REMOTE CONTROL LED RGB LAMP WITH AUTO BRIGHTNESS ADJUSTER

A project report submitted in partial fulfillment of the requirements for the award of the degree of

B.TECH in

Electronics And Communication Engineering

By

ABHAY TIWARI (2021UEC2608)

DHRUV SAINI (2021UEC2626)

DIVYE GANDHI(2021UEC2638)

Under the supervision of

D.V GADRE

ELECTRONICS AND COMMUNICATION ENGINEERING

Netaji Subhas University of Technology, Delhi

DIVISION OF ELECTRONICS AND COMMUNICATION ENGINEERING
NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

CERTIFICATE

This is to certify that the project titled **REMOTE CONTROL LED RGB LAMP WITH AUTO BRIGHTNESS ADJUSTER** is a bonafide record of the work done by

ABHAY TIWARI(2021UEC2608)

DHRUV SAINI (2021UEC2626)

DIVYE GANDHI(2021UEC2638)

under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering** in **ELECTRONICS AND COMMUNICATION ENGINEERING** of the **Netaji Subhas Institute of Technology, DELHI-110078**, during the year 2023-2024.

Their work is genuine and has not been submitted for the award of any other degree to the best of my knowledge.

DATE:

PROFESSOR NAME : D.V GADRE

Division of Electronics And Communication Engineering

Netaji Subhas Institute of Technology

DECLARATION

This is to certify that the work which is being hereby presented by us in this project titled "**REMOTE CONTROL LED RGB LAMP WITH AUTO BRIGHTNESS ADJUSTER**" in partial fulfilment of the award of the Bachelor of Engineering submitted at the Department of Electronics And Commucation Engineering, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, is a genuine account of our work carried out during the period from January 2023 to April 2023 under the guidance of D.V.GADRE , Department of Electronics And Communication Engineering, Netaji Subhas Institute of Technology, New Delhi.

The matter embodied in the project report to the best of our knowledge has not been submitted for the award of any other degree elsewhere.

DATE:

Abhay Tiwari	2021UEC2608
Dhruv Saini	2021UEC2608
Divye Gandhi	2021UEC2638

Index

Introduction

Throughout the ECE Lab portion, we learned a variety of electronic ideas, and we practically all applied them to our EdW Lab project. Additionally, after learning a lot from this course, we wanted to create a project that would push our academic and intellectual limits. This paper includes a thorough overview of our invention, the "Smart LED Lamp," a lamp that can be controlled remotely to change colour.

DESCRIPTION

We aim to create a 'REMOTE CONTROL LED RGB LAMP WITH AUTO BRIGHTNESS ADJUSTER' lamp that can be controlled by a TV remote. The lamp will have the ability to change the intensity of red, blue, and green colors to create the desired color output. It will also be capable of generating different patterns that can be changed using the TV remote. Additionally, it will include an auto brightness feature that can adjust the lamp's intensity based on the room's brightness. Finally, the lamp will have a memory feature that allows it to remember the last color stored in the lamp, in the event of a power outage. According to electronic theory, the project will include all six blocks of electronic projects, as stated below:

1)Input Devices

2)Output Devices

3)Signal Processing

4)Data Visualization And storage

5)Communication with External World

6)Power Supply

Motivation

During an interactive discussion with our professor D.V. Gadre, we decided to undertake this project. The project is aimed at elderly people who may not have extensive knowledge of technology. To make it accessible, we decided to use a TV remote control as it is simpler than a mobile phone or other technologies. This Smart RGB lamp can also be used as a home accessory, which is easy to use and interesting.

Components List

1) Arduino Nano

Arduino Nano is a compact and user-friendly microcontroller board that is ideal for a wide range of electronic projects. It is based on the ATmega328P microcontroller and offers 14 digital input/output pins, 8 analog inputs, and 6 PWM pins. Additionally, it has a built-in USB interface for programming and power supply purposes.

The small form factor of the Arduino Nano makes it easy to integrate into a wide range of electronic devices, including small robots, home automation systems, and wearable electronics. It is also an excellent choice for prototyping and experimenting with new electronic projects.

Figure 1: Arduino Nano board

2) TSOP1838 IR Receiver

The TSOP1838 is a miniature IR receiver IC from the TSOP18xx series. This particular model TSOP1838 will respond to 38kHz IR signals from remote control devices. It is normally used as an IR receiver in remote control applications like TV, AC, etc. This version is small in size and hence can be used in compact devices.

The TSOP1838 is capable of receiving infrared signals from a distance of up to 10 meters and is compatible with most infrared remote control protocols.

TSOP1838 Features

- Operating Voltage: 2.5V to 5.5V
- Carrier Frequency (38kHz)
- Operating Current: 5mA
- High Range and Wide Coverage Area
- Improved Immunity Against HF and RF Noise
- Has Built-in Pre-Amplifier
- TTL and CMOS Compatible

Figure 2: TSOP1838 IR receiver IC

3) LDR Sensor

An LDR (Light-Dependent Resistor) is a sensor that changes its resistance based on the amount of light it is exposed to. LDRs are commonly used in light-sensing applications, such as controlling streetlights or detecting when a room is occupied.

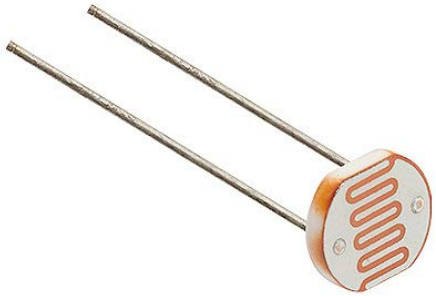

Figure 3: LDR sensor

4) LCD Display

An LCD (Liquid Crystal Display) is a type of flat panel display commonly used in digital watches, calculators, and other electronic devices. They are also commonly used in micro-controller based projects to display information, such as sensor readings or system status.[Dimension of LCD:16X4]

Figure 4: LCD display

4) Neopixel Led [WS2812B]

Neopixel LED is a type of RGB LED that is made up of individual programmable pixels, each of which can display any color by mixing red, green, and blue light. Neopixel LEDs are controlled by a microcontroller or computer and are commonly used in a wide range of electronic projects, including DIY home lighting systems, art installations, and wearable electronics.

Neopixel LEDs were created by Adafruit Industries and are based on the WS2812 LED driver chip. Each Neopixel LED contains a red, green, and blue LED, as well as a controller chip that handles the color mixing and communication with the microcontroller. The controller chip can be programmed to display a wide range of colors and patterns, and multiple Neopixel LEDs can be connected together to create larger displays.

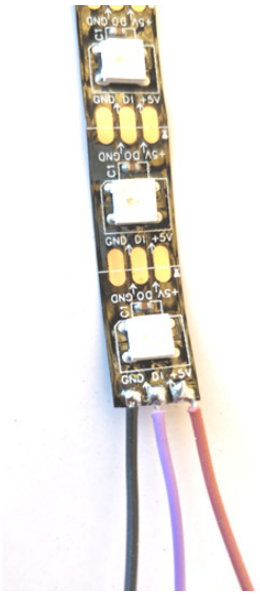

Figure 5: LCD display

Working

Block Diagram

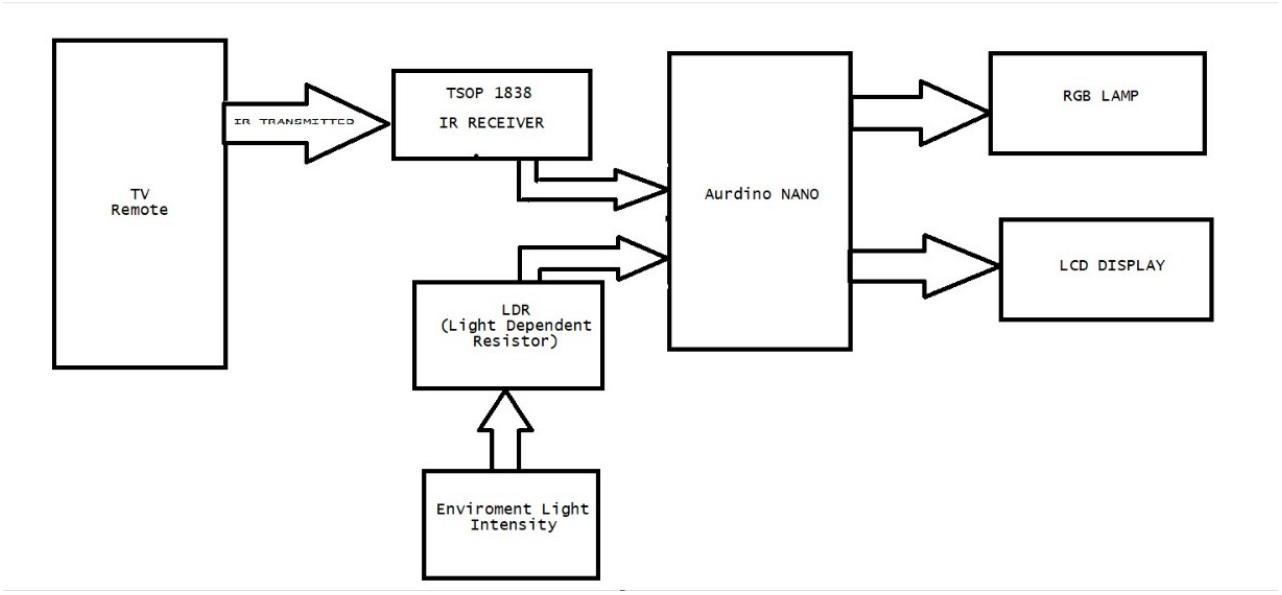

Figure 6: Block Diagram

Designing

We will now move on to the designing phase, starting with drawing the schematic using Eagle CAD software. After that, we will proceed to the PCB (Printed Circuit Board) design layout.

1) Schematic Diagram :

To design the circuit for our SMART RGB LED Lamp, we used EAGLE CAD software. The following image shows the complete circuit diagram for our project.

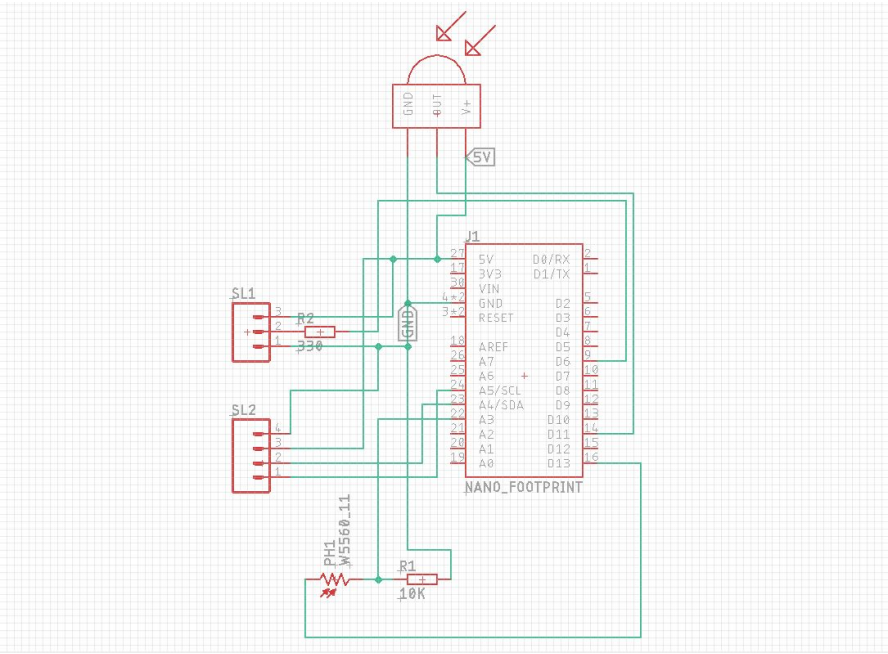

Figure 7: Schematic Diagram

2) Code Attachments :

```
#include <IRremote.h>
#include<EEPROM.h>
```

```
#include <Adafruit_NeoPixel.h>
#ifdef AVR
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif
#include <LCD_I2C.h>
LCD_I2C lcd(0x27, 16, 4);
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
Adafruit_NeoPixel pixels(170, 6, NEO_GRB + NEO_KHZ800);
float R=0,B=0,G=0;
float b=100;
void setup()
{
  pixels.begin();
  pinMode(6,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(A3,INPUT);
  lcd.begin();
  lcd.backlight();
  Serial.begin(9600);
  // In case the interrupt driver crashes on setup, give a clue
  // to the user what's going on.
  Serial.println("Enabling IRin");
  irrecv.enableIRIn(); // Start the receiver
  Serial.println("Enabled IRin");
}
void writeEEPROM(int number , int address){
  byte bytee=number;
```

```
  EEPROM.write(address,bytee);
}
int readEEPROM(int address){
  byte bytee=EEPROM.read(address);
  return bytee;
}
void loop() {
//  pulseWhite(5);
  if (irrecv.decode(&results)) {

    if(results.value==0xA70){
      lcd.clear();
      delay(500);
      lcd.print("SET TO LAST SAVED COLOUR");
      delay(500);
      lcd.clear();
      R=readEEPROM(18404);
      G=readEEPROM(18405);
      B=readEEPROM(18406);
      for(int i=0; i<170; i++) {
        pixels.setPixelColor(i, pixels.Color(R, G, B));
      }

    }
    if(results.value==0xDD0){
      lcd.clear();
      delay(500);
      lcd.print("SAVING COLOUR...");
      delay(500);
```

```
  lcd.clear();
  writeEEPROM(R,18404);
  writeEEPROM(G,18405);
  writeEEPROM(B,18406);
}
else if(results.value==0x6CE9){

}
else if (results.value==0x2CE9){
  colorWipe(pixels.Color(255,   0,   0), 50);
}
else if (results.value==0x1CE9){
  whiteOverRainbow(75, 5);
}
else if(results.value==0xCD05 ){
   if(b+10<=100)
   {b=b+10;}
   else{
    b=100;
   }
   pixels.setBrightness(b);
}
else if(results.value==0xCD0){
   if(b+10<=100)
   {b=b+10;}
   else{
    b=100;
   }
   pixels.setBrightness(b);
```

```
}
else if(results.value==0x2D0){
   if(b-10>0)
   {b=b-10;}
   pixels.setBrightness(b);
}
else if(results.value==0x10)
{
  for(int i=0; i<170; i++) {
    R=90;
    G=0;
    B=0;
    pixels.setPixelColor(i, pixels.Color(R, G, B));

  }
  lcd.clear();
  lcd.print("RED");
  delay(500);
  lcd.clear();
}
else if(results.value==0x810){
  for(int i=0; i<170; i++) {
    R=0;
    G=90;
    B=0;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("GREEN");
```

```
  delay(500);
  lcd.clear();
}
else if(results.value==0x410){
  for(int i=0; i<170; i++) {
    R=0;
    G=0;
    B=90;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("BLUE");
  delay(500);
  lcd.clear();
}
else if(results.value==0xC10){
  for(int i=0; i<170; i++) {
    R=0;
    G=90;
    B=90;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("AQUA");
  delay(500);
  lcd.clear();
}
else if(results.value==0x210){
  for(int i=0; i<170; i++) {
```

```
    R=90;
    G=90;
    B=0;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("YELLOW");
  delay(500);
  lcd.clear();
}
else if(results.value==0xA10){
  for(int i=0; i<170; i++) {
    R=90;
    G=0;
    B=90;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("PINK");
  delay(500);
  lcd.clear();
}
else if(results.value==0x610){
  for(int i=0; i<170; i++) {
    R=60;
    G=60;
    B=60;
    R=R-7;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
```

```
  }
  lcd.clear();
  lcd.print("WHITE");
  delay(500);
  lcd.clear();
}
else if(results.value==0xE10){
  for(int i=0; i<170; i++) {
    R=0;
    G=90;
    B=70;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("TERQUOISE");
  delay(500);
  lcd.clear();
}
else if(results.value==0x110){
  for(int i=0; i<170; i++) {
    R=90;
    G=10;
    B=0;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("ORANGE");
  delay(500);
  lcd.clear();
```

```
}
else if(results.value==0xA90){
  for(int i=0; i<170 ; i++) {
  lcd.clear();
    R=0;
    G=0;
    B=0;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("LED OFF");
  delay(500);
  lcd.clear();
}
else if (results.value==0x490){
  if(R+15<=90)
  {R=R+15;}
  else{
    R=90;
  }
  for(int i=0; i<170; i++) {
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
}
else if (results.value==0x90){
  if(G+15<=90)
  {G=G+15;}
  else
  {
```

```
    G=90;
  }
  for(int i=0; i<170; i++) {
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }

}
else if (results.value==0x2F0){
  if(B+15<=90){
   B=B+15;
  }
  else{
    B=90;
  }
  for(int i=0; i<170; i++) {
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
}
else if(results.value==0x910){
  for(int i=0; i<170; i++) {
    R=80;
    G=80;
    B=60;
    R=R-7;
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
  lcd.clear();
  lcd.print("WARM WHITE");
  delay(500);
```

```
  lcd.clear();
}
else if (results.value==0xC90){
  if(R-15>=0)
  {R=R-15;}
  else{
    R=0;
  }
  for(int i=0; i<170; i++) {
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
}
else if (results.value==0x890){
  if(G-15>=0)
  {G=G-15;}
  else{
    G=0;
  }
  for(int i=0; i<170; i++) {
    pixels.setPixelColor(i, pixels.Color(R, G, B));
  }
}
else if (results.value==0xAF0){
  if(B-15>=0)
  {B=B-15;}
  else{
    B=0;
  }
  for(int i=0; i<170; i++) {
```

```
        pixels.setPixelColor(i, pixels.Color(R, G, B));
      }

    }
    else if (results.value==0x290){
      digitalWrite(13,HIGH);
      float light=analogRead(A3);
      lcd.clear();
      lcd.print("ADJUST TO ROOM BRIGHTNESS =");
      delay(900);
      float vall=(1-(light/977))*100;
      lcd.print(vall);
      delay(900);
      lcd.clear();

      lcd.print(vall);
      b=vall;
      pixels.setBrightness(vall);
      for(int i=0; i<170; i++) {
        pixels.setPixelColor(i, pixels.Color(R, G, B));
//        pixels.show();
      }
    }
    pixels.show();
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
    lcd.setCursor(0, 1);
//    lcd.print(results.value, HEX);
```

```
//    delay(900);
//    lcd.clear();
   lcd.clear();
      lcd.print("RED=");
//      lcd.setCursor(0, 1); // Or setting the cursor in the desired
      float val = (R/90.0)*100;
    lcd.print(val);
    lcd.print("%");
//      delay(250);
      lcd.setCursor(0, 1);
//    lcd.clear();
      lcd.print("GREEN=");
      val = (G/90.0)*100;
    lcd.print(val);
    lcd.print("%");
//      delay(250);
      lcd.setCursor(0, 2);
      lcd.print("BLUE=");
      val = (B/90.0)*100;
    lcd.print(val);
    lcd.print("%");
    lcd.setCursor(0, 3);
      lcd.print("BRIGHT.=");
    lcd.print(b);
    lcd.print("%");

//      delay(250);
  }
  delay(90);
```

```
}
void colorWipe(uint32_t color, int wait) {
  for(int i=0; i<pixels.numPixels(); i++) { // For each pixel in str
    pixels.setPixelColor(i, color);         //  Set pixel's color (i
    pixels.show();                          //  Update strip to matc
    delay(wait);                           //  Pause for a moment
  }
}
void whiteOverRainbow(int whiteSpeed, int whiteLength) {

  if(whiteLength >= pixels.numPixels()) whiteLength = pixels.numPixe

  int      head          = whiteLength - 1;
  int      tail          = 0;
  int      loops         = 3;
  int      loopNum       = 0;
  uint32_t lastTime      = millis();
  uint32_t firstPixelHue = 0;

  for(;;) { // Repeat forever (or until a 'break' or 'return')
    for(int i=0; i<pixels.numPixels(); i++) {  // For each pixel in
      if(((i >= tail) && (i <= head)) ||      //  If between head &
         ((tail > head) && ((i >= tail) || (i <= head)))) {
        pixels.setPixelColor(i, pixels.Color(0, 0, 0, 255)); // Set
      } else {                                             // else s
        int pixelHue = firstPixelHue + (i * 65536L / pixels.numPixel
        pixels.setPixelColor(i, pixels.gamma32(pixels.ColorHSV(pixel
      }
    }
```

```
    pixels.show(); // Update strip with new contents
    // There's no delay here, it just runs full-tilt until the timer
    // counter combination below runs out.

    firstPixelHue += 40; // Advance just a little along the color wh

    if((millis() - lastTime) > whiteSpeed) { // Time to update head/
      if(++head >= pixels.numPixels()) {      // Advance head, wrap
        head = 0;
        if(++loopNum >= loops) return;
      }
      if(++tail >= pixels.numPixels()) {      // Advance tail, wrap
        tail = 0;
      }
      lastTime = millis();                   // Save time of last mo
    }
  }
}
```

Setup and Fabrication

1) Board Design Layout

We use Autodesk Eagle CAD in order to get board layout.The board layout is as follows:

We print this Layout on glossy paper and then it is used for PCB Fabrication

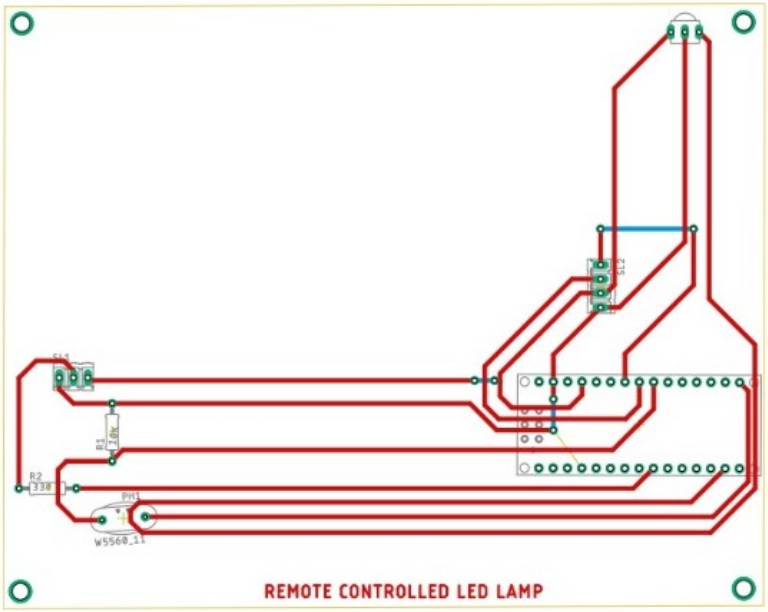

Figure 8: Board layout Design

2) PCB frabrication:

Following are the step of PCB Frabrication:

- Copper Clad Cutting: We use Shopbot machine in order to cut Copper Clad .The Dimensions of copper clad is:

- Sanding And Filing: The copper surface that has been cut so far may have acquired grooves and scratches, which could create issues during the process of PCB (Printed Circuit Board) fabrication. To address this problem, sanding and filling is performed on the copper clad.

- Pressing the board Layout with Iron: To transfer the layout onto the PCB, we usually use a method called ”iron-on transfer.” This involves printing the layout onto a special transfer paper and then using a hot iron to press the paper onto the PCB, transferring the layout onto the board.

- Removing Paper: After pressing, we remove the paper from the PCB by dipping it into the water.

- Etching: After removing paper ,we put PCB (Printed Circuit Board) in FeCl3 (Ferric Chloride) to etch away the copper that is not protected by the resist.The resist is a protective layer applied to the copper surface to protect the desired copper traces and pads from being etched away. When the PCB is placed in the FeCl3 solution, the exposed copper that is not protected by the resist is chemically etched away by the ferric chloride, leaving behind the copper traces and pads that form the circuit pattern.After etching, the PCB is thoroughly cleaned and the resist is removed, leaving only the copper traces and pads that form the circuit pattern on the PCB.

- Drilling: Once we the board has been etched, it is drilled to create holes for the components to be inserted.We drills holes using Drilling machine.

- Soldering: After drilling the components are solder on the board.After that necessary wiring are done.

.

3) Outer Case:

After PCB fabrication and testing ,we make cutouts of Arcylic Sheets in order to make Outer Case in Our Pcb: follows:

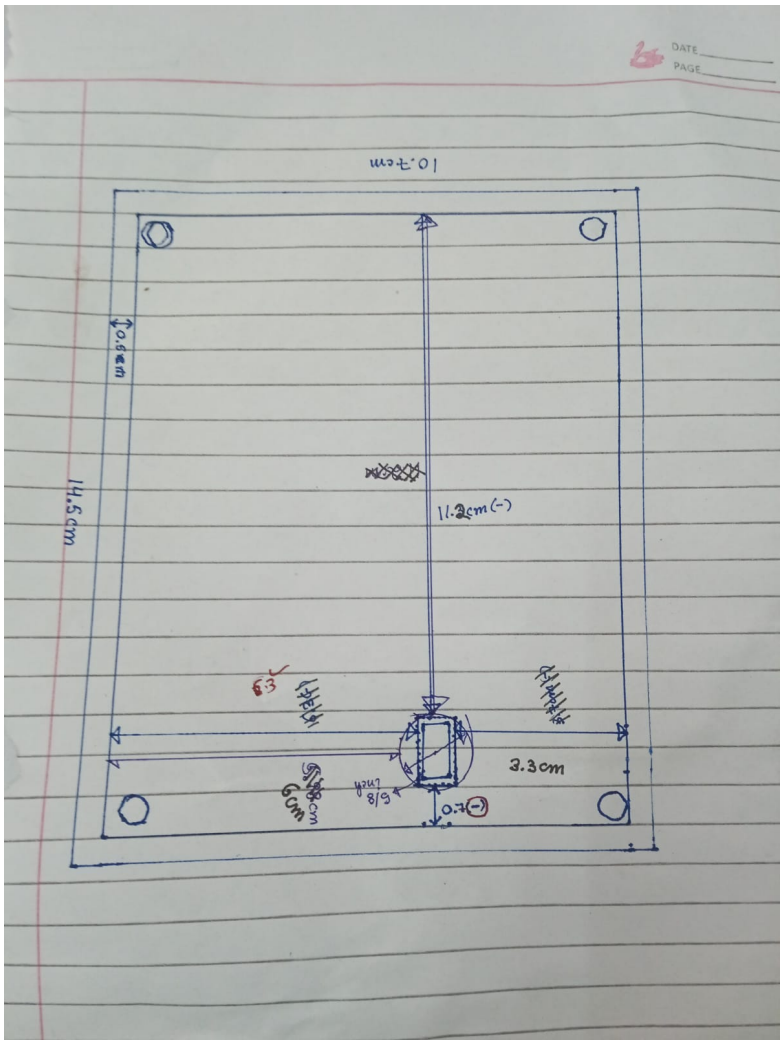

Figure 9: Dimension of outer case

Conclusion

The fully functioning model of our project ”Smart LED Lamp ” has been successfully developed with all functions working properly, it gives us all the feature that is discussed in this project domentation.

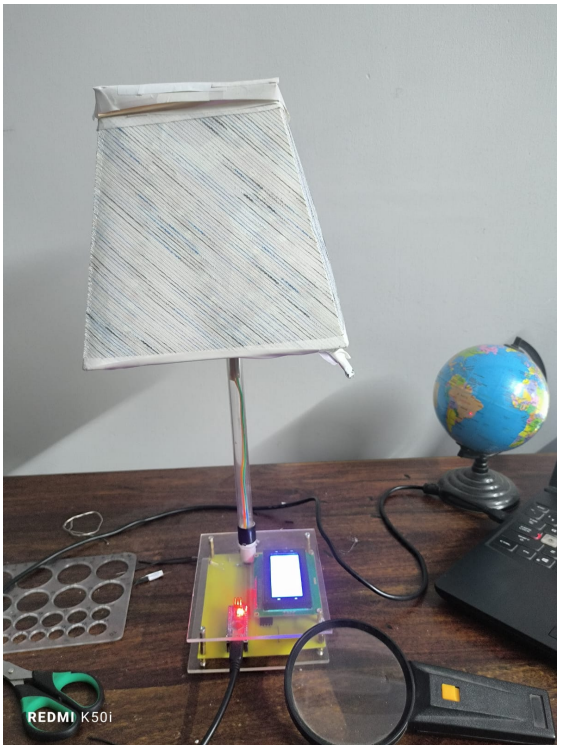

Figure 10: Block diagram of the system

Bibliography