#include <Adafruit\_NeoPixel.h>

#define PIAN 1

#define PIBN 2

#define PICN 3

#define PIDN 4

#define PIEN 5

#define PIFN 6

#define PIGN 7

#define PIHN 8

#define PIIN 9

#define PIJN 10

#define PIKN 11

#define PILN 12

#define PIMN 13

// input pin Neopixel is attached to

#define NUMPIXELS 12 // number of neopixels in strip

Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(NUMPIXELS, PIAN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels1 = Adafruit\_NeoPixel(NUMPIXELS, PIBN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels2 = Adafruit\_NeoPixel(NUMPIXELS, PICN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels3 = Adafruit\_NeoPixel(NUMPIXELS, PIDN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels4 = Adafruit\_NeoPixel(NUMPIXELS, PIEN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels5 = Adafruit\_NeoPixel(NUMPIXELS, PIFN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels6 = Adafruit\_NeoPixel(NUMPIXELS, PIGN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels7 = Adafruit\_NeoPixel(NUMPIXELS, PIHN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels8 = Adafruit\_NeoPixel(NUMPIXELS, PIIN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels9 = Adafruit\_NeoPixel(NUMPIXELS, PIJN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels10 = Adafruit\_NeoPixel(NUMPIXELS, PIKN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels11 = Adafruit\_NeoPixel(NUMPIXELS, PILN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels12 = Adafruit\_NeoPixel(NUMPIXELS, PIMN, NEO\_GRB + NEO\_KHZ800);

int delayval = 100; // timing delay in milliseconds

int redColor = 0;

int greenColor = 0;

int blueColor = 0;

void setup() {

// Initialize the NeoPixel library.

pixels.begin();

}

void loop() {

pixels.setPixelColor(0, pixels.Color(255, 0, 0));//Sixers Colors

pixels.setPixelColor(1, pixels.Color(0, 0, 255));

pixels.setPixelColor(2, pixels.Color(255, 17, 95));//brithstone ruby

pixels.setPixelColor(3, pixels.Color(0, 150, 200));//eye light blue

pixels.setPixelColor(4, pixels.Color(165, 100, 75));//hair brownish

pixels.setPixelColor(5, pixels.Color(255, 0, 0));//favorite color red

pixels.setPixelColor(6, pixels.Color(255, 0, 0));

pixels.setPixelColor(7, pixels.Color(255, 0, 0));

pixels.setPixelColor(8, pixels.Color(255, 0, 0));

pixels.setPixelColor(9, pixels.Color(255, 0, 0));

pixels.setPixelColor(10, pixels.Color(255, 0, 0));

pixels.setPixelColor(11, pixels.Color(255, 0, 0));

pixels.setPixelColor(12, pixels.Color(255, 0, 0));

pixels.setPixelColor(13, pixels.Color(255, 0, 0));

pixels.show();

//setColor

//for (int i=0; i < NUMPIXELS; i++) {

// pixels.Color takes RGB values, from 0,0,0 up to 255,255,255

// pixels.setPixelColor(i, pixels.Color(redColor, greenColor, blueColor));

// This sends the updated pixel color to the hardware.

// pixels.show();

// Delay for a period of time (in milliseconds).

// delay(delayval);

// }

//}

// setColor()

// picks random values to set for RGB

//void setColor(){

//redColor = random(0, 255);

//greenColor = random(0,255);

//blueColor = random(0, 255);

}

**DAY1 \*\*\*\*OUTPUT VERIFIED LIBRARY**

#include <Adafruit\_NeoPixel.h>

#define PIAN 1

#define PIBN 2

#define PICN 3

#define PIDN 4

#define PIEN 5

#define PIFN 6

#define PIGN 7

#define PIHN 8

#define PIIN 9

#define PIJN 10

#define PIKN 11

#define PILN 12

#define PIMN 13

// input pin Neopixel is attached to

#define NUMPIXELS 12 // number of neopixels in strip

Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(NUMPIXELS, PIAN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels1 = Adafruit\_NeoPixel(NUMPIXELS, PIBN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels2 = Adafruit\_NeoPixel(NUMPIXELS, PICN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels3 = Adafruit\_NeoPixel(NUMPIXELS, PIDN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels4 = Adafruit\_NeoPixel(NUMPIXELS, PIEN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels5 = Adafruit\_NeoPixel(NUMPIXELS, PIFN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels6 = Adafruit\_NeoPixel(NUMPIXELS, PIGN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels7 = Adafruit\_NeoPixel(NUMPIXELS, PIHN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels8 = Adafruit\_NeoPixel(NUMPIXELS, PIIN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels9 = Adafruit\_NeoPixel(NUMPIXELS, PIJN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels10 = Adafruit\_NeoPixel(NUMPIXELS, PIKN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels11 = Adafruit\_NeoPixel(NUMPIXELS, PILN, NEO\_GRB + NEO\_KHZ800);

Adafruit\_NeoPixel pixels12 = Adafruit\_NeoPixel(NUMPIXELS, PIMN, NEO\_GRB + NEO\_KHZ800);

int delayval = 100; // timing delay in milliseconds

int redColor = 0;

int greenColor = 0;

int blueColor = 0;

void setup() {

// Initialize the NeoPixel library.

pixels.begin();

}

void loop() {

pixels.setPixelColor(0, pixels.Color(255, 0, 0));//Sixers Colors

pixels.setPixelColor(1, pixels.Color(0, 0, 255));

pixels.setPixelColor(2, pixels.Color(255, 17, 95));//brithstone ruby

pixels.setPixelColor(3, pixels.Color(0, 150, 200));//eye light blue

pixels.setPixelColor(4, pixels.Color(165, 100, 75));//hair brownish

pixels.setPixelColor(5, pixels.Color(255, 0, 0));//favorite color red

pixels.setPixelColor(6, pixels.Color(255, 0, 0));

pixels.setPixelColor(7, pixels.Color(255, 0, 0));

pixels.setPixelColor(8, pixels.Color(255, 0, 0));

pixels.setPixelColor(9, pixels.Color(255, 0, 0));

pixels.setPixelColor(10, pixels.Color(255, 0, 0));

pixels.setPixelColor(11, pixels.Color(255, 0, 0));

pixels.setPixelColor(12, pixels.Color(255, 0, 0));

pixels.setPixelColor(13, pixels.Color(255, 0, 0));

pixels.show();

//setColor

//for (int i=0; i < NUMPIXELS; i++) {

// pixels.Color takes RGB values, from 0,0,0 up to 255,255,255

// pixels.setPixelColor(i, pixels.Color(redColor, greenColor, blueColor));

// This sends the updated pixel color to the hardware.

// pixels.show();

// Delay for a period of time (in milliseconds).

// delay(delayval);

// }

//}

// setColor()

// picks random values to set for RGB

//void setColor(){

//redColor = random(0, 255);

//greenColor = random(0,255);

//blueColor = random(0, 255);

}