Parcial2

Informe de implementación

Nombres y Apellidos del autor
Julián David Quintero Marín

Informática II

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
September de 2021

Índice

1. Clases implementadas

En este proyecto implementamos las clase QImage, con esta clase se puede almacenar una imagen al obtener su dirección y asi leerla y modificarle el color de cada pixel que hace parte de la imagen. Aunquepara poder modificar el color de un pixel en específico de una posición en específico,los parámetros que exige dicho método (setPixelColor) corresponde al uso de un objeto de la clase QColor, en el cual se almacenan los datos de los colores RGB que se le quieren asignar al pixel de la imagen en la posición (x,y) que se le defina. De este modo, la clase QColor también es usada dentro del proyecto. Como segunda clase, como se había planteado anteriormente en el informe de diseño de la solución, se plantea la creación de la clase "pixmat", la cual se encarga de almacenar como elementos los punteros auxiliares necesarios para la lectura de la imagen, así como los contenedores tipo vector necesarios para poder guardar los datos y acceder a ellos fácilmente, pues deben ser escritos en el archivo de texto para usar su contenido como la información que el usuario ingrese en el proyecto en Tinkercad cuando éste lo requiera.

2. Esquema de la estructura final de las clases implementadas

3. Módulos de código implementado donde interactúan las diferentes clases

4. Estructura del circuito montado

En caso de modificaciones, el último montaje de la matriz de Neopixeles es la versión final del proyecto en Tinkercad. La primera versión corresponde a una matriz de Neopixeles de 16 x 16 en la cual se usa el pin digital 2 del Arduino Uno como pin de salida para transmitir la información leída del archivo de texto y guardada en un arreglo tridimensional de valores enteros, usando los métodos de la librería correspondiente de dichos componentes para su efectiva configuración a partir del uso del pin 2 del Arduino Uno. En esta matriz, a pesar de estas dispuestos como tiras de 16 neopixeles una encima de la otra, en cuanto a conexiones, están conectados de modo que la entrada de la tira de la fila superior está conectada en su puerto de entrada al pin 2 y su puerto negativo y positivo a los que le corresponden al arduino, haciendo uso a su vez de un suministro de energía para asegurar la transmisión de suficiente voltaje para las 16 tiras en el circuito. Los segundos puertos de positivo, negativo y el puerto de salida están conectados, respectivamente, a los puertos positivo, negativo y de entrada de la tira de 8 neopixeles de la fila inferior a ella. Y así sucesivamente hasta llegar a la tira de neopixeles de la última fila, cuyo puerto de salida, uno de los positivos y uno de los negativos se encuentra desconectado pues corresponde a la última fila. En primera instancia, se espera que, en dado

caso que se modifique el circuito, no implique un cambio en las conexiones del circuito drásticamente, sino únicamente en la cantidad de neopixeles que se desea usar por fila o columna.

5. Problemas presentados durante la implementación

Al realizar pruebas con código en el proyecto en Tinkercad, en primer lugar, hubo una serie de problemas al usar un buffer de un tamaño lo suficientemente grande para que el usuario ingresase línea por línea el contenido del archivo de texto de tal modo que cada línea representara una de las filas de la matriz, con el fin de que el usuario tenga que hacer el menor esfuerzo posible, no fue viable debido a que al usar tanto memoria en Stack como por memoria dinámica, el buffer solo alcanzaba a captar un poco más de la mitad de la línea que se ingresaba por medio del monitor en serie, por lo cual decidí que tocaba hacer que cada línea representara la información de los 3 colores de 4 neopixeles de una misma fila, de modo que por cada 2 ingresos de datos, se cambia el número de la fila. Dicha información ingresada por el usuario es almacenada en una matriz tridimensional de enteros llamada "pixeles", la cual va a ser indexada con el objetivo de configurar los neopixeles con el método "setPixelColor"que proporciona la librería de Adafruit Neopixel. De esta forma, se configuró todo dentro de una función, además de organizar dentro de una función las instrucciones que se le dan al usuario una vez termina de representar su bandera así como cuando inicializa el programa. De la misma forma, se creó una función para que el usuario tenga la opción de verificar el correcto funcionamiento de los neopixeles de la matriz de 8 x 8. A continuación, la primera versión organizada del código en Tinkercad descrito anteriormente: