

Informe de análisis y diseño

Parcial 2

Angie Paola Jaramillo Ortega

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

Índice

1. Sección introductoria	2
2. Análisis	2
2.1. Tareas	2
2.2. Alternativa de solución	3
2.3. Algoritmo	3
3. Consideraciones	3

1. Sección introductoria

Se ha asignado el trabajo de crear una pantalla de LED's para la representación de distintas banderas. El programa debe ser capaz de mostrar la bandera escogida por el usuario sin importar su tamaño y debe ser reconocible.

2. Análisis

Una imagen seleccionada por el usuario debe ser leída por el programa en Qt, se debe procesar la imagen por submuestreo o sobremuestreo para adaptarse a una matriz de LEDs y se debe generar un archivo txt con la porción de código que represente la imagen a mostrar en la pantalla, la información del archivo txt será pasada al código de arduino y al iniciar la simulación la bandera debe ser mostrada en la pantalla LED.

Datos de entrada: Nombre de un archivo imagen.jpg

Datos de salida en Qt: Segmento de código a ser agregado en el controlador de la matriz de LEDs de arduino

Datos de salida en arduino: Representación de la imagen ingresada en la matriz de LEDs

Para implementar esta solución se hará uso de la clase QImage para abrir la imagen y leer el su contenido. Se usará los métodos de la clase QImage, width() y heigh() para obtener el ancho y la altura de la imagen respectivamente. El método pixelColor() de la clase QImage servirá para obtener los valores RGB de cada pixel. Para el código en arduino se utilizará principalmente la función setPixelColor() para especificar el color de cada pixel.

2.1. Tareas

Lectura de imagen

Obtener medidas de la imagen

Definir si es necesario submuestreo o sobremuestreo

Hacer proceso de transformación

Crear archivo txt

En arduino leer la nueva imagen

establecer valores RGB de cada pixel

2.2. Alternativa de solución

para el procesamiento de imagenes dependerá de si es necesario realizar submuestreo o sobremuestreo.

Submuestreo: En caso de que se necesite reducir la cantidad de pixeles de la imagen ingresada se realizará promediando los valores RGB de un conjunto de pixeles para ubicarlo en un pixel correspondiente de la matriz de LEDs.

Sobremuestreo: En caso de tener que aumentar el tamaño de la bandera se usará la técnica de interpolación bilinial, esta calcula los valores de una ubicación utilizando las cuatro posiciones mas cercanas para realizar un promedio ponderado usando como pesos la distancia a estos cuatro puntos

2.3. Algoritmo

El siguiente algoritmo 2.3, es un diseño general de la solución planteada.

```
{
  Escribir "Ingrese nombre de archivo imagen"
  Leer imagen
  Crear objeto imagen

  ancho←leer ancho de imagen
  altura←leer altura de imagen

  SI altura>10 OR ancho>10 entonces
    submuestrear(imagen)
  SINO altura<10 OR ancho<10 entonces
    sobremuestrear(imagen)
  SINO altura=10 AND ancho=10 entonces
    entero array[10][10][3]
    Desde fila=0 hasta 10 con paso 1 hacer
      Desde columna=0 hasta 10 hacer
        ColorR←leer color rojo de pixel (fila ,columna)
        ColorG←leer color verde de pixel (fila ,columna)
        ColorB←leer color azul de pixel (fila ,columna)
        array [ fila ][ columna]←[colorR ,ColorG ,ColorB]
  FIN SI
}
```

3. Consideraciones

La solución del problema se realizará en una matriz de LEDs 10x10 y la cantidad de pixeles a promediar durante el submuestreo dependerá de la dimensión

de la imagen original. Los promedios sacados para el submuestreo como para el sobremuestreo se realizarán para cada componente R, G y B.