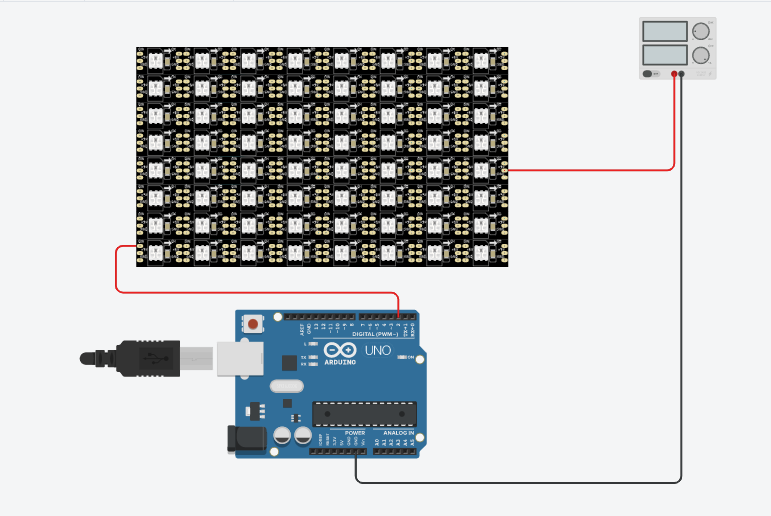
**17/09/2021**

Se hace la lectura del documento y se comienza con el prototipado del circuito

<https://www.tinkercad.com/things/30irhVGsUJx-exquisite-vihelmo/editel?sharecode=QeH9-G-r-W29xbARQEft_GBHeOOqjtv8OLDDFRzYSH8>



**18/09/2021**

Se actualiza un cable que faltaba del circuito

Diagrama

Descripción generada automáticamente

Inicio de la codificación

#include <Adafruit\_NeoPixel.h>

const int pinDatos=2;

const int numPixeles=8\*8;

void setup(){

  //Ya tenemos el serial para recibir la información del imagen de

  //la bandera desde Qt

  Serial.begin(9600);

  pinMode(2,OUTPUT);

}

void loop()

{

  if (Serial.available()>0){

  //Se Establece el menu para el usuario

  Serial.print("BIENVENIDO");

  Serial.print("Ingrese los datos de la bandera");

    //Falta determinar como se orgnizará esta información

  }

**19/09/2021**

Se hace la conexión de la parte trasera de la placa. Se le envía el voltaje y la GND a cada uno de los LEDs. El circuito se ve igual, pero con la segunda foto acercada se ven las conexiones.

Diagrama

Descripción generada automáticamente con confianza mediaDiagrama

Descripción generada automáticamente con confianza media

IDEACIÓN PARA EL INFORME INICIAL Y PRUEBAS DEL CIRCUITO

#include <Adafruit\_NeoPixel.h>

const int pinDatos=2;

const int numPixeles=8\*8;

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

Serial.begin(9600);

pinMode(2,OUTPUT);

}

int main(){

for(int i =0; i<64; i++){

digitalWrite(2,1);

}

}

void loop()

{

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Ingrese los datos de la bandera");

//Falta determinar como se orgnizará esta información

}

}

**20/09/2021**

Actualización de las conexiones del circuito. Se conectaron los voltajes y tierras en serie a los inputs de las tiras de LEDs. Además, se cambió la conexión de los DIN para que queden desde cada output de una tira hacia el input de la siguiente.

Interfaz de usuario gráfica

Descripción generada automáticamente

Actualización del código de prueba. Se intenta probar el encendido de los LEDs pero se tiene que la tira de la parte superior no funciona aún. Se intentó con un color elegido aleatoriamente.

Interfaz de usuario gráfica

Descripción generada automáticamente

CÓDIGO

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 64

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

//Serial.begin(9600);

//pinMode(2,OUTPUT);

leds.begin();

for( int i = 0; i<LED\_COUNT; i++){

leds.setPixelColor(i, 20, 241, 55);

};

leds.show();

}

void loop()

{

/\*

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Ingrese los datos de la bandera");

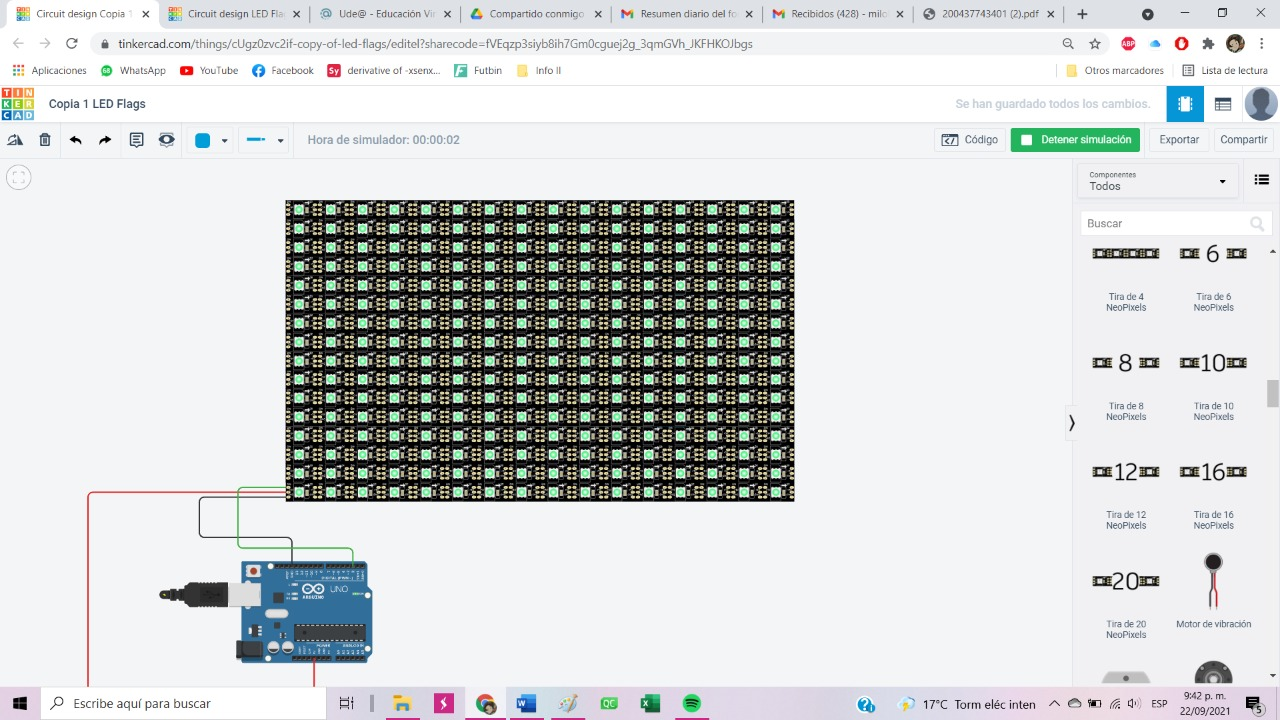
//Falta determinar como se orgnizará esta información

\*/

}

**22/09/2021**

Actualización y ampliación de la matriz



Actualización con la matriz de 16x16

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 256

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

//Serial.begin(9600);

//pinMode(2,OUTPUT);

leds.begin();

for( int i = 0; i<LED\_COUNT; i++){

leds.setPixelColor(i, 20, 241, 55);

};

leds.show();

}

void loop()

{

/\*

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Ingrese los datos de la bandera");

//Falta determinar como se organizará esta información

\*/

}

**23/09/2021**

Se comienza con la organización de la información para Arduino.

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 256

//Definimos el LED NeoPixel

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

//Serial.begin(9600);

//pinMode(2,OUTPUT);

leds.begin();

for( int i = 0; i<LED\_COUNT; i++){

leds.setPixelColor(i, 20, 241, 55);

};

leds.show();

}

void loop()

{

/\*

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Ingrese los datos de la bandera");

//Falta determinar como se organizará esta información

\*/

}

void salida(\*\*\*int arreglo){

int NumLed = 255;

int red =0;

int green =0;

int blue =0;

for(int a=0;a<16;a++){

for(int b=0;b<16;b++){

for(int c=0;c<3;c++){

red = \*(\*(\*(arreglo+a)+b)+0);

green = \*(\*(\*(arreglo+a)+b)+1);

blue = \*(\*(\*(arreglo+a)+b)+2);

leds.setPixelColor(NumLed, red, green, blue);

}

NumLed--;

}

}

}

COMIENZO DE PRUEBAS CON ARREGLOS ALEATORIOS

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 256

//Definimos el LED NeoPixel

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

//Definimos el arreglo que se pega de Qt

int arr[16][16][3]=

{

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

};

int\*\*\* arreglo = arr;

void salida(\*\*\*int arreglo){

int NumLed = 0;

int red =0;

int green =0;

int blue =0;

for(int a=15;a=>0;a--){

for(int b=0;b<16;b++){

for(int c=0;c<3;c++){

red = \*(\*(\*(arreglo+a)+b)+0);

green = \*(\*(\*(arreglo+a)+b)+1);

blue = \*(\*(\*(arreglo+a)+b)+2);

leds.setPixelColor(NumLed, red, green, blue);

}

NumLed++;

}

}

}

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

//Serial.begin(9600);

//pinMode(2,OUTPUT);

leds.begin();

salida(\*\*\*arreglo);

leds.show();

}

void loop()

{

/\*

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Ingrese los datos de la bandera");

//Falta determinar como se organizará esta información

\*/

}

**24/09/2021**

AVANCE CÓDIGO DE LA MATRIZ EN QT

int prueba[12][12]; //tamanio matriz que simula la imagen

int p=1;

for(int i=0 ; i<4;i++){ //este for es para rellenar la matriz, simulando una imagen

for(int j=0; j<4; j++){

prueba[i][j]=p++; //Rellene la matriz que esta simulando la imagen

}

}

int matriz[4][4]; //Matriz de LEDs

int altura=12;

int ancho=12;

AVANCE CÓDIGO ARDUINO TINKERCAD

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 256

//Definimos el LED NeoPixel

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

//Definimos el arreglo que se pega de Qt

int arr[16][16][3]=

{

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

};

int \*\*\*arreglo=arr;

void salida(\*\*\*int arreglo, Adafruit\_NeoPixel leds){

int NumLed = 0;

int red =0;

int green =0;

int blue =0;

for(int a=15;a=>0;a--){

for(int b=0;b<16;b++){

for(int c=0;c<3;c++){

red = \*(\*(\*(arreglo+a)+b)+0);

green = \*(\*(\*(arreglo+a)+b)+1);

blue = \*(\*(\*(arreglo+a)+b)+2);

leds.setPixelColor(NumLed, red, green, blue);

}

NumLed++;

}

}

}

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

Serial.begin(9600);

pinMode(2,OUTPUT);

}

void loop()

{

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Presione uno para imprimir la información ingresada");

Serial.flush();

int caso = Serial.parseInt();

if(caso==1){

leds.begin();

salida(\*\*\*arreglo);

leds.show();

}

}

**25/09/2021**

Nueva versión Arduino

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 256

//Definimos el LED NeoPixel

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

//Definimos el arreglo que se pega de Qt

int arr[16][16][3]=

{

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

};

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

Serial.begin(9600);

pinMode(2,OUTPUT);

}

void loop()

{

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Presione uno para imprimir la información ingresada");

Serial.flush();

int caso = Serial.parseInt();

if(caso==1){

leds.begin();

for(int a=15;a=>0;a--){

for(int b=0;b<16;b++){

for(int c=0;c<3;c++){

red = arr[a][b][c];

green = arr[a][b][c+1];

blue = arr[a][b][c+2];

leds.setPixelColor(NumLed, red, green, blue);

}

NumLed++;

}

}

leds.show();

}

}

**CODIGO SIN ERRORES (SOLUCION ERROR CONDICIONALES)**

#include <Adafruit\_NeoPixel.h>

//Icluimos el pin de control de la salida

#define LED\_PIN 2

//Incluimos el numero de pixeles a controlar

#define LED\_COUNT 256

//Definimos el LED NeoPixel

Adafruit\_NeoPixel leds(LED\_COUNT, LED\_PIN, NEO\_GRB + NEO\_KHZ800);

//Definimos el arreglo que se pega de Qt

int arr[16][16][3]=

{

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

{{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200},{20,47,200}},

};

void setup(){

//Ya tenemos el serial para recibir la información del imagen de

//la bandera desde Qt

Serial.begin(9600);

pinMode(2,OUTPUT);

}

void loop()

{

if (Serial.available()>0){

//Se Establece el menu para el usuario

Serial.print("BIENVENIDO");

Serial.print("Presione uno para imprimir la información ingresada");

Serial.flush();

int caso = Serial.parseInt();

if(caso==1){

int NumLed = 0;

int red =0;

int green =0;

int blue =0;

leds.begin();

for(int a=15;a>0 || a==0;a--){

for(int b=0;b<16;b++){

for(int c=0;c<3;c++){

red = arr[a][b][c];

green = arr[a][b][c+1];

blue = arr[a][b][c+2];

leds.setPixelColor(NumLed, red, green, blue);

}

NumLed++;

}

}

leds.show();

}

}

}

.