

UNIVERSIDAD DEL VALLE DE GUATEMALA

IM2015 - Manufactura Digital

Sección 11

Ing. José Bagur



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Proyecto Mood Lamp

Diego Ramírez, 21579

Javier Villatoro, 21159

Axel Coronado, 21533

Cristian Ramírez, 21564

GUATEMALA, 20 de septiembre de 2021

Índice

Materiales	3
Técnicas de Manufactura Digital	3
Código de la Lámpara	4
Circuito de la Lámpara	4
Anexos	4

1. Materiales

- Arduino Nano
- Sensor de temperatura y humedad DHT11
- Sensor de Humo MQ-135 ***
- Cautin 30w
- Cautin 100w
- Pasta para soldar
- Alambre galvanizado calibre 14
- Estaño
- Roldana $\frac{3}{8}$ 3 cm
- Tuerca $\frac{3}{8}$
- Base de madera* revisar planos en anexos
- 15 piezas de MDF* revisar planos en anexos
- 2 barras de metal de 32 centímetros, rosca $\frac{3}{8}$
- Cables de conexión para arduino (jumpers)
- Resistencias 220 Ω
- Placa perforada
- Tela
- Tubo termocontraible
- Barniz
- Botón
- Switch
- tubo pvc

***Este sensor se averió sin razón al realizar las últimas pruebas, por lo que se tuvo que desoldar y modificar el código. El código original está al final del documento y debe tomarse en cuenta que funcionaba en conjunto con un buzzer para alertar al usuario de la presencia de humo y demás gases nocivos.

2. Técnicas de Manufactura Digital

Se utilizó un programa CAD, Inventor, para crear el diseño a escala que tendrá la lámpara, los planos de estas piezas se utilizaron para mandar a cortar las piezas de madera y MDF de la lámpara. Asimismo, se programó el código que tendría el circuito de la lámpara, ese circuito tiene 2 funciones, modo manual y modo automático.

El modo manual funciona con 3 potenciómetros que permiten cambiar el color de los neopixels de la lámpara (RGB). El modo automático funciona con un sensor de temperatura, este tiene como función cambiar el color mediante el ambiente, es decir, mientras más temperatura haya la intensidad del color rojo de los neopixels aumentará. Continuando en la estructura de la lámpara, en el interior se realizó una

“base” que sostendrá un tubo de PVC que es en donde irán pegados los neopixeles, esta base es hecha con alambre galvanizado calibre 14 para que tenga una mejor resistencia y pueda sostener sin problema el tubo pvc, De igual manera, el circuito principal del arduino nano, se encuentra debajo de la base manteniéndose escondido mostrando solamente los potenciómetros, el switch y el botón.

Una vez terminado el cableado conectándolo a los neopixels y haber barnizado con dos capas las piezas de madera y MDF, se comienza a armar la estructura que rodea la lámpara. Para esto, se colocan las 14 piezas de MDF, una sobre otra poniendo entre cada una, una roldana, una tuerca encima y otra roldana, arriba y luego se repite el proceso con la otra pieza. Después de esto, se cubre con tela el tubo que está pegado a la estructura de alambre (la tela debe quedar con una forma cuadrada. Este tubo es el que sostiene los neopixels y por debajo pasa el circuito con el arduino para esconderlo en la base de madera. Se tapa la base por debajo con la base hecha con cortadora láser. Por último, las piezas colocadas en la barra roscada se ponen en la base de madera para estar alrededor de la tela y darle el diseño final a la lámpara.

3. Código de la Lámpara

```
#include <Adafruit_NeoPixel.h>
#include "DHT.h"
#define DHTPIN 6
#define button 12

// Pin al que está conectado los neopixeles
int neoPixelPin = 3;

// No. de Neopixeles
int numPixels = 60;

// Inicializar los neopixeles desde la librería
Adafruit_NeoPixel strip = Adafruit_NeoPixel(numPixels, neoPixelPin, NEO_GRB +
NEO_KHZ800);

// Valor globales R, G, B
int r = 0;
int g = 0;
int b = 0;

//-----
```

```

//Manual
//Potenciometros

const int redControl = A0;
const int greenControl = A1;
const int blueControl = A2;

int redPotenciometerValue = 0;
int greenPotenciometerValue = 0;
int bluePotenciometerValue = 0;

int redLedBright = 0;
int greenLedBright = 0;
int blueLedBright = 0;

//Automático
//Humo
int gas = A6;
int buzzer = 5;
int delayval = 550;

//Temperatura
#define DHTTYPE DHT11
int temperatura;
int humedad;

DHT dht(DHTPIN, DHTTYPE);

//-----

//Botón cambio de función
int contador = 0;
int buttonPoll = 0;

void setup(){
  Serial.begin(9600);
  pinMode (button, INPUT_PULLUP);

  pinMode(redControl, INPUT);
  pinMode(greenControl, INPUT);
  pinMode(blueControl, INPUT);

```

```

pinMode(r, OUTPUT);
pinMode(g, OUTPUT);
pinMode(b, OUTPUT);

dht.begin();

strip.begin(); // Inicializar la tira
strip.show(); // Hacerla visible
strip.clear(); // Tira en off
}
void loop() {
  // put your main code here, to run repeatedly:
  buttonPoll = digitalRead(button);
  Serial.println(buttonPoll);

  if (buttonPoll == LOW) {
    delay(250);
    if (contador == 1) {
      contador = 0;
      Serial.println(contador);
    } else {
      contador = 1;
      Serial.println(contador);
    }
  }
}

switch (contador){
  case 0:
    for( int i = 0; i < numPixels; i++ )
    {
      // Lectura del valor de los potenciometros
      redPotenciometerValue = analogRead(redControl);
      delay(5);
      greenPotenciometerValue = analogRead(greenControl);
      delay(5);
      bluePotenciometerValue = analogRead(blueControl);

      r = map(redPotenciometerValue, 0, 1023, 0, 255);
      g = map(greenPotenciometerValue, 0, 1023, 0, 255);
      b = map(bluePotenciometerValue, 0, 1023, 0, 255);;
    }
  }
}

```

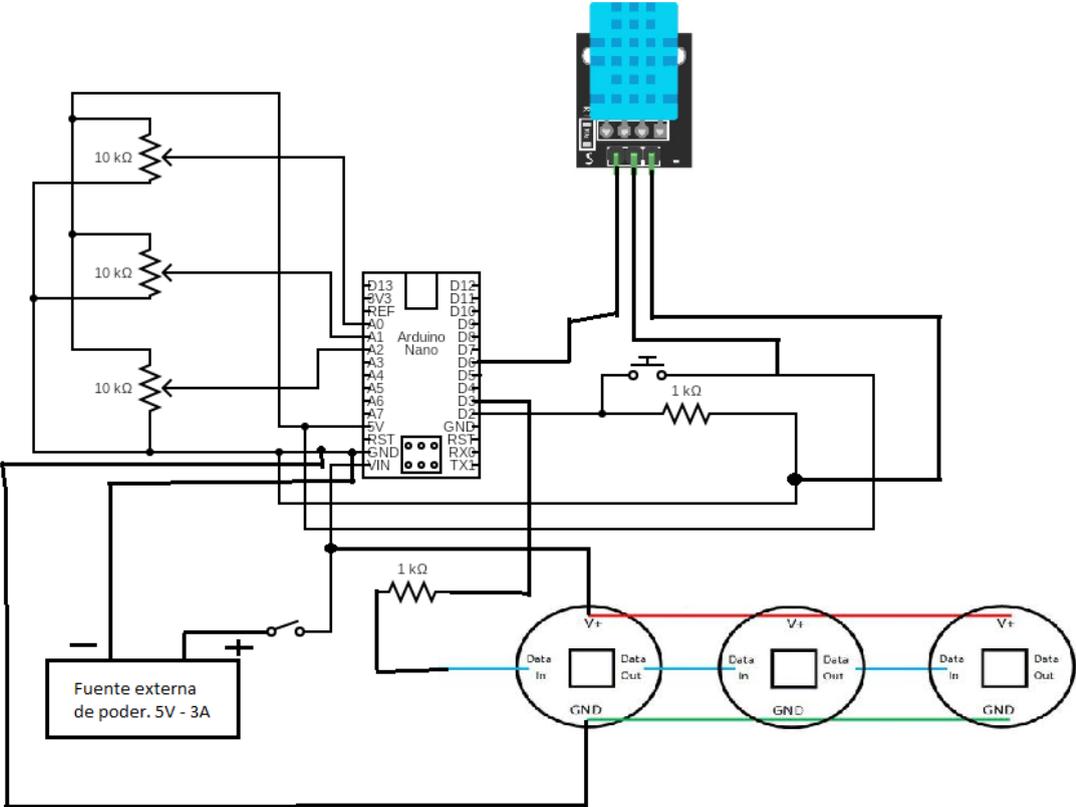
```
strip.setPixelColor(i, strip.Color(r, g, b));
strip.show();

}
break;

case 1:
int h = dht.readHumidity();
int t = dht.readTemperature();
temperatura = map(t, 20, 33, 0, 255);
humedad = map(h, 20, 33, 0, 255);
for(int i=0; i<numPixels; i++){
strip.setPixelColor(i, strip.Color(temperatura,humedad, 0));
strip.show();
}
break;

}
}
```

4. Diagrama esquemático de la Lámpara



5. Anexos

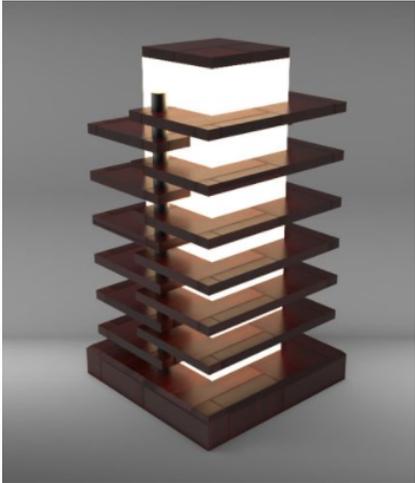


Figura 1. Diseño de la lámpara en inventor.

6. Archivos:

Aquí pueden ver los modelos 3D de las partes y el ensamblaje de la lámpara. Esto les permitirá tener una mejor imagen de cómo se arma. Los archivos son .iam y .ipt; por lo que debes utilizar inventor para poder verlos.

https://drive.google.com/file/d/1zlkOBojUmmnJ_ors840FdFZ6-T4GYp5R/view?usp=sharing

7. Código original (Con buzzer y sensor de humo MQ-135)

```
#include <Adafruit_NeoPixel.h>
#include "DHT.h"
#define DHTPIN 6
#define button 12

// Pin al que está conectado los neopixeles
int neoPixelPin = 3;

// No. de Neopixeles
int numPixels = 60;

// Inicializar los neopixeles desde la librería
Adafruit_NeoPixel strip = Adafruit_NeoPixel(numPixels, neoPixelPin, NEO_GRB +
NEO_KHZ800);

// Valor globales R, G, B
int r = 0;
int g = 0;
int b = 0;

//-----

//Manual
//Potenciómetros

const int redControl = A0;
const int greenControl = A1;
const int blueControl = A2;

int redPotenciometerValue = 0;
int greenPotenciometerValue = 0;
int bluePotenciometerValue = 0;

int redLedBright = 0;
int greenLedBright = 0;
```

```

int blueLedBright = 0;

//Automático
//Humo
int gas = A6;
int buzzer = 5;
int delayval = 550;

//Temperatura
#define DHTTYPE DHT11
int temperatura;
int humedad;

DHT dht(DHTPIN, DHTTYPE);

//-----

//Botón cambio de función
int contador = 0;
int buttonPoll = 0;

void setup(){
  Serial.begin(9600);
  pinMode (button, INPUT_PULLUP);

  pinMode(redControl, INPUT);
  pinMode(greenControl, INPUT);
  pinMode(blueControl, INPUT);

  pinMode(r, OUTPUT);
  pinMode(g, OUTPUT);
  pinMode(b, OUTPUT);

  dht.begin();

  strip.begin(); // Inicializar la tira
  strip.show(); // Hacerla visible
  strip.clear(); // Tira en off
}
void loop() {
  // put your main code here, to run repeatedly:
  buttonPoll = digitalRead(button);

```

```
Serial.println(buttonPoll);
```

```
if (buttonPoll == LOW) {  
  delay(250);  
  if (contador == 1) {  
    contador = 0;  
    Serial.println(contador);  
  } else {  
    contador = 1;  
    Serial.println(contador);  
  }  
}
```

```
switch (contador){  
  case 0:  
    for( int i = 0; i < numPixels; i++ )  
    {  
      // Lectura del valor de los potenciometros  
      redPotenciometerValue = analogRead(redControl);  
      delay(5);  
      greenPotenciometerValue = analogRead(greenControl);  
      delay(5);  
      bluePotenciometerValue = analogRead(blueControl);  
  
      r = map(redPotenciometerValue, 0, 1023, 0, 255);  
      g = map(greenPotenciometerValue, 0, 1023, 0, 255);  
      b = map(bluePotenciometerValue, 0, 1023, 0, 255);  
  
      strip.setPixelColor(i, strip.Color(r, g, b));  
      strip.show();  
    }  
    break;  
  
  case 1:  
    if(gas>450) {  
      for(int i=0; i<numPixels; i++)  
        strip.setPixelColor(i, strip.Color(255,0,0));  
      strip.show();  
      delay(delayval);  
      tone(buzzer,3450,500);  
    }
```

```
}  
else {  
  int h = dht.readHumidity();  
  int t = dht.readTemperature();  
  temperatura = map(t, 0, 40, 0, 255);  
  humedad = map(h, 0, 150, 0, 255);  
  Serial.println(h);  
  for(int i=0; i<numPixels; i++){  
    strip.setPixelColor(i, strip.Color(temperatura,humedad, 0));  
    strip.show();  
    noTone(buzzer);}  
  strip.show();  
  noTone(buzzer);  
  }  
break;  
  
}  
}
```