

Informe Análisis y Diseño

Parcial 2

Manuela Gutiérrez Rodríguez

Daniela Andrea Gallego Díaz

Departamento de Ingeniería Electrónica y
Telecomunicaciones

Universidad de Antioquia

Medellín

21 de Septiembre de 2021

Índice

1. Análisis del problema	2
2. Esquema de tareas	2
3. Algoritmo diseñado	3
3.1. Función submuestreo:	3
3.2. Funcion sobremuestreo:	3
4. Consideraciones	4

1. Análisis del problema

Al comprender el objetivo y funcionamiento del problema hemos concluido que algunos de los retos que se nos presentaran a lo largo del desarrollo de este proyecto seran:

- Uno de los primeros puntos a tener en cuenta sera saber conectar correctamente el circuito y configurarlo adecuadamente, tener claro con que tipo de leds se trabajara y hacer un estudio previo de como es el funcionamiento de este montaje.

- Crear clases para el modelamiento de objetos:

Una de las clases mas importantes para este desafio sera la de submuestreo o sobremuestreo, en donde se podran adecuar las dimensiones de las imagenes leidas teniendo previamente claro las dimensiones de los leds ya montados en el circuito. Para este punto en especifico se tendria pensado en multiplicar y eliminar, segun sea el caso, los pixeles cada cierta cantidad.

Tambien es importante modelar de manera correcta una clase para la lectura de las imagenes y que en esta misma se genere el archivo txt de salida con la informacion correspondiente, para ello, tenemos en cuenta que otro de los retos sera construir y modelar una clase que genere las matrices de manera eficiente y teniendo en cuenta la gama de colores del RGB para que al pasar la informacion al proyecto en arduino se pueda apreciar gran parte de la imagen original.

- Por ultimo, en la plataforma de arduino tenemos el desafio de poder escribir la informacion generada por Qt de manera correcta (que los colores y sus limites esten bien definidos) y que no se pierda tanto detalle a comparacion de la imagen original, esto teniendo en cuenta de que ya se ha hecho un estudio de todo el funcionamiento de este tipo de leds.

2. Esquema de tareas

Lista de tareas:

1. En primer lugar diseñar las funciones en qt para el cargue de imagenes y para leer cada uno de los pixeles correspondientes a ésta de acuerdo al model RGB.

2. Realizar un análisis detallado sobre los algoritmos que van a ofrecer la posibilidad de ajustar las imágenes (agrandar o reducir) de acuerdo a las dimensiones de la matriz de leds.

3. A partir del análisis realizado, implementar un algoritmo en qt que permita la realización y aplicación de dichas funciones.

4. Analizar la matriz que corresponde al modelo RGB de la imagen y como esta debe ser almacenada en un archivo .txt

5. Diseñar el circuito en thinkercad, conectar la cantidad de leds o de tiras de leds para la correcta implementación del algoritmo.

6. Implementar en thinkercad las funciones que a partir del contenido del archivo txt van a permitir mostrar en la matriz de led el color deseado.

7. Realizar el segundo informe de implementación.

8. Realizar un manual de usuario, para el correcto uso del arduino.
9. Realizar el video explicando todo lo realizado.

3. Algoritmo diseñado

3.1. Función submuestreo:

```

Vector de vectores de enteros vetorRGB
AnchoFinal=ancho/12
AltoFinal=alto/12
Mientras i!=AnchoFinal*AltoFinal
    Para x=0, hasta x menor que AnchoFinal, x=x+1
        Para y=0 hasta y menor que AltoFinal, y=y+1
            Obtener color rojo
            Obtener color verde
            Obtener color azul
            Almacenar los valores en el vector de enteros
            Y este vector almacenarlo en el vector de vectores
        Fin para
    Fin para
tamVec=tamaño vector de vectores
Para k=0 menor que tamVec
    rojos=rojos+vector de vectores[ posicion k] [posicion 0]
    verdes=verdes+vector de vectores[ posicion k] [posicion 1]
    azules=azules+vector de vectores[ posicion k] [posicion 2]
Fin para
rojos/tamVec
verdes/tamVec
azules/tamVec
vectorRGB[0]=rojos
vectorRGB[1]=verdes
vectorRGB[2]=azules
matrizFinal[Posicion]=vectorRGB
i=i+1    posicion=posicion+1
Fin mientras

```

3.2. Funcion sobremuestreo:

```

inicializacion de contenedores
vector de vectores (almacena una matriz completa)
vector fila (se almacena cada fila)
De acuerdo a la lectura de la imagen se empiezan a almacenar las matrices
matrices de acuerdo al RGB con sus dimensiones originales
Matriz valores color rojo
Matriz valores color verde

```

```

Matriz valores color azul
dimensiones de la matriz=
Ancho original(valor obtenido)xAlto original(valor obtenido)
Si las dimensiones de la matriz final es de 12x12 entonces:
numero de elementos del vector de vectores= 12 (hay 12 vectores fila)
numero de elementos del vector fila= 12 (es decir 12 columnas)
se itera sobre la matriz:
(Se repite el ciclo con la matriz valores color rojo,verde y azul)
i=0, i<alto original, i++
    j=0,j<ancho original, j++
    se llena el vector de vectores y cada vector fila
    (se itera sobre los vectores)
        vector de vectores[i]
        vector fila posicion i[j]
    Se asigna el mismo valor a varias posiciones seguidas en el vector
    Esto hasta que numero de elementos vector de vectores= 12
    Y hasta que el numero de elementos de cada vector fila= 12
    (Es decir, se multiplica cada valor para hacer mas grande
    la matriz hasta que las dimensiones sean de 12x12)
fin del ciclo
vector de vectores (para cada color)= matriz final en dimension 12x12

```

4. Consideraciones

-Con respecto al proyecto en Tinkercard hay que tener claro las dimensiones del circuito de neopixeles para que en base de estas se realicen las demas funciones para el programa, ademas del correcto funcionamiento del circuito y las funciones implementadas alli (impresion de la matriz en los neopixeles)

-Tener en cuenta los diferentes tipos de contenedores y elegir el adecuado para este programa para una adecuada implementacion. Tambien tener en cuenta la libreria Qimage que ayudara en varios procesos del programa.

-Tener claro el proceso de implementacion en las funciones de submuestreo y sobremuestreo ya que son las funciones mas importantes en todo el programa, en conjunto con el proceso de implementacion en tinkercard.