

# Sensor de Cores TCS3200

O sensor de cores utiliza o chip TCS3200 para detectar os níveis de luzes RGB dos objetos através do envio de dados para um microcontrolador, como o Arduino, Rasbperry, Pic, entre outros. Esse chip tem 64 fotodiodos, sendo com 16 filtros para a cor vermelha, 16 para a verde, 16 para a azul e 16 sem filtro, permitindo a criação de sistemas detecção de cores

> eficientes. Também possui quatro LEDs brancos para iluminação e oito pinos de conexão.

## Como o Sensor de Cores TCS3200 funciona?

Os 64 fotodiodos que estão distribuídos de forma uniformemente no sensor, são responsáveis por captar a luminosidade, filtrar as cores recebidas e gerar na saída do sensor um sinal de onda quadrada que traz informações sobre a intensidade das cores RGB (Red, Green, Blue). Ele atua efetuando uma análise do comprimento de onda de cada uma as cores RGB, e torna essas informações disponíveis para verificação de qual cor foi identificada. Além disso, com a identificação dos valores RGB lidos, pode-se efetuar também mudanças no código de programação, de modo que sensor reconheça não só as cores RGB, mais também as demais cores.

#### Descrição de pinagem do Sensor



GND: Aterramento da fonte de alimentação. Todas as tensões são referenciadas para GND.

OE: Habilita para fo (Ativo em nível baixo). Pino I/O tipo Entrada

OUT: Frequência de saída (fo). Pino I/O tipo Saída

SO | S1: Entradas de seleção da escala de frequência de saída. Pino I/O tipo Entrada

S2 | S3: Entradas de seleção do tipo de fotodiodo. Pino I/O tipo Entrada

VDD: Tensão de alimentação.

Tabela de Nível Lógico disponibilizada no Datahseet do chip TCS3200

| Tabela de Nível Lógico |           |                                    |  |  |  |  |  |  |
|------------------------|-----------|------------------------------------|--|--|--|--|--|--|
| <b>SO</b>              | S1        | Escala de Frequência de saída (FO) |  |  |  |  |  |  |
| L                      | L         | 0% (Desligado)                     |  |  |  |  |  |  |
| L                      | Н         | 2%                                 |  |  |  |  |  |  |
| Н                      | L         | 20%                                |  |  |  |  |  |  |
| Н                      | Н         | 100%                               |  |  |  |  |  |  |
| <b>S2</b>              | <b>S3</b> | Tipo do Fotodiodo                  |  |  |  |  |  |  |
| L                      | L         | Vermelho                           |  |  |  |  |  |  |
| L                      | Н         | Blue                               |  |  |  |  |  |  |
| Н                      | L         | Claro (Sem Filtro)                 |  |  |  |  |  |  |
| н                      | Н         | Verde                              |  |  |  |  |  |  |



## Conexão de montagem para teste do sensor de Cores TCS3200 utilizando LED RGB na exibição do reconhecimento das cores RGB



#### Conexão Sensor de Cor:

Pino GND do sensor: Conectado ao GND do Arduino Pino OE do sensor: Não conectado Pino S0 do sensor: Conectado ao pino 9 do Arduino Pino S1 do sensor: Conectado ao pino 10 do Arduino Pino S2 do sensor: Conectado ao pino 11 do Arduino Pino S3 do sensor: Conectado ao pino 12 do Arduino Pino OUT do sensor: Conectado ao pino 8 do Arduino Pino VCC do sensor: Conectado ao 5V do Arduino



**Atenção:** Devido as diferentes variações de LED RGB Anodo Comum disponíveis no mercado, pode-se haver a inversão dos pinos G e B. É aconselhável a efetuação do teste pino a pino para averiguação da ordem das cores, de modo a garantir o correto funcionamento do sistema de exibição de reconhecimento da cor.

#### **Conexão LED RGB:**

Pino 1 do LED (R - Vermelho): Conectado ao pino 2 do Arduino, e ao resistor de 150 ohms, o outro lado do resistor deve ser conectado ao GND do Arduino.
Pino 2 do LED (Anodo Comum): Conectado ao VCC do Arduino.
Pino 3 do LED (B - Azul): Conectado ao pino 3 do Arduino, e ao resistor de 100 ohms, o outro lado do resistor deve ser conectado ao GND do Arduino.
Pino 4 do LED (G - Verde): Conectado ao pino 4 do Arduino, e ao resistor de 100 ohms, o outro lado do resistor deve ser conectado ao GND do Arduino.



Após efetuada a montagem, como demonstrado acima, compile o código "teste do sensor de Cores TCS3200 utilizando LED RGB na exibição do reconhecimento das cores RGB" disponibilizado no fim do roteiro, para o microcontrolador Arduino como demonstrado abaixo, e faça o teste de reconhecimento das cores.

```
basico | Arduino 1.8.16
                                                          Х
Arquivo Editar Sketch Ferramentas Ajuda
    +
            ₳
                ÷
                                                                 Q.
  basico
     \land
******** Teste do Sensor de cor TCS3200 utilizando LED RGB **************
*********** na exibição do reconhecimento das cores RGB *****************
 *****
//Definição de conexão dos pinos do módulo ao Arduino
const int s0 = 9;
const int s1 = 10;
const int s2 = 11;
const int s3 = 12;
const int out = 8;
//Definição de conexão dos pinos do LED RGB ao Arduino
int pinoledverm = 2;
int pinoledverd = 4;
int pinoledazul = 3;
//Variaveis cores
int red = 0;
int green = 0;
int blue = 0;
void setup()
{ //Definicão de entradas e saídas
 pinMode(s0, OUTPUT);
 pinMode(s1, OUTPUT);
 pinMode(s2, OUTPUT);
 pinMode(s3, OUTPUT);
 pinMode(out, INPUT);
 pinMode(pinoledverm, OUTPUT);
 pinMode(pinoledverd, OUTPUT);
 pinMode(pinoledazul, OUTPUT);
 Serial.begin(9600);
 digitalWrite(s0, HIGH);
  digitalWrite(sl, LOW);
ł
void loop()
Ł
  //Detecão de cores
```





Conexão de montagem para teste do sensor de Cores TCS3200 utilizando Display LCD na exibição do reconhecimento das cores



## Conexão Sensor de Cor:

Pino GND do sensor: Conectado ao GND do Arduino
Pino OE do sensor: Não conectado
Pino S0 do sensor: Conectado ao pino 9 do Arduino
Pino S1 do sensor: Conectado ao pino 10 do Arduino
Pino S2 do sensor: Conectado ao pino 11 do Arduino
Pino S3 do sensor: Conectado ao pino 12 do Arduino
Pino OUT do sensor: Conectado ao pino 8 do Arduino
Pino VCC do sensor: Conectado ao 5V do Arduino

## **Conexão Display:**

Pino 1 do Display: Conectado ao GND do Arduino
Pino 2 do Display: Conectado ao 5V do Arduino
Pino 3 do Display: Conectado ao pino 2 do Potenciômetro
Pino 4 do Display: Conectado ao pino 2 do Arduino
Pino 5 do Display: Conectado ao pino GND do Arduino
Pino 6 do Display: Conectado ao pino 3 do Arduino
Pino 11 do Display: Conectado ao pino 5 do Arduino
Pino 12 do Display: Conectado ao pino 5 do Arduino
Pino 13 do Display: Conectado ao pino 5 do Arduino
Pino 13 do Display: Conectado ao pino 7 do Arduino
Pino 14 do Display: Conectado ao pino 7 do Arduino
Pino 15 do Display: Conectado ao pino 7 do Arduino
Pino 15 do Display: Conectado ao pino 7 do Arduino

## **Conexão Potenciômetro:**

Pino 1 Potenciômetro: Conectado ao GND do Arduino Pino 2 Potenciômetro: Conectado ao pino 3 do Display Pino 3 Potenciômetro: Conectado ao 5V do Arduino



Após efetuada a montagem, como demonstrado acima, compile o código "teste do sensor de Cores TCS3200 utilizando Display LCD para reconhecimento das cores" disponibilizado no fim do roteiro, para o microcontrolador Arduino, e faça o teste de reconhecimento das cores, como demonstrado no vídeo. *Vale ressaltar que o teste efetuado foi feito em um laboratório com luminosidade ambiente, sofrendo interferência direta da iluminação externa. Para uso do sensor utilizando esse software, indicamos uma proteção ao sensor contra luminosidade externa, garantindo assim uma fiel leitura do sensor sem variações de iluminação pois a variação de iluminação afeta diretamente no reconhecimento e definição das cores pelo reconhecimento utilizando o código disponibilizado.* 

| Deteccao_de_cor_com_Display   Arduino 1.8.16                                   |       |   |  |  |  |  |  |  |
|--|-------|---|--|--|--|--|--|--|
| Arquivo Editar Sketch Ferramentas Ajuda  |       |   |  |  |  |  |  |  |
|  | P     |   |  |  |  |  |  |  |
| Deteccao_de_cor_com_Display  |       |   |  |  |  |  |  |  |
| /*************************************   |       | ^ |  |  |  |  |  |  |
| ******** Teste do Sensor de cor TCS3200 utilizando ***********                 |       |   |  |  |  |  |  |  |
| **************************************   |       |   |  |  |  |  |  |  |
|  |       |   |  |  |  |  |  |  |
|  |       |   |  |  |  |  |  |  |
| <pre>#include <liquidcrystal.h></liquidcrystal.h></pre>                        |       |   |  |  |  |  |  |  |
| LiquidCrystal lcd(2, 3, 4, 5, 6, 7);   |       |   |  |  |  |  |  |  |
|  |       |   |  |  |  |  |  |  |
| #define outPin 8   |       |   |  |  |  |  |  |  |
| #define s0 9   |       |   |  |  |  |  |  |  |
| #define s1 10  |       |   |  |  |  |  |  |  |
| #define s3 12  |       |   |  |  |  |  |  |  |
|  |       |   |  |  |  |  |  |  |
| boolean DEBUG = true;  |       |   |  |  |  |  |  |  |
|  |       |   |  |  |  |  |  |  |
| // Variáveis   |       |   |  |  |  |  |  |  |
| int Vermelho, Verde, Azul;   |       |   |  |  |  |  |  |  |
| String cor ="";  |       |   |  |  |  |  |  |  |
| long startTiming = 0;  |       |   |  |  |  |  |  |  |
| Tong etapsedTime -0,   |       |   |  |  |  |  |  |  |
| <pre>void setup() {</pre>  |       |   |  |  |  |  |  |  |
| <pre>Serial.begin(9600);</pre>   |       |   |  |  |  |  |  |  |
|  |       |   |  |  |  |  |  |  |
| <pre>pinMode(s0, OUTPUT); //Definição de entrada e saída dos pinos do as</pre> | rduin |   |  |  |  |  |  |  |
| <pre>pinMode(s1, OUTPUT);</pre>  |       |   |  |  |  |  |  |  |
| pinMode(s2, OUTPUT);   |       |   |  |  |  |  |  |  |
| pinMode(ss, OUTPUT);   |       |   |  |  |  |  |  |  |
| primode (outrin, infor,  |       | ~ |  |  |  |  |  |  |
|  | >     |   |  |  |  |  |  |  |
| Compilação terminada.  |       |   |  |  |  |  |  |  |

Com o código compilado para o Arduino, abra o monitor serial, ou analise no próprio Display LCD os valores lidos. Inicialmente aparecerá "Sem cor detectado!", com os valores de Vermelho, Verde e Azul no monitor serial, ou R, G e B no display. Como demostrado a seguir:

| 💿 COM3       |     |                |                   |                  | _               |                                    |
|--------------|-----|----------------|-------------------|------------------|-----------------|------------------------------------|
|              |     |                |                   |                  |                 | Enviar                             |
| vermeino=    | 4   | verae= 8       | Azui= / - sem cor | detectado:       |                 | ^                                  |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 6 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 9       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 |                                    |
| Vermelho=    | 4   | verde= 8       | Azul= 7 - Sem cor | detectado!       |                 | ~                                  |
| Auto-rolager | n [ | Show timestamp | Am                | bos, NL e CR 🛛 🗸 | 9600 velocidade | <ul> <li>Deleta a saida</li> </ul> |

Adicione o cartão com a cor a ser detectada, e analise novamente os valores lidos. Os valores lidos são os valores referentes a cor a ser detectada, assim, no campo "readRGB" do software, edite esses valores para o range dentro dos valores lidos.

Como exemplo, falaremos sobre a leitura efetuada da cor Vermelho. Os valores lidos, como pode ser visto no display da primeira amostragem de cor, foi R=10, G=21 e B=17; sendo assim, vamos editar os valores da linha da cor vermelho, colocando os valores dentro do range, dando um pequeno intervalo entre os valores. A variável R (Vermelho) possui valor de 10, então colocamos como mínimo > 8, e como máximo <12 garantindo um range de 9 a 11. Na variável G (Verde) colocamos como mínimo > 19, e como máximo <23 garantindo um range de 20 à 22, e Na variável B (Azul) colocamos como mínimo > 15, e como máximo <19 garantindo um range de 16 à 18, assim, a leitura dentro do intervalo dos valores setados retorna a leitura da cor Vermelho.





```
void getcor(){
   readRGB();
```

|      | if <mark>(Vermelho&gt;8</mark> | & & | Vermelho<12 | & & | Verde>19  | & & | Verde<23   | & &          | Azul>15  | 88  | Azul<19)  | cor = | "Vermelho";  |
|------|--------------------------------|-----|-------------|-----|-----------|-----|------------|--------------|----------|-----|-----------|-------|--------------|
| else | if(Vermelho>14                 | &&  | Vermelho<18 | & & | Verde>10  | & & | Verde<15   | &&           | Azul>6   | &&  | Azul<10)  | cor = | "Azul";      |
| else | if(Vermelho>13                 | &&  | Vermelho<17 | & & | Verde>11  | & & | Verde<15   | &&           | Azul>11  | &&  | Azul<16)  | cor = | "Verde";     |
| else | if(Vermelho>24                 | &&  | Vermelho<29 | & & | Verde>30  | & & | Verde<35   | &&           | Azul>24  | &&  | Azul<28)  | cor = | "Preto";     |
| else | if(Vermelho>6                  | &&  | Vermelho<10 | & 8 | & Verde>7 | &   | & Verde<11 | <u>&amp;</u> | & Azul>5 | & ( | 🛛 Azul<9) | cor   | = "Branco";  |
| else | if(Vermelho>5                  | & & | Vermelho<8  | & & | Verde>7   | & & | Verde<10   | & &          | Azul>10  | & ( | Azul<14   | ) cor | = "Amarelo"; |
| else | cor = "Sem con                 | :"; |             |     |           |     |            |              |          |     |           |       |              |
| }    |                                |     |             |     |           |     |            |              |          |     |           |       |              |

Para detecção das demais cores, efetuamos a leitura dos valores referentes a cada cor, e reconfiguramos os intervalos dos valores na linha da respectiva cor, assim como efetuamos com vermelho.

Efetuado a reconfiguração de todas as cores, recompilamos o código e testamos as cores como se segue, sendo as cores detectadas com êxito, comprovando assim, a aprovação do sensor de cores.

Vale ressaltar que, como o teste efetuado não foi feito em um ambiente controlado, a cada mudança externa do clima como o clarear ou o escurecer do dia chuvoso em que efetuamos o teste, foi-se necessário refazer a reconfiguração dos valores das cores. Para a solução desse problema pode-se implementar uma função filtro no código de programação, ou garantir uma proteção contra a luminosidade externa ao sensor.



#### SOFTWARES utilizados para Teste:

#### SOFTWARE 1: Utilizando LED RGB



//Definicão de conexão dos pinos do LED RGB ao Arduino int pinoledverm = 2; int pinoledverd = 4; int pinoledazul = 3; //Variaveis cores int red = 0;int green = 0;int blue = 0;void setup() {//Definicão de entradas e saídas pinMode(s0, OUTPUT); pinMode(s1, OUTPUT); pinMode(s2, OUTPUT); pinMode(s3, OUTPUT); pinMode(out, INPUT); pinMode(pinoledverm, OUTPUT); pinMode(pinoledverd, OUTPUT); pinMode(pinoledazul, OUTPUT); Serial.begin(9600); digitalWrite(s0, HIGH); digitalWrite(s1, LOW); } void loop() { //Deteção de cores color(); //Exibição dos valores das cores no serial monitor Serial.print("Vermelho :"); Serial.print(red, DEC); Serial.print(" Verde : "); Serial.print(green, DEC); Serial.print(" Azul : "); Serial.print(blue, DEC); Serial.println(); //Verifica se a cor vermelha foi detectada if (red < blue && red < green && red < 100) { Serial.println("Vermelho"); digitalWrite(pinoledverm, LOW); //Acende o led vermelho digitalWrite(pinoledverd, HIGH); digitalWrite(pinoledazul, HIGH); } //Verifica se a cor azul foi detectada else if (blue < red && blue < green && blue < 1000) { Serial.println("Azul"); digitalWrite(pinoledverm, HIGH); digitalWrite(pinoledverd, HIGH); digitalWrite(pinoledazul, LOW); //Acende o led azul }



```
//Verifica se a cor verde foi detectada
else if (green < red && green < blue)
{
 Serial.println("Verde");
  digitalWrite(pinoledverm, HIGH);
  digitalWrite(pinoledverd, LOW); //Acende o led verde
  digitalWrite(pinoledazul, HIGH);
}
Serial.println();
//Delay para reiniciar o processo
delay(50);
digitalWrite(pinoledverm, HIGH);
digitalWrite(pinoledverd, HIGH);
digitalWrite(pinoledazul, HIGH);
}
void color()
{
//Rotina que lê o valor das cores
digitalWrite(s2, LOW);
digitalWrite(s3, LOW);
//count OUT, pRed, RED
red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
digitalWrite(s3, HIGH);
//count OUT, pBLUE, BLUE
blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
digitalWrite(s2, HIGH);
//count OUT, pGreen, GREEN
green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
}
SOFTWARE 2: Utilizando Display LCD
```

#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

#define outPin 8
#define s0 9
#define s1 10
#define s2 11
#define s3 12

boolean DEBUG = true;

// Variáveis
int Vermelho, Verde, Azul;
String cor ="";



```
long startTiming = 0;
long elapsedTime =0;
```

```
void setup(){
Serial.begin(9600);
```

```
pinMode(s0, OUTPUT); //Definição de entrada e saída dos pinos do arduino
pinMode(s1, OUTPUT);
pinMode(s2, OUTPUT);
pinMode(s3, OUTPUT);
pinMode(outPin, INPUT);
```

```
// Configurando a escala de frequência para 100% em nível alto
digitalWrite(s0,HIGH);
digitalWrite(s1,HIGH);
```

```
lcd.begin(16, 2);
 lcd.clear();
 lcd.setCursor (1,0);
 lcd.print("---Eletodex---");
 lcd.setCursor (1,1);
 lcd.print("Sensor de cor");
 delay(6000);
 lcd.clear();
 startTiming = millis();
}
void loop(){
 getcor();
 if(DEBUG)printData();
 elapsedTime = millis()-startTiming;
 if (elapsedTime > 1000) {
  showDataLCD();
  startTiming = millis();
 }
}
/* Leitura RGB */
void readRGB(){
 Vermelho = 0, Verde=0, Azul=0;
 int n = 10;
 for (int i = 0; i < n; ++i){
  //Leitura do vermelho
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  Vermelho = Vermelho + pulseIn(outPin, LOW);
 //leitura do verde
  digitalWrite(s2, HIGH);
  digitalWrite(s3, HIGH);
  Verde = Verde + pulseIn(outPin, LOW);
```

```
//leitura do azul
```



```
digitalWrite(s2, LOW);
 digitalWrite(s3, HIGH);
 Azul = Azul + pulseIn(outPin, LOW);
}
Vermelho = Vermelho/n;
Verde = Verde/n;
Azul = Azul/n;
}
  Mostrando os dados da leitura dos valores RBG no Serial Monitor
void printData(void){
Serial.print("Vermelho= ");
Serial.print(Vermelho);
Serial.print(" verde= ");
Serial.print(Verde);
Serial.print(" Azul= ");
Serial.print(Azul);
Serial.print (" - ");
Serial.print (cor);
Serial.println (" detectado!");
}
Mostrando os dados da leitura dos valores RBG no LCD
void showDataLCD(void){
lcd.clear();
lcd.setCursor (0,0);
lcd.print("R");
lcd.print(Vermelho);
lcd.setCursor (6,0);
lcd.print("G");
lcd.print(Verde);
lcd.setCursor (12,0);
lcd.print("B");
lcd.print(Azul);
lcd.setCursor (0,1);
lcd.print("cor: ");
lcd.print(cor);
}
void getcor(){
readRGB();
```

if(Vermelho>3 && Vermelho<12 && Verde>19 && Verde<23 && Azul>15 && Azul<19) cor = "Vermelho"; else if(Vermelho>14 && Vermelho<18 && Verde>10 && Verde<15 && Azul>6 && Azul<10) cor = "Azul"; else if(Vermelho>13 && Vermelho<17 && Verde>11 && Verde<15 && Azul>11 && Azul<16) cor = "Verde"; else if(Vermelho>24 && Vermelho<29 && Verde>30 && Verde<35 && Azul>24 && Azul<28) cor = "Preto"; else if(Vermelho>6 && Vermelho<10 && Verde>7 && Verde<11 && Azul>5 && Azul<9) cor = "Branco"; else if(Vermelho>5 && Vermelho<8 && Verde>7 && Verde<10 && Azul>10 && Azul<14) cor = "Amarelo"; else cor = "Sem cor";}