

Parcial II 2021-1

Informática 2

Informe de Implementación de la Solución del
Desafío

Juan Diego Cabrera Moncada

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

Índice

1. Clases implementadas	2
2. Esquema de la estructura final de las clases implementadas	2
3. Módulos de código implementado donde interactúan las diferentes clases	2
4. Estructura del circuito montado	2
5. Problemas presentados durante la implementación	2

1. Clases implementadas
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
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 "píxeles", 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:

```
#include <Adafruit_NeoPixel.h>

#define LED_PIN 2

#define LED_COUNT 64

Adafruit_NeoPixel leds(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

void verificacion();
void ingreso_info();
```

```

void main_instruc();

void setup()
{
    leds.begin();
    Serial.begin(9600);
    int pixeles[3][LED_COUNT/8][LED_COUNT/8] = {0};
    main_instruc();
}

void loop()
{
    if(Serial.available()>0){
        int numero = Serial.parseInt();
        if(numero == 1){
            verificacion();
        }
        else if(numero == 2){
            Serial.println("Por_favor ,_ingrese _en _el _monitor _en _serie _el _contenido _del
            Serial.println("de _texto _dado _en _el _primer _programa _usado _ingresando _linea
            Serial.println("(Es _decir _el _archivo _de _texto _que _se _le _menciona _mantener _
            Serial.println("exactamente _como _se _encuentra _escrito . _No _modifique _nada _d
            Serial.println("En _el _monitor _en _serie , _se _le _va _a _mostrar _un _aviso _para _p
            Serial.println("y _pegarla _en _el _recuadro _del _monitor _en _serie , _en _su _parte
            Serial.println("Una _vez _verifique _que _la _linea _ha _sido _copiada _y _pegada _co
            Serial.write("\n\n\n");
            ingreso_info();
            main_instruc();
        }
        else{
            Serial.println("El _numero _ingresado _no _se _encuentra _entre _las _opciones _dad
            Serial.println("Por_favor ,_ingrese _otro _numero.");
            main_instruc();
        }
    }
}

void main_instruc(){
    Serial.println("Estimado _cliente , _para _hacer _uso _de _esta _matriz _de _neopixeles ,
    Serial.println("por_favor , _en _el _monitor _en _serie , _escriba:");
    Serial.println("1 _para _verificar _el _correcto _funcionamiento _de _la _matriz");
    Serial.println("2 _para _representar _en _la _matriz _la _imagen _de _la _bandera _escogi
}

void verificacion(){
    Serial.println("Verificacion _de _Matriz _de _Neopixeles _8x_8");
}

```

```

    for(int i = 0; i < LED_COUNT; i += 1){
        leds.setPixelColor(i, 207, 97, 62);
    }
    leds.show();
    delay(3000);
    Serial.println("Finalizando Verificacion...");
    for(int i = 0; i < LED_COUNT; i += 1){
        leds.setPixelColor(i, 0, 0, 0);
    }
    leds.show();
}

void ingreso_info(){
    int *ptr_cont= NULL;
    ptr_cont = new int;
    *ptr_cont = 0;
    int num_color = 0, f = 0, c = 0;
    while(*ptr_cont <= 15){
        delay(1000);
        if(Serial.available()>0){
            const int length = 48;
            char *fil = new char[length];
            int user = Serial.readBytes(fil, length);
            int valColor[3] = {0};
            for(int i = 0; i<=length-1; i++){
                Serial.println(int(fil[i]));
                if(int(fil[i]) == 44){ // ,
                    num_color++;
                    int col = 0;
                    for(int i = 2; i>=0; i--){
                        int mult = 1;
                        if(valColor[i] != 46){ // No hay mas .
                            col = col + (valColor[i]-48)*mult;
                            mult = mult*10;
                        }
                    }
                    pixeles[num_color][f][c] = col; //Asignacion color
                }
            }
            else if(int(fil[i]) == 59){ // ;
                for(int i = 0; i<=2;i++) valColor[i] = 0;
                num_color = 0;
                c++;
            }
        }
        *ptr_cont++;
    }
    delete [] fil;
}

```

```
        if(*ptr_cont %2!=0){  
            f++;  
            c=0;  
        }  
        *ptr_cont = *ptr_cont + 1;  
    }  
    }  
    delete [] ptr_cont;  
}
```

Referencias