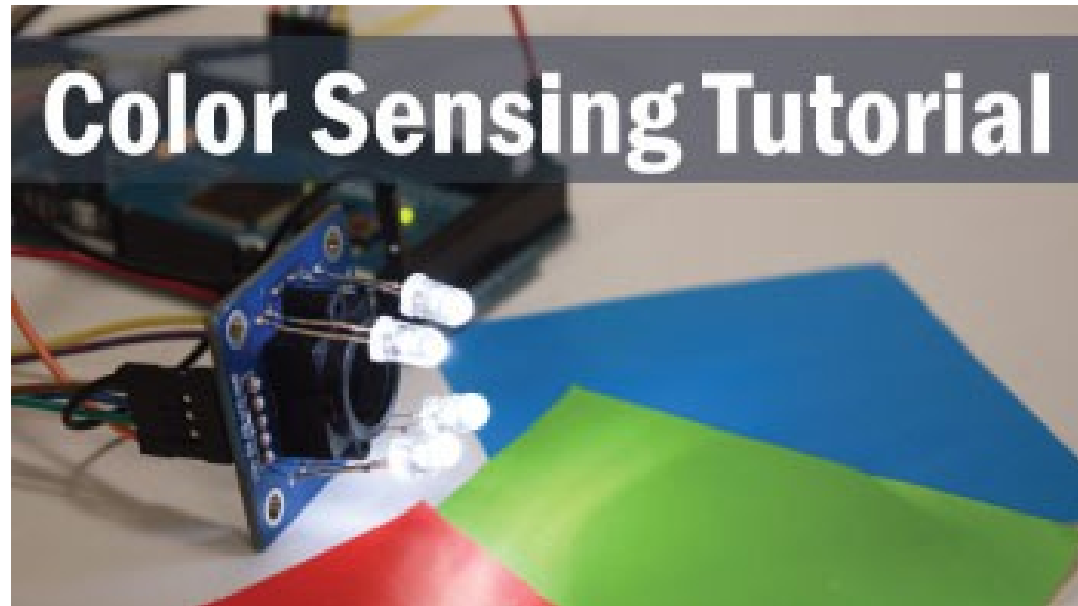


# ARDUINO COLOR SENSOR– TCS230 / TCS3200

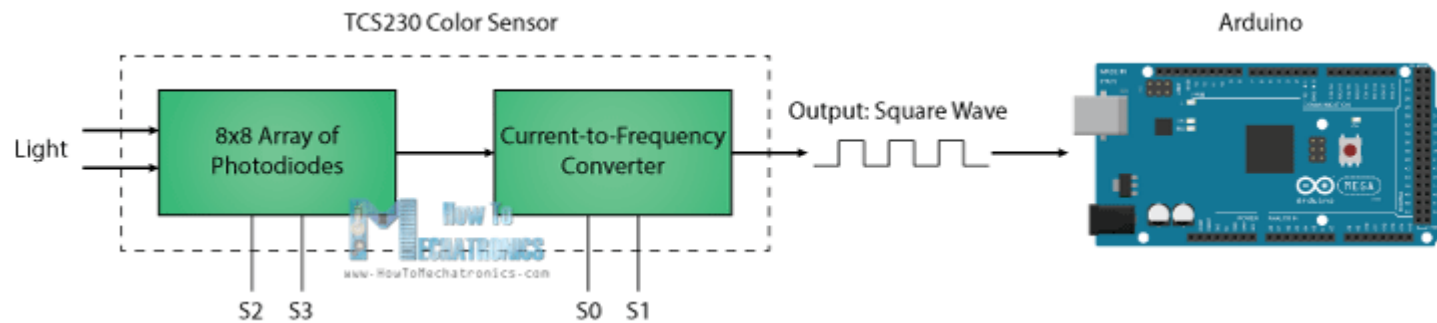
Per imparare come rilevare i colori utilizzando il sensore di colore TCS3200 o il TCS230, si può guardare il seguente video: <https://www.youtube.com/watch?v=CPUXxuyd9xw>



*dal sito....* <http://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/>

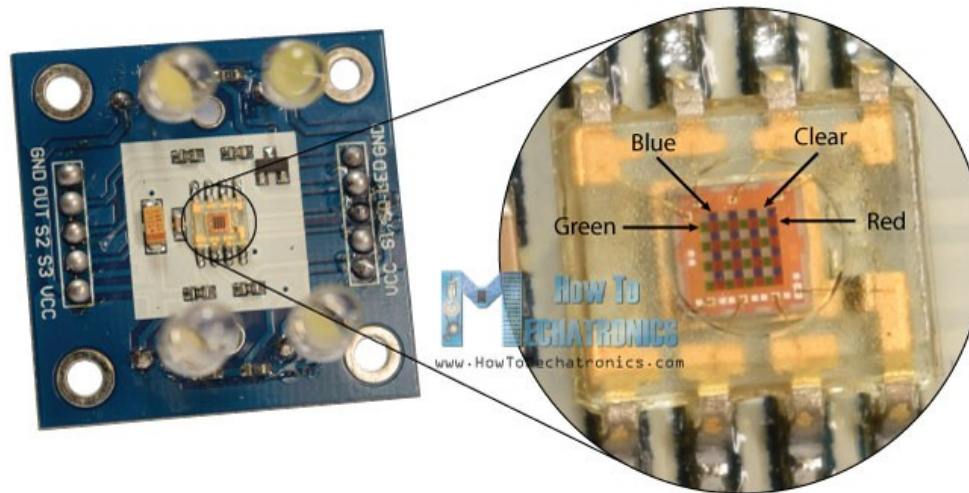
# FUNZIONAMENTO DEL SENSORE DI COLORE TCS3200 /TCS230

Il TCS230 rileva un colore con l'aiuto di una matrice 8 x 8 di fotodiodi. Quindi utilizzando un convertitore corrente-frequenza le letture dei fotodiodi sono convertiti in un'onda quadra con una frequenza direttamente proporzionale all'intensità della luce. Infine, utilizzando la scheda Arduino possiamo leggere l'output dell' onda quadra e ottenere i risultati per il colore.



I fotodiodi hanno tre diversi filtri di colore. Sedici di loro hanno filtri rossi, un altro 16 hanno filtri verdi, altri 16 hanno filtri blu e gli altri 16 fotodiodi sono chiari senza filtri.





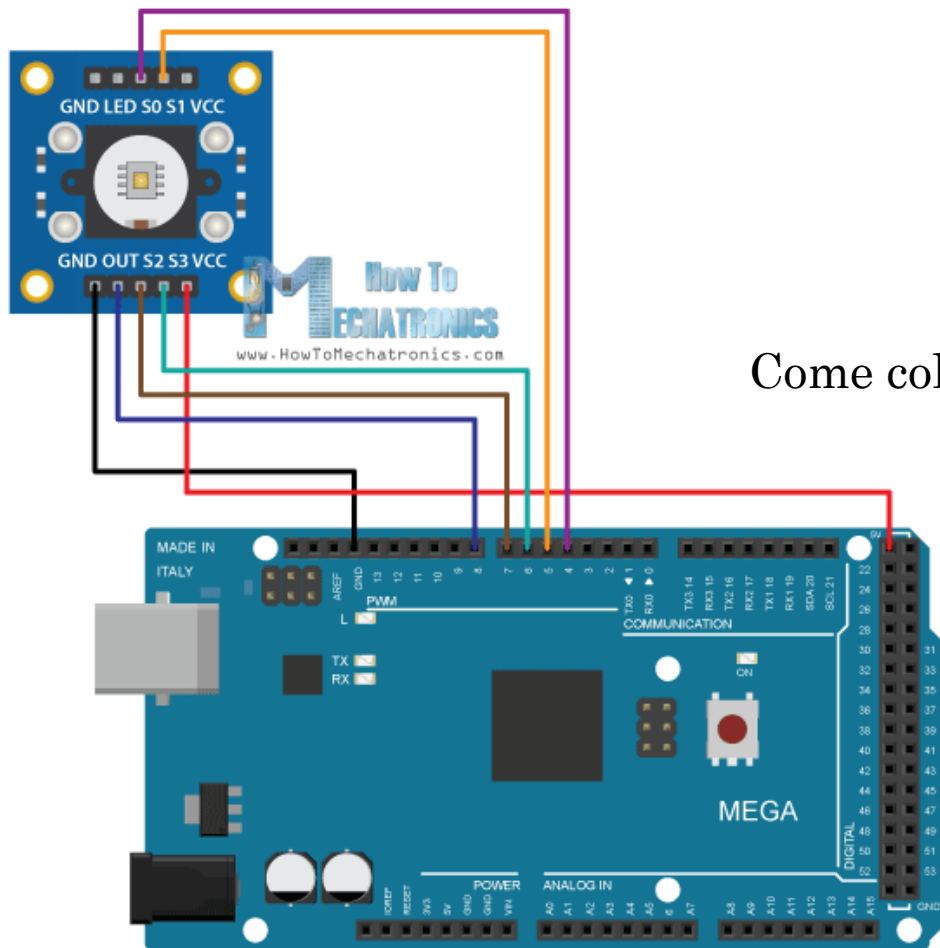
Ciascuno dei 16 fotodiodi sono collegati in parallelo, così utilizzando i due pin di controllo S2 e S3 possiamo selezionare quale di essi verrà letto. Così, per esempio, se vogliamo rilevare il colore rosso, è sufficiente utilizzare i 16 fotodiodi rossi filtrati settando i due pin al livello logico low in base alla seguente tabella.

<i>S0</i>	<i>S1</i>	<i>Output Frequency Scaling</i>
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

<i>S2</i>	<i>S3</i>	<i>Photodiode Type</i>
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green



Il sensore ha due pin di controllo, S0 e S1 che vengono utilizzati per scalare la frequenza di uscita. La frequenza può essere scalata a tre diversi valori preimpostati del 100%, 20% o 2%. Questa funzione di frequency-scaling permette di ottimizzare l'uscita del sensore per vari contatori di frequenza o microcontrollori.



Come collegare il sensore TCS230



# DESCRIZIONE DEL CODICE

Prima bisogna definire i pin a cui è collegato il sensore e definire una variabile per leggere la frequenza. Nella sezione di setup abbiamo bisogno di definire i quattro pin di controllo come uscite e l'uscita del sensore come input di Arduino. Qui abbiamo anche bisogno di impostare la frequenza-scaling, per questo esempio impostiamo il 20%, e avviamo la comunicazione seriale per la visualizzazione dei risultati nel monitor seriale

Nella sezione loop , si inizierà con la lettura dei fotodiodi rossi filtrati. Per questo scopo imposteremo i due pin di controllo S2 e S3 a livello logico basso. Quindi, utilizzando la funzione "pulseIn ()" leggeremo la frequenza di uscita che verrà posta nella variabile "frequency". Utilizzando la funzione Serial.print () verrà stampato il risultato sul monitor seriale. La stessa procedura vale per gli altri due colori, abbiamo solo bisogno di regolare i pin di controllo per il colore appropriato secondo la precedente tabella.



# IL CODICE

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
int frequency = 0;

void setup() {
    pinMode(S0, OUTPUT);
    pinMode(S1, OUTPUT);
    pinMode(S2, OUTPUT);
    pinMode(S3, OUTPUT);
    pinMode(sensorOut, INPUT);
    // Setting frequency-scaling to 20%
    digitalWrite(S0,HIGH);
    digitalWrite(S1,LOW);
    Serial.begin(9600);
}
```



# IL CODICE

```
void loop() {  
  // Setting red filtered photodiodes to be read  
  digitalWrite(S2,LOW);  
  digitalWrite(S3,LOW);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  // Printing the value on the serial monitor  
  Serial.print("R= "); //printing name  
  Serial.print(frequency); //printing RED color frequency  
  Serial.print(" ");  
  delay(100);  
  // Setting Green filtered photodiodes to be read  
  digitalWrite(S2,HIGH);  
  digitalWrite(S3,HIGH);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  // Printing the value on the serial monitor  
  Serial.print("G= "); //printing name  
  Serial.print(frequency); //printing RED color frequency  
  Serial.print(" ");  
  delay(100);  
}
```



# IL CODICE

```
// Setting Blue filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
// Printing the value on the serial monitor
Serial.print("B= ");//printing name
Serial.print(frequency);//printing RED color frequency
Serial.println(" ");
delay(100);
}
```







Così se ora vogliamo rappresentare i colori rilevati con un RGB Model, che ha valori da 0 a 255, useremo la funzione della mappa () per mappare o convertire le letture con valori da 0 a 255.

**// Remaping il valore della frequenza al modello RGB di 0 a 255**

**frequency = map( frequency, 25 , 70 , 255 , 0 ) ;**

Il valore di 70 verrà associato a 0, e il valore di 25 a 255. La stessa procedura vale per gli altri due colori.

```
#define S0 4
```

```
#define S1 5
```

```
#define S2 6
```

```
#define S3 7
```

```
#define sensorOut 8
```

```
int frequency = 0;
```

```
void setup() {
```

```
pinMode(S0, OUTPUT);
```

```
pinMode(S1, OUTPUT);
```

```
pinMode(S2, OUTPUT);
```

```
pinMode(S3, OUTPUT);
```

```
pinMode(sensorOut, INPUT);
```

```
// Setting frequency-scaling to 20%
```

```
digitalWrite(S0,HIGH);
```

```
digitalWrite(S1,LOW);
```

```
Serial.begin(9600); }
```



```
void loop() {  
  // Setting red filtered photodiodes to be read  
  digitalWrite(S2,LOW);  
  digitalWrite(S3,LOW);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  //Remaping the value of the frequency to the RGB Model of 0 to 255  
  frequency = map(frequency, 25,72,255,0);  
  // Printing the value on the serial monitor  
  Serial.print("R= ");//printing name  
  Serial.print(frequency);//printing RED color frequency  
  Serial.print(" ");  
  delay(100);  
  // Setting Green filtered photodiodes to be read  
  digitalWrite(S2,HIGH);  
  digitalWrite(S3,HIGH);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  //Remaping the value of the frequency to the RGB Model of 0 to 255  
  frequency = map(frequency, 30,90,255,0);  
  // Printing the value on the serial monitor  
  Serial.print("G= ");//printing name  
  Serial.print(frequency);//printing RED color frequency  
  Serial.print(" ");  
  delay(100);  
}
```



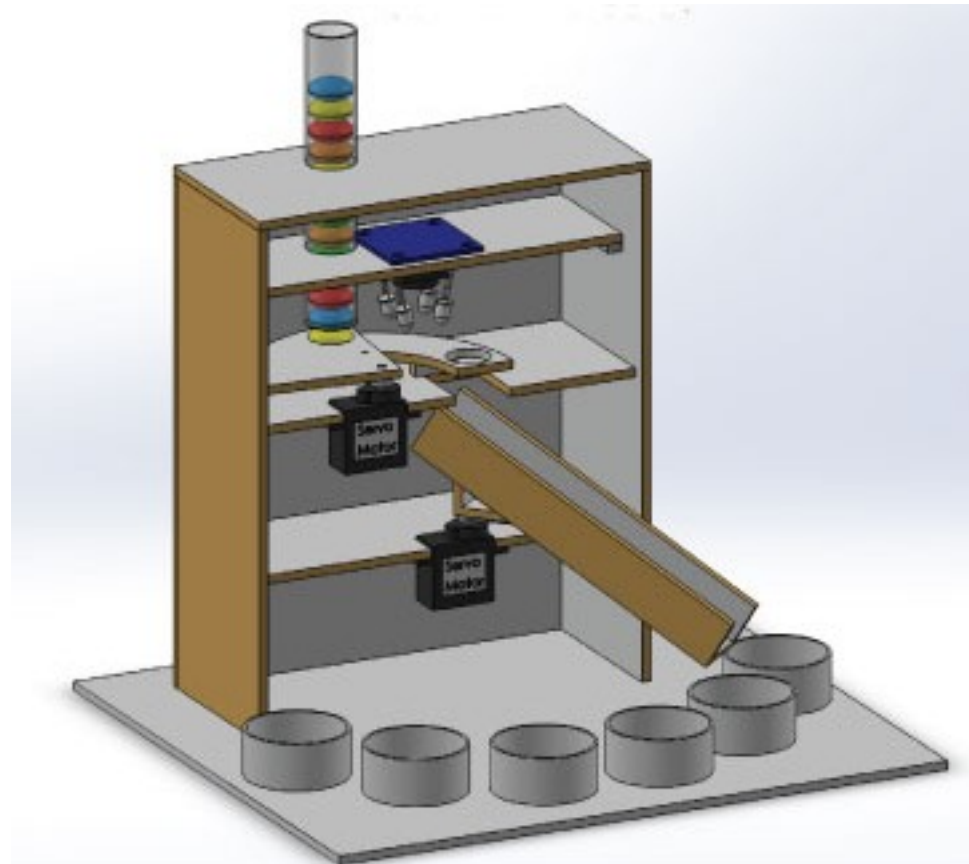
```
// Setting Blue filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
// Reading the output frequency
frequency = pulseIn(sensorOut, LOW);
//Remaping the value of the frequency to the RGB Model of 0 to 255
frequency = map(frequency, 25,70,255,0);
// Printing the value on the serial monitor
Serial.print("B= ");//printing name
Serial.print(frequency);//printing RED color frequency
Serial.println(" ");
delay(100);
}
```

Si noti che i colori non sono molto accurati ma sono accettabili per progetti semplici.



# LABORATORIO

## DISTRIBUTORE DI PALLINE COLORATE



*dal sito....* <http://howtomechatronics.com/projects/arduino-color-sorter-project/>

# FUNZIONAMENTO

- Inizialmente, le palline colorate vengono inserite nel caricatore e fatte cadere nella piattaforma attaccata sul servomotore superiore.
- Poi il servomotore ruota e porta la pallina al sensore di colore TCS3200 che ne rileva il colore.
- Dopo che il servomotore inferiore ruota nella posizione particolare corrispondente al contenitore che dovrà accogliere la pallina del colore individuato il servomotore superiore ruota fino alla caduta della pallina nella guida.
- Ecco il filmato  
.....<https://www.youtube.com/watch?v=g3i51hdfLaw>

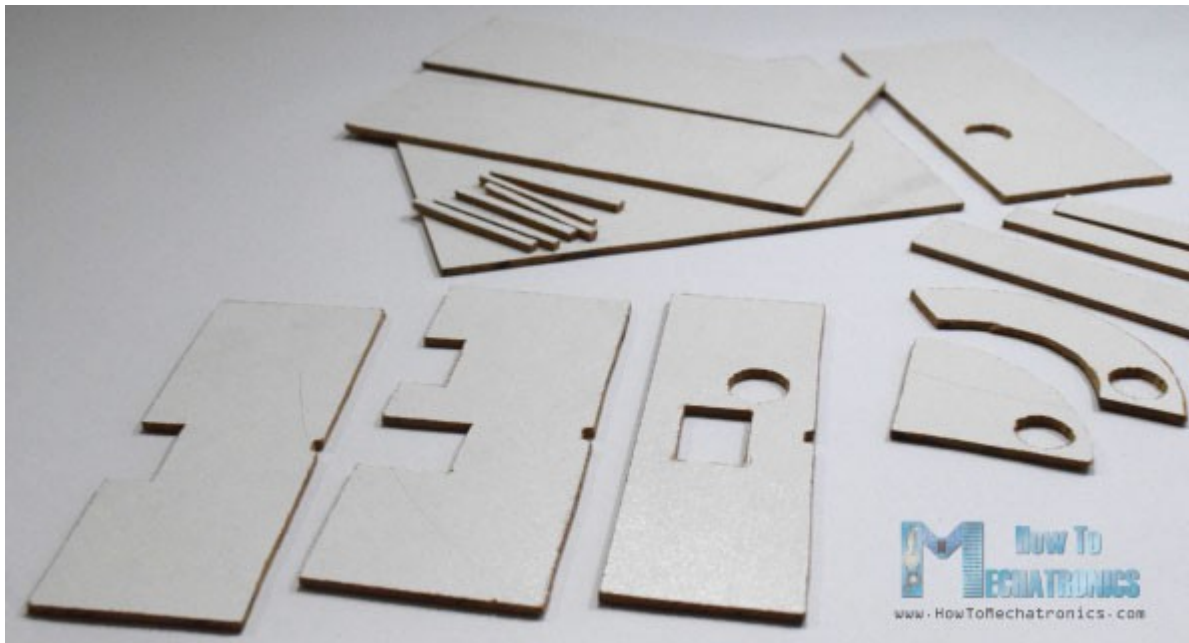


# COMPONENTI

Tutto quello di cui abbiamo bisogno per questo progetto è un sensore colore (TCS3200) e due servomotori.

Utilizzando della masonite o altro materiale possiamo disegnare e ritagliare i vari pezzi della struttura.

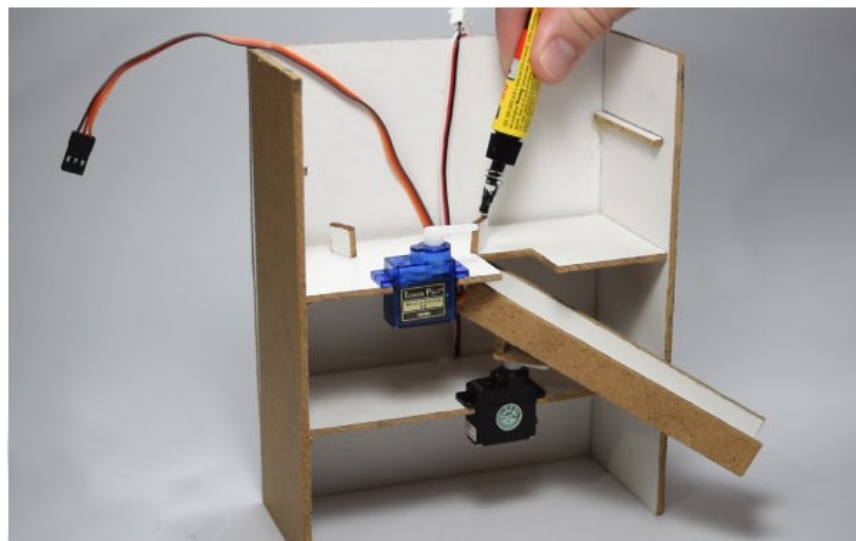
[Scarica i disegni](#)



Ottenute tutte le varie parti si assemblano con una pistola per colla partendo dalle parti esterne.

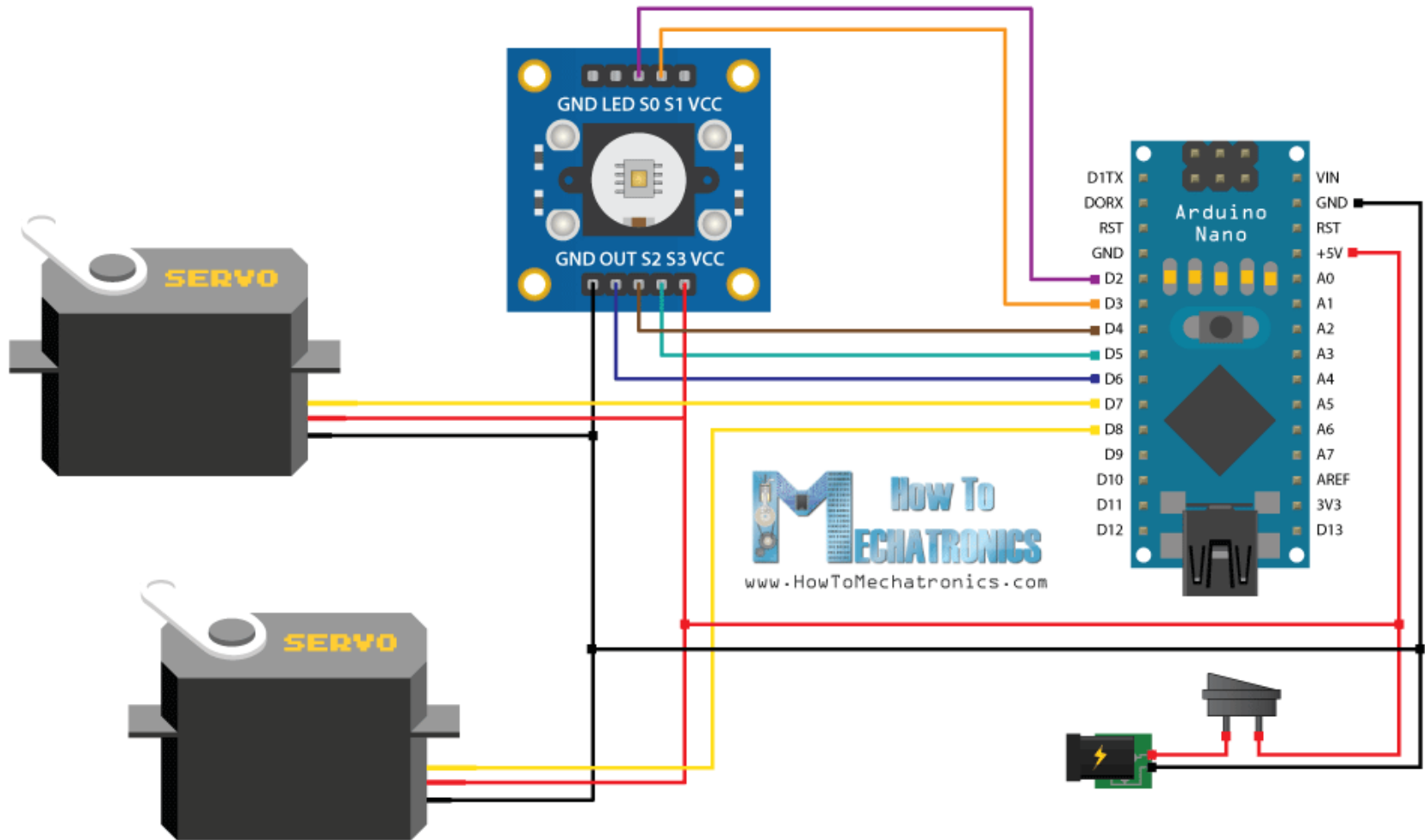


Quindi, si fissano i due servomotori sulle loro piattaforme e la guida sul servomotore inferiore così come il supporto e la piattaforma necessaria per il servomotore superiore.





Successivamente, possiamo inserire un interruttore e un jack di alimentazione per alimentare Arduino con un adattatore a 5V e sulla terza piattaforma inseriamo il sensore di colore.



Terminato il montaggio delle varie parti , ecco il codice Arduino:

```
#include <Servo.h>  
#define S0 2  
#define S1 3  
#define S2 4  
#define S3 5  
#define sensorOut 6  
  
Servo topServo;  
Servo bottomServo;  
int frequency = 0;  
int color=0;  
  
void setup() {  
    pinMode(S0, OUTPUT);  
    pinMode(S1, OUTPUT);  
    pinMode(S2, OUTPUT);  
    pinMode(S3, OUTPUT);  
    pinMode(sensorOut, INPUT);  
    digitalWrite(S0, HIGH);  
    digitalWrite(S1, LOW);  
    topServo.attach(7);  
    bottomServo.attach(8);  
    Serial.begin(9600); }
```



```
void loop() {  
topServo.write(115);  
delay(500);  
for(int i = 115; i > 65; i--)  
    { topServo.write(i);  
      delay(2); }  
delay(500);  
color = readColor();  
delay(10);  
switch (color) {  
case 1:  
    bottomServo.write(50); break;  
case 2:  
    bottomServo.write(75);break;  
case 3:  
    bottomServo.write(100);break;  
case 4:  
    bottomServo.write(125);break;  
case 5:  
    bottomServo.write(150);break;  
case 6:  
    bottomServo.write(175);break;  
case 0: break; }  
delay(300);
```



```
for(int i = 65; i > 29; i--) {
    topServo.write(i);
    delay(2); }

delay(200);
for(int i = 29; i < 115; i++) {
    topServo.write(i);
    delay(2); }

color=0;
}
// Custom Function - readColor()
int readColor() {
// Setting red filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, LOW);
// Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int R = frequency;
// Printing the value on the serial monitor
    Serial.print("R= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.print(" ");
delay(50);
```



```
// Setting Green filtered photodiodes to be read
    digitalWrite(S2, HIGH);
    digitalWrite(S3, HIGH);
// Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int G = frequency;
// Printing the value on the serial monitor
    Serial.print("G= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.print(" ");
    delay(50);
// Setting Blue filtered photodiodes to be read
    digitalWrite(S2, LOW);
    digitalWrite(S3, HIGH);
// Reading the output frequency
    frequency = pulseIn(sensorOut, LOW);
    int B = frequency;
// Printing the value on the serial monitor
    Serial.print("B= "); //printing name
    Serial.print(frequency); //printing RED color frequency
    Serial.println(" ");
    delay(50);
```




```
if(R<45 & R>32 & G<65 & G>55) { color = 1; // Red }
if(G<55 & G>43 & B<47 & B>35) { color = 2; // Orange }
if(R<53 & R>40 & G<53 & G>40) { color = 3; // Green }
if(R<38 & R>24 & G<44 & G>30) { color = 4; // Yellow }
if(R<56 & R>46 & G<65 & G>55) { color = 5; // Brown }
if (G<58 & G>45 & B<40 & B>26) { color = 6; // Blue }
return color;
}
```

### Descrizione del codice:

Quindi, abbiamo bisogno di includere la libreria "Servo.h", definire i pin a cui verrà collegato il sensore di colore, creare gli oggetti servo e dichiarare alcune variabili necessarie per il programma. Nella sezione di **setup** abbiamo bisogno di definire i pin di ingresso e di uscita, impostare la frequenza di scala per il sensore di colore, definire i pin per i servo e avviare la comunicazione seriale per la stampa dei risultati del colore di leggere sul monitor seriale.

Nella sezione **loop**, il programma inizia con lo spostamento del servomotore superiore nella posizione del caricatore bowling. Si noti che il valore di 115 si adatta alle parti ma lo si potrebbe variare in base al servomotore che si utilizza.



Nel successivo ciclo “for” si ruoterà in modo da portare la pallina in prossimità del sensore di colore.

Si utilizza un ciclo "for" in modo da poter controllare la velocità di rotazione, modificando il delay nel loop.

Successivamente, dopo mezzo secondo di ritardo, utilizzando la funzione di misura, `readColor ()` leggeremo il colore della pallina

Utilizzando i quattro pin di controllo e il pin di uscita frequenza del sensore di colore si legge colore della pallina Il sensore legge i 3 valori diversi rosso, verde e blu e in base a questi valori ci dice qual è il colore reale. Per maggiori dettagli su come il sensore di colore TCS3200 funziona è possibile controllare il relativo tutorial.

<https://www.youtube.com/watch?v=CPUXxuyd9xw>

I valori RGB ricevuti dal sensore per ogni pallina possono variare perché i sensori non sempre sono precisi. Pertanto, l'utilizzo di questi “if” permettono al sensore di un errore di circa  $\pm 5$  % del valore testato per quel particolare colore. Così, per esempio se abbiamo una pallina rossa, il primo "if" sarà vero e la variabile "colore" otterrà il valore 1. Ecco, questo è ciò che la funzione personalizzata `readColor ()` fa e dopo con l'utilizzo di un "switch-case “ ruotiamo il servo inferiore nella posizione voluta. Alla fine si ruota ulteriormente il servomotore superiore per fare in modo che la pallina cada nella guida per farlo ritornare nella posizione iniziale in modo che il processo possa ripetersi.

Fissato Arduino con un pò di silicone a caldo, con un tubo di plastica possiamo realizzare il caricatore come in figura:

