

# Implementacion

**Juan Jose Arboleda  
Cardona**

Departamento de Ingeniería Electrónica y  
Telecomunicaciones  
Universidad de Antioquia  
Medellín  
Septiembre de 2021

# Índice

<b>1. descripcion</b>	<b>2</b>
<b>2. implementacion</b>	<b>2</b>
2.1. clases, atributos y metodos . . . . .	2
2.2. main . . . . .	2
<b>3. conexiones</b>	<b>3</b>
<b>4. problemas durante la implementacion</b>	<b>3</b>
<b>5. manual de uso</b>	<b>3</b>

## 1. descripcion

problema: En esta ocasión su tarea consiste en recibir una imagen en formato jpg y realizar el procesamiento de la información contenida en esta, de tal forma que se haga un ajuste de sus dimensiones, para que pueda ser presentada en una matriz de LEDs RGB. plantear: - conexiones en Arduino - funcion para redimensionar la imagen de entrada

## 2. implementacion

para mi solucion, tome la imagen de entrada y la dividi en partes iguales y con el punto medio de estas podia promediar los pixeles que estaban a los lados de estas.

con esta aproximacion empee a construir la matriz de 12 x 12 con sus respectivos valores de RGB promediandolos (  $r_1+r_2+...+r_n$  ) /  $num_{pixeles}$ .

### 2.1. clases, atributos y metodos

mi implementacion la sintetize en una clase "imagen" desde la cual se puede acceder a los metodos de sobremuestreo y submuestreo

los atributos son 2, un String ruta , que contiene la direccion de la imagen y es el unico parametro del constructor. QImage png, este ojeito de tipo QImage se utilizará para acceder a las características de la imagen ( tamaño, rgb de cada pixel, etc)

como metodos tenemos

1. constructor: recibe un String con la ruta de la imagen y lo almacena en el atributo ruta; su vez cre un objeto tipo QImage y lo almacena en "png".

2. void resize(); esta funcion toma el QImage "png", submuestra y escribe en un archivo de texto los correspondientes colores RGB para una matriz de 12 x 12, los imprime para mas comodidad .

2. void m\_resize(); esta funcion toma el QImage "png", sobremuestra y escribe en un archivo de texto los correspondientes colores RGB para una matriz de 12 x 12, los imprime para mas comodidad .

### 2.2. main

el main contiene un cin para le ingreso por parte del usuario, de la ruta de la imagen, crea un ojeito tipo imagen y realiza el ajuste de su tamaño

### 3. conexiones

mostrare mi planteamiento para la conexion de arduino para las conexiones,

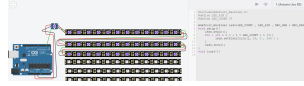


Figura 1: imagen conexiones

desde el pin D2 del arduino va hacia la entrada de la tira de neopixeles de 12, y luego la potencia y tierra.

con la otra tira de neopixeles la conexion es la misma (salida,potencia,tierra) se dirigen hacia ( entrada, potencia,salida) el inicio de la otra tira de neopixel A continuación, se presenta el código de las clases definidas para la solución

### 4. problemas durante la implementacion

durante la implementacion el problema principal fue la desinformacion acerca de como sobremuestrear o submuestrear una imagen. este represento una gran parte investigativa para poder dar con un algoritmo que pudiera satisfacer esta necesidad.

frente a las conexiones no hubo ningun inconveniente, fue facil intuir como conectar entre si las tiras de neopixel.

### 5. manual de uso

el uso es sencillo, al ejecutar el programa en la consola ingresa la ruta de la imagen.

el programa le hara la conversion automaticamente e imprimira de emergencia (por si ha un error abriendo el archivo de salida) una matriz debe copiarla sin los siguientes simbolos "" que se encuentra al inicio y "" que se encuentra al final. esto lo debe pegar entre las llaves que se encuentran en el codigo de tinterkat.

el mismo proceso es para lo que se encuentra en el txt