

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. Alternativa de solución	2
2.2. Algoritmo	2
3. Consideraciones	3

1. Sección introductoria

Se ha asignado el trabajo de crear una pantalla de leds para la representación de distintas banderas. El programa debe permitir la representación de una bandera aleatoria 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

2.1. Alternativa de solución

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

Submuestreo: En caso de que se necesite reducir la cantidad de píxeles de la imagen ingresada se realizará promediando los valores RGB de un conjunto de píxeles para ubicarlo en un píxel 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 más cercanas para realizar un promedio ponderado usando como pesos la distancia a estos cuatro puntos

2.2. Algoritmo

El siguiente algoritmo 2.2, 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
    array[ancho][altura]<-leer y guardar colores de cada pixel
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.