

Projeto Arduino para alimentação automatizada de animais.

Matheus de Souza Amorim, Eduardo Scheibner.

Escolas e Faculdades QI – Análise e desenvolvimento de sistemas –
Organização de Computadores
Gravataí – RS – Brasil

Souzasantos.matheus@yahoo.com.br

Eduardoscheibner@hotmail.com

***Abstract.** This article describes the use of Arduino and its components in the development of an automated feeder for animal feed. In this article we will cover the types of materials used and their functions.*

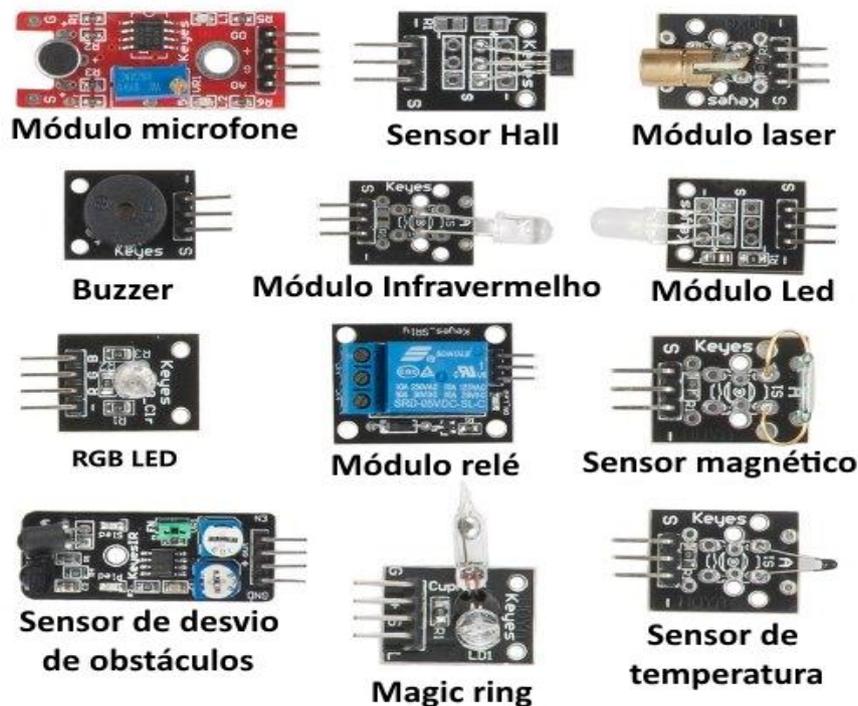
***Resumo.** Este artigo descreve a utilização do Arduino e seus componentes no desenvolvimento de um alimentador automatizado para alimentação de animais, neste artigo abordaremos os tipos de materiais usados e suas funções, abordaremos também os tipos de códigos aplicados..*

Depois de programado, o microcontrolador Arduino pode ser usado de forma independente, ou seja, você pode colocá-lo para controlar um robô, uma lixeira, um ventilador, as luzes da sua casa, a temperatura do ar condicionado, pode utilizá-lo como um aparelho de medição ou qualquer outro projeto que vier à cabeça.

O que você pode fazer com o Arduino?

A lista de possibilidades é praticamente infinita. Você pode automatizar sua casa, seu carro, seu escritório, criar um novo brinquedo, um novo equipamento ou melhorar um já existente. Tudo vai depender da sua criatividade.

Para isso, o Arduino possui uma quantidade enorme de sensores e componentes que você pode utilizar nos seus projetos. Grande parte do material utilizado no Arduino está disponível em módulos, que são pequenas placas que contém os sensores e outros componentes auxiliares como resistores, capacitores e leds.



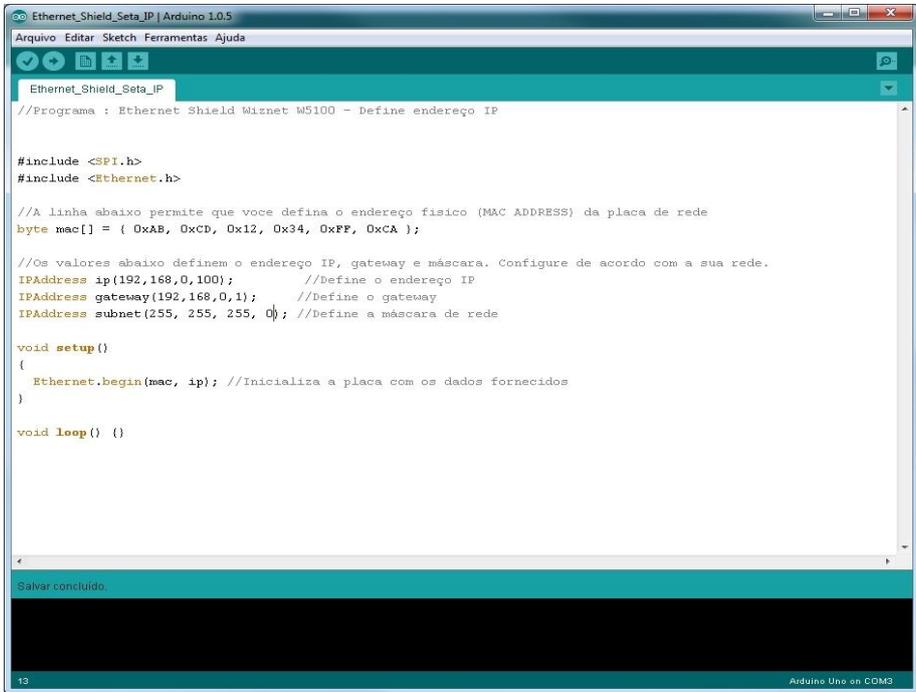
(Na imagem: Módulos extras para Arduino)

Existem também os chamados Shields, que são placas que você encaixa no Arduino para expandir suas funcionalidades. A imagem abaixo mostra um **Arduino Ethernet Shield** encaixado no Arduino Mega 2560. Ao mesmo tempo que permite o acesso do Arduino à uma rede ou até mesmo à internet, mantém os demais pinos disponíveis para utilização, assim você consegue, por exemplo, utilizar os pinos para receber dados de temperatura e umidade de um ambiente, e consultar esses dados de qualquer lugar do planeta.

Estrutura de um programa em Arduino

Escrever um programa em Arduino é muito simples. Tudo o que você precisa é conectar o Arduino ao computador por meio de um cabo USB e utilizar um ambiente de programação chamado IDE, onde você digita o programa, faz os testes para encontrar eventuais erros e transfere o programa para o Arduino.

Na imagem abaixo temos a IDE já com um programa carregado. No site oficial do Arduino (arduino.cc), você pode fazer o download da IDE gratuitamente:



```
Arduino IDE - Ethernet_Shield_Seta_IP | Arduino 1.0.5
Arquivo Editar Sketch Ferramentas Ajuda
Ethernet_Shield_Seta_IP
//Programa : Ethernet Shield Wiznet W5100 - Define endereço IP

#include <SPI.h>
#include <Ethernet.h>

//A linha abaixo permite que voce defina o endereço fisico (MAC ADDRESS) da placa de rede
byte mac[] = { 0xAB, 0xCD, 0x12, 0x34, 0xFF, 0xCA };

//Os valores abaixo definem o endereço IP, gateway e máscara. Configure de acordo com a sua rede.
IPAddress ip(192,168,0,100); //Define o endereço IP
IPAddress gateway(192,168,0,1); //Define o gateway
IPAddress subnet(255, 255, 255, 0); //Define a máscara de rede

void setup()
{
  Ethernet.begin(mac, ip); //Inicializa a placa com os dados fornecidos
}

void loop() {}

Salvar concluído.
13 Arduino Uno on COM3
```

(Na imagem: IDE Arduino em utilização.)

Uma vez feito o programa, basta transferí-lo para o Arduino e o mesmo começa a funcionar.

Você não precisa ser expert em linguagem C para programar o Arduino. Além da grande quantidade de exemplos que você encontra aqui no blog, você pode começar um programa utilizando a estrutura básica do Arduino, que é composta por duas partes, ou dois blocos.

setup() – É nessa parte do programa que você configura as opções iniciais do seu programa: os valores iniciais de uma variável, se uma porta será utilizada como entrada ou saída, mensagens para o usuário, etc.

loop() – Essa parte do programa repete uma estrutura de comandos de forma contínua ou até que alguma comando de “parar” seja enviado ao Arduino.

Projeto Arduino para alimentação de cães de forma automatizada.

Componentes:

01 - Placa Uno
01 - Cabo USB
01 - Protoboard
01 - Sensor de Luz LDR
01 – LED azul*
01 – Resistor
03 – Jumpers

- O LED nesse caso foi utilizado para simular um servo, cujo papel seria de abrir o dispenser de alimento.

Placa Uno e Cabo USB:

O Arduino Uno é uma placa de microcontrolador baseado no ATmega328 ([datasheet](#)). Ele tem 14 pinos de entrada/saída digital (dos quais 6 podem ser

usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP e um botão de reset. Ele contém todos os componentes necessários para suportar o microcontrolador, simplesmente conecte a um computador pela porta USB ou alimente com uma fonte ou com uma bateria e tudo pronto para começar.



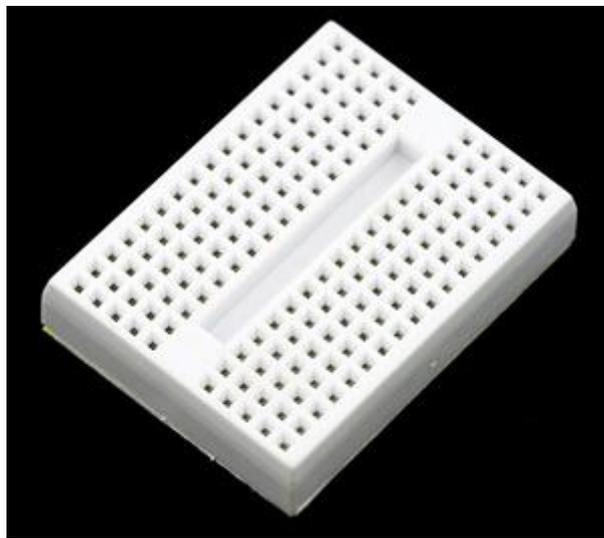
(Na imagem: Placa UNO arduino.)

Protoboard

As protoboards talvez sejam umas das ferramentas mais importantes para quem esteja começando com eletrônica e montagem de circuitos, pois com ela é possível montar dezenas de circuitos sem a necessidade de soldar qualquer componente. Neste tutorial apresentaremos o porque utilizar e como utilizar

uma protoboard.

Como dissemos acima, a grande mágica das protoboards é a possibilidade de montar, desenvolver e testar diversos circuitos eletrônicos, tudo sem ter que soldar os componentes. Sendo assim, se você não tem certeza de como um determinado circuito irá se comportar durante seu funcionamento, a protoboard é o lugar mais recomendado para montar este circuito e efetuar todos os testes necessários. Uma outra utilização muito comum é interligar sensores e circuitos integrados (CIs) aos diversos microcontroladores disponíveis como, por exemplo, o Arduino.



(Na imagem: Protoboard Arduino básica.)

Sensor de luminosidade

O LDR (Light Dependent Resistor) é um componente cuja resistência varia de acordo com a intensidade da luz. Quanto mais luz incidir sobre o componente, menor a resistência. Este sensor de luminosidade pode ser utilizado em projetos com arduino e outros microcontroladores para alarmes, automação residencial, sensores de presença e etc.



(Na imagem: Sensor de luminosidade para Arduino.)

LED



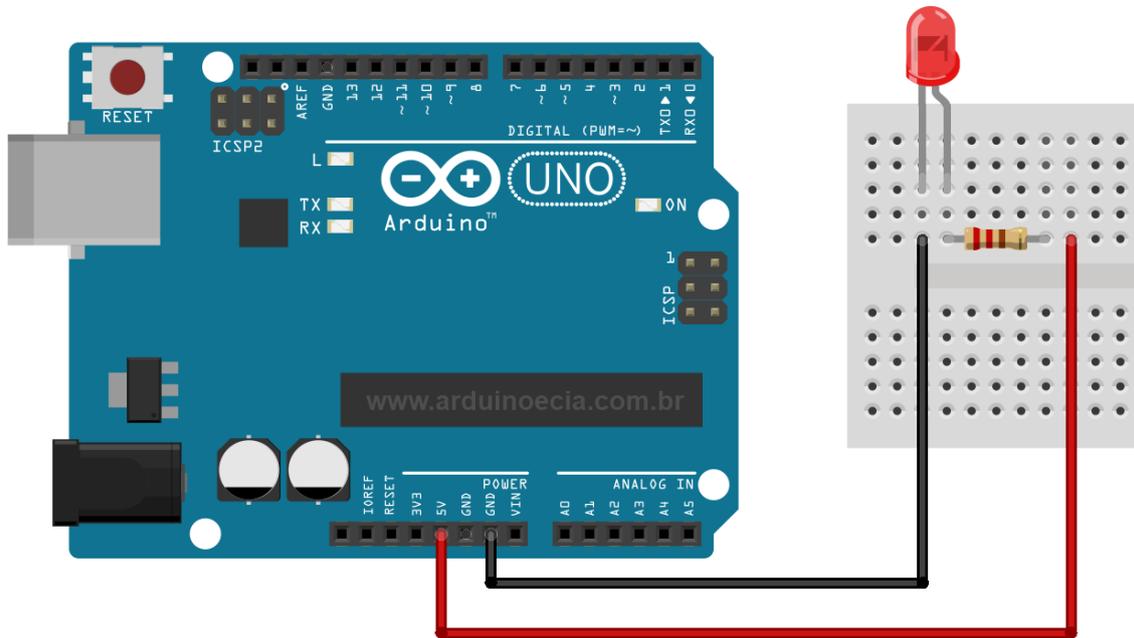
(Na imagem: LED azul para Arduino.)

Resistores

Um componente muito utilizado nos circuitos do Arduino são os **resistores**. Eles são utilizados principalmente para limitar a corrente em determinadas partes do circuito, evitando queima de outros componentes como leds, circuitos integrados e displays.

Se usarmos um led, por exemplo, que trabalha com 2 volts, e ligarmos esse led

diretamente nos 5 volts do Arduino, provavelmente iremos danificá-lo. O uso de uma resistência em série com o led limita a corrente no componente, evitando a queima do mesmo, como nesse pequeno exemplo :



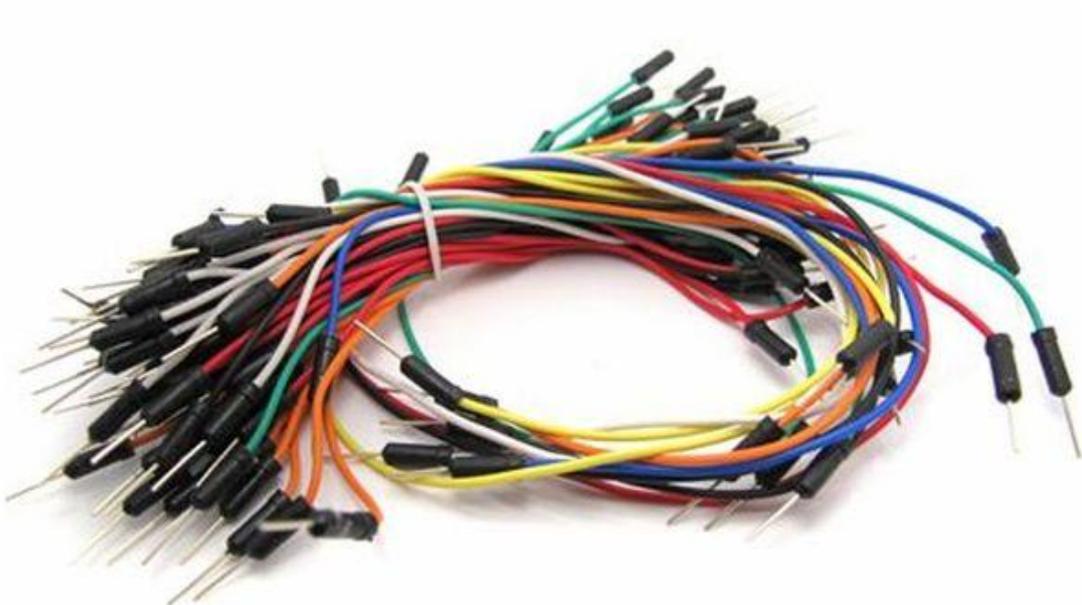
(Na imagem: Exemplo de conexão Arduino utilizando uma placa UNO, uma protoboard, um LED, dois jumpers e um resistor.)



(Na imagem: Resistor para Arduino.)

Cabos Jumpers

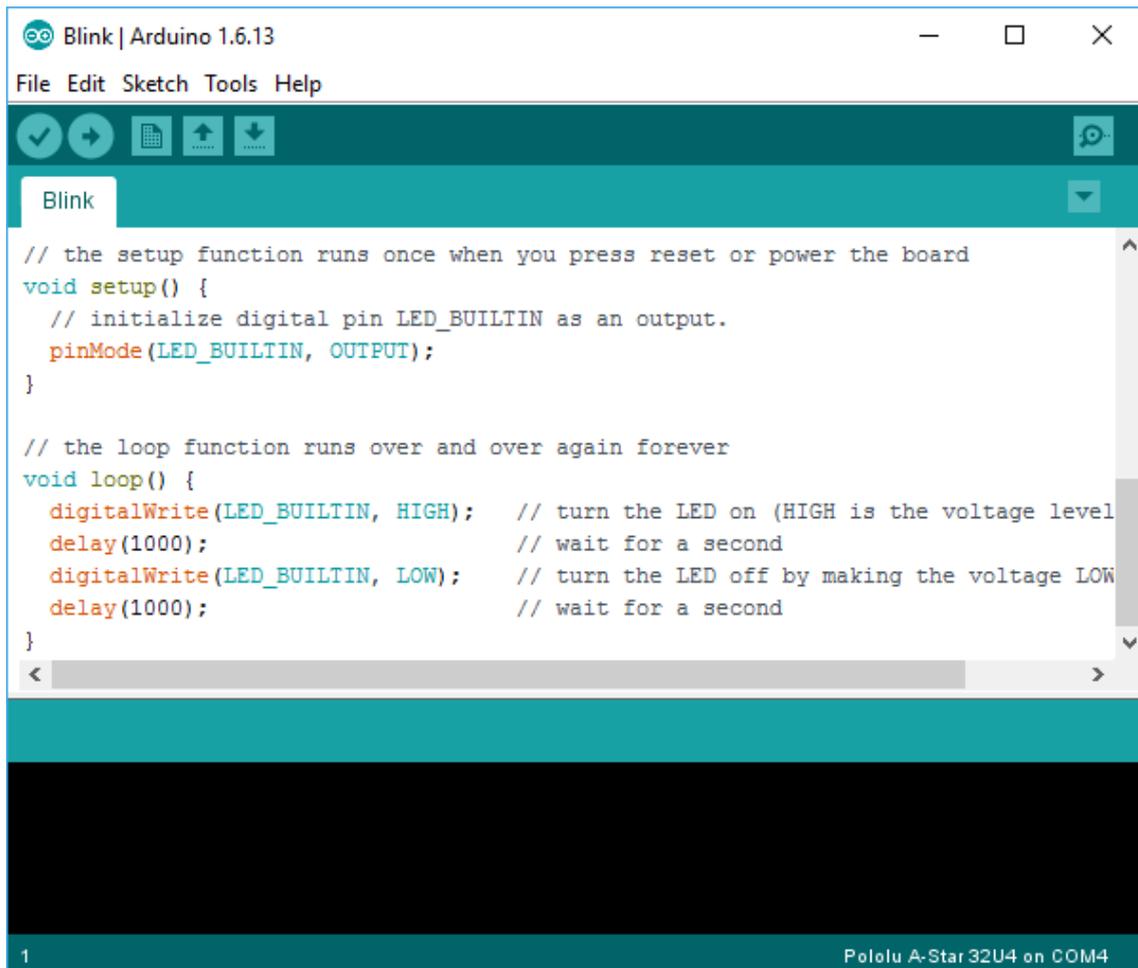
Os Cabos Jumpers Macho x Macho foram desenvolvidos para ligação entre sensores bricks e módulos externos as placas Arduino.



(Na imagem: Jumpers de diferentes cores e tamanhos para Arduino.)

A IDE Arduino para programação da placa UNO.

A IDE Arduino é distribuída através do site oficial(www.arduino.cc) de forma gratuita, sendo uma ferramenta de fácil utilização como mostra a imagem abaixo.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.6.13". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, uploading, and downloading. The main editor area shows the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

The status bar at the bottom indicates "1" on the left and "Pololu A-Star 32U4 on COM4" on the right.

(Na imagem: IDE Arduino em utilização.)

Em nosso projeto utilizamos um LED para simular um servo, para normalmente ligar um LED utilizamos o código a baixo:

```

void setup()
{
  // Configura o pino d13 como saída
  pinMode(13, OUTPUT);
}

void loop()
{
  // Configura o pino 13 como HIGH
  digitalWrite(13, HIGH);
  // Espera 1000 ms (1 segundo)
  delay(1000);
  // Configura o pino 13 como LOW
  digitalWrite(13, LOW);
  // Espera 1000 ms (1 segundo)
  delay(1000);
}

```

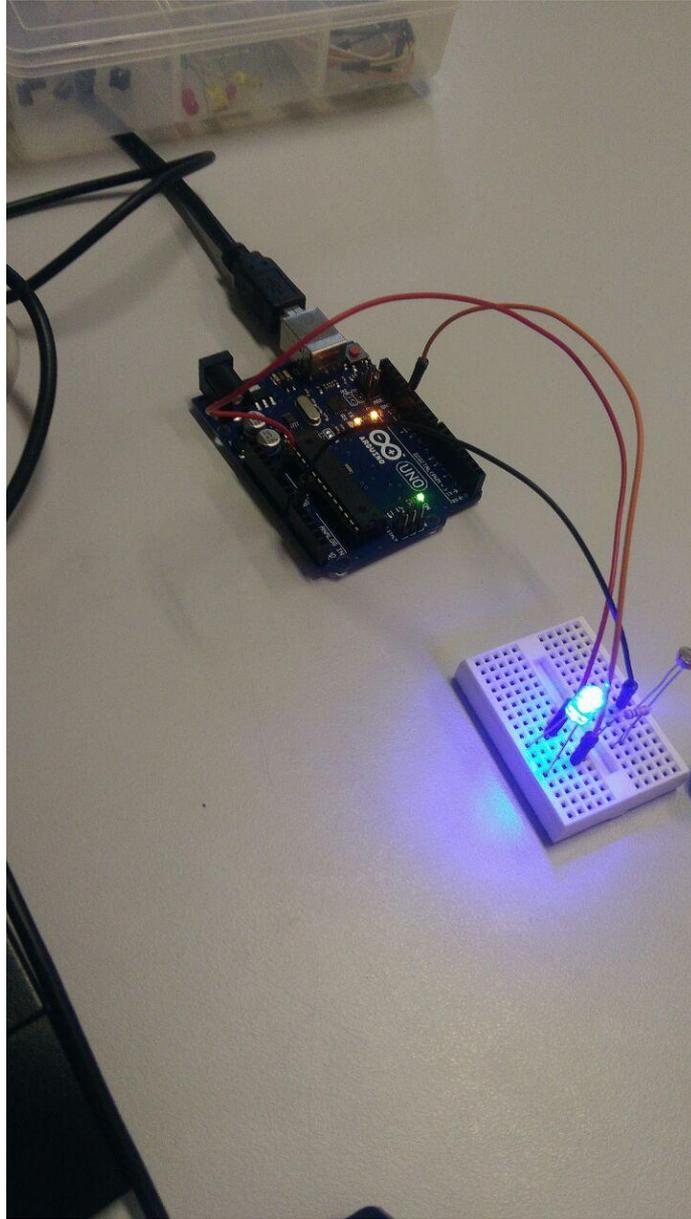
Para utilização do Sensor de Luminosidade, normalmente utilizamos uma codificação padrão como a da imagem abaixo:

```

1 //Sensor de luz com LDR
2
3 int ledPin = 7; //Led no pino 7
4 int ldrPin = 0; //LDR no pino analógico 8
5 int ldrValor = 0; //Valor lido do LDR
6
7 void setup() {
8   pinMode(ledPin,OUTPUT); //define a porta 7 como saída
9   Serial.begin(9600); //Inicia a comunicação serial
10 }
11
12 void loop() {
13   ///ler o valor do LDR
14   ldrValor = analogRead(ldrPin); //O valor lido será entre 0 e 1023
15
16   //se o valor lido for maior que 500, liga o led
17   if (ldrValor >= 800) digitalWrite(ledPin,HIGH);
18   // senão, apaga o led
19   else digitalWrite(ledPin,LOW);
20
21   //imprime o valor lido do LDR no monitor serial
22   Serial.println(ldrValor);
23   delay(100);
24 }

```

Projeto aplicado



Na foto podemos ver a ligação da placa UNO na Protoboard através de jumpers, o LED azul ligado informando que há pouca luz no ambiente.

A utilização do sensor de luz se dá pela necessidade de saber quando não há alimento no alimentador, com a falta do alimento o sensor fica exposto a luz ambiente e o dispenser se abre.

Bibliografia:

wikipedia (<https://pt.wikipedia.org/>)

Site oficial IDE Arduino (<https://www.arduino.cc/>)

filipeflop (<https://www.filipeflop.com>)

Circuitar (<https://www.circuitar.com.br/>)