Ain shams university
Faculty of engineering

Senior 1 Mechatronics

Sensors and measurements (MCT334)

Task 1

Team 18 “Industrial Sorting”

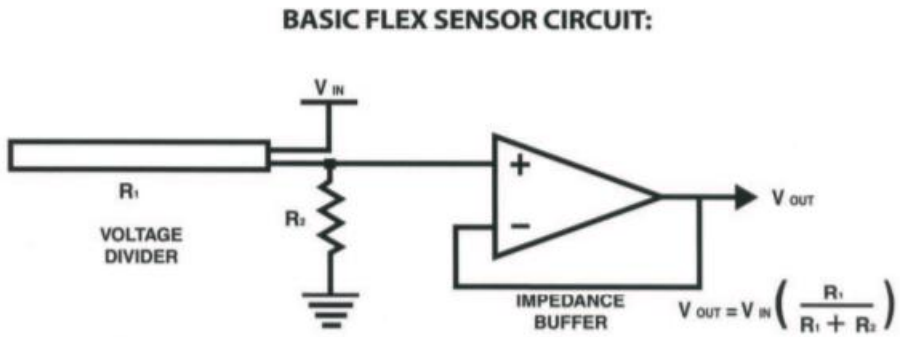
Name	ID	Sec
Elsayed Ayman Elsayed Ali Habib	**1804765**	**4**
Mohamed Emad Mohamed Moslem	**1806770**	**2**
Anas Ahmed Talaat	**1806766**	**4**

First Sensor : Flex Sensor 2.2" (SEN-10264) :

Overview:

- Flex sensors are sensors that change in resistance depending on the amount of bend on the sensor.
- They convert the change in bend to electrical resistance - the more the bend, the more the resistance value.
- When the sensor straightens out again, the resistance returns to the original value.
- By measuring the resistance, you can determine how much the sensor is being bent.

Datasheet details of flex sensor .



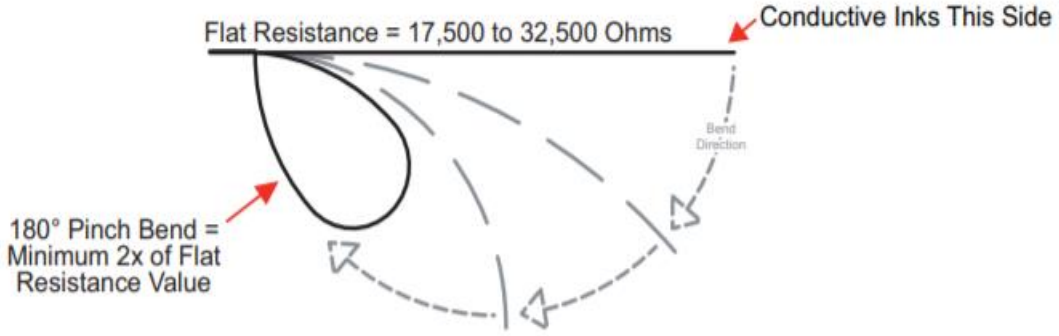
➢ Mechanical Specifications

- Life Cycle: >1 million
- Height: 0.43mm (0.017")
- Temperature Range: -35°C to +80°C

➢ Electrical Specifications

- Flat Resistance: 25K Ohms
- Resistance Tolerance: ±30%
- Bend Resistance: minimum 2 times greater than the flat resistance at 180° pinch bend
- Power Rating : 0.50 Watts continuous.
- 1 Watt Peak

How it work :



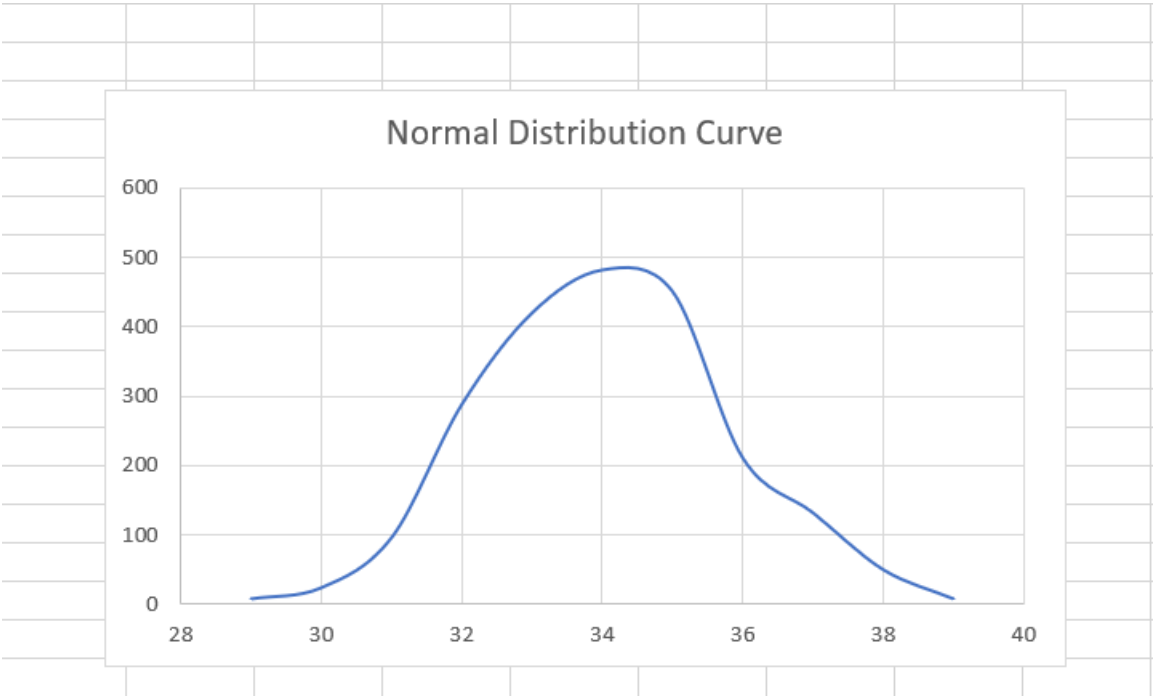
Datasheet details that we use during calibration & practical working :

- Calibrated directly in kilohms by using ***R2*** = 1000 kilohms.
- **Reference** : Change in resistance from approximately **10 to 50 kilohms** when the sensor is **bent to 90º.**

Our analysis on the sensor Readings :

	Value	No.of repetitions
	29	7
	30	23
	31	95
	32	287
	33	419
	34	481
	35	450
	36	210
	37	131
	38	49
	39	7
MIN	29	
MAX	39	
MEAN (avrage)	34.04122279	
STANDARD DEVIATION	1.720687403	
MEDIAN	34	

Normal Distribution curve :

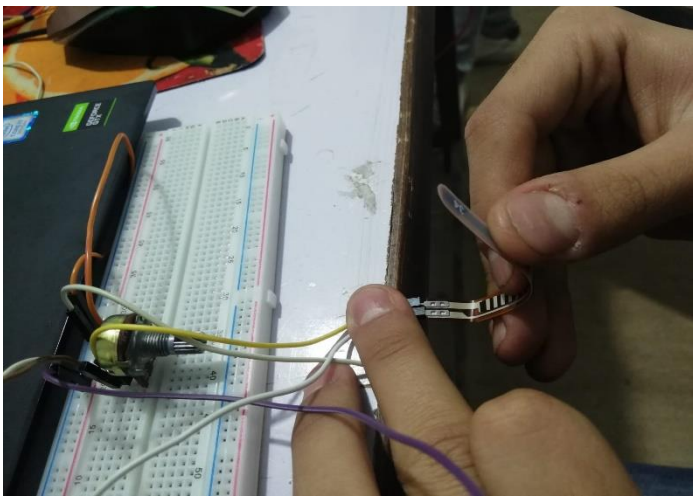


The circuit used in the calibration process :

lm35 §

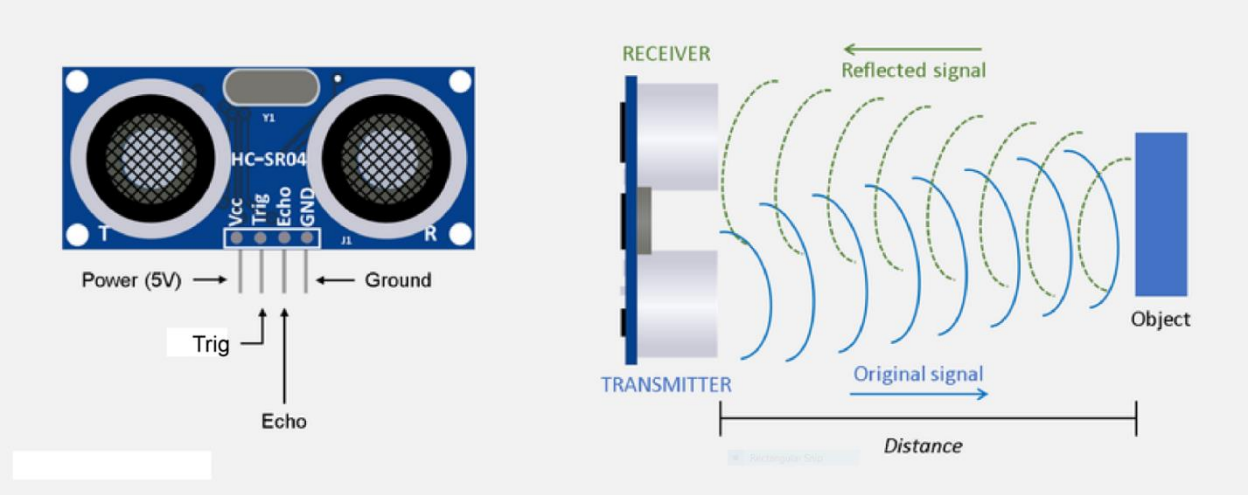
```
int flex = A0;
int data = 0;
void setup()
{
  Serial.begin(9600);
  pinMode(flex,INPUT);
}
void loop()
{
  data = analogRead(flex);
  Serial.println(data);
  delay(10);

}
```



Second sensor: Ultrasonic Ranging Module (HC - SR04).

Overview:

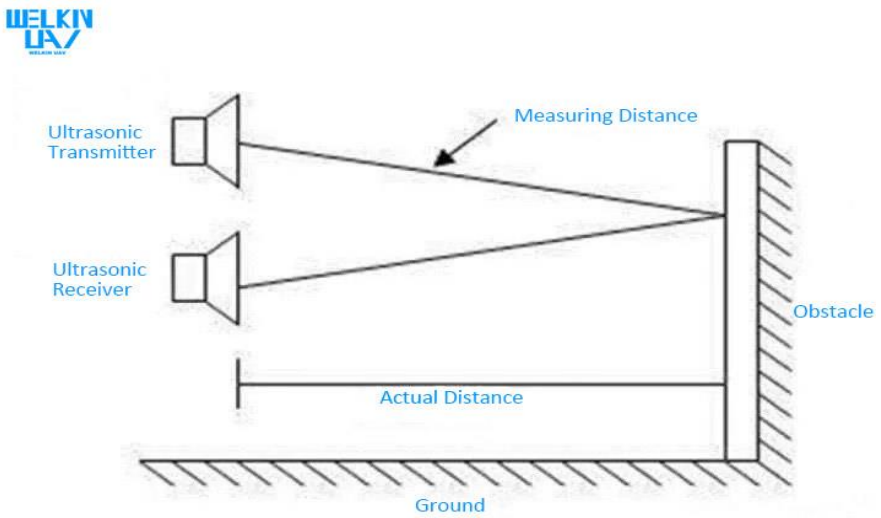


- The basics of using ultrasound are like this: you shoot out a sound, wait to hear it echo back, and if you have your timing right, you'll know if anything is out there and how far away it is. This is called echolocation and it's how bats and dolphins find objects in the dark and underwater, though they use lower frequencies than you can use with your Arduino.

The principle of ultrasonic ranging :

- sound waves are reflected by obstacles, and the speed of sound waves is known, so it is only necessary to know the time difference between transmission and reception, and the measurement distance can be easily calculated, combined with transmitter and receiver. The distance of the device can calculate the actual distance of the obstacle

Datasheet details of ultrasonic :



Electrical Specifications

- Operating Voltage: 3.3Vdc ~ 5Vdc
- Quiescent Current: <2mA
- Operating Current: 15mA
- Operating Frequency: 40KHz
- Operating Range & Accuracy: 2cm ~ 400cm (1in ~ 13ft) ± 3mm
- Sensitivity: -65dB min
- Sound Pressure: 112dB
- Effective Angle: 15°

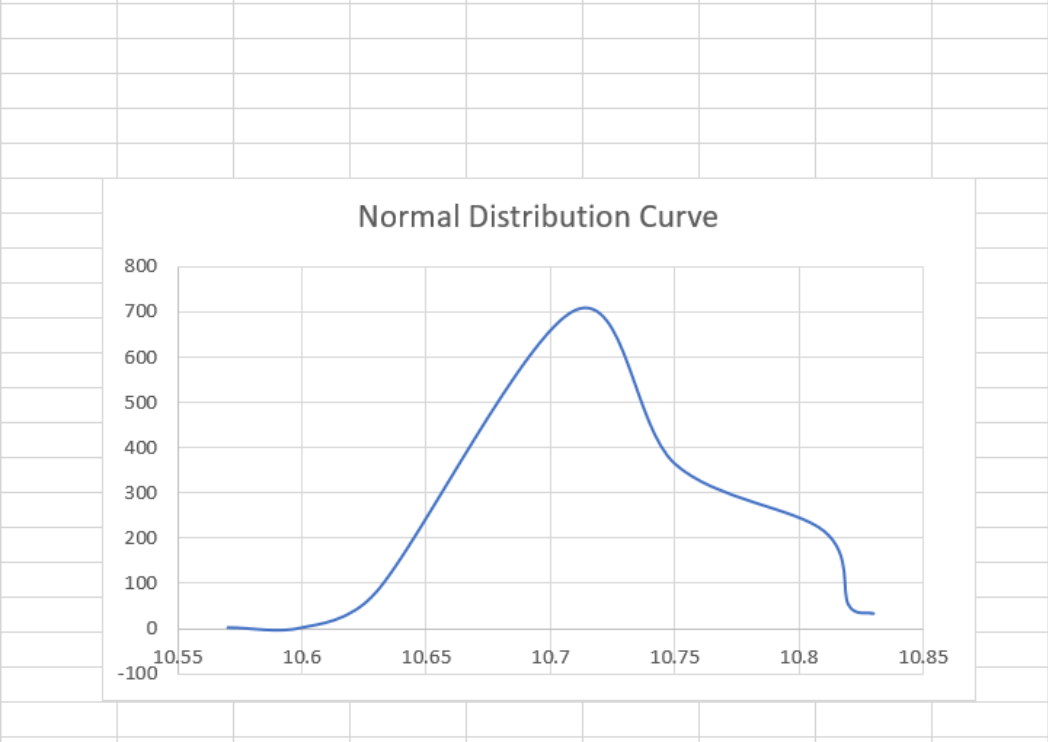
Datasheet details that we use during calibration & practical working:

- Fixation of the target at **10.7 cm** from the ultrasonic.

Our analysis on the sensor Readings:

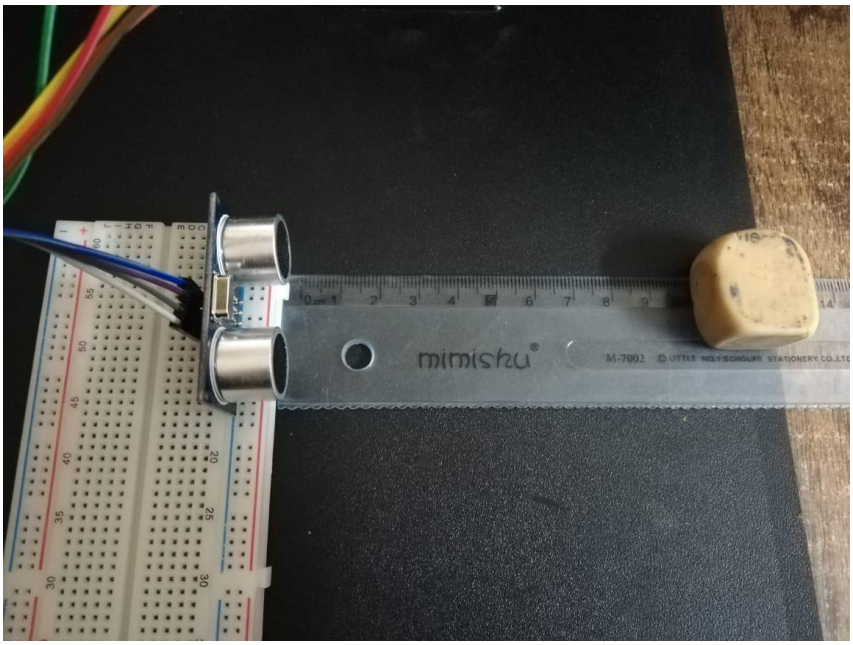
	value	repeated
	10.57	1
	10.6	1
	10.63	81
	10.71	705
	10.75	364
	10.81	215
	10.82	50
	10.83	32
MIN	10.57	
MAX	10.83	
MEAN (avrage)	10.73669	
STANDARED DEVIATION	0.047857	
MEDIAN	10.71	

Normal Distribution curve:



The circuit used in the calibration process:

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
}
void loop() {
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance= duration*0.034/2;
// Prints the distance on the Serial Monitor
Serial.println(distance);
}
```



Third Sensor: Capacitive Sensor(LM12-2004A)

Overview:

surveillance of metal elements motion. They can also be used in other machine systems applications as no contact
control sensors for level of liquids, control sensors for the speed and position of rotating chains, etc. They are
developed on the base of PNP and NPN transition. They have small dimensions and feature a cylindrical metal shell
resistant to vibrations and a plastic lid which is oil and water resistant.

Datasheet details of Capacitive Sensor:

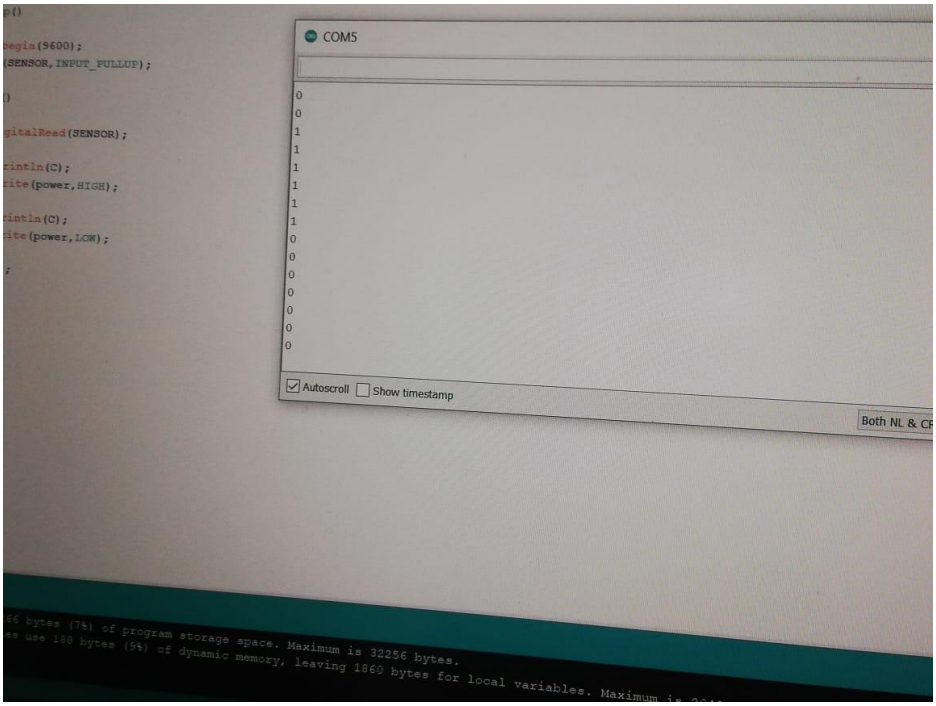
Electrical Specifications

- Insulating resistance: ≥ 50 M Ω
- Gearing distance: from 2 mm to 7 mm
- Precision of repetition: 0.01
- Ambient temperature: -25˚C to +65˚C
- Gearing speed: 5mm/s
- Rated voltage: 6~36VDC; 90~230VAC

Datasheet details that we use during calibration & practical working:

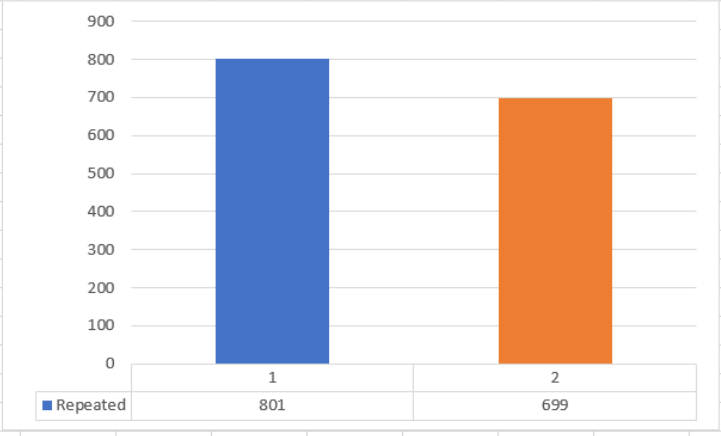
- Fixation of the target at 4,00mm from the capacitive sensor.
- Usingrelay 5VDC.

Our analysis on the sensor Readings:



Normal Distribution curve:

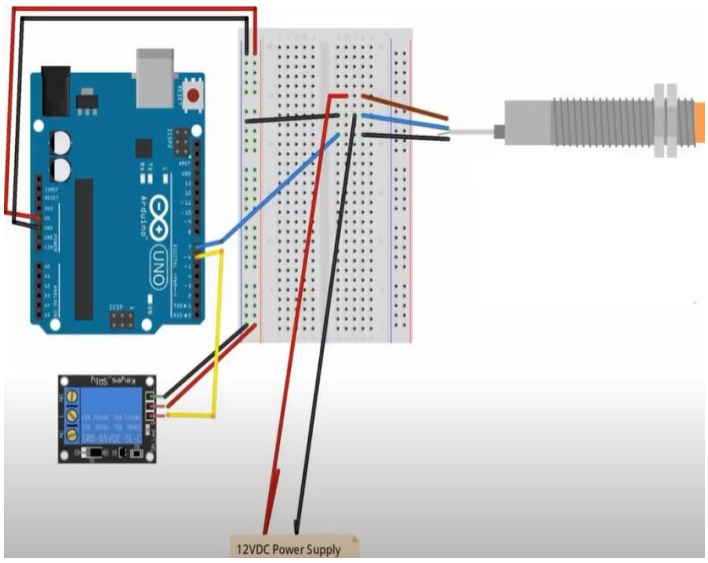
	Reading	Repeated
	0	801
	1	699
Standared devitation	0.499009	
Mean (Average)	0.466	



The circuit used in the calibration process:

lm35

```
#define SENSOR 7
#define power 5
void setup()
{
  Serial.begin(9600);
  pinMode(SENSOR,INPUT_PULLUP);
}
void loop()
{
 int C =digitalRead(SENSOR);
if (C==0) {
  Serial.println(C);
  digitalWrite(power,HIGH);
}else{
  Serial.println(C);
  digitalWrite(power,LOW);
}
delay(1000);
}
```



Forth sensor: Colour Sensor Module (TCS3200).

Overview:

This **Arduino compatible TCS3200 color sensor module** consist of a TAOS TCS3200 RGB sensor chip and 4 white LEDs. The main part of the module is the **TCS3200 chip** which is a Color Light-to-Frequency Converter. The white LEDs are used for providing proper lighting for the sensor to detect the object colour correctly. This chip can sense a wide variety of colours and it gives the output in the form of corresponding frequency. This module can be used for making colour sorting robots, test strip reading, colour matching tests, etc.

The **TCS3200 chip** consist of an 8 x 8 array of photodiodes. Each photodiode have either a red, green, or blue filter, or no filter. The filters of each color are distributed evenly throughout the array to eliminate location bias among the colors. Internal circuits includes an oscillator which produces a square-wave output whose frequency is proportional to the intensity of the chosen color.

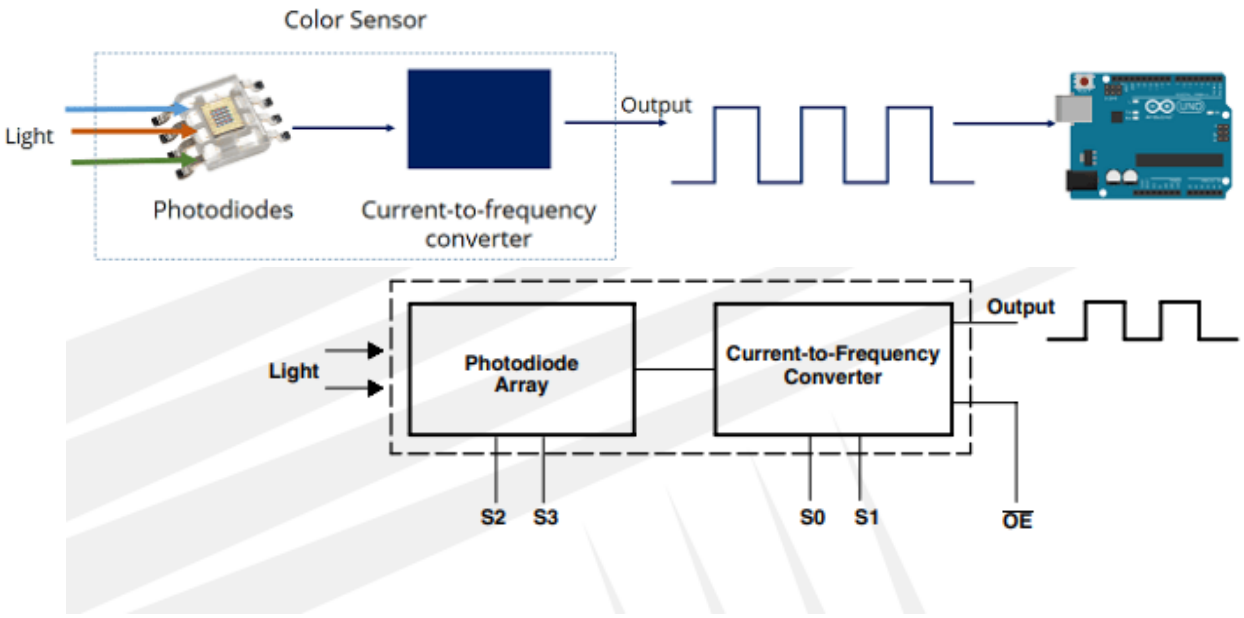
The principle of Color sensor working :

The TCS3200 has an array of photodiodes with 4 different filters. A photodiode is simply a semiconductor device that converts light into current. The sensor has:

- 16 photodiodes with red filter – sensitive to red wavelength
- 16 photodiodes with green filter – sensitive to green wavelength
- 16 photodiodes with blue filter – sensitive to blue wavelength
- 16 photodiodes without filter

If you take a closer look at the TCS3200 chip you can see the different filters.

By selectively choosing the photodiode filter's readings, you're able to detect the intensity of the different colors. The sensor has a current-to-frequency converter that converts the photodiodes' readings into a square wave with a frequency that is proportional to the light intensity of the chosen color. This frequency is then, read by the Arduino – this is shown in the figure.



Datasheet details of Colour Sensor(TCS3200):

Features and Specifications

- Input voltage: (2.7V to 5.5V)
- Interface: Digital TTL
- High-resolution conversion of light intensity to frequency
- Programmable colour and full-scale output frequency
- No need of ADC(Can be directly connected to the digital pins of the microcontroller)
- Power down feature
- Working temperature: -40oC to 85oC
- Size: 28.4x28.4mm(1.12x1.12")

Datasheet details that we use during calibration & practical working:

Detecting an object that have Green Color

Ain shams university
Faculty of engineering

Our analysis on the sensor Readings :

Red	repeated for Red
44	2
46	4
47	4
48	8
49	10
50	22
51	32
52	42
53	50
54	55
64	79
65	80
71	60
72	50
73	18
74	8
MEAN	60.92647059
Standard Dev.	8.639821594

Red Filter

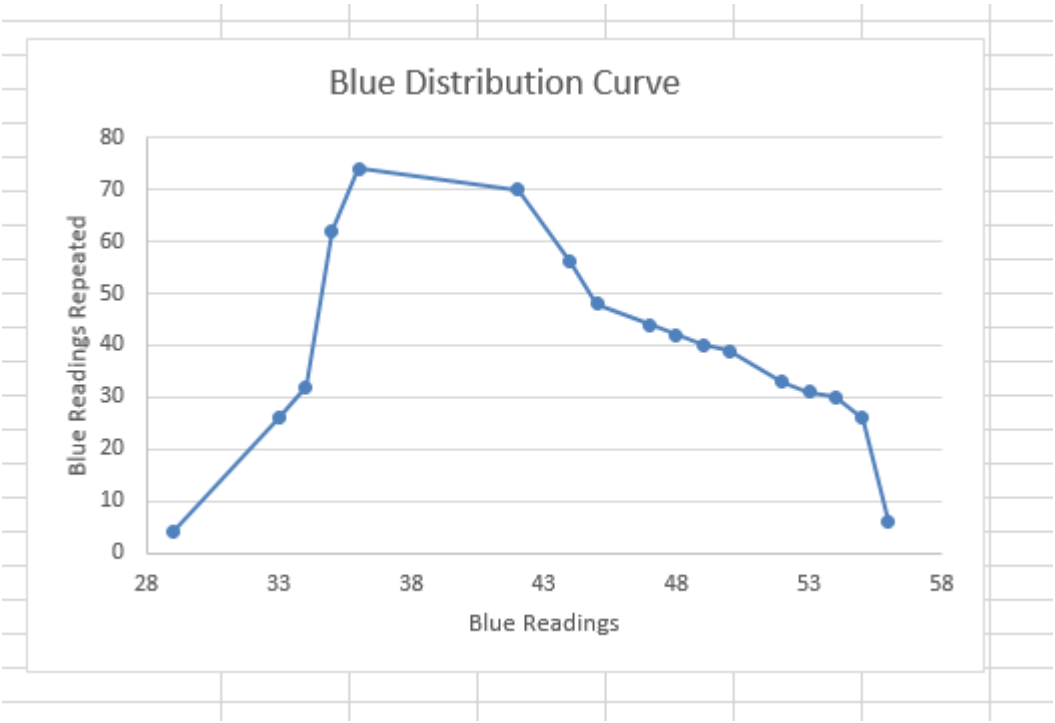
Green	Reapeted for Green
23	2
24	2
25	4
28	4
29	6
30	50
31	82
32	88
33	92
34	100
35	95
36	89
38	66
40	58
41	57
42	51
44	42
45	39
46	20
48	4
MEAN	36.1719457
Standard Dev.	4.906586538

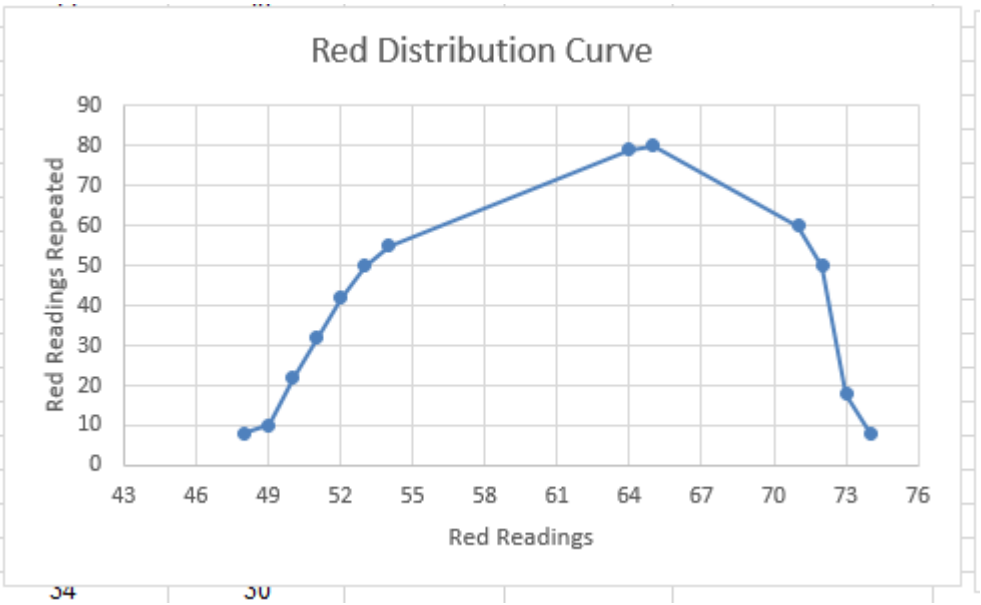
Green Filter

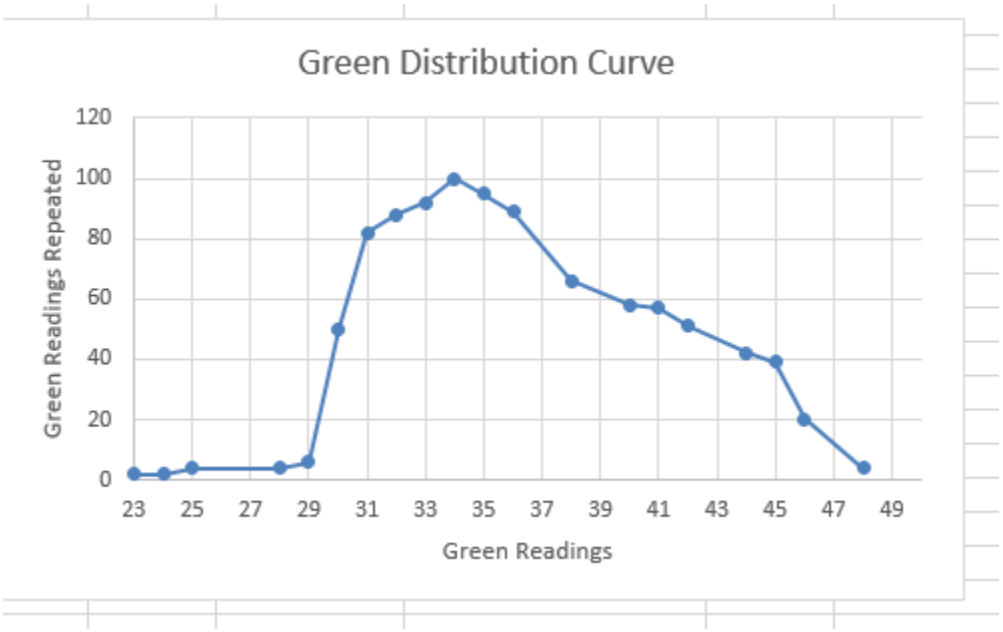
Blue	Reapeted for Blue
29	4
33	26
34	32
35	62
36	74
42	70
44	56
45	48
47	44
48	42
49	40
50	39
52	33
53	31
54	30
55	26
56	6
MEAN	44.94878706
Standard Dev.	7.121469119

Blue Filter

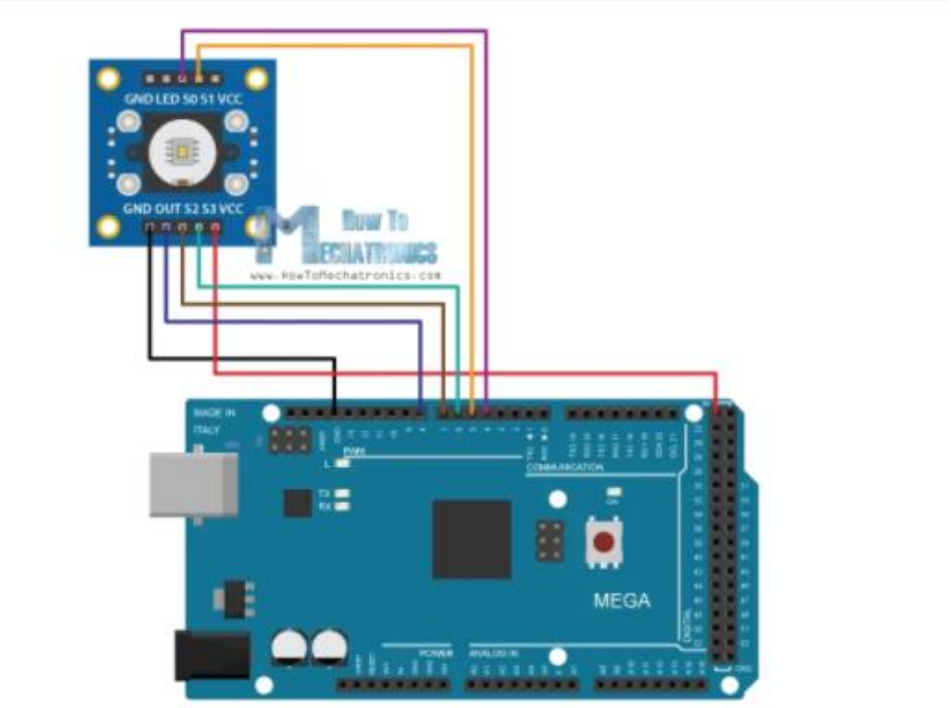
Normal Distribution curve :







The circuit used in the calibration process :



```
#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("Red= ");//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("Green= ");//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("Blue= ");//printing name
  Serial.print(frequency);//printing RED color frequency
  Serial.println("  ");
  delay(100);
}
```