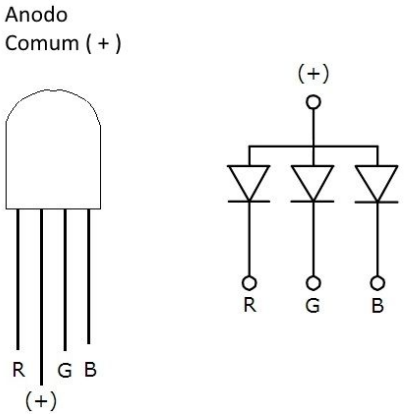
Arduino e Shields

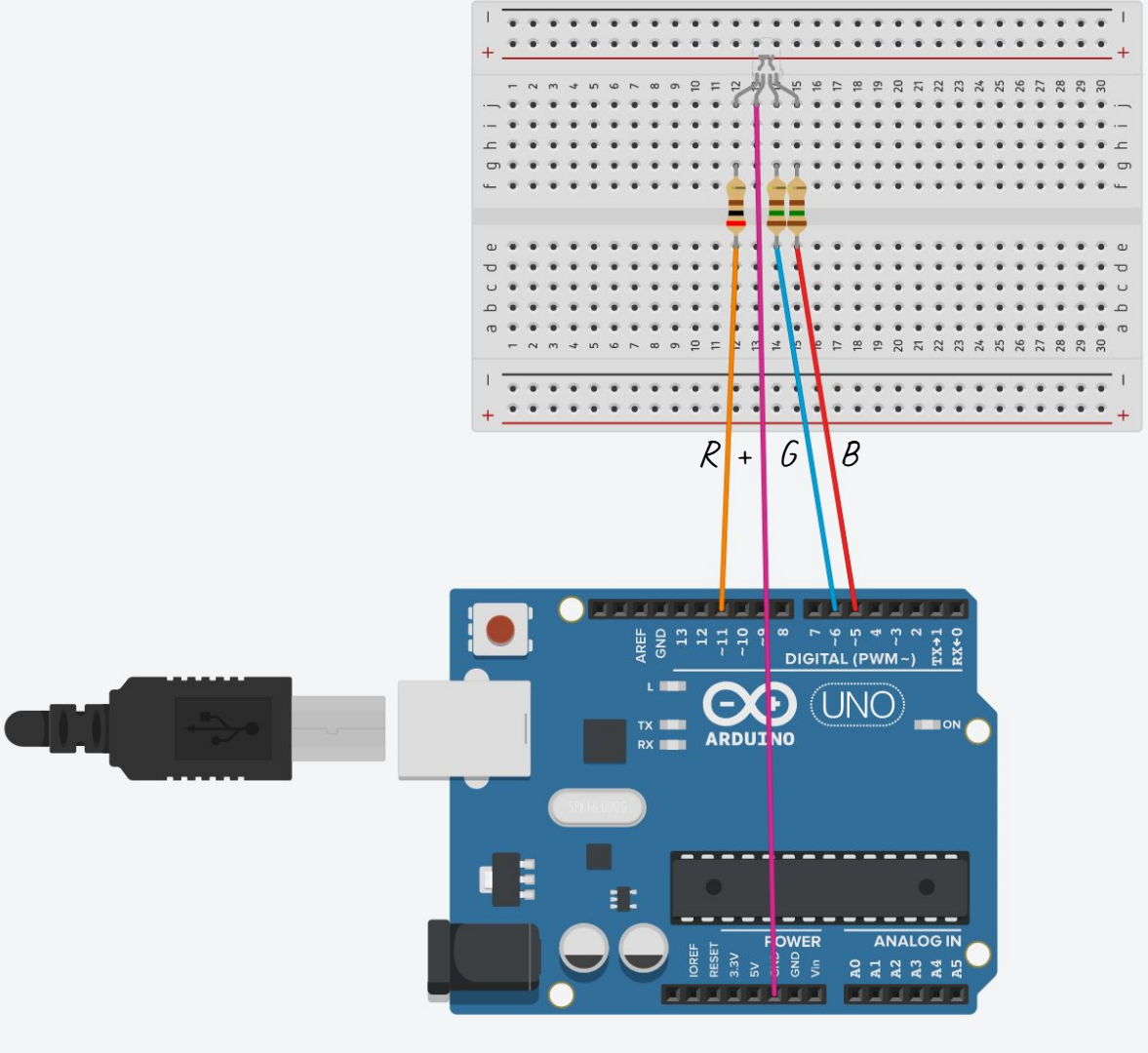
Prof. André Luis Meneses Silva

Trabalhando com LED RGB

Luz emitida: Vermelho, verde ou azul

- Vermelho – Tensão: 1.8 – 2.0V
- Verde – Tensão :3.2 – 3.4V
- Azul – Tensão :3.2 – 3.4V




R + G B
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
Vin
A0
A1
A2
A3
A4
A5

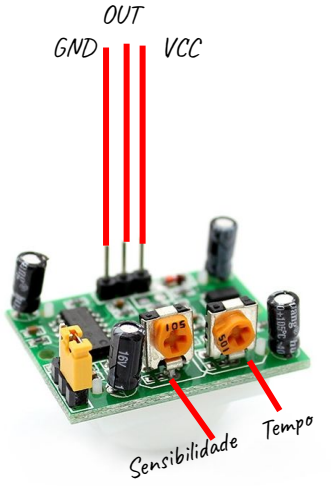
Código

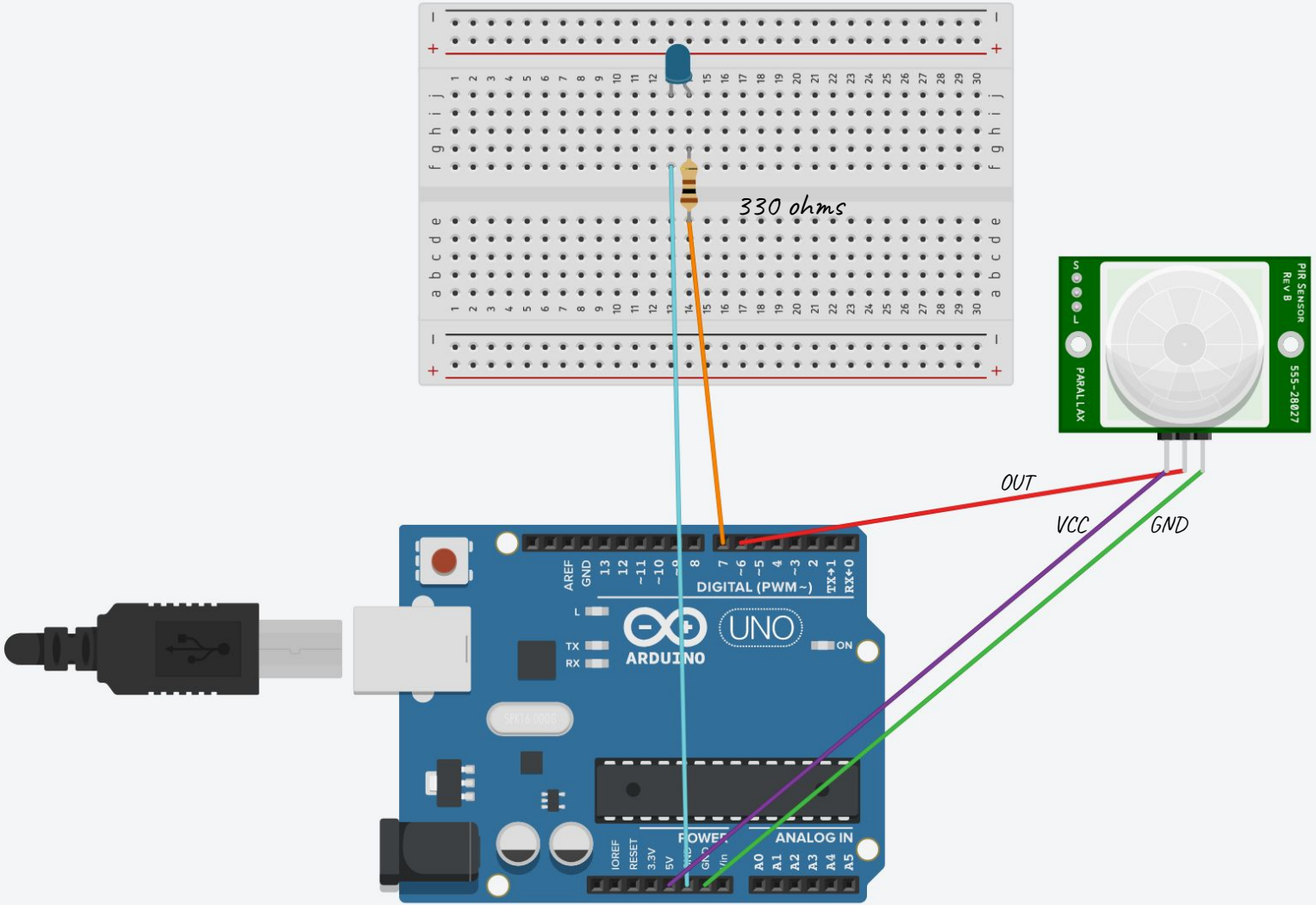
```
void setup() {
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}
```

```
void loop() {
  analogWrite(3, 220);
  delay(1000);
  analogWrite(3, 255);
  delay(1000);
  analogWrite(5, 220);
  delay(1000);
  analogWrite(5, 255);
  delay(1000);
  analogWrite(6, 220);
  delay(1000);
  analogWrite(6, 255);
  delay(1000);
}
```

Trabalhando com o sensor de presença

- Sensor de Movimento PIR DYP-ME003 detecta o movimento de objetos que estejam em uma área de até 7 metros
- Caso algo ou alguém se movimente nesta área o pino de alarme é ativado.
- Características:
  - Tensão de Operação: 4,5-20V
  - Tensão Dados: 3,3V (Alto) – 0V (Baixo)
  - Distância detectável: 3-7m (Ajustável)
  - Tempo de Delay: 5-200seg (Default: 5seg)
  - Tempo de Bloqueio: 2,5seg (Default)
  - Trigger: (L)-Não Repetível (H)-Repetível (Default: H)




330 ohms
OUT
VCC
GND
PIR Sensor
Rev B
PARALLAX
555-28027
DIGITAL (PWM~)
ARDUINO
UNO
ON
TX
RX
POWER
ANALOG IN

Código

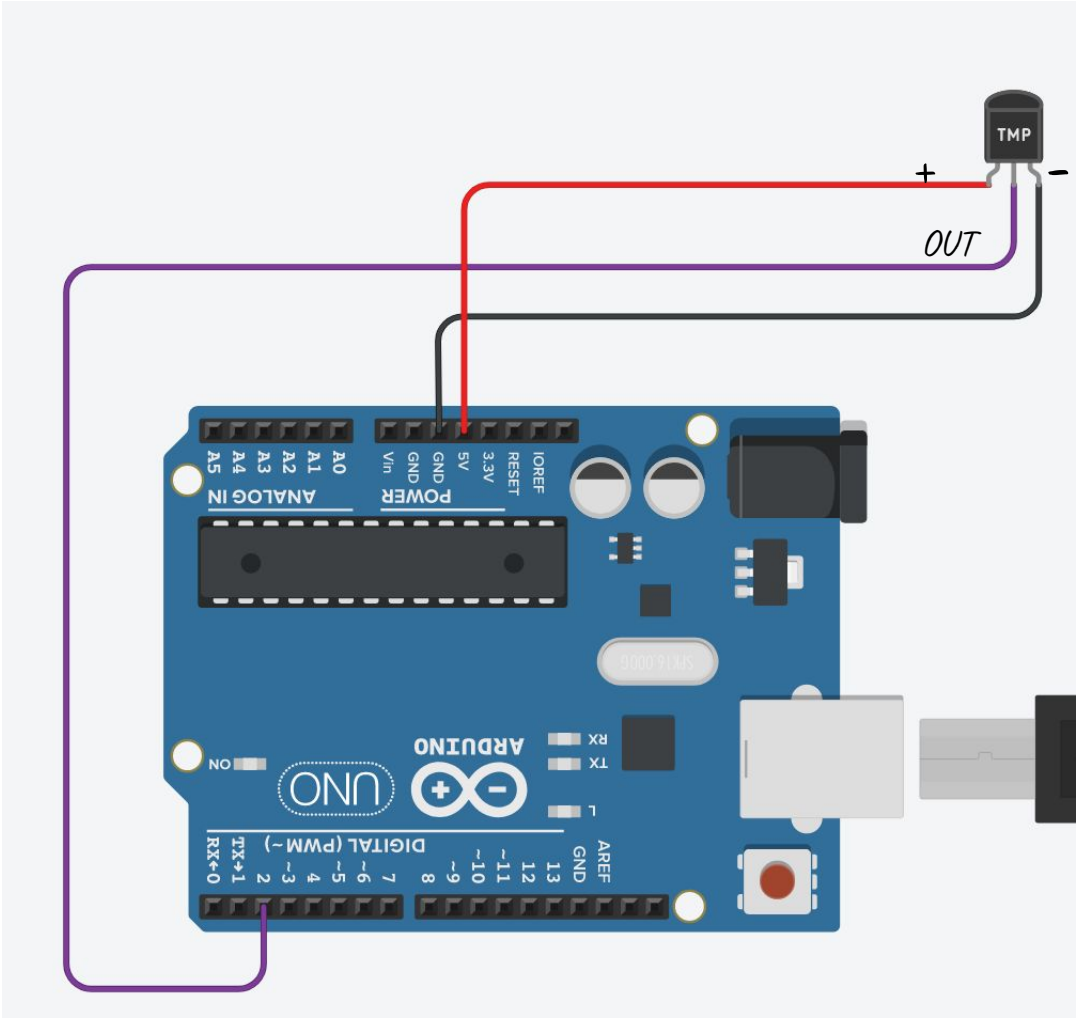
```
void setup() {
  pinMode(6, INPUT);
  pinMode(7, OUTPUT);
}

void loop() {
  if (digitalRead(6) == 1)
    digitalWrite(7, HIGH);
  else
    digitalWrite(7, LOW);
}
```

Sensor de Temperatura

- O DHT11 é um sensor de temperatura e umidade de saída de sinal digital garantindo alta confiabilidade e estabilidade a longo prazo.

- Características:
  - Alimentação: 3,0 a 5,0 VDC (5,5 Vdc máximo)
  - Faixa de medição de umidade: 20 a 95% UR
  - Faixa de medição de temperatura: 0º a 50ºC
  - Precisão de umidade de medição: ± 5,0% UR
  - Precisão de medição de temperatura: ± 2.0 ºC


TMP
+
-
OUT
ANALOG IN
A0
A1
A2
A3
A4
A5
POWER
Vin
GND
GND
5V
3.3V
RESET
IOREF
ARDUINO
UNO
ON
RX
TX
L
DIGITAL (PWM~)
RX←0
TX→1
2
~3
4
~5
~6
7
8
~9
~10
~11
12
13
GND
AREF

Código

```
//  Adaptado de Fábrica de Programas Autoria
//  Flavio Guimaraes
//                  int.0   int.1   int.2   int.3   int.4   int.5
// Uno              2       3
// Mega2560         2       3       21      20      19      18

#include <idDHT11.h>
int idDHT11pin = 2;       //Porta Digital do Arduino onde o Sinal do Sensor DHT esta conectado
int idDHT11intNumber = 0; //Número da interrupção respectiva à porta definida no parametro anterior (veja tabela
acima)
void dht11_wrapper();     // Declaração da funcão de controle da interrupção.
void loopDHT();           // Atualiza a leitura do sensor
idDHT11 DHT11(idDHT11pin, idDHT11intNumber, dht11_wrapper);   //Instanciação do Objeto de Controle do Sensor
void setup(){
  Serial.begin(9600);
  Serial.println("Inicio do Sketch");
}
//Variaveis que irao conter os valores lidos no Sensor DHT11
float temperaturaC;
float umidade;
float dewPoint;
float dewPointSlow;
```

Código

```
void loop(){
  loopDHT();
  Serial.print("Temperatura Celcius: ");
  Serial.println( temperaturaC );
  Serial.print("Umidade Relativa: ");
  Serial.println( umidade );
  Serial.print("Ponto de Orvalho: ");
  Serial.println( dewPoint );
  Serial.println();
}

void dht11_wrapper() {
  DHT11.isrCallback();
}

void loopDHT() {
#define tempoLeitura 1000
static unsigned long delayLeitura = millis() + tempoLeitura + 1;
static bool request = false;
  if ((millis() - delayLeitura) > tempoLeitura) {
      if (!request) {
         DHT11.acquire();
         request = true;
      }
  }
```

```
if (request && !DHT11.acquiring()) {
  request = false;
  int result = DHT11.getStatus();
  switch (result){
  case IDDHTLIB_OK:
     Serial.println("Leitura OK");
     break;
  case IDDHTLIB_ERROR_CHECKSUM:
     Serial.println("Erro\n\r\tErro Checksum");
     break;
  case IDDHTLIB_ERROR_ISR_TIMEOUT:
     Serial.println("Erro\n\r\tISR Time out");
     break;
  case IDDHTLIB_ERROR_RESPONSE_TIMEOUT:
     Serial.println("Erro\n\r\tResponse time out");
     break;
  case IDDHTLIB_ERROR_DATA_TIMEOUT:
     Serial.println("Erro\n\r\tData time out erro");
     break;
  case IDDHTLIB_ERROR_ACQUIRING:
     Serial.println("Erro\n\r\tAcquiring");
     break;
  case IDDHTLIB_ERROR_DELTA:
     Serial.println("Erro\n\r\tDelta time to small");
     break;
  case IDDHTLIB_ERROR_NOTSTARTED:
     Serial.println("Erro\n\r\tNao iniciado");
     break;
  default:
     Serial.println("Erro Desconhecido");
     break;
  }
    float valor = DHT11.getCelsius();
    if (!isnan(valor)) {
      temperaturaC = valor;
    }
    valor = DHT11.getHumidity();
    if (!isnan(valor)) {
      umidade = valor;
    }
    valor = DHT11.getDewPoint();
    if (!isnan(valor)) {
      dewPoint = valor;
    }
    delayLeitura = millis();
  }
}
```

Sensor de Luz LDR 5mm

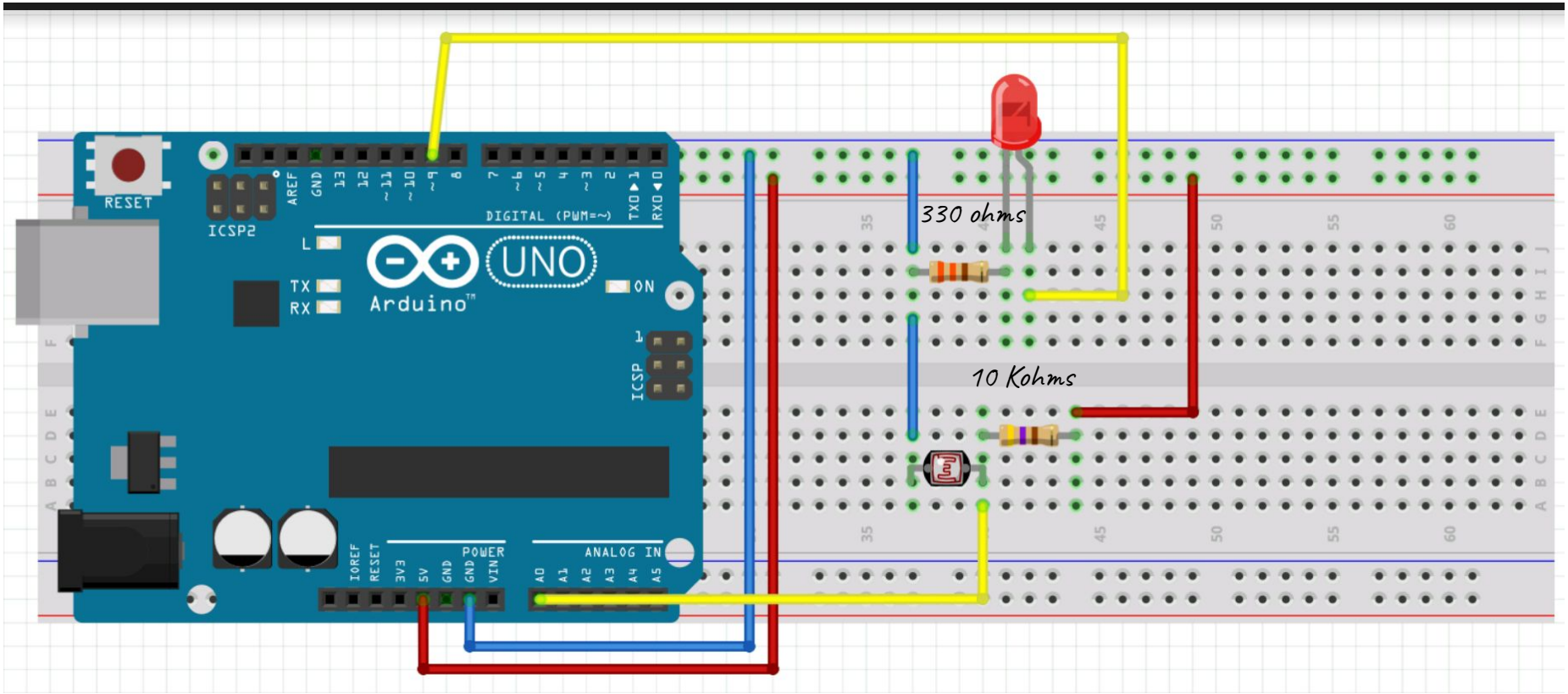
O Sensor de Luminosidade LDR (Light Dependent Resistor) é um componente cuja resistência varia de acordo com a intensidade da luz. Quanto mais luz incidir sobre o componente, menor a resistência.

Características:

– Tensão máxima: 150VDC

– Resistência no escuro: 1 MΩ (Lux 0)

– Resistência na luz: 10-20 KΩ (Lux 10)



extraído de arduinoecia.com.br

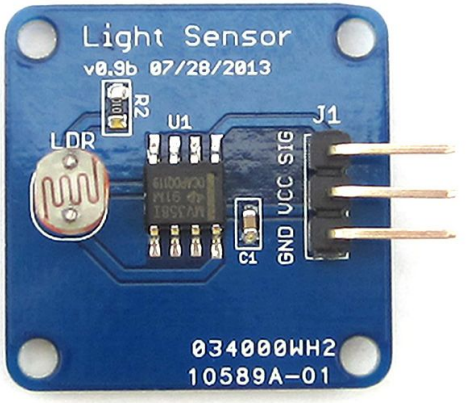
Código

```
// Programa : LDR - Sensor de Iluminação
// Autor : Arduino e Cia

int portaLed = 9; //Porta a ser utilizada para ligar o led
int portaLDR = A0; //Porta analógica utilizada pelo LDR

void setup()
{
  pinMode(portaLed, OUTPUT); //Define a porta do Led como saída
}
void loop()
{
  int estado = analogRead(portaLDR);  //Lê o valor fornecido pelo LDR
  // Caso o valor lido na porta analógica seja maior do que
  // 800, acende o LED
  // Ajuste o valor abaixo de acordo com o seu circuito
  if (estado > 500)
  {
    digitalWrite(portaLed, HIGH);
  }
  else  //Caso contrário, apaga o led
  {
    digitalWrite(portaLed, LOW);
  }
}
```

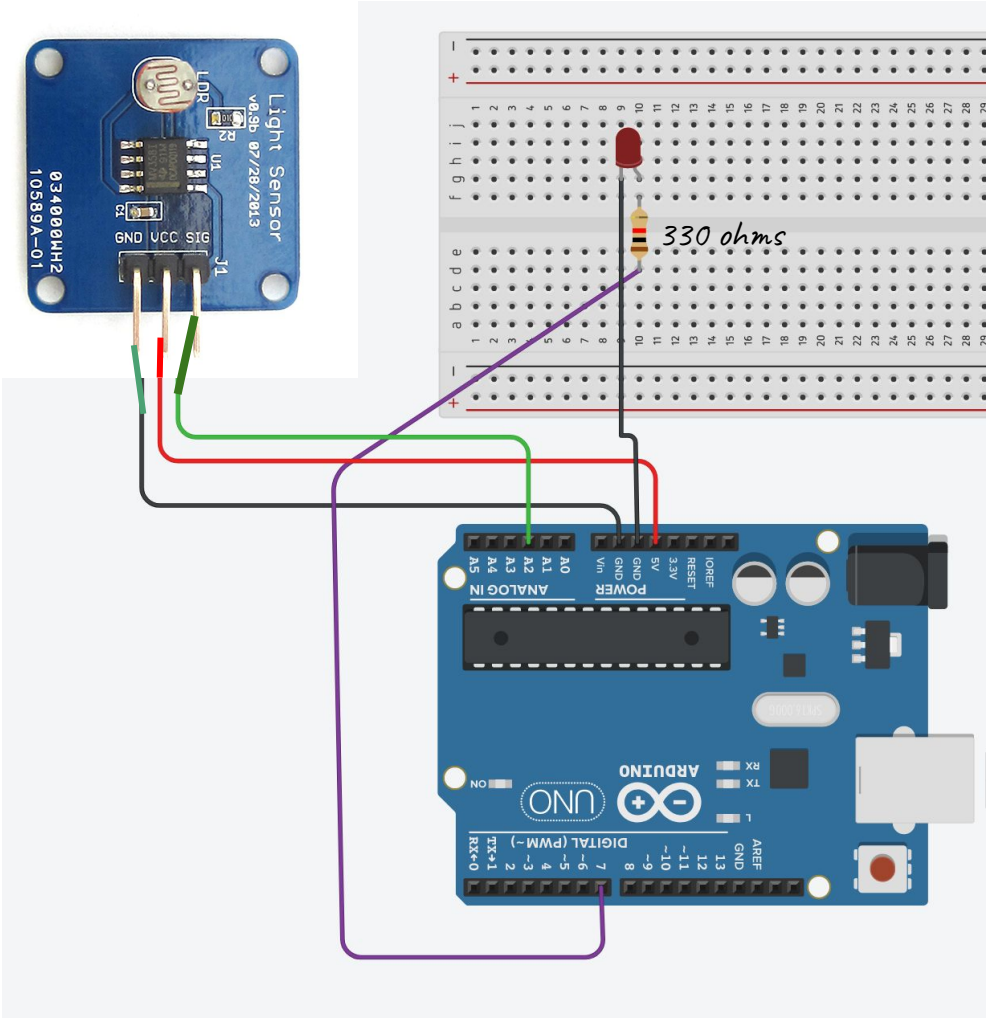
Sensor de Luz LDR



O módulo do sensor de luz possui o fotoresistor GL5528 para detectar a intensidade da luz do ambiente. A resistência do sensor diminui quando a intensidade da luz do ambiente aumenta. A tensão de saída deste módulo aumenta de acordo com a flutuação da intensidade da luz.

Características

VCC: 2.7 - 5.5 V


Light Sensor
v0.9b 07/28/2013
LDR
R2
U1
R1
GND VCC SIG
J1
034000WH2
10589A-01
330 ohms
ANALOG IN
POWER
ARDUINO
UNO
DIGITAL (PWM~)

Código

```
int limit = 70;
int LED  = 7;
int PORTA = A2;
void setup () {
  Serial.begin (9600);
 pinMode(LED, OUTPUT);
  Serial.begin (9600);
}
void loop () {
  int sensorValue = analogRead (PORTA);
  if (sensorValue < limit)
     digitalWrite(LED, HIGH) ;
  else
    digitalWrite(LED, LOW) ;
  float Rsensor;
  Rsensor = (float) (1023-sensorValue) * 10 / sensorValue;
  Serial.println ("The resistance of the Light sensor is");
  Serial.print (Rsensor, 1);
  Serial.println ("KOhm");
  Serial.println (sensorValue);
}
```