Esteira para Separação de Tomates

Celso Roberto Lima dos Santos/ [celsorugby@hotmail.com](mailto:celsorugby@hotmail.com)

Heloanny Fernanda F. M. de Souza Monteiro/ heloanny.fernanda@hotmail.com

Isabela de Souza Ribeiro/ isabeladesouzaribeiro2309@gmail.com

Profª Helcy Moreira Martins Aguiar

Orientador: Profº.MSc Diego Henrique Emygdio Lázaro

**RESUMO**

No projeto foi realizada uma adaptação em uma esteira para separação dos tomates, sendo vermelhos ou verdes, contribuindo assim para o processo de triagem realizado pelo agricultor, desta forma ganha praticidade na hora da separação.

Quando o tomate for colocado na esteira automaticamente passará por um sensor que distinguirá sua cor, se for vermelho passará direto e caíra em uma caixa, e se for verde receberá um toque da alavanca e cairá em outra. Sensores fazem o controle dos tomates que passam na esteira gerenciados pelo micro controlador Arduíno.

**Palavras-Chave**

Esteira, sensor, separação, triagem.

# INTRODUÇÃO

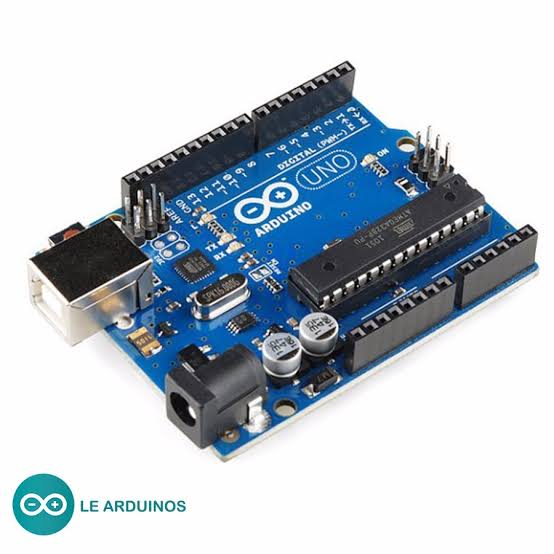
# ESTUDO DE CASO

Em geral, a separação dos tomates nas estufas ocorre de forma manual, acarretando risco ao trabalhador, e onerando o tempo produtivo, em virtude da falta de uma tecnologia que propicie ganho de escala. Para amenizar esse fato esteiras são utilizadas. A esteira será implementada na estufa. O principal resultado a ser alcançado com esse trabalho é a melhoria no trabalho realizado pelos funcionários, com ganho para a segurança do trabalhador, e em escala produtiva e financeira. Vale ressaltar, que para este projeto será realizado um protótipo que simula em escala a realidade que o mesmo pode trazer ao ser implementado na estufa.

# TECNOLOGIAS UTILIZADAS

Utilizou-se um microcontrolador Arduíno, módulo sensor de cor RGB, motor de corrente contínua, esteira pré-moldada, servo motor, jumpers, resistores e programação na linguagem de arduíno.

**A seguir são apresentados os conceitos de cada componente utilizado:**



**Micro controlador Arduíno:**

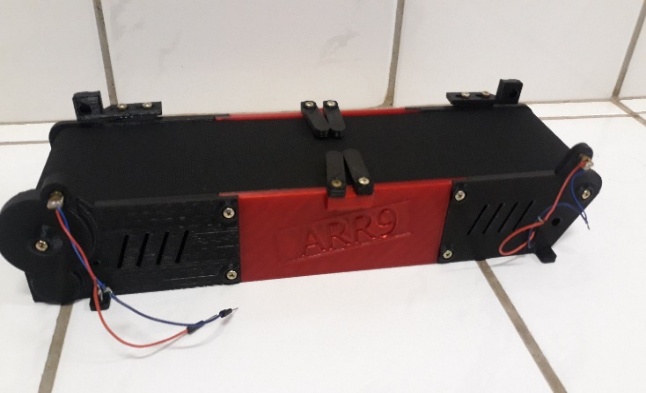
**A placa de um Arduino consiste em um microcontrolador** (que é um microcomputador de um único circuito, ele processa o código desenvolvido pelo desenvolvedor), com componentes complementares para facilitar a programação e incorporação para outros circuitos.

**Motor de corrente contínua:**

****

Um [motor cc](https://www.citisystems.com.br/motor-cc/) nada mais é do que um motor alimentado por [corrente contínua](https://www.citisystems.com.br/corrente-continua/) (CC), sendo esta alimentação proveniente de uma bateria ou qualquer outra de alimentação CC. A sua comutação (troca de energia entre rotor e estator) pode ser através de escovas (escovado) ou sem escovas (brushless) e com relação a velocidade, o [motor cc](https://www.citisystems.com.br/motor-cc/) pode ser controlado apenas variando a sua tensão, diferentemente de um [motor elétrico](https://www.citisystems.com.br/motor-eletrico/) de [corrente alternada](https://www.citisystems.com.br/corrente-alternada/) (CA) cuja a velocidade é variada pela frequência.

**Esteira pré-moldada:**

****

**Servo motor:**

****

 Os **Servo motores**, também chamados de **servos**, são muito utilizados quando o assunto é robótica. De forma simplificada, um servomotor é um motor na qual podemos controlar sua posição angular através de um sinal [PWM](https://portal.vidadesilicio.com.br/grandezas-digitais-e-analogicas-e-pwm/).

**Jumpers:**

****

Os jumpers são chaves elétricas utilizadas em placas e alguns dispositivos, como discos rígidos para ativar, regular ou desativar funções específicas do sistema que não são acessíveis via software.

**Resistores:**

****

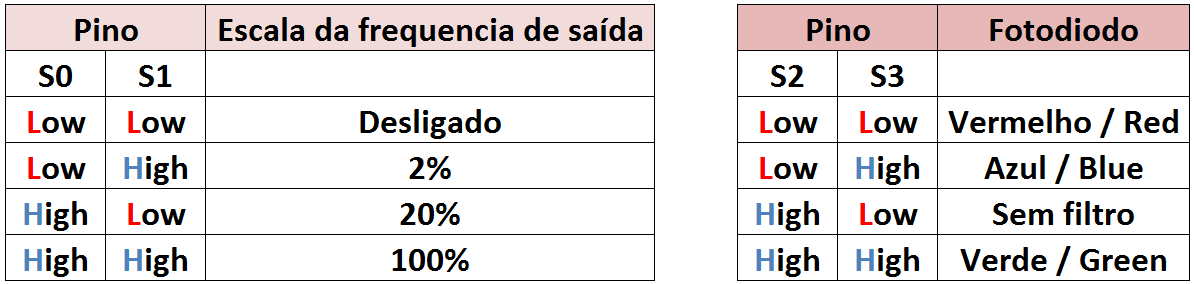
Os resistores são componentes eletrônicos passivos, muito comuns no mundo da eletrônica e a letra R é usada para representá-lo. Os resistores não são polarizados, ou seja, não possuem polo positivo e negativo. A função do resistor é limitar o fluxo de corrente elétrica que passa por ele, e a essa limitação se dá o nome de resistência, medida em ohms, e ela define qual a facilidade ou dificuldade que os eletrons terão que enfrentar para passar pelo resistor. Quanto maior o valor da resistência (em ohms) mais difícil será para os eletrons passarem pelo resistor e quanto menor o valor da resistência (em ohms) mais fácil será para os eletrons passarem. A limitação do fluxo da corrente elétrica que o resistor impõe causa também uma queda na tensão.

**O sensor de cor TCS3200**

O sensor de cor TCS3200 tem duas fileiras de 5 pinos, onde encontramos os pinos de controle (S0, S1, S2, S3), saída (OUT), controle do Led (LED) e alimentação (VCC e GND). Os pinos Vcc e GND estão em duplicidade, e você pode usar qualquer um deles para alimentar o módulo.



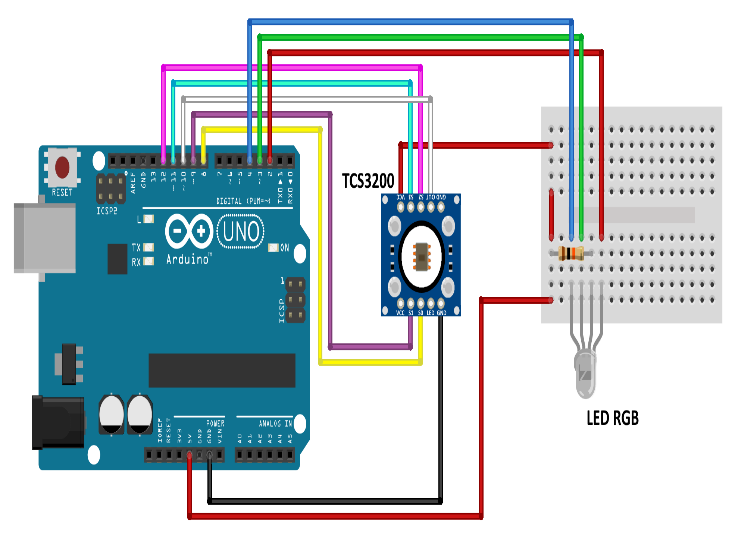
Os pinos S0 e S1 determinam a frequência de saída, e os pinos S2 e S3 determinam qual nível de cor será detectado no momento, segundo a tabela abaixo:



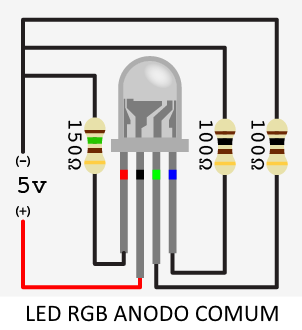
O pino LED serve para ligar (nível 0 – LOW) ou desligar (nível 1 – HIGH), os leds brancos em volta do sensor.

**Circuito Arduino e Sensor TCS3200**

No nosso circuito, vamos utilizar um Arduino Uno, um sensor de cor TCS3200 e um [led RGB difuso 5mm](https://www.filipeflop.com/produto/led-rgb-difuso-5mm/), que vai acender nas cores vermelho, verde ou azul conforme a cor do objeto que for colocado em frente ao sensor:



O led RGB é do tipo anodo comum, com o esquema de ligação abaixo:



Você pode fazer a ligação com 3 resistores, conforme a imagem acima, ou utilizar, para efeitos de testes, apenas um resistor no anodo (pino maior). No nosso circuito utilizamos apenas um resistor de 10K.

**Programa o sensor de cor TCS3200**

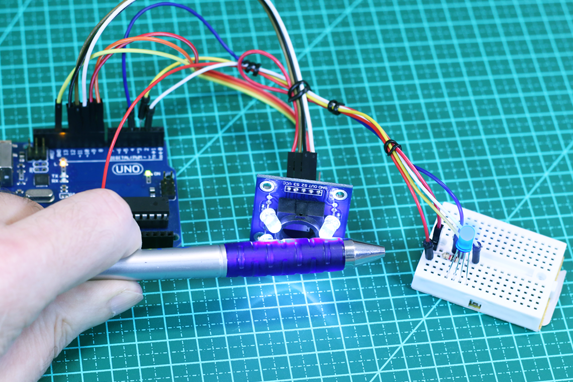
A princípio o sensor de cor TCS3200 não usa nenhuma biblioteca específica, já que vamos acionar os pinos S0, S1, S2 e S3 pelos pinos digitais, e ler o valor da saída OUT também por um pino digital.

A cada ciclo do loop é efetuada uma leitura das cores, e os valores lidos são mostrados no serial monitor. Esses valores também são utilizados para determinar qual das três cores do led serão acesas.

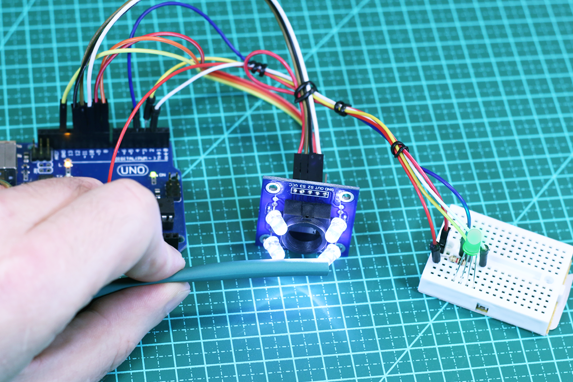
[?](https://www.filipeflop.com/blog/sensor-de-cor-tcs3200-rgb-arduino/)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97 | **//Programa: Sensor TCS3200 e led RGB**    **//Pinos de conexao do modulo**  const int s0 = 8;  const int s1 = 9;  const int s2 = 12;  const int s3 = 11;  const int out = 10;    **//Pinos do led RGB**  int pinoledverm = 2;  int pinoledverd = 3;  int pinoledazul = 4;    **//Variaveis cores**  int red = 0;  int green = 0;  int blue = 0;    void setup()  {    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()  {    //**Detecta a cor**    color();  **//Mostra valores 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 apagar os leds e reiniciar o processo**    delay(50);    digitalWrite(pinoledverm, HIGH);    digitalWrite(pinoledverd, HIGH);    digitalWrite(pinoledazul, HIGH);  }    void color()  {  **//Rotina que le 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);  } |

O programa lê as cores primárias (RGB), mas você pode configurar o programa para detectar outras cores. Nas imagens abaixo, temos o sensor detectando a cor azul...



e também a cor verde...



REFERÊNCIAS:

<https://www.hostgator.com.br/blog/o-que-e-arduino/> ARDUÍNO

<https://www.sick.com/br/pt/sensores-de-registro/sensores-de-cor/c/g113666> SENSORES

<https://www.citisystems.com.br/motor-cc/>MOTOR DE CORRENTE

<https://portal.vidadesilicio.com.br/o-que-e-servomotor/> SERVO MOTOR

<https://www.techtudo.com.br/noticias/noticia/2015/03/jumpers-entenda-o-que-sao-e-para-que-servem-essas-pecas.html> JUMPERS

<http://www.comofazerascoisas.com.br/resistor-o-que-e-e-para-que-serve-introducao-aplicacao.html>RESISTOR