SENSOR DE LUMINOSIDADE

Com indicador de intervalo

ARDUINO UNO R3

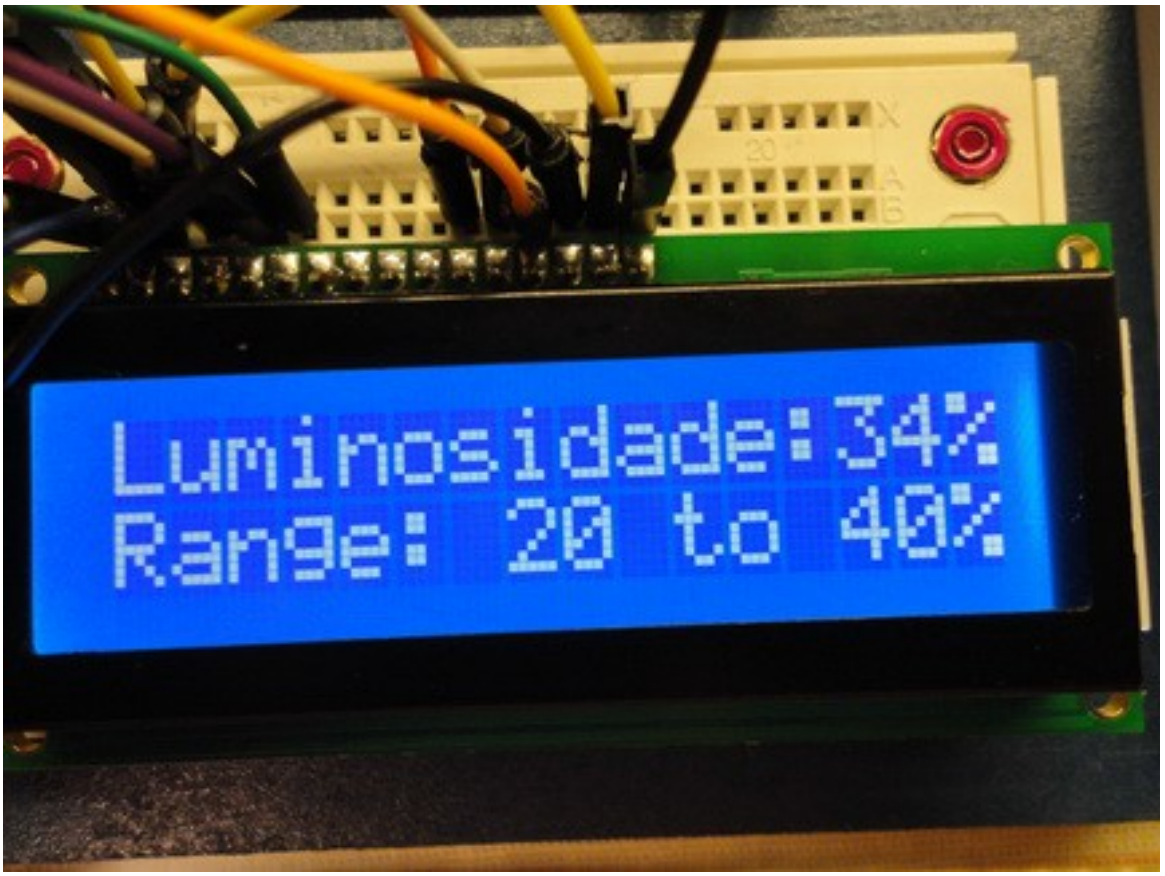
Este projeto refere-se à construção de um sensor de luminosidade com o uso de um LDR (Light Dependent Resistor), com as seguintes características:

1. Luminosidade especificada em percentual;

2. Range (intervalo) de luminosidade especificado entre 20 a 50%;

3. Dois leds indicadores, sendo um quando a luminosidade estiver abaixo de 20% (led azul) e outro quando a luminosidade estiver acima de 40% (led vermelho).

Isto significa que os leds estarão apagados somente quando a luminosidade estiver dentro de um intervalo de 20 a 40%.

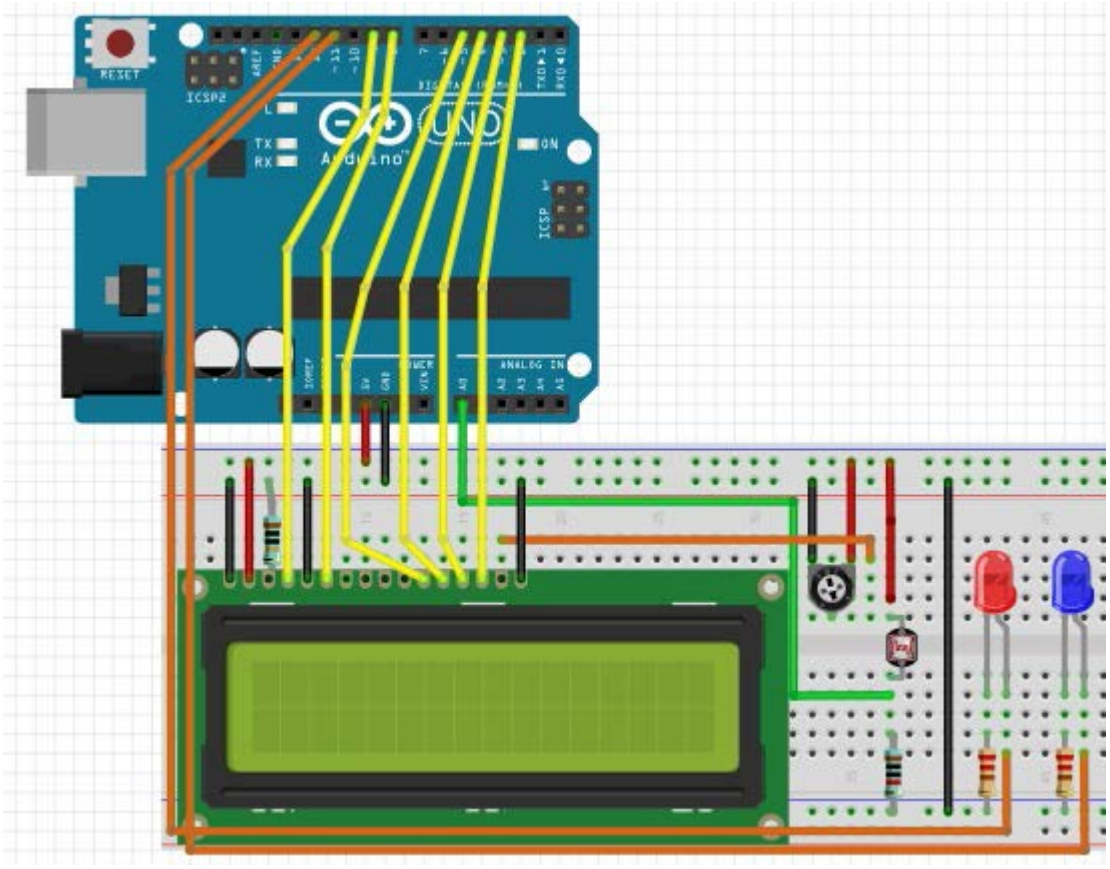


O sensor de luminosidade usado é um LDR, cuja resistência apresenta um coeficiente negativo, ou seja, à medida que a luminosidade aumenta a sua resistência diminui.

O sinal proveniente do LDR será introduzido em uma das entradas analógicas do Arduino: A0~A5.

A figura acima ilustra um LDR (também conhecido como foto resistor) largamente utilizado como sensor de luminosidade.

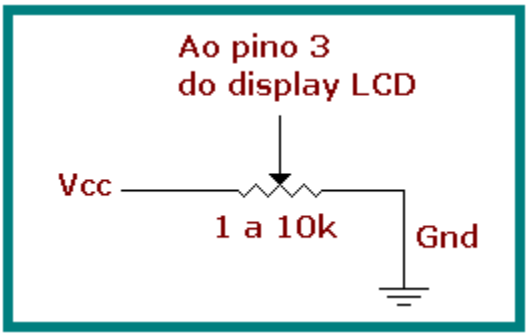
A figura a seguir ilustra o projeto desenvolvido no *Fritzing*.



O trimpot (valor de 1 a 10k) ajusta a luminosidade de fundo (backlight) do display. *No layout mostrado acima o pino 15 do display está desconectado por ser uma ligação opcional. O pino 15 poderá ser conectado diretamente ao VCC ou ao pino central do trimpot, permitindo assim o ajuste do brilho de fundo do display.*

O contraste é fixado no pino 3 do display, através de um resistor de 10 ohms, no entanto, dependendo do modelo do display, o pino 3 poderá ser ligado diretamente ao terra.

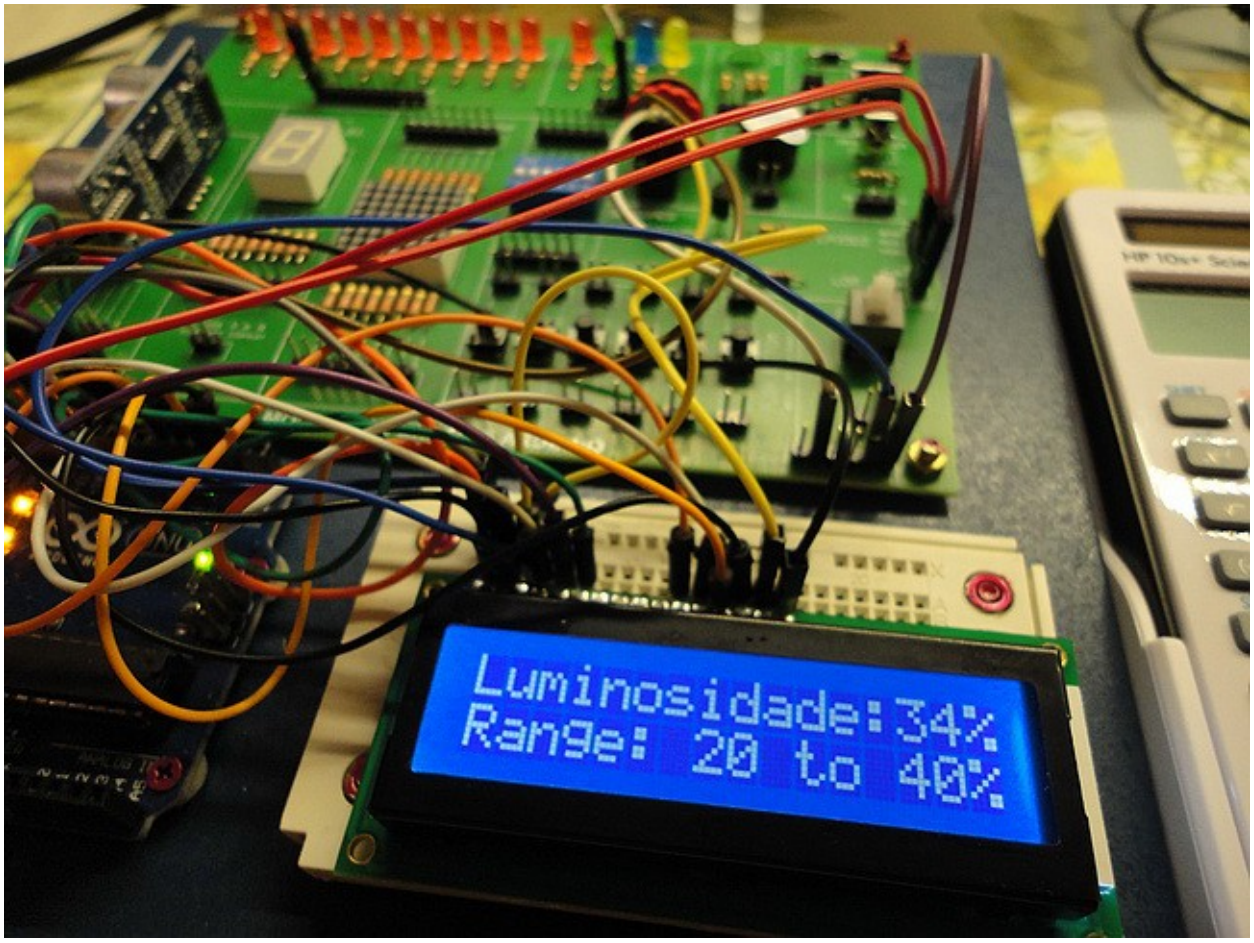
O contraste também pode ser ajustado através de um potenciômetro, conforme indica o diagrama:



O sinal do LDR para o pino A0 do Arduino é referenciado ao terra através de um resistor de 10kΩ.

Os resistores de 220Ω que fecham o circuito dos *leds* ao terra atuam como limitadores de corrente.

Veja abaixo o aspecto da montagem do projeto no Módulo de Ensaios Arduino.

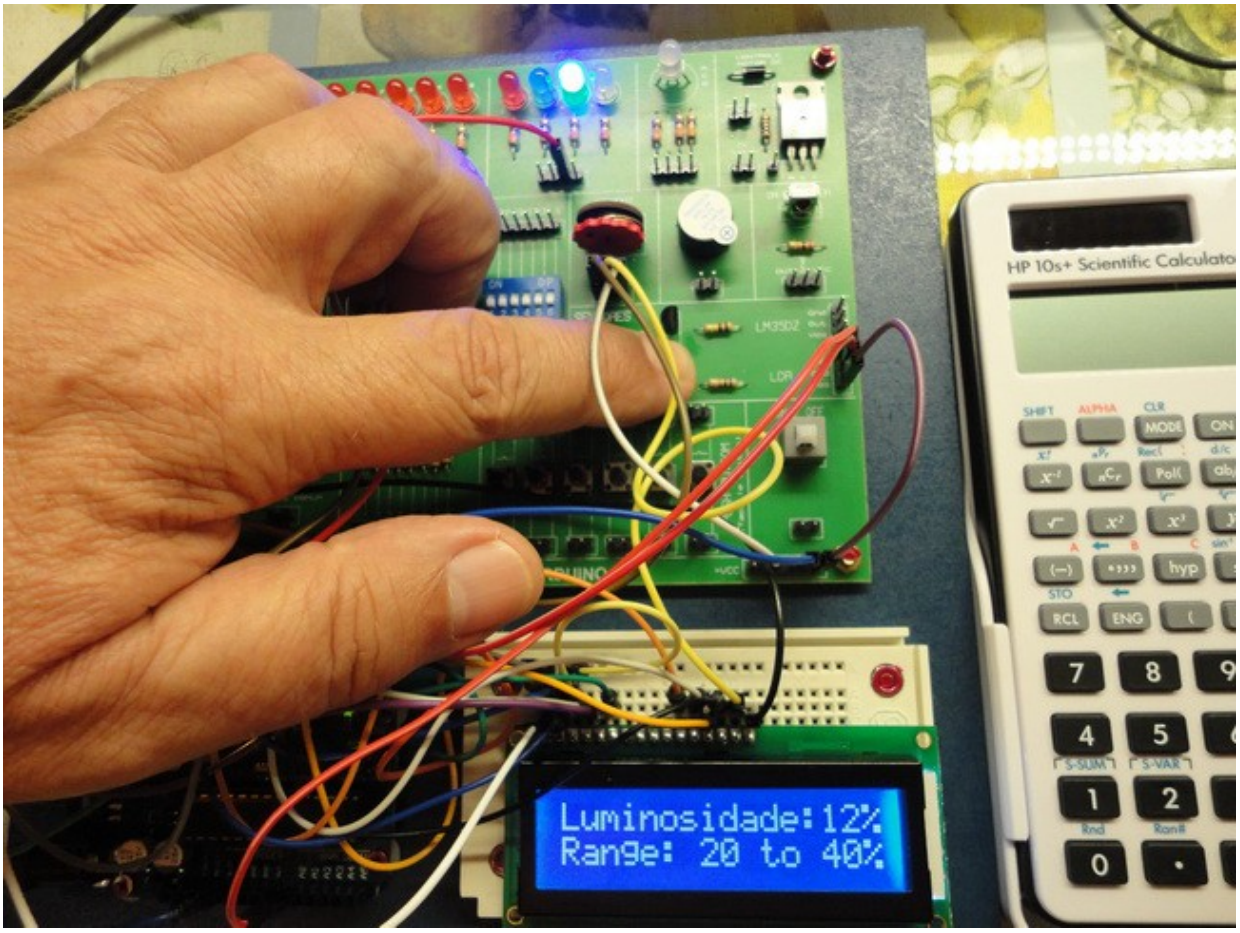


Observe que a luminosidade indica 34%, portanto, dentro do intervalo especificado e nenhum led acende.

O intervalo (range) é especificado pelas linhas de programação:

```
if (lumValor>=40) digitalWrite(ledPinVM,HIGH);
else digitalWrite(ledPinVM,LOW);
if (lumValor<=20) digitalWrite(ledPinAZ,HIGH);
else digitalWrite(ledPinAZ,LOW);
```

A figura a seguir mostra a diminuição da incidência luminosa sobre o LDR, diminuindo o percentual de luminosidade para 12%, acendendo o led azul.



Criando o símbolo (caractere) de porcentagem:

```
byte x[8]={B11000,B11001,B00010,B00100,B01000,B10011,B00011,B00000};
```

Os códigos acima referem-se a 8 linhas e 5 colunas. As linhas são definidas por:

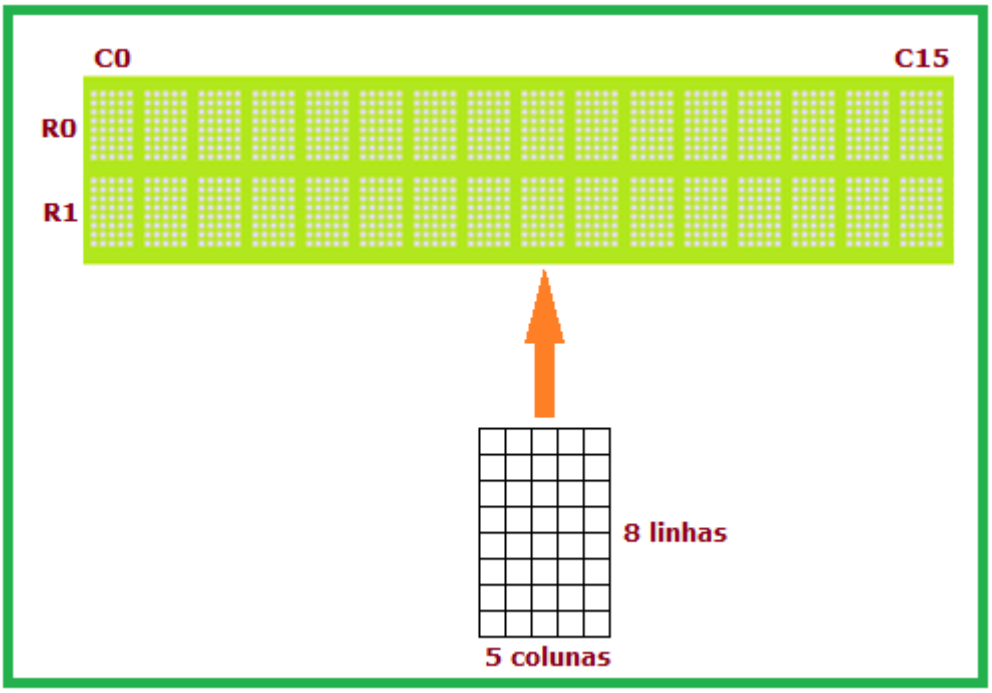
Byte x[8]

Enquanto que as colunas são definidas por B11000, B11001...

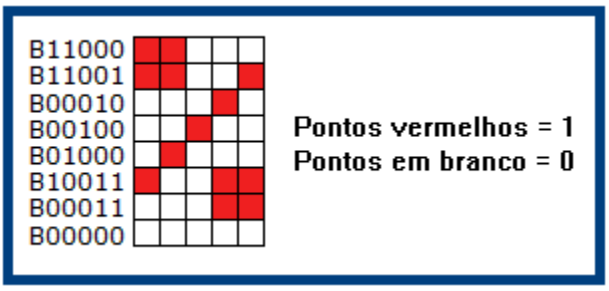
Para melhor entender, vamos mostrar os passos para criar o símbolo de porcentagem.

Esse procedimento vale para a criação de qualquer caractere.

O display 16x2 usado neste projeto, possui 16 colunas (0 a 15) e 2 linhas (0 e 1) e cada caractere é impresso em uma matriz com pontos, com 8 linhas e 5 colunas, totalizando 40 pontos, conforme ilustra a figura a seguir:



Criando o caractere da porcentagem:



Para que o caractere seja escrito no display, as linhas abaixo deverão ser acrescentadas:

lcd.createChar(1,x); ➔ cria o caractere
lcd.setCursor(15,0); ➔ posiciona o cursor
lcd.write(1); ➔ escreve o caractere no display

Programação:

```
#include <LiquidCrystal.h> // biblioteca do display

int ldrPin=0; // define a entrada analógica do LDR (neste caso, A0)
int ledPinVM=12; // define o pino para alimentar o led vermelho
int ledPinAZ=11; // define o pino para alimentar o led azul
int ldrValor=0; // leitura do valor lido do LDR
```

```
int lumValor=0; // armazena a leitura do LDR (luminosidade)
LiquidCrystal lcd(9,8,5,4,3,2); //pinos que serão ligados no display LCD
byte x[8]={B11000,B11001,B00010,B00100,B01000,B10011,B00011,B00000}; // cria o
caractere de porcentagem

void setup() {
pinMode(ledPinVM,OUTPUT); // define o pino 12 do Arduino como saída
pinMode(ledPinAZ,OUTPUT);  // define o pino 11 do Arduino como saída
Serial.begin(9600); //inicia a comunicação serial
lcd.begin(16,2); // inicia e identifica o tipo de display (16 colunas, 2 linhas)
lcd.print("Luminosidade: ");
lcd.setCursor(0,1); // ajusta a posição do cursor
lcd.print("Range: 20 to 40%");
lcd.createChar(1,x);  // identifica o caractere de porcentagem criado anteriormente
lcd.setCursor(15,0);
lcd.write(1); // escreve o caractere porcentagem
}

void loop() {
ldrValor=analogRead(ldrPin)/4; // divide 1024 por 4 (conversão 10-8 bits)
lumValor=(analogRead(ldrPin)/4)*100/512; // calcula a porcentagem da luminosidade
if (lumValor>=40) digitalWrite(ledPinVM,HIGH); // define o limiar máximo da luminosidade
else digitalWrite(ledPinVM,LOW);
if (lumValor<=20) digitalWrite(ledPinAZ,HIGH); // define o limiar mínimo da luminosidade
else digitalWrite(ledPinAZ,LOW);
Serial.println(ldrValor); // monitoramento pelo serial monitor
Serial.println(lumValor); // monitoramento pelo serial monitor
lcd.setCursor(13,0); // posiciona o cursor
lcd.print(lumValor); // escreve no display
delay(200);

}
```