Automatic Light Intensity Controlled Curtains:

By TEAM MELHOR

PROJECT REPORT

TEAM MEMBERS AND CONTRIBUTIONS:

Keshav Garg

Coordination, Circuits and Calculations

Karan Nag

Software and Coding

Shelly Aggarwal

Circuits, Research Work and Mechanical parts assembly

Aditya Chandrikapure

Hardware and Assembly

Akshat Chaudhary

Presentations, Lab Report and Calculations

Total Parts and Expenses

1. Bluetooth Module- 282 Rs
2. Jumper Wires- Around 100 Rs
3. Breadboard- Around 80 Rs
4. Arduino uno R3- Around 550 Rs
5. 2 resistors - 1 k(ohm) - 185 Rs for 5
6. 2 photoresistor- 95 Rs for 5
7. Servo Motors- 207 Rs
8. Curtains- Around 100 Rs
9. Strings- Inexpensive

Total Amount needed is Rs. 1600 (Max.)

PROJECT OVERVIEW:

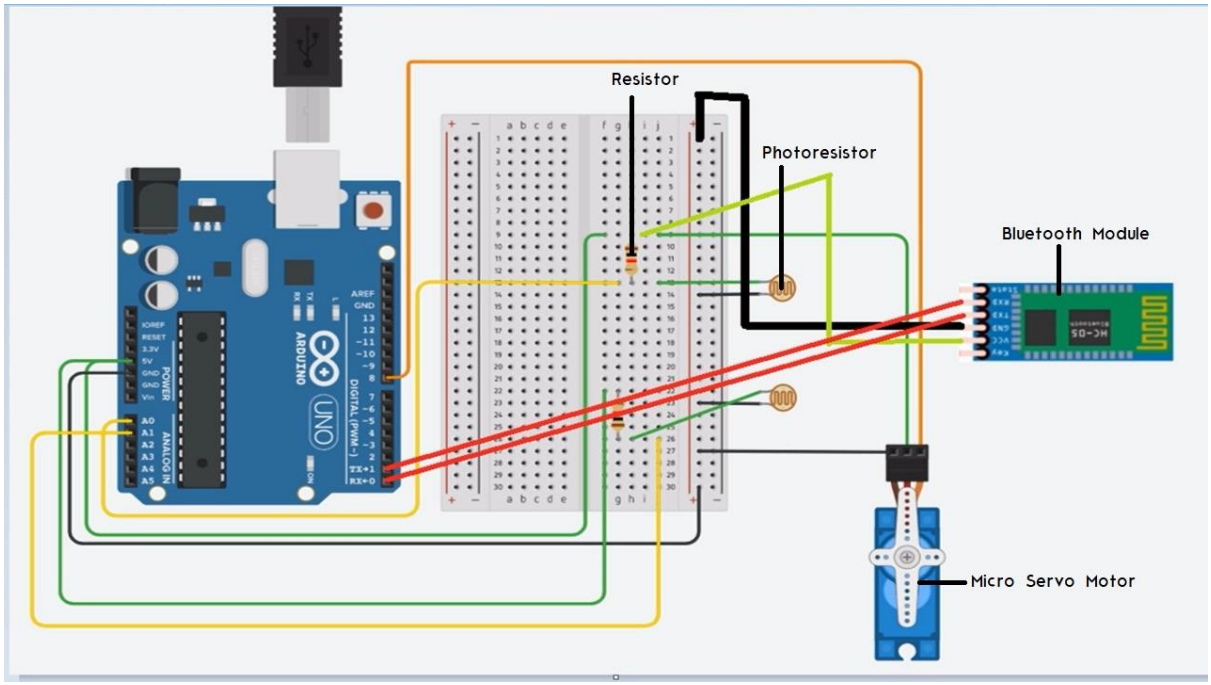
We have made automatic curtains with light sensor and voice command to make it easy to control them from anywhere. These curtains adjust themselves accordingly according to the light levels outside or we can simply give them voice commands to control them manually from a distance.

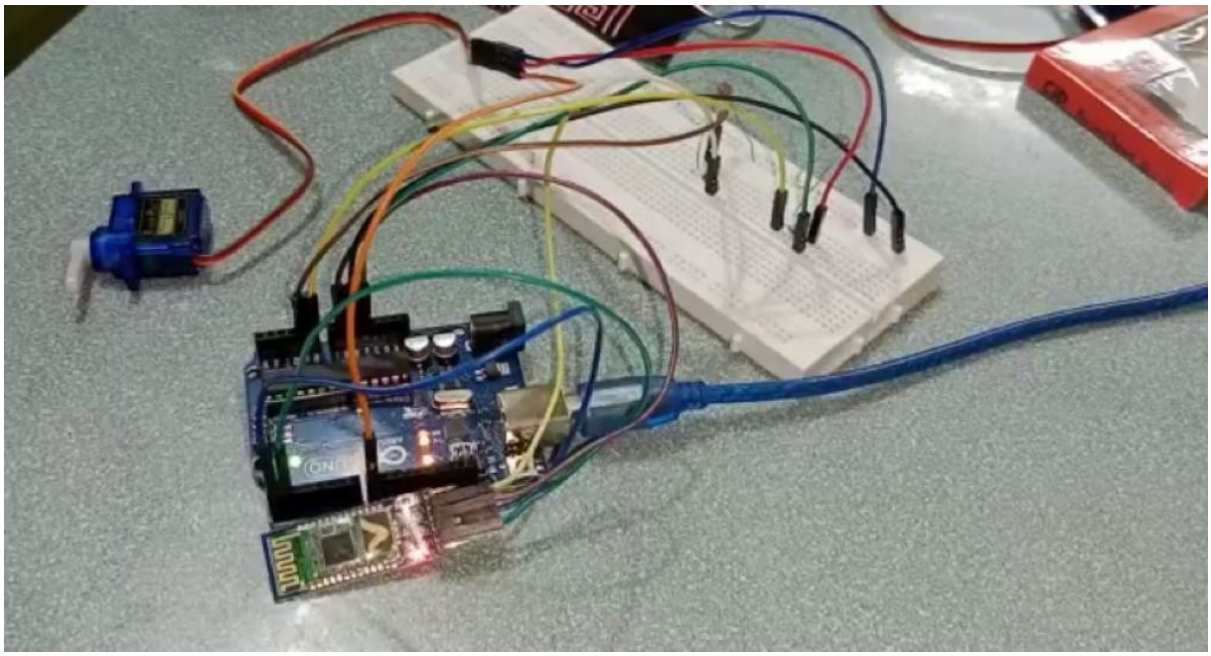
We used a photoresistor and Bluetooth module. The Bluetooth module transfers the voice commands to the Arduino which gives the output to the servo motors. In automatic mode, the photoresistors take the light values and give them to the Arduino board.

Our Automatic Curtains include several modes such as Automatic, Manual, Fully Off, Fully On and Voice Command. These modes are useful at different situations and weather

Cost of the total project components comes under just Rs. 2000 which makes it feasible to make and assemble.

CIRCUIT DIAGRAM and PICTURE





UPPER- Circuit diagram in TinkerCAD showing the Automatic Curtain Circuit

LOWER- Actual circuit of the project.

Software Code

```
#include <Servo.h>
#include <SoftwareSerial.h>

String voice;
int light1=0;
int light2=0;
int TxD=0;
int RxD=1;
Servo servo_8;
SoftwareSerial BT(TxD,RxD); //TX, RX respetively

void setup() {
 BT.begin(9600);
 Serial.begin(9600);
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(8,OUTPUT);
  servo_8.attach(8,500,2500);
}

void automatic()
{
 delay(10);
 light1 = analogRead(A0);
 light2 = analogRead(A1);
 if(344<=light1 && 540>=light1)
 {
  if(344<=light2 && 540>=light2)
  {
    digitalWrite(8,HIGH);
    servo_8.write(0);
  }
  else if(540<light2 && 650>=light2)
  {
    digitalWrite(8,HIGH);
    servo_8.write(30);
  }
  else if(650<light2 && 780>=light2)
  {
    digitalWrite(8,HIGH);
    servo_8.write(60);
  }
  else if(780<light2 && 1017>=light2)
  {
    digitalWrite(8,HIGH);
    servo_8.write(90);
  }
 }
 else if(540<light1 && 650>=light1)
 {
  if(344<=light2 && 540>=light2)
  {
    digitalWrite(8,HIGH);
    servo_8.write(30);
```

```
    }
    else if(540<light2 && 650>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(30);
    }
    else if(650<light2 && 780>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(60);
    }
    else if(780<light2 && 1017>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(90);
    }
   }
   else if(650<light1 && 780>=light1)
   {
    if(344<=light2 && 540>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(60);
    }
    else if(540<light2 && 650>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(60);
    }
    else if(650<light2 && 780>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(60);
    }
    else if(780<light2 && 1017>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(90);
    }
   }
   else if(780<light1 && 1017>=light1)
   {
    if(344<=light2 && 540>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(0);
    }
    else if(540<light2 && 650>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(30);
    }
    else if(650<light2 && 780>=light2)
    {
```

```
      digitalWrite(8,HIGH);
      servo_8.write(60);
    }
    else if(780<light2 && 1017>=light2)
    {
      digitalWrite(8,HIGH);
      servo_8.write(90);
    }
   }
   voice=BT.readString();
   if(voice == "Stop")
   {
    voice="";
    loop();
   }
   else
   {
    voice="";
    automatic();
   }
   }

  void loop() {
    while (BT.available()){  //Check if there is an available byte to read
    delay(10); //Delay added to make thing stable
    char c = BT.read(); //Conduct a serial read
    if (c == 'X') {break;} //Exit the loop when the X is detected after the word
    voice += c; //Shorthand for voice = voice + c
    }
    if (voice.length() > 0) {
      Serial.println(voice);
      if (BT.available())
      {
        voice=BT.readString();

     if(voice == "manual")
    {
     digitalWrite(8,LOW);
    }
    else if(voice == "open")
    {
     digitalWrite(8, HIGH);
     servo_8.write(90);
    }
    else if(voice == "close")
    {
     digitalWrite(8,HIGH);
     servo_8.write(0);
    }
    else if(voice == "automatic")
    {
    automatic();
    }
  voice="";
```

Last three line (163,164,165) contain 3 }.

SUMMARY-

This report will discuss the operation and working of the light-controlled curtains in detail with illustrations, diagrams and circuits of the project.

CONTENTS-

INTRODUCTION-

The curtains will have 2 modes of operation that are light intensity controlled and voice command mode. Arduino board is used in the main part of the circuit. In voice control mode, a Bluetooth module is used to record voice commands and send them to the servo motors to roll the curtains.

Details about parts used-

The project contains the following parts-

1. Bluetooth Module HC05
2. Jumper Wires
3. Breadboard
4. Arduino uno R3
5. 2 resistors - 1 k(ohm)
6. 2 photoresistor
7. Servo Motors
8. Curtains

9. Strings
10. Rods

Bluetooth Module HC05-

WORKING OF A BLUETOOTH MODULE

Bluetooth devices use radio waves to connect mobile phones and computers. Bluetooth products include a small Bluetooth module and Bluetooth radio and software that support connection. When two Bluetooth devices want to communicate with each other, they need to be paired. The communication between Bluetooth devices takes place in a short-range temporary network (called a piconet, which refers to a network connected by devices using Bluetooth technology). This kind of network can accommodate two to eight devices to connect. When the network environment is successfully created, one device acts as the master device, and all other devices act as slave device.

Bluetooth uses radio waves instead of wires or cables to transmit information between electronic devices over short-distances.

When Bluetooth-enabled devices are close enough, they can connect with each other through a tiny computer chip inside them that emits the special Bluetooth radio waves. But first, you have to turn on this chip, which you can usually do by pressing a specific button or flipping a marked switch. Then, the communication between the two Bluetooth devices happens over a short-range network called a piconet (pico means really really small in the metric system). This piconet is essentially a network of Bluetooth connected devices.

BLUETOOTH HC 05 MODULE USED IN OUR CIRCUIT

The HC-05 module can be operated within 4-6V of power supply. It supports baud rate of 9600, 19200, 38400, 57600, etc. Most importantly it can be operated in Master-Slave mode which means it will neither send or receive data from external sources.

DIFFERENT PINS OF BLUETOOTH HC 05

- Enable- This pin is used to set the Data Mode or and AT command mode (set high). {This feature is not required here so this pin is not used}
- VCC- This is connected to +5V power supply.
- Ground- Connected to the ground of powering system
- Tx (Transmitter)- This pin transmits the received data serially.
- Rx (Receiver)- Used for broadcasting data serially over Bluetooth.
- State- Used to check if the Bluetooth is working properly

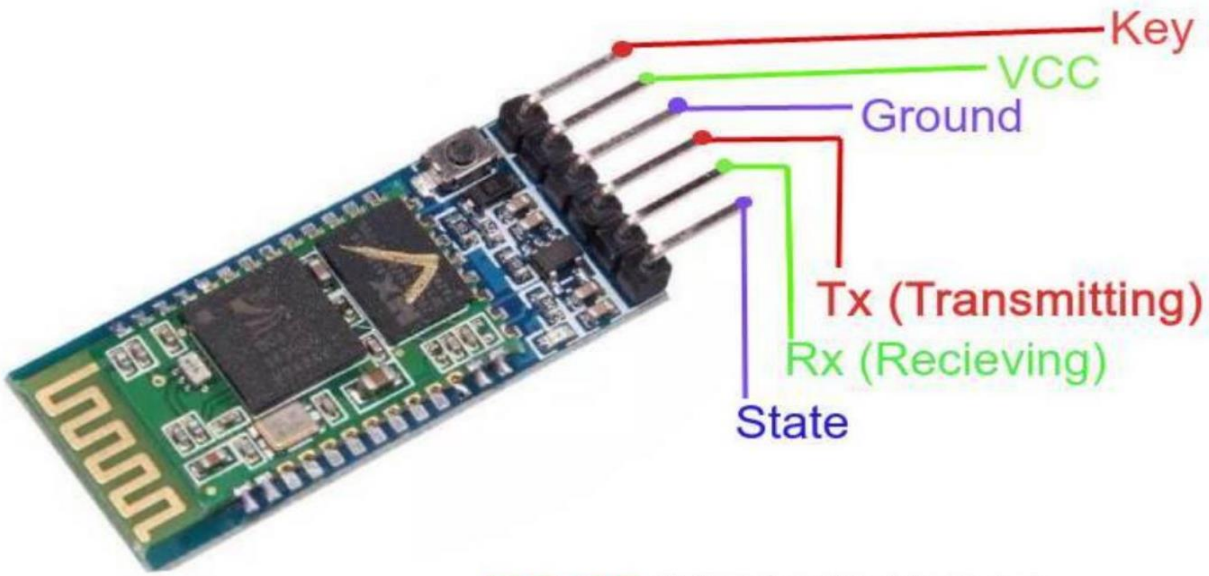
DIFFERENT MODES OF HC 05 MODULE

Command Mode-

In Command Mode, you can communicate with the Bluetooth module through AT Commands for configuring various settings and parameters of the Module like get the firmware information, changing Baud Rate, changing module name, it can be used to set it as master or slave. A point about HC-05 Module is that it can be configured as Master or Slave in a communication pair. In order to select either of the modes, you need to activate the Command Mode and sent appropriate AT Commands.

Data Mode-

Coming to the Data Mode, in this mode, the module is used for communicating with other Bluetooth device i.e. data transfer happens in this mode.



HC-05 (Bluetooth Module)

Breadboard-

The breadboard is a construction base for prototyping of circuits, to try the circuit logic first and check for any errors.

The breadboard consists of two types of strips-

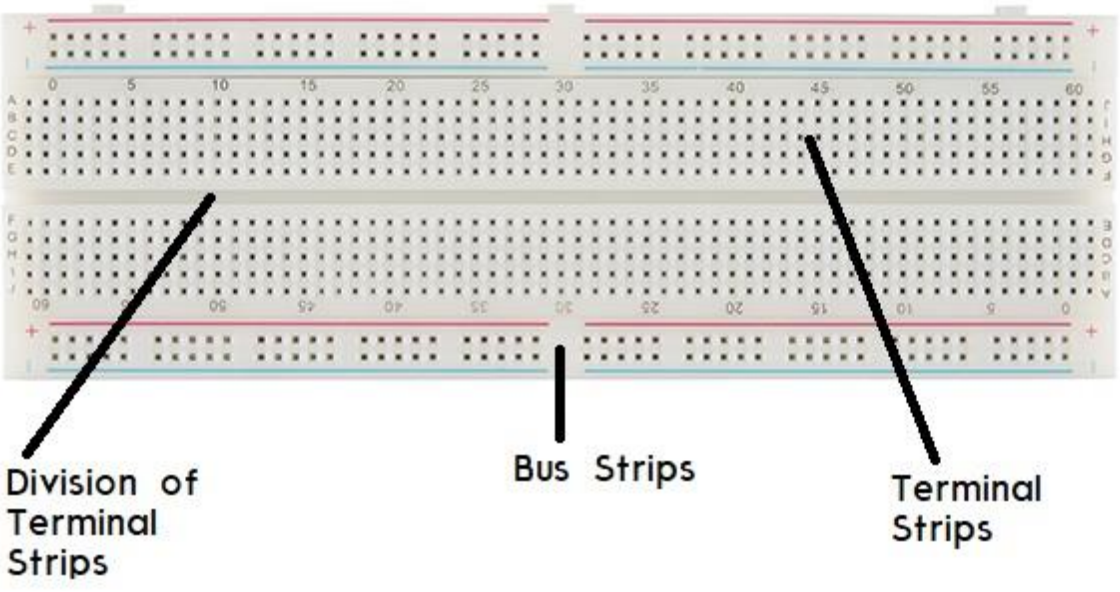
1. Bus Strips
2. Terminal Strips

BUS STRIPS-

Used to power electronic components. It usually contains 2 columns, one for ground and the other for supply voltage. Generally, the dots of a single column of a strip are all interconnected to draw power easily to different parts of the board.

TERMINAL STRIPS-

These are the main strips where all the circuit elements are connected. The strips are divided into 2 parts. Each part has several columns and all the rows are interconnected.



Arduino Uno R3

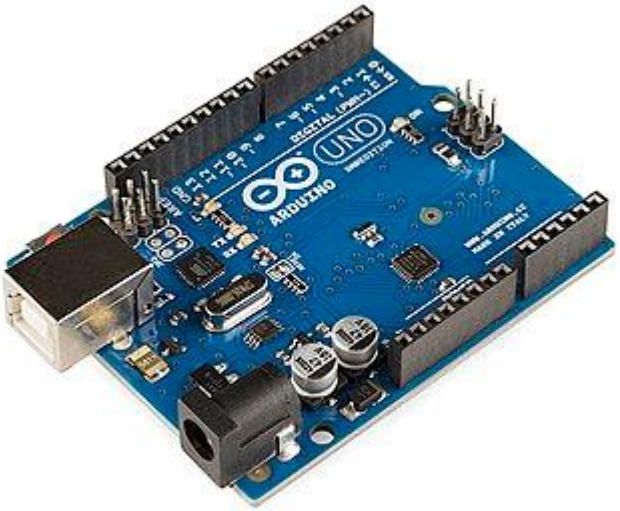
The **Arduino Uno** is an **open-source microcontroller board** based on the **Microchip ATmega328P** microcontroller and developed by **Arduino.cc**. The board is equipped with sets of digital and

analog input/output (I/O) pins that receive and transfer information using written code in the Arduino IDE.

The board has 14 Digital pins, 6 Analog pins and is programmable with Arduino IDE. It can be powered by USB cable or external 9V battery.

DIGITAL PINS-

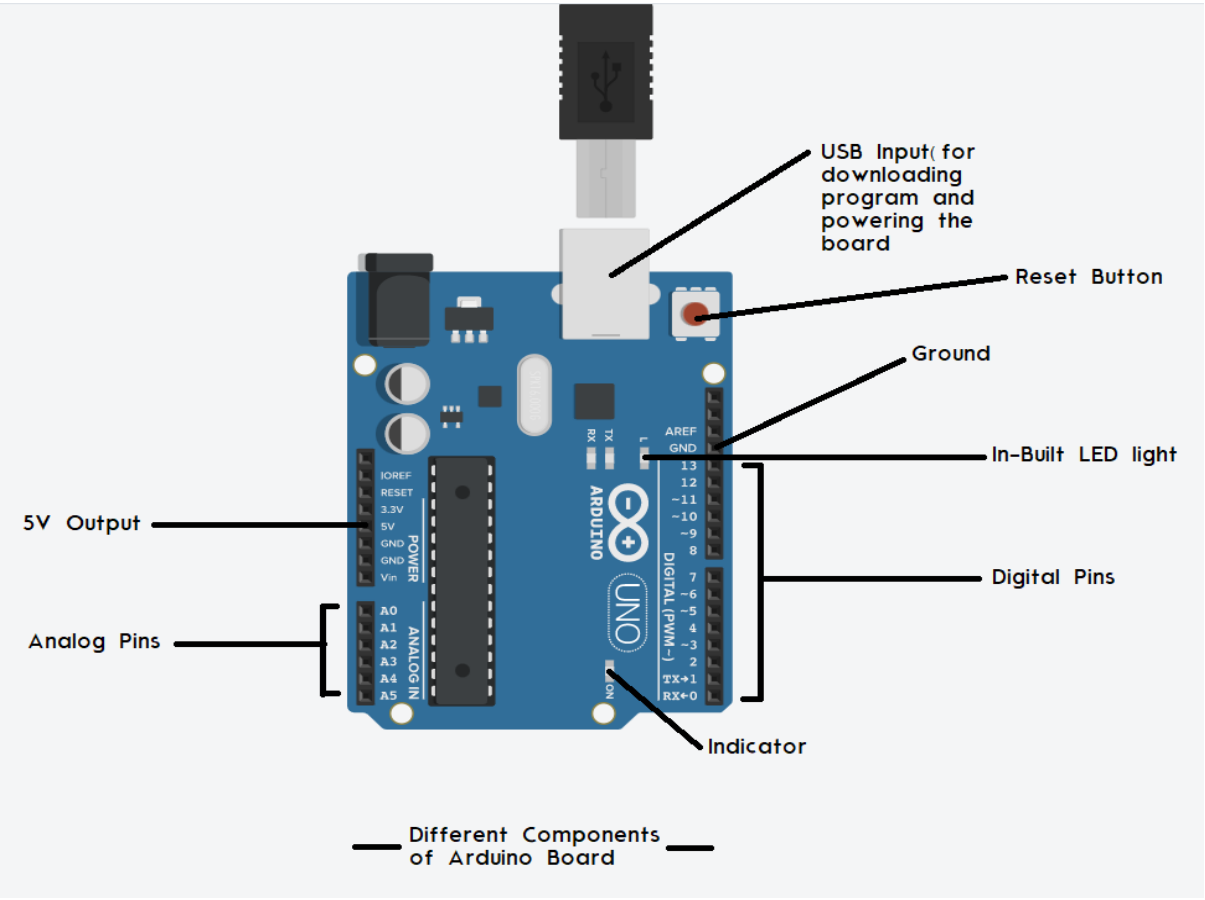
These pins are used to power the circuit elements connected to the Arduino board by changing the voltage provided by the pins from 0-5V. These pins are only used to transfer information (in form of voltage).



Arduino Board

ANALOG PINS-

These pins are use to record and write (in discrete steps) information.



Photoresistor-

WHAT IS A PHOTORESISTOR

LDR (Light Dependent Resistor) as the name states it is a special type of resistor that works on the photoconductivity principle means

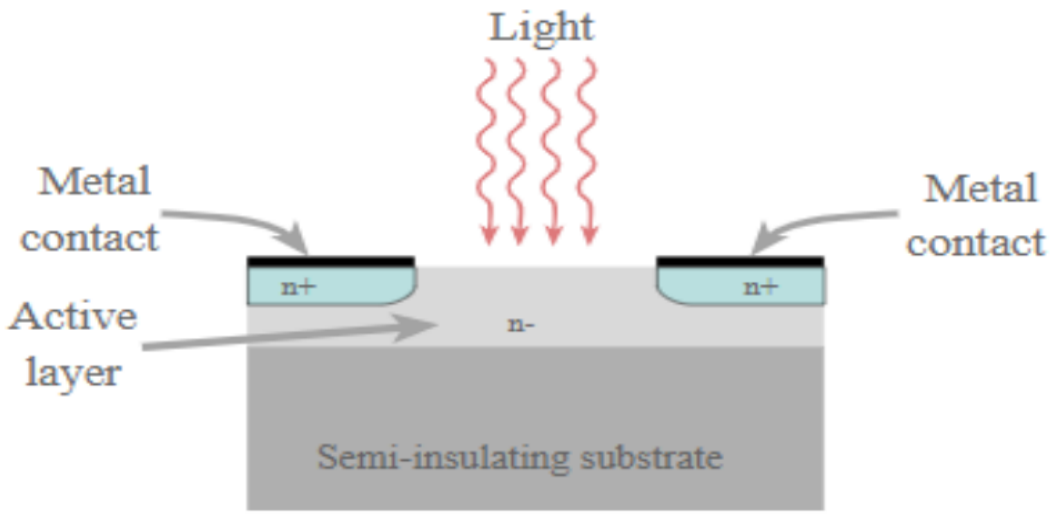
that resistance changes according to the intensity of light. Its resistance decreases with an increase in the intensity of light. It is often used as light sensor, light meter, Automatic Street lights and in area where we need to have light sensitivity. Here we have utilized for new application: automatic curtains.

PRINCIPLE

It works on the principle of photoconductivity whenever the light on its photoconductive material falls it absorbs its energy and the electrons of that photoconductive material that is in the valence band get excited and go to the conduction band and thus increases the conductivity as per the increase in light intensity. Also, the energy in incident light should be greater than the bandgap gap energy so that the electrons from the valence band got excited and go to conduction band. The LDR has the highest resistance in dark around 1012 Ohm and this resistance decreases with the increase in Light.

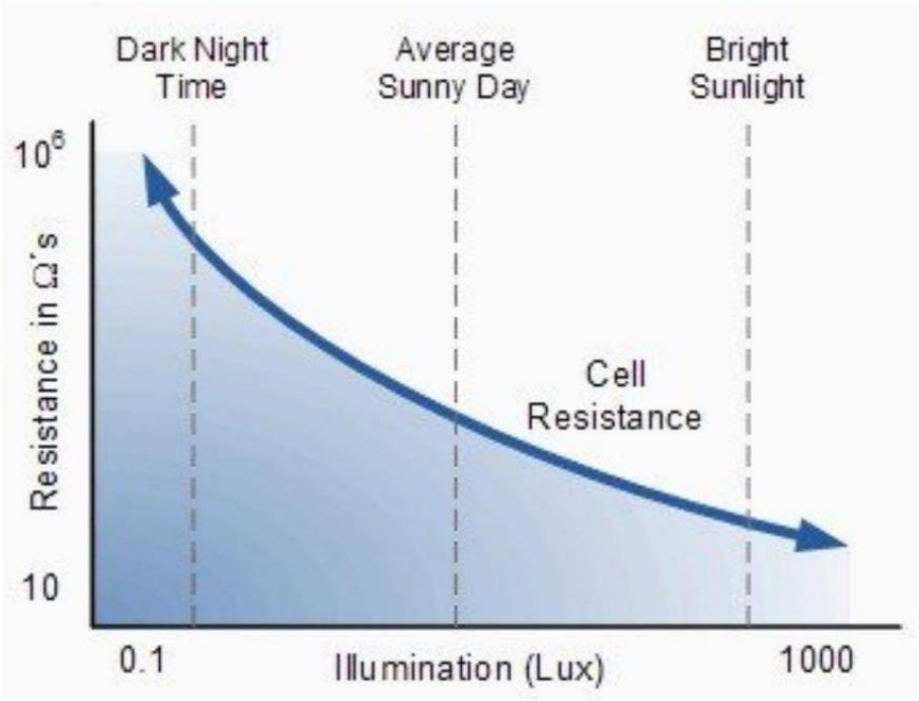
STRUCTURE

Structurally the photoresistor is a light sensitive resistor that has a horizontal body that is exposed to light. The basic format for a photoresistor is that shown below:



Photoresistor structure

GRAPH



We searched for data on internet and various articles on intensity needed for photoresistor, when we finally came across this data based on which we have developed our project.

Position	Sunrise	Upward	Sunset
Maximum (Lux)	980 Lux at 2.00pm	970 Lux at 11.00am	950 Lux at 12.00pm
Minimum (Lux)	700 Lux at 5.00pm	350 Lux at 5.00pm	830 Lux at 5.00pm

Maximum and minimum light intensity.

https://www.researchgate.net/figure/The-resistance-VS-illumination-Graph-of-LDR_fig3_323359222 (Reference for Information)

This data gave us an idea on what intensities should we specify in our code based on which we decided how much we want our servo motor to be rotated.

We used two photoresistors to account for the changing positions of the sun during the day and hence more accuracy.

CHALLENGE-

Now using two photoresistors made the rotation of servo motor dependent on two intensities as measured by the two photoresistors. They could be different most of the time as the light falling on the photoresistors depend on the position and distance of light source too. Directly specifying the intensity would lead to failure of our project as that would create confusion between do different measured values.

For e.g.: In case of a circular light source intensity is inversely proportional to square of the distance.

$$I \propto 1/d2$$

CALCULATIONS-

To solve this challenge, our knowledge of permutations and combinations came handy.

Intensity measured by Photoresistor 1= I1

Intensity measured by Photoresistor 2=I2

We specified four levels at which our curtain would be open so four intensity ranges were decided.

Hence, we had **four values of each I1 and I2.**

So total combinations that would occur at any given day=4*4=16

The Table of intensities vs servo motor rotation angle (in degrees) is as follows

S.no	I1	I2	Degrees by which rotation would occur
1	344-540	344-540	0
2	344-540	540-650	30
3	344-540	650-780	60
4	344-540	780-1017	90
5	540-650	344-540	30
6	540-650	540-650	30

7	540-650	650-780	60
8	540-650	780-1017	90
9	650-780	344-540	60
10	650-780	540-650	60
11	650-780	650-780	60
12	650-780	780-1017	90
13	780-1017	344-540	90
14	780-1017	540-650	90
15	780-1017	650-780	90
16	780-1017	780-1017	90

Information about lux:

A light dependent resistor (LDR) measures the intensity of light which is denoted by Lux

Illuminance	Example
0.002 lux	Moonless clear night sky
0.2 lux	Design minimum for emergency lighting (AS2293).
0.27 - 1 lux	Full moon on a clear night
3.4 lux	Dark limit of civil twilight under a clear sky
50 lux	Family living room
80 lux	Hallway/toilet
100 lux	Very dark overcast day
300 - 500 lux	Sunrise or sunset on a clear day. Well-lit office area.
1,000 lux	Overcast day; typical TV studio lighting
10,000 - 25,000 lux	Full daylight (not direct sun)
32,000 - 130,000 lux	Direct sunlight

What is lux?

The lux (symbol: lx) is the SI derived unit of illuminance, measuring luminous flux per unit area.

In photometry, this is used as a measure of the intensity, as perceived by the human eye, of light that hits or passes through a surface.

The illuminance as a measure of the intensity of illumination on a surface. A given amount of light will illuminate a surface more dimly if it is spread over a larger area, so illuminance is inversely proportional to area when the luminous flux is held constant.

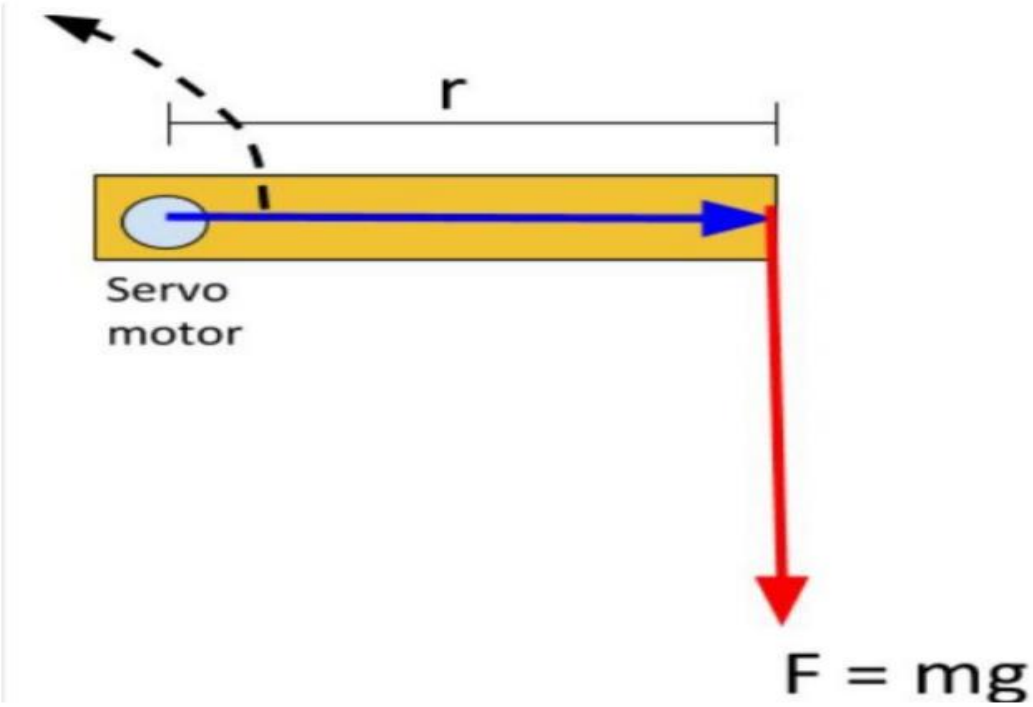
Servo Motors

Model Used: SG90 Micro Servo Motor

SPECIFICATIONS-

- Operating Voltage: 4.8-6V
- Operating Speed: 0.12sec/60 degrees
- Output Torque: 1.6kg/cm at 4.8V
- Weight: 9g

OUTPUT TORQUE



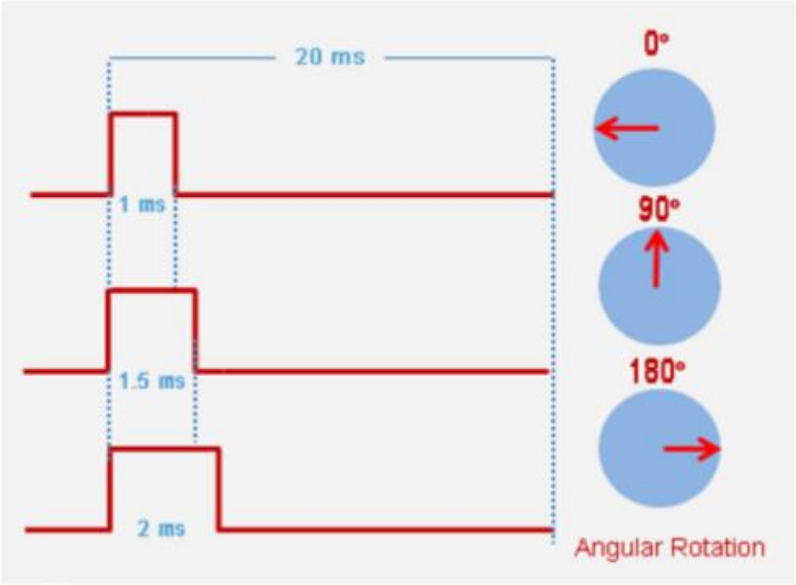
TORQUE=Force*Radial Distance of Force from Rotation axis

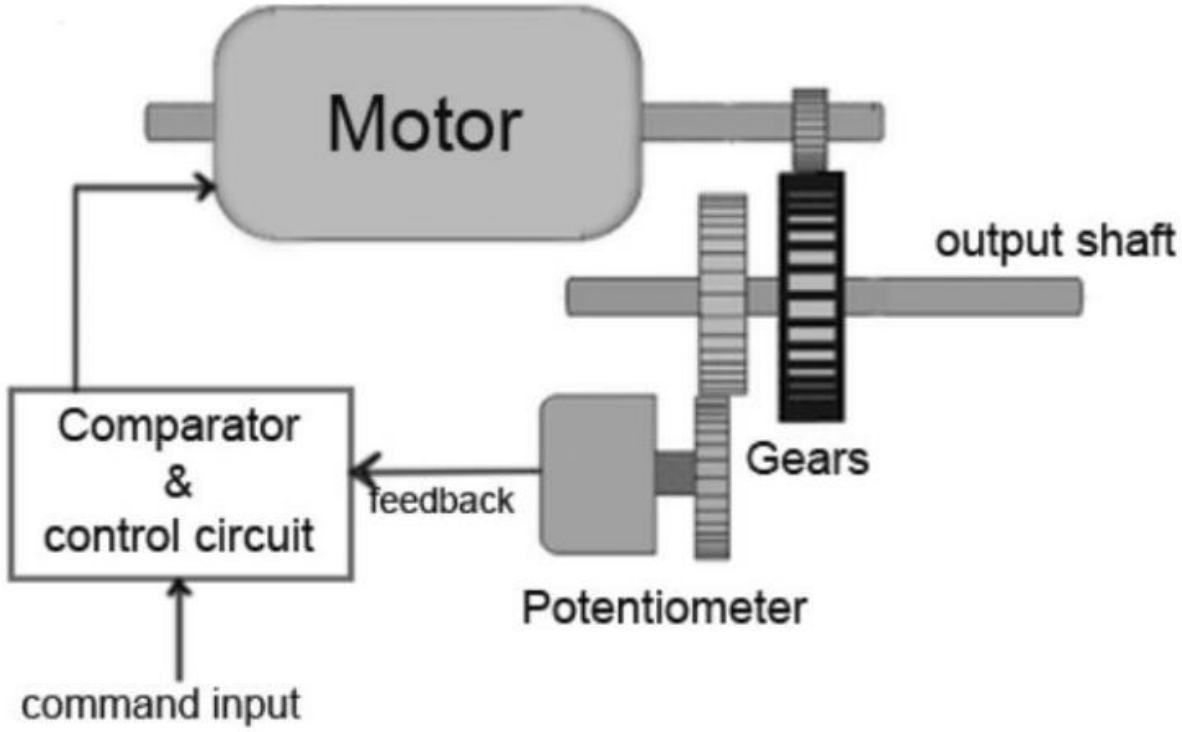
SIGNIFICANCE OF OUTPUT TORQUE-

If the specification of servo motor is x kg/cm, the servo motor will stop rotating when it is trying to move a x kg weight at a radial distance of 1.0 cm.

WORKING AND PRINCIPLE:

Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears.** High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.





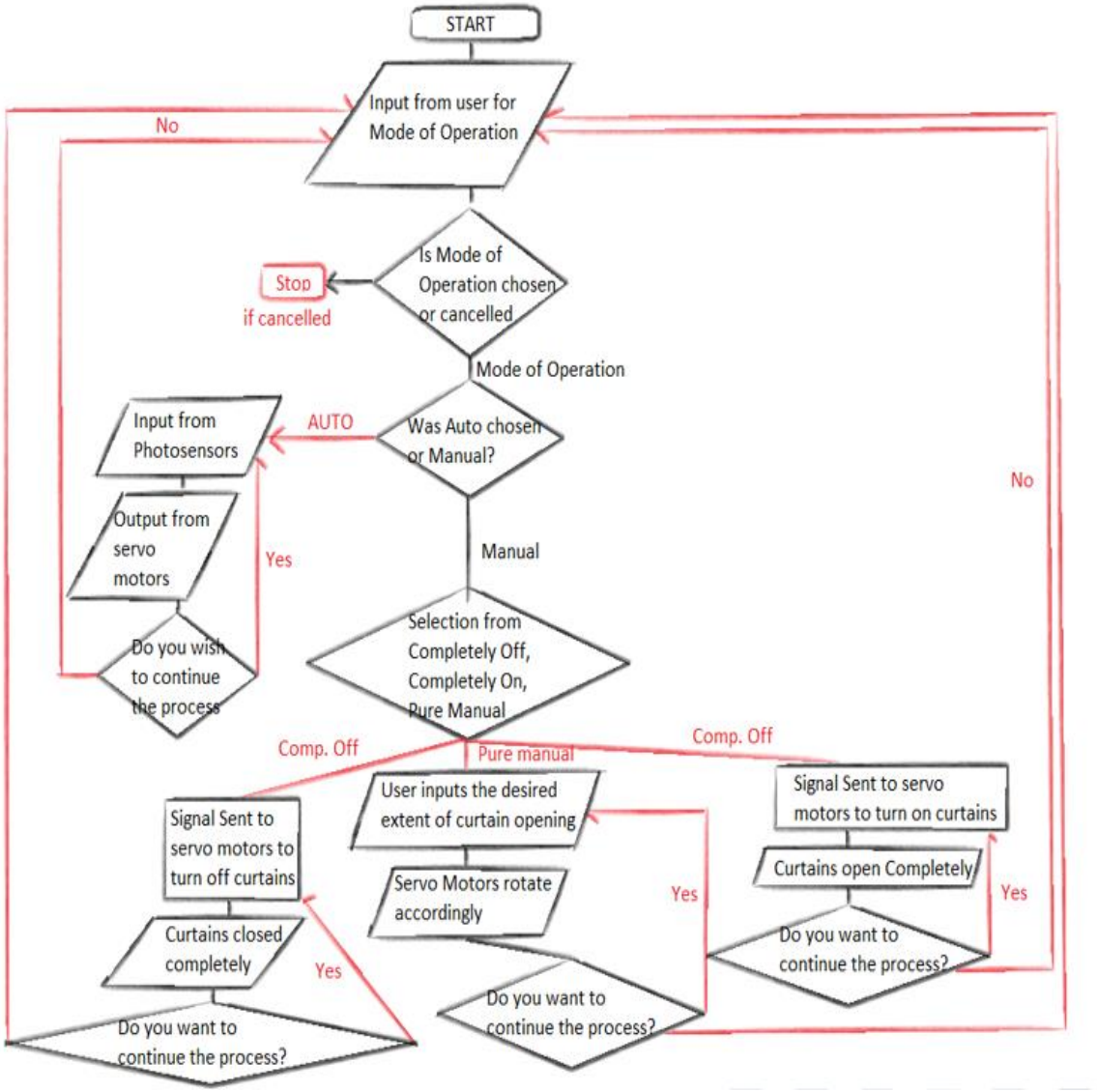
Servo motors are used to control position and speed very precisely, but in a simple case, only position may be controlled. Mechanical position of the shaft can be sensed by using a potentiometer, which is coupled with the motor shaft through gears. The current position of the shaft is converted into electrical signal by the potentiometer, and the compared with the command input signal. In modern servo motors, electronic encoders or sensors are used to sense the position of the shaft.

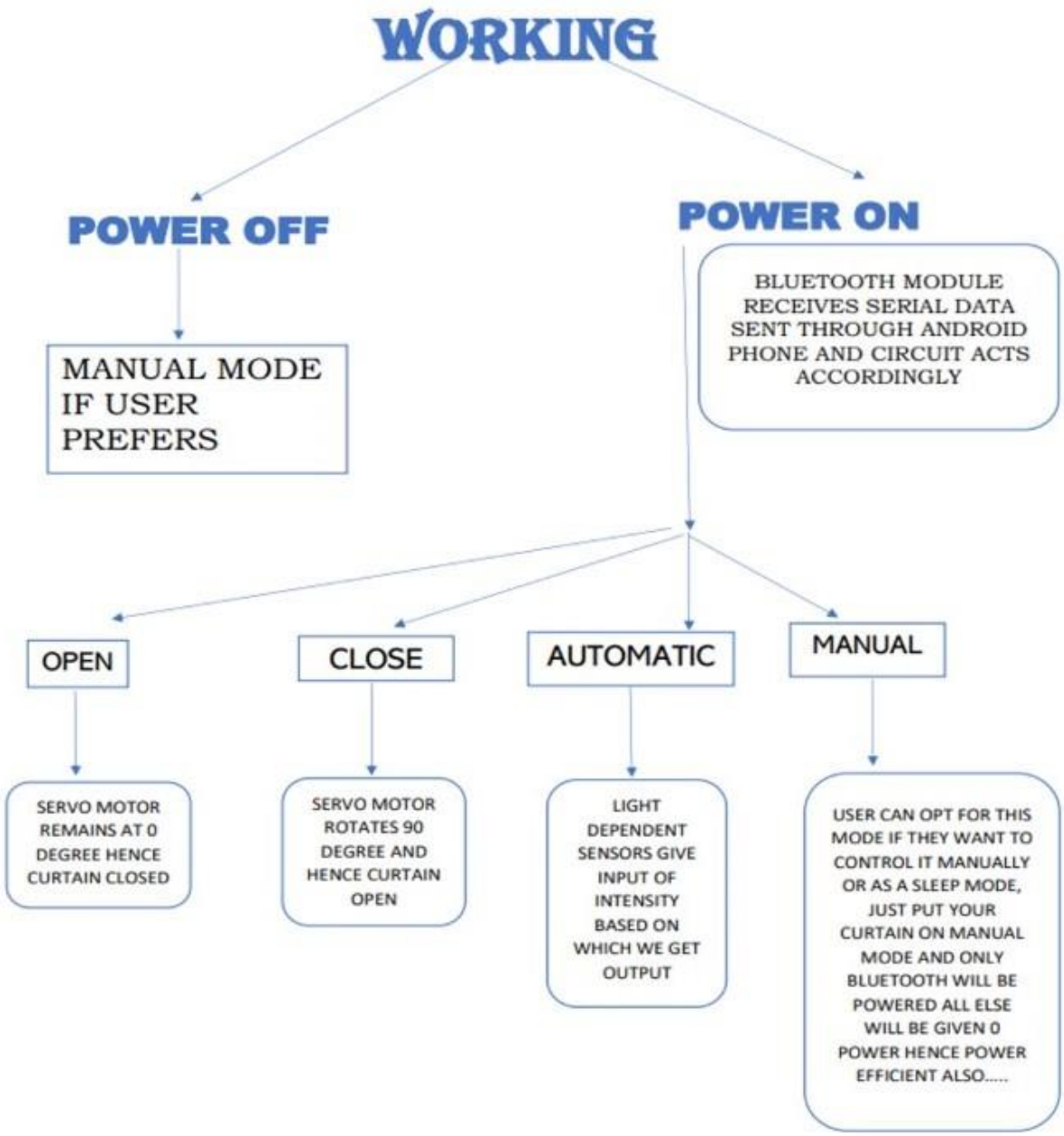
Command input is given according to the required position of the shaft. If the feedback signal differs from the given input, an error signal is generated. This error signal is then amplified and applied as

the input to the motor, which causes the motor to rotate. And when the shaft reaches to the required position, error signal becomes zero, and hence the motor stays standstill holding the position.

The command input is given in the form of electrical pulses. As the actual input applied to the motor is the difference between feedback signal (current position) and applied signal (required position), speed of the motor is proportional to the difference between the current position and the required position. The amount of power required by the motor is proportional to the distance it needs to travel.

WORKING




WORKING
POWER OFF
MANUAL MODE IF USER PREFERS
POWER ON
BLUETOOTH MODULE RECEIVES SERIAL DATA SENT THROUGH ANDROID PHONE AND CIRCUIT ACTS ACCORDINGLY
OPEN
CLOSE
AUTOMATIC
MANUAL
SERVO MOTOR REMAINS AT 0 DEGREE HENCE CURTAIN CLOSED
SERVO MOTOR ROTATES 90 DEGREE AND HENCE CURTAIN OPEN
LIGHT DEPENDENT SENSORS GIVE INPUT OF INTENSITY BASED ON WHICH WE GET OUTPUT
USER CAN OPT FOR THIS MODE IF THEY WANT TO CONTROL IT MANUALLY OR AS A SLEEP MODE, JUST PUT YOUR CURTAIN ON MANUAL MODE AND ONLY BLUETOOTH WILL BE POWERED ALL ELSE WILL BE GIVEN 0 POWER HENCE POWER EFFICIENT ALSO.....

SOURCES AND REFRENCES:

Circuit Diagram:

- from TinkerCAD

Bluetooth Module:

- https://www.geeksforgeeks.org/all-about-hc-05-bluetooth-module-connection-with-android/
- https://www.gme.cz/data/attachments/dsh.772-148.1.pdf
- https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-
- https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/

Breadboard:

- Wikipedia

Arduino Uno:

- Wikipedia

Photoresistor:

- https://www.electronicsforu.com/technology-trends/learn-electronics/ldr-light-dependent-resistors-basics
- Wikipedia

Servo Motor:

- https://circuitdigest.com/article/servo-motor-working-and-basics
- https://automaticaddison.com/how-to-determine-what-torque-you-need-for-your-servo-motors/