EK 210 Desk Lamp Final Report

Varsha Singh
Ranely Henriquez
Matthew Hendsch
Erin Lynch

Executive Summary

The goal of this desk lamp is to provide users with mobility constraints an accessible, functional lamp that provides a full spectrum of visible lighting. To satisfy this goal, the lamp will require no applied force by the user while also allowing the user a range of space to turn on/off the lamp. To toggle between on/off states, the user places their hand above the proximity sensor's target box for two seconds. The lamp changes its output light depending on the time of day. At 9 P.M, the light will slowly dim over the span of 2 minutes from high to low blue light emissions, effectively minimizing the user's blue light intake at night time.

This device utilizes a RTC module, 10W RGB, and a dual proximity and ambient light sensor. An arduino UNO microcontroller drives logic for our lamp – a necessary implementation for an effective smart device. Additionally, this is strategically housed in a 3D printed cylinder with an attached gooseneck, a laser printed support, and a lampshade.

At this point in time, we have developed a functional and stylish lamp which delivers adequate and uniform illumination to a desk. As part of our potential considerations, we began this project by researching the requirements for a MVP.[1] Once we decided on objectives to meet basic requirements, we then iteratively changed our implementations in meeting the requirements.

Introduction:

Desk lamps operating with small switches or buttons are not usable by those with motor control impairments. Any lamp which functions through applied force fails to account for a wide customer base with disabilities. Similarly, a lamp with high blue light emission can be harmful to those with autism or ADHD[2]. To combat this, we set out to create a desk lamp, catered to providing for those with a lack of fine motor skills or sensory sensitivities to lighting

Problem Statement:

To design a desk lamp that provides lighting and adjusts blue light output based on the time of day, for those with limited motor skills, cognitive and/or sensory sensitivities.

Key Objectives:

- [ ] Illuminates desk (3 to 4 feet diameter)
- [ ] Safe from electrical/fire hazards
- [ ] Reduces blue light exposure closer to bedtime
- [ ] Usable by those with limited motor skills

Important/Critical Design Elements:

- **10W RGB** CHANZON 1DGL-JC-10W-RGB

[1] Minimum viable product.
[2] Autism and blue light sensitivity

Being able to modulate blue light emission while producing bright, yet organic lighting was a key focus in this project; as such, we decided on using a 10 W RGB. This LED works with mosfets and our code to modulate brightness and produce varying color, and requires a minimum of 12V to operate safely.[3] The emitting angle for this component is 120-140°, a feature that synergizes with our developed lamp shade.[4]

- **CLOCK MODULE:** Adafruit PCF8523 Clock Module

This clock module was chosen for this project because when used in the circuit it is able to keep time— this is a necessary step to our project as the blue light dimming and changing is time sensitive. This particular clock module has a coin battery in it so that it will keep track of time even when the lamp is unplugged from the wall power source.

- **LIGHT/PROXIMITY SENSOR :** Adafruit APDS9960 Proximity Light/Gesture Sensor

We utilized this sensor because it enables us to find two different real-time data in parallel: proximity and ambient light.[5] An added consideration that stems from utilizing this component is the placement of the sensor – requiring strategic placement for reachability while avoiding ambient light picked up from the LED.

- **GOOSENECK:** ASIN B08HNH9YR4

This gooseneck allows for the desk lamp to be flexible, while providing a flexible, yet rigid, casing for the wires, and connects the base to the lamp shade.

- **3D PRINTED PARTS:** Base, Lid and Lamp Shade

These parts house our design: the base houses the microcontroller, RTC module, neck support, proximity/light sensor, solderable breadboard, and all wired connections. The lid supports the neck, safely containing the electrical components of the lamp. The shade houses the 10W RGB LED. The internal part of the shade is threaded to securely fasten the gooseneck.[6] PLA was used to 3D print these parts due to accessibility, affordability, and stability.

- **LASER CUT GOOSENECK HOLDER:** Housing made with 1/8 inch MDF

This part supports the gooseneck to the base. It was designed on Makercase using the polygon box feature; which allows for designing boxes with an uneven number of sides.[7]A triangle design was used for this part to minimize the space needed to support the gooseneck as the other components needed to fit into the box. This triangle attaches together via finger slots. Setting the kerf compensation to .0035 allows for the sides to fit together snuggly. The holder is attached to the base via hot gorilla glue.

---

3 RBG LED product link

4 ibid.

5 Copy of adafruit-apds9960-breakout.pdf

6 Figure 1

7 Figure 3

Description of Final Product:

The final project is a desk lamp with a circular 3D printed base and lid, a gooseneck and a 3D printed lamp shade. The circular base houses the arduino UNO and the circuit which consists of a 10W RGB LED, gesture/ambient light sensor, clock module and mosfets. On the base lid, there is a hole for the gesture/ambient light sensor. In the middle of the base, a laser cut housing triangle is included to support the desk lamp. The base and the lid are attached with velcro to make accessing the arduino and circuit easy as we continue to improve our design.

Evaluation of Results:

To rate and iterate our design, we began this project by researching quantitative and standardized metrics for general functionality of a lamp, and then created additional metrics – slightly more qualitative – as based on our client's needs. To meet our metrics, we vastly utilized our technical skills developed in EK131: (circuit wiring, KVL, arduino coding, etc.). Additionally, we also developed metrics, like: low intensity of blue light, can withstand fall from table, and overall cost being less than $200 – based on our client's responses.[8]

Our final design faced many iterations – both in physical and logical choices. Our first prototype design involved 3D-printing parts for the base and lamp shade, as well as PVC pipes for the neck of the lamp. The code was in state like that of an MVP. After our first presentation, we took inspiration from our peers' designs and worked towards implementing a sleek and thinner design – remodeling the base, implementing a gooseneck, and developing a more user-friendly logic for our electronic components. Based on feedback, our biggest change was our time-based light dimming as well as an energy-efficient implementation of turning off the light if the room is bright enough.[9] After finishing our final design, we tested our effective beam spreading through the use of lux meters, as well as testing the reliability and accuracy of the proximity sensor.[10]

Lessons Learned:

We learned a lot of practical skills in this project. We gained a deeper understanding of the engineering design process. We refined our skills with CAD, circuit design, arduino, and soldering. Another important skill we learned was how to read technical documentation. Additionally, we learned open communication skills, as coordinating information between every member of the team was difficult at times. We learned how to keep each other informed in order to ensure that there was no redundancy or inefficiency in our design process.

---

[8] Smart desk lamp for users with specific needs

[9] Lamp Dimming

[10] EK210 Final Results

References:

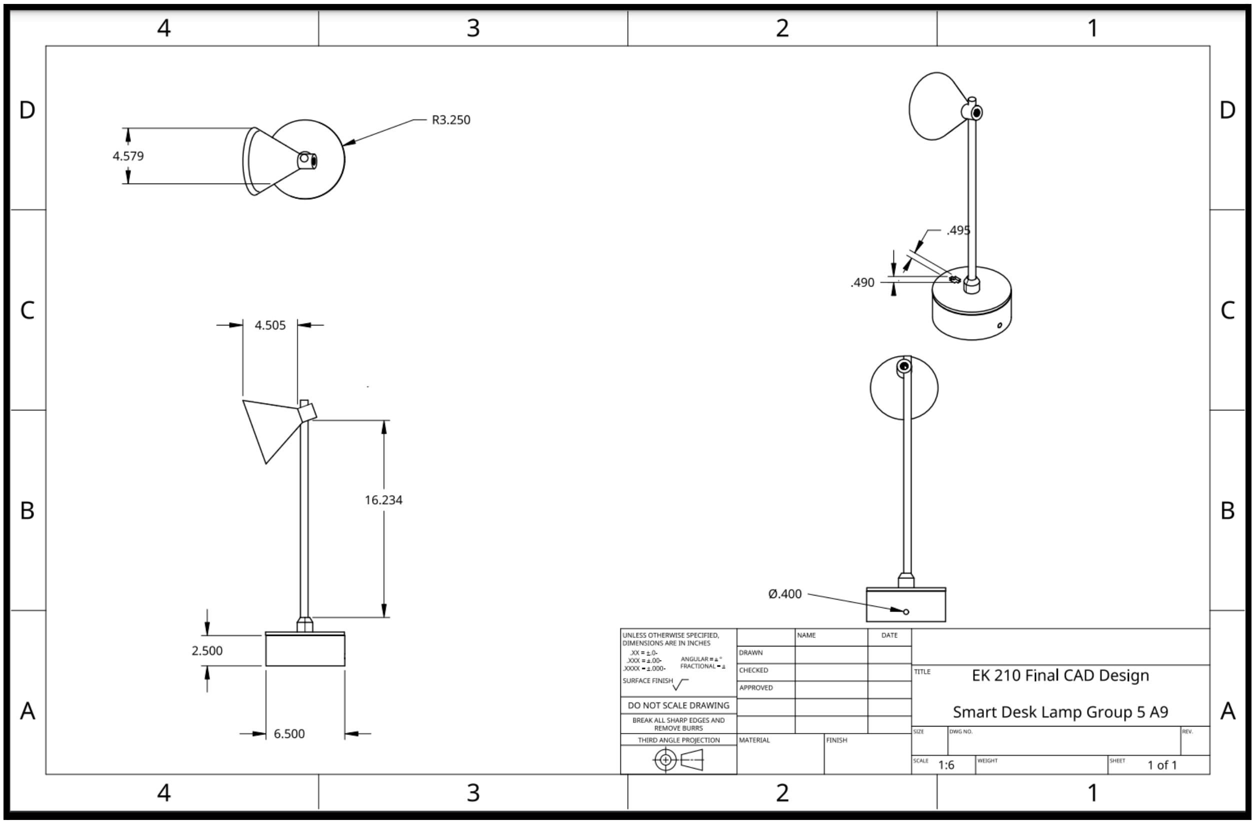
"Autism Light Sensitivity: Causes, Symptoms, Solution." *Axon Optics*, Axon Optics, 9 Aug. 2021, axonoptics.com/blogs/post/autism-light-sensitivity-causes-symptoms-solution.

Chanzon High Power Led Chip 10W RGB Common Anode (300mA - 350mA) *Amazon.Com*, www.amazon.com/Chanzon-Power-Common-Anode-300mA/dp/B01DBZK64K. Accessed 4 Dec. 2023.

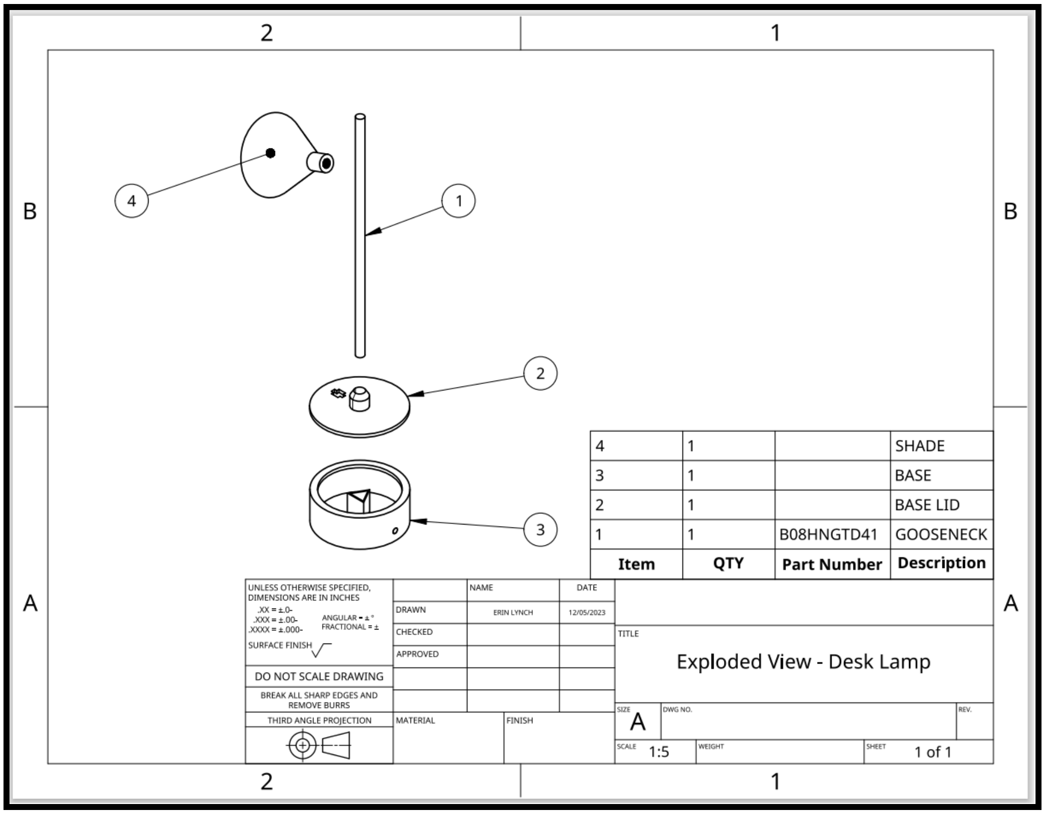
Cooper, Justin. "Adafruit APDS9960 Breakout." *Adafruit Learning System* learn.adafruit.com/adafruit-apds9960-breakout. Accessed 4 Dec. 2023.

"Minimum Viable Product." *Wikipedia*, Wikimedia Foundation, 6 Dec. 2023, en.wikipedia.org/wiki/Minimum_viable_product.

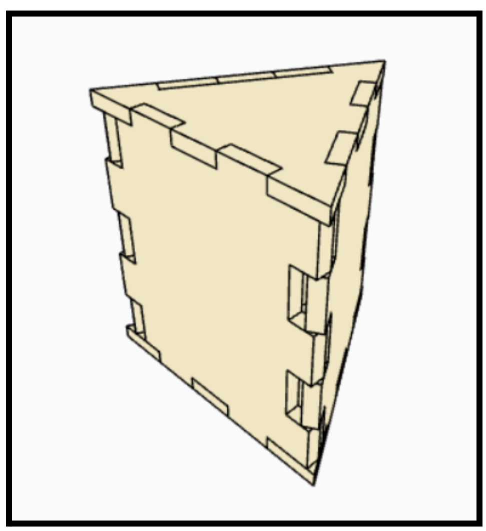
Appendix



(Figure 1: Assembly Drawing)

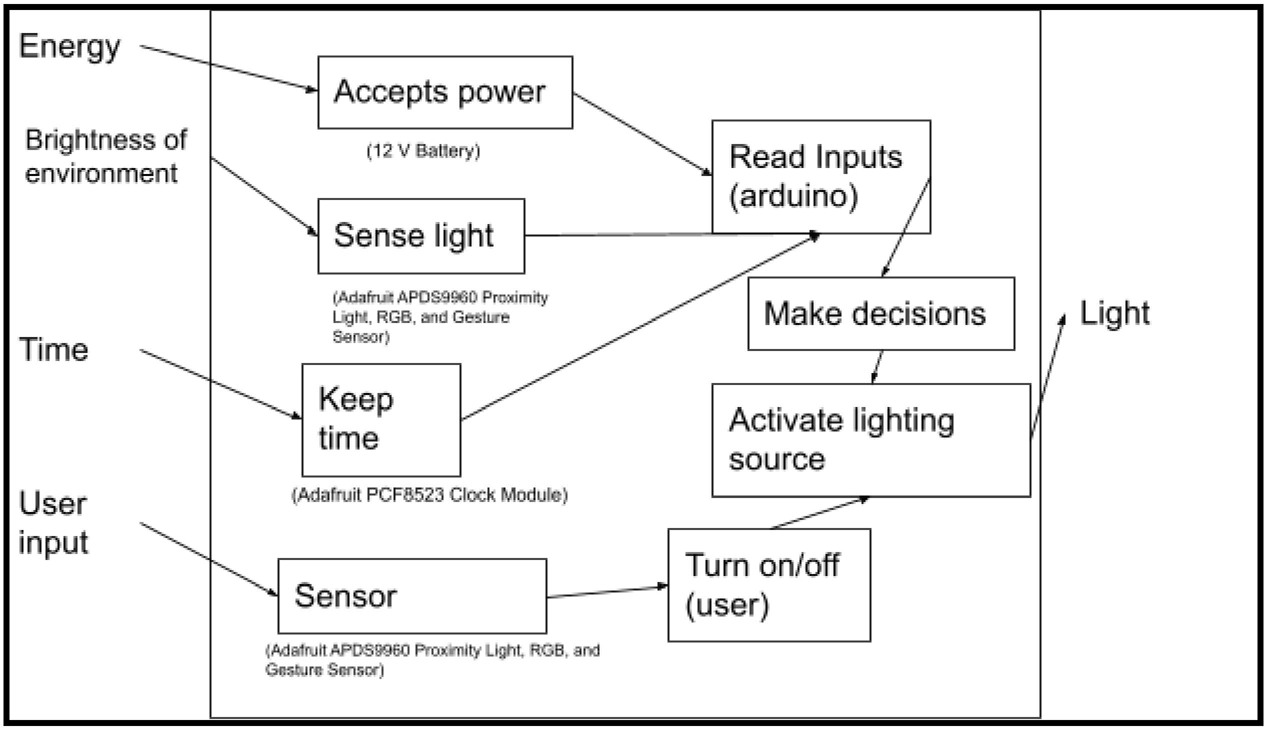


(Figure 2: Exploded View CAD Drawing)

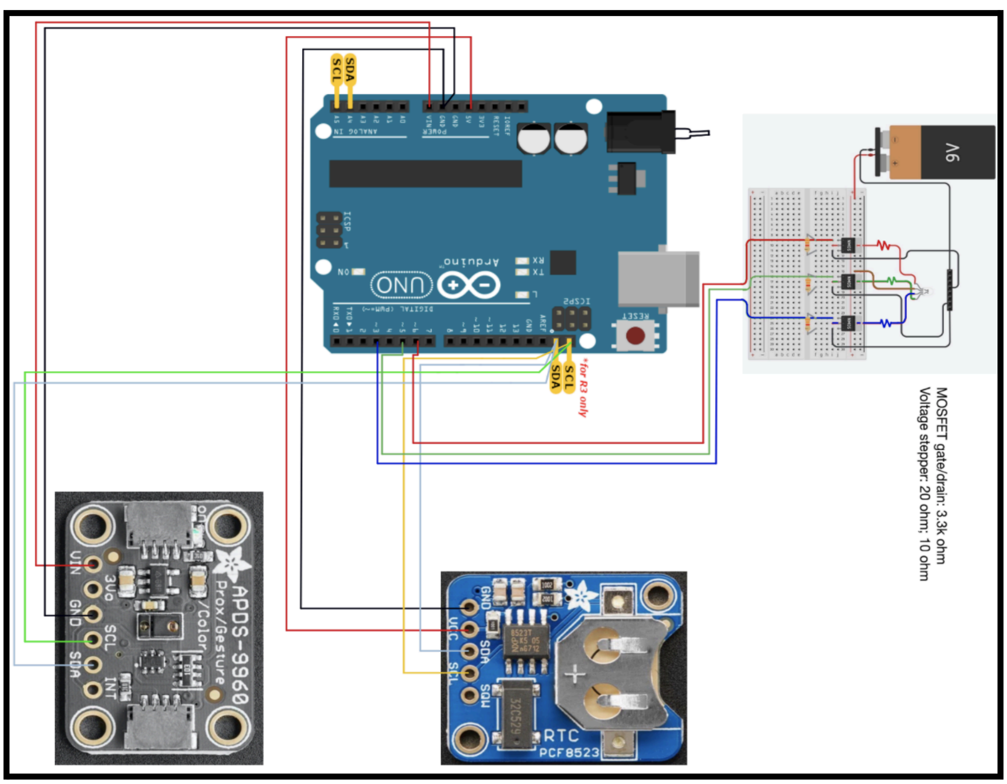


(Figure 3: Gooseneck Holder Design)[11]

[11] Missing a hole on the lid to fit the gooseneck spout



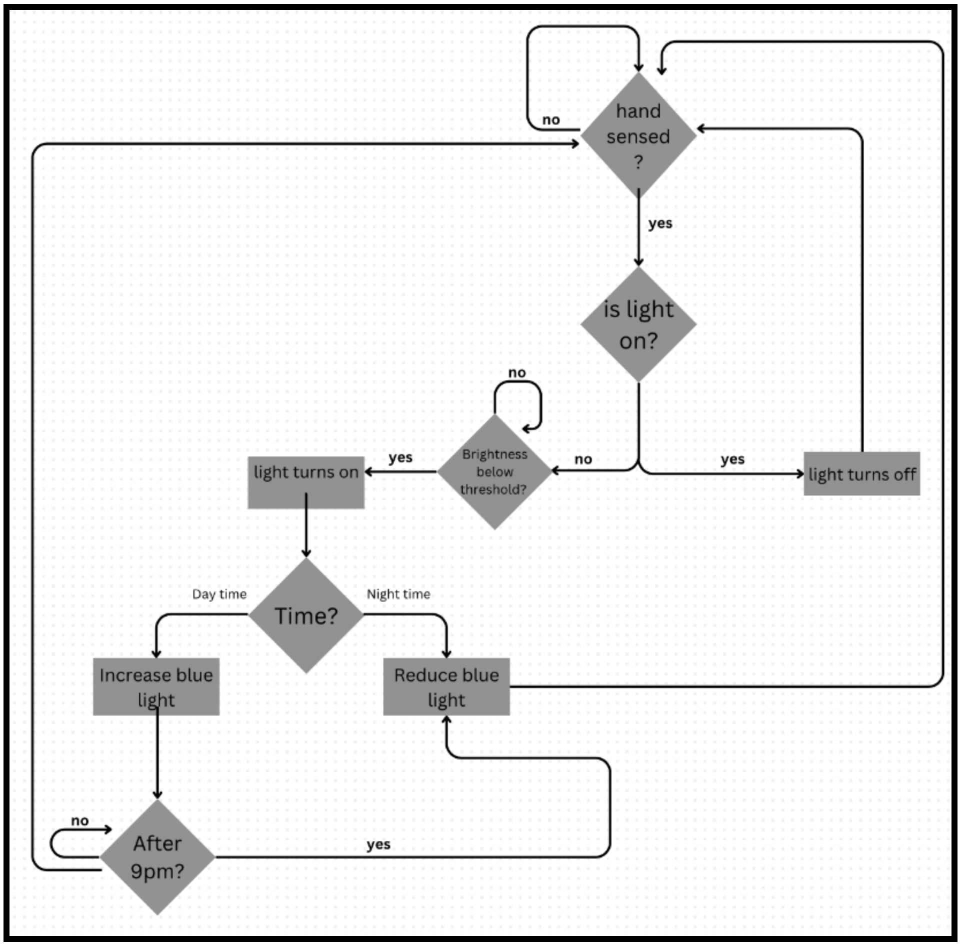
(Figure 4: Glass Box)



(Figure 5: Circuit Diagram)

Objective	Metric	Support
Safe (from electrical/ fire/ shock)	Does not catch fire	Heat sink and appropriate wire gauge
	Will not shock user	No exposed wires, no access to circuity,
	Can withstand a temperature up to 100 degrees	Heat sink, PLA can withstand up to 410 F
Usable by those with limited motor skills	On/ Off is touchless/ minimal force	Gesture/ambient light sensor
Reduces blue light	Low intensity of blue light	1/6th blue light from 9pm-9am
Easily moveable	Less than 5 pounds	<4.5 lbs
Durable	Lifespan of 5+ years	LED lifespan is 5000 hrs = 5.7 years non-stop
	Can withstand fall off table	Survived fall from table
Affordable	Less than $200	Used <$100

(Figure 6: Metrics/Objectives Chart)



(Figure 7: Code Flow Chart)

```
#include "RTClib.h"
#include "Adafruit_APDS9960.h"
RTC_DS3231 rtc;
//create the APDS9960 object
Adafruit_APDS9960 apds;
//the pin that the interrupt is attached to
#define INT_PIN 3
#define BEGINNING_NIGHT 20
#define END_NIGHT 8
//char daysOfTheWeek[7][12] = { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" };
bool proxSensed, resetProx;
int startTime, endTime, timeElapsed, startDim;
bool lightOn, beginDim;
bool alreadyDetected;
bool brightnessOffed;
bool lowed;
void setup() {
  lowed = false;
  brightnessOffed = false;
  lightOn = false;
  proxSensed = false;
  resetProx = false;
  beginDim = false;
  alreadyDetected = false;
  Serial.begin(57600);
  pinMode(INT_PIN, INPUT_PULLUP);

  if (!apds.begin()) {
    Serial.println("failed to initialize device! Please check your wiring.");
  } else Serial.println("Device initialized!");

  //enable proximity mode
  apds.enableProximity(true);

  //enable color sensing mode
  apds.enableColor(true);


  //set the interrupt threshold to fire when proximity reading goes above 175
  apds.setProximityInterruptThreshold(0, 1);

  //enable the proximity interrupt
  apds.enableProximityInterrupt();


#ifndef ESP8266
  while (!Serial)
    ;  // wait for serial port to connect. Needed for native USB
#endif

  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    Serial.flush();
    while (1) delay(10);
  }

  if (rtc.lostPower()) {
    Serial.println("RTC lost power, let's set the time!");
    // When time needs to be set on a new device, or after a power loss, the
    // following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  }
```

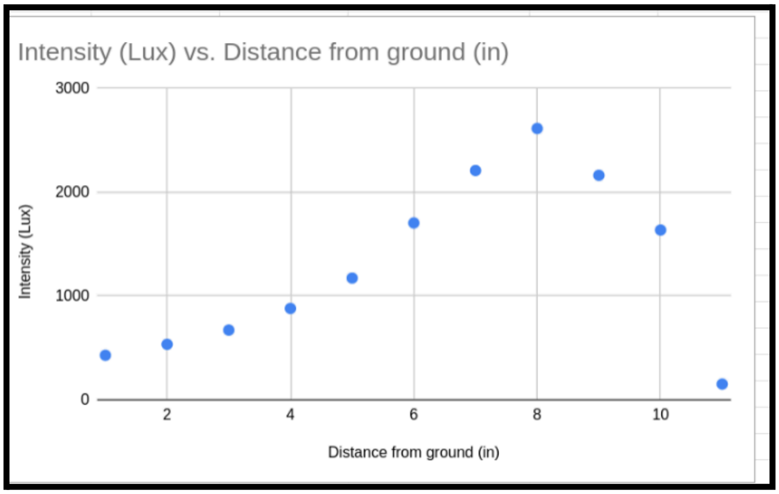
(Figure 8a: Code Section 1)

```
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(9, OUTPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  DateTime now = rtc.now();
  int currentHour = now.hour();
  int currentMinute = now.minute();
  int currentSecond = now.second();
  uint16_t r, g, b, c;
  //bool interrupted;
  int distance = apds.readProximity();

  //setHigh();
  //wait for color data to be ready
  while (!apds.colorDataReady()) {
    delay(5);
  }
  apds.getColorData(&r, &g, &b, &c);


  if (c < 200) {
    if (brightnessOffed) {
      if (currentHour > BEGINNING_NIGHT || currentHour < END_NIGHT) {
        setLow();  //Low blue light
        lightOn = true;
        lowed = true;
      } else {
        setHigh();  //High blue light
        lightOn = true;
        lowed = false;
      }
      brightnessOffed = false;
    }
    if ((currentHour > BEGINNING_NIGHT || currentHour < END_NIGHT) && lightOn && !lowed) {
      //Serial.println("Changing");
      if (!beginDim) {
        //Serial.println("Not GOOD");
        beginDim = true;
        startDim = (int)now.unixtime();
      }
    } else {
      beginDim = false;
    }

    //Serial.println(now.unixtime() - startDim);
    if (beginDim && ((int)now.unixtime() - startDim) <= 125) {
      analogWrite(6, 150 - (((int)now.unixtime() - startDim)));
      //Serial.println((int)now.unixtime() - startDim);
      //Serial.println("DIMMING");
    }
    if (distance < 2) {
      proxSensed = false;
      alreadyDetected = false;
      timeElapsed = 0;
    }
    if (!alreadyDetected && proxSensed && (distance > 1)) {
      endTime = (int)now.unixtime();
      //Serial.print("End: ");
      //Serial.println(endTime);
      timeElapsed = endTime - startTime;
      //Serial.println("Time elapsed: ");
      //Serial.println(timeElapsed);
```

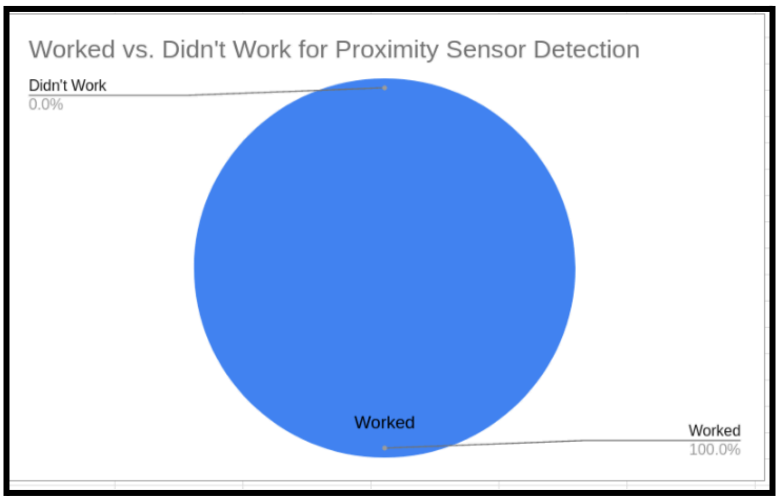
(Figure 8b: Code Section 2)

```
      //Serial.println(timeElapsed);
      if ((timeElapsed > 1) && !alreadyDetected) {
        //timeElapsed = 0;
        //Serial.println("TRYING");
        alreadyDetected = true;
        if (lightOn) {
          ledOff();
          lightOn = false;
          Serial.println("WORKED");
        } else {
          if (currentHour > BEGINNING_NIGHT || currentHour < END_NIGHT) {
            setLow();  //Low blue light
            //setHigh();
            lightOn = true;
            lowed = true;
            Serial.println("WORKED");
            //lowed = false;
          } else {
            setHigh();  //High blue light
            lightOn = true;
            lowed = false;
            Serial.println("WORKED");
          }
        }
        //Serial.println("REGISTERED");
      }
    }
    if ((distance > 1) && !alreadyDetected && !proxSensed) {

      startTime = (int)now.unixtime();

      proxSensed = true;
    }
  } else {
    brightnessOffed = true;
    ledOff();
    Serial.print("Light level: ");
    Serial.println(c);
  }
  if (resetProx) {
    resetProx = false;
    proxSensed = false;
  }
}

void setLow() {
  //150 150 50
  analogWrite(5, 150);
  analogWrite(9, 150);
  analogWrite(6, 25);
}

void setHigh() {
  //150 150 150
  analogWrite(5, 150);
  analogWrite(9, 150);
  analogWrite(6, 150);
}

void ledOff() {
  analogWrite(5, 0);
  analogWrite(6, 0);
  analogWrite(9, 0);
}
```

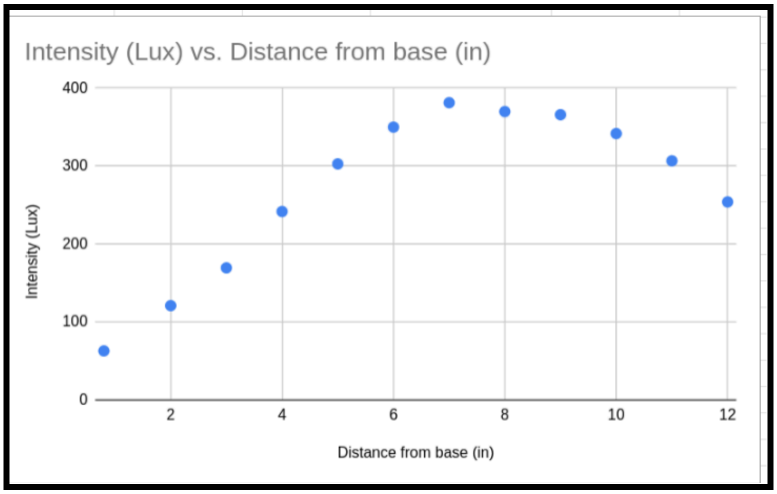
(Figure 8c: Code Section 3)



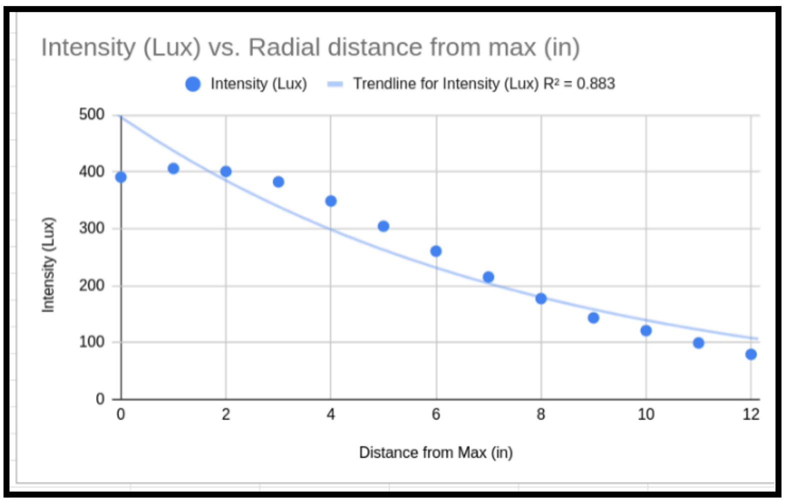
(Figure 9: Light measurement, lux from tabletop)



(Figure 10: Testing proximity sensor)



(Figure 11: Light measurement, lux from base)



(Figure 12: Light measurement, lux radially)