ativado.

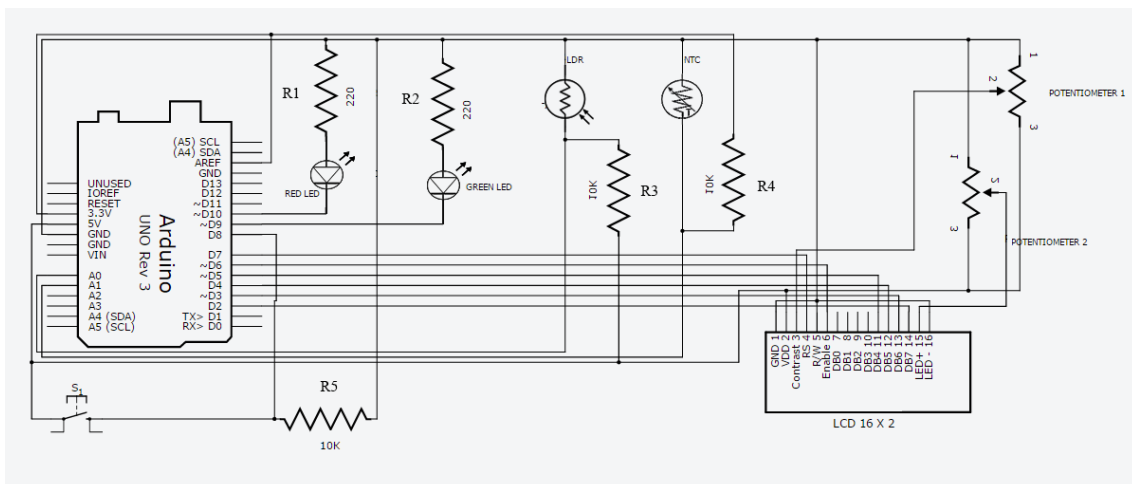
Para a montagem do circuito, foi preciso o uso de certos componentes além dos básicos já familiarizados, como o sensor de luminosidade (LDR), termistor (NTC), potenciômetro (10K $\Omega$ ) e o display LCD (16x2), que dá um diferencial ao projeto por exibir informações captadas pelo Arduino e melhorando a interação objeto-pessoa.

## 2. Esquemático do Circuito Eletrônico

2.1. Ilustração do Arduino com suas componentes utilizadas durante o projeto, obtida a partir de uma simulação na plataforma 123D Circuits (Autodesk 123D).



2.2. Esquemático do circuito elétrico, obtida a partir de uma simulação na plataforma 123D Circuits (Autodesk 123D).



### 3. Memória de Cálculos

$$R = (V_a - V_l)/i$$

**V<sub>a</sub>** [Tensão de alimentação do circuito em Volts]

**V<sub>l</sub>** [Tensão de funcionamento do componente, também em Volts]

**i** [Corrente de funcionamento do componente em Ampère]

**3.0.1. Observação:** para todos os componentes, a corrente de funcionamento será a mesma com exceção do NTC que usará tensão de 3,3V.

**3.0.2. Observação:** uma tensão de 5V é fornecida pelo circuito, e será igual para todos os LEDs, porém o termistor NTC funcionará com uma corrente de 3,3V enquanto o ajuste de brilho e contraste do LCD irá variar de acordo com o ajuste manual do potenciômetro e com isso a variação da tensão.

#### 3.1. Cálculo da resistência

##### 3.1.1. Em relação aos LEDs:

- Os LEDs funcionam a uma tensão de 2,0V e uma corrente elétrica de 20,0mA.

Logo, sendo  $i = 20\text{mA}$  e convertendo para Ampère, temos que  $i = 0,02\text{A}$ , então, pela fórmula apresentada anteriormente:

$$R = (5\text{V} - 2\text{V}) / 0.02\text{A} = 150\Omega$$

Essa será a resistência aproximada que deverá ser ligada em série com os LEDs. Nesse caso, será o valor dos resistores R1 e R2.

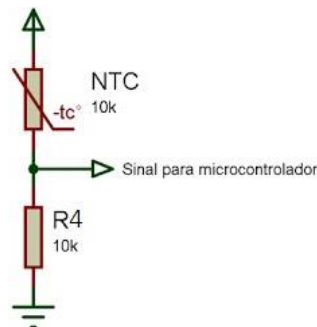
##### 3.1.2. Em relação ao botão:

- Foi usada uma resistência de  $10\text{K}\Omega$  para o botão, com o intuito de assegurar que a corrente chegue a entrada digital 08, pois a resistência de  $10\text{K}\Omega$  será maior que a resistência interna do próprio Arduino, esse será o valor do resistor R5.

##### 3.1.3. Em relação ao termistor NTC:

- Utilizando as características do sensor resistivo foi montado um circuito divisor de tensão com a adição de um resistor de  $10\text{K}\Omega$  em série ao NTC. Este divisor tem como tensão de entrada 5V – o limite permitido à entrada

analógica do microcontrolador – e o sinal de saída do divisor de tensão será injetado à entrada analógica do microcontrolador conforme a figura abaixo:



Para este circuito divisor de tensão determina-se a equação para o sinal de saída injetado ao microcontrolador da seguinte maneira:

$$i = \frac{VCC}{R4 + R_{NTC}}$$

Note que:

**i** é a corrente em *Ampère*;

**VCC** é a tensão fornecida pelo Arduino (5v);

**R4** é o resistor de 10KΩ;

**RNTC** é o resistor interno do NTC (10KΩ).

Usando a Formula acima podemos calcular a corrente, temos que:

$$i = 0,00025mA = 2,5 \cdot 10^{-4}mA = 2,5 \cdot 10^{-7}A$$

#### 3.1.4. Em relação ao Sensor de Luminosidade LDR:

- Será usada uma resistência de 10KΩ ao resistor R3, que funcionará como divisor de tensão. O valor de 10KΩ para R3 é justificada pelo fato de que possa ocorrer uma situação em que a luminosidade faça com que o LDR diminua bastante a sua resistência e com isso a porta do Arduino seja danificada.

## 4. Descrição do Código Solução

### 4.1. Explicação do Código Solução

O código faz leituras de luminosidade e temperatura através do valor oferecido pelo sensor LDR, e pelo NTC. Os valores lidos estão entre 0 e 1023, mas são convertidos para porcentagem (luminosidade) e graus Celsius (temperatura).

Um LED vermelho e outro verde acompanham o funcionamento do sensor de luminosidade, funcionando em três níveis: o baixo (de 0% a 40%), o médio (de 40% a 80%), e o alto (80% a 100%).

O cálculo do valor do termistor envolve o coeficiente beta, e a resistência, além disso utiliza uma tensão de referência, através do pino AREF da placa microcontroladora. O código cálculo é oferecido no site oficial do Arduino na sessão *playground*.

### 4.2. Descrição Narrativa do Código Solução

- O LCD acende, e os dois LEDs acendem;
- A função loop começa;
- Os níveis de luminosidade e a temperatura são mostrados no display. Enquanto isso os LEDs ficam variando de acordo a incidência de luz: de 0% a 40% acende somente o verde, de 40% a 80% acendem os dois, de 80% a 100% acende somente o vermelho);
- Caso o botão seja pressionado, chama a função printa;
- A função printa fixa no display os valores lidos no momento que o botão foi pressionado.;
- Os valores ficam fixados enquanto o botão estiver em nível logico alto. Quando o botão estiver em nível logico baixo, voltamos para a função loop.

## **5. Conclusão**

O projeto requereu maior uso das componentes disponíveis para o Arduino que, para serem bem aproveitadas, é necessário conhecimento sobre elas e seus funcionamentos. Tal aprendizado sobre esses componentes e o modo de usá-las nos possibilitou não somente a resolver o problema pedido pelo trabalho, mas também a combinar os itens e suas especificidades de inúmeras maneiras de forma a criarmos uma vasta possibilidade de uso para eles.

Com isso, levamos à prática o conhecimento adquirido em sala de aula, aprendendo várias funções das componentes usadas, o que nos apta a construir tanto o circuito quanto seu programa.



## 6. Anexo I: Código Solução

```
// Include das Bibliotecas
#include <LiquidCrystal.h>

// Definindo o Botão
const int botao = 8;

// Definindo os pinos da interface (LCD) (RS, Enable, D4,D5,D6,D7)
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

// Definindo as pinos dos Sensores
const float SensorDeLuz = 0; // Define o pino do sensor de luminosidade como o pino
analogico 0
const float SensorDeTemp = 1; // Define o pino do sensor de temperatura como o pino
analogico 1
float ValorSensorLuz = 0; // Variável auxiliar ao valor do sensor de luminosidade
float ValorSensorTemp = 0; // Variável auxiliar ao valor do sensor de temperatura

// Definindo os pinos dos LEDs
int LedVerde = 9; // LED vermelho do sensor de luz no pino 9
int LedVermelho = 10; // LED verde do sensor de luz no pino 10

// Definindo a temperatura para resistência nominal (25°C)
const float TemperaturaNominal = 25;

// Definindo o valor do termistor a 25 °C
const float TermistorNominal = 10000;

// Definindo a quantidade de amostras utilizadas para a media da temperatura
const int NumeroDeAmostras = 5;

// Definindo o coeficiente beta do termistor
const float CoeficienteBeta = 3435;

// Definindo o valor do Resistor em Serie com o Sensor de Temperatura (10K)
const float Resistor = 10000;

// Definindo as amostras como um vetor de valores inteiros
int Amostras[NumeroDeAmostras];

// Definindo as variáveis que auxiliarão na função printa
int i;
float MediaT;
float TemperaturaCelsius;
float LuminosidadePorcentagem;

// Inicialização da Interface
void setup() {
    // Definindo o número de colunas (16) e linhas (2) no LCD
    lcd.begin(16, 2);

    //Definindo os LEDS como saidas
    pinMode(LedVermelho, OUTPUT);
    pinMode(LedVerde, OUTPUT);

    // Definindo a taxa de comunicação em bits por segundo
    Serial.begin(9600);

    // Valor de tensão aplicada no pino AREF, usada como referência externa
    analogReference(EXTERNAL);

    delay(10);
}

// Função Loop
void loop() {
    // Chama a função printa se o botão for pressionado
    if (digitalRead (botao) == HIGH)
        printa();
    delay(10);

    // Ler o valor atual do Sensor de Luz
    ValorSensorLuz = analogRead(SensorDeLuz);
```

```

delay(10);

// Colocando em Porcentagem
ValorSensorLuz = (1023 - ValorSensorLuz);
LuminosidadePorcentagem = (ValorSensorLuz / 1023) * 100;

// Pega o valor do sensor de temperatura e armazena em amostras
for (i = 0; i < NumeroDeAmostras; i++) {
    Amostras[i] = analogRead(SensorDeTemp);
    delay(10);
}

// Media de Todas as amostras de temperatura
MediaT = 0;
for (i = 0; i < NumeroDeAmostras; i++) {
    MediaT += Amostras[i];
}
MediaT /= NumeroDeAmostras;

// Escreve a media na porta serial
Serial.print("Media da Temperatura ");
Serial.println(MediaT);

// Converte o valor para Resistência
MediaT = 1023 / MediaT - 1;
MediaT = Resistor / MediaT;

// Escreve a resistência na porta serial
Serial.print("Resistencia do Termistor: ");
Serial.println(MediaT);

// (1/T)=(1/To)+(1/B)ln(R/Ro)
// Calcula a Temperatura em Celsius
TemperaturaCelsius = MediaT / TermistorNominal; // (R/Ro)
TemperaturaCelsius = log(TemperaturaCelsius); // ln(R/Ro)
TemperaturaCelsius /= CoeficienteBeta; // 1/B * ln(R/Ro)
TemperaturaCelsius += 1.0 / (TemperaturaNominal + 273.15); // + (1/To)
TemperaturaCelsius = 1.0 / TemperaturaCelsius; // Inverte
TemperaturaCelsius -= 273.15; // Converte para C°

// Escreve a temperatura na porta serial
Serial.print("Temperatura em Celsius ");
Serial.print(TemperaturaCelsius);
Serial.println("C");

// Escreve a luminosidade na porta serial
Serial.print("Luminosidade Relativa ");
Serial.println(LuminosidadePorcentagem);
delay(10);

// Condições para que os LEDs que indicam os níveis de luminosidade acendam e apaguem
// Nível Baixo: de 0% a 40% (Acende só o LED verde)
// Nível Médio: de 40% a 80% (Acende o LED vermelho e o verde)
// Nível Alto: de 80% a 100% (Acende só o LED vermelho)
if (LuminosidadePorcentagem > 40) {
    digitalWrite(LedVerde, LOW);
    if (LuminosidadePorcentagem >= 40 && LuminosidadePorcentagem < 80) {
        digitalWrite(LedVerde, HIGH);
        digitalWrite(LedVermelho, HIGH);
    }
    if (LuminosidadePorcentagem >= 80) {
        digitalWrite(LedVermelho, HIGH);
        digitalWrite(LedVerde, LOW);
    }
}
else {
    digitalWrite(LedVerde, HIGH);
    digitalWrite(LedVermelho, LOW);
}

// Escreve o valor da Temperatura
lcd.setCursor(0, 0); //Define o cursor na 1ª linha e 1ª coluna
lcd.print("Temp.: ");
lcd.print(TemperaturaCelsius);
lcd.write(B11011111);
lcd.print("C");

```

```

// Escreve o valor da Luminosidade
lcd.setCursor(0, 1); //Define o cursor na 2° linha e 1° coluna
lcd.print("Lumi.: ");
lcd.print(LuminosidadePorcentagem);
lcd.print("%");
delay(1000);

}

// Função que escreve os valores de temperatura e luminosidade
void printa() {

    // Escreve o valor da Temperatura
    lcd.setCursor(0, 0); //Define o cursor na 1° linha e 1° coluna
    lcd.print("Temp.: ");
    lcd.print(TemperaturaCelsius);
    lcd.write(B11011111);
    lcd.print("C");

    // Escreve o valor da Luminosidade
    lcd.setCursor(0, 1); //Define o cursor na 2° linha e 1° coluna
    lcd.print("Lumi.: ");
    lcd.print(LuminosidadePorcentagem);
    lcd.print("%");
    delay(1000);

    // Se o botão continuar pressionado ele fica "preso" dentro da função printa
    // Se o botão for solto voltamos para a função loop
    if (digitalRead (botao) == HIGH)
        printa();
    else
        loop();
    delay(10);
}

```