

```

#include <Servo.h>
Servo myservo1;
Servo myservo2;
int sensorPin = A0; // select the input pin for LDR
int sensorValue = 0; // variable to store the value coming from the sensor
int pos = 1;
int barState;
int timeDelay = 5000;
int displaceAngle = 35;
int barDelay = 40;
int checkDelay = 1000;
int lightOnCount = 0;
int lightOnMax = 5;

void setup() {
  Serial.begin(9600); //sets serial port for communication
  myservo1.attach(9);
  myservo2.attach(10);
  barState = 1;
  myservo1.write(0);
  myservo2.write(0);
  up();
}

void upMove()
{
  delay(timeDelay);
  for (pos = 0; pos <= displaceAngle; pos += 1) { // goes from 0 degrees to displaceAngle degrees
    // in steps of 1 degree
    myservo1.write(pos);
    myservo2.write(pos); // tell servo to go to position in variable 'pos'
    delay(barDelay); // waits 30ms for the servo to reach the position
    barState = 0;
  }
}

void downMove()
{
  for (pos = displaceAngle; pos >= 0; pos -= 1) { // goes from 0 degrees to displaceAngle degrees
    // in steps of 1 degree
    myservo1.write(pos);
    myservo2.write(pos); // tell servo to go to position in variable 'pos'
    delay(barDelay); // waits 30ms for the servo to reach the position
    barState = 1;
  }
}
}

```

```

void up()

{
  {
    if (barState == 1){ Serial.print("Count = "); Serial.println(lightOnCount);
                        Serial.print("Max = "); Serial.println(lightOnMax);
                        upMove();
    }
  }
}

void down()
{
  {
    if (barState == 0) downMove();
  }
}

void loop() {
  { delay(checkDelay);
    sensorValue = analogRead(sensorPin);          //define photocellReading as pin 0 input from
LDR
    sensorValue = map(sensorValue, 0, 1023, 0, 179); //map the LDR input to a value between 1-
180 so the servo can understand it
    if (sensorValue <= 90) //if the LDR is showing less than half light intensity
    {
      up();
    }
    else if (sensorValue > 90) //if the LDR is showing more than half light intensity
    {
      down(); //then tell the servo to rotate forwards at a steady rate
    }
  }
}
}

```