```
/*************************************************************
 * LED-Farbwechsel-Leuchte (19.08.2021)
 *
 * Gedacht für einen ATTiny 85-20. Jede Leuchte bekommt einen
 * eigenen Controller
 *
 * Grundlage = Strandtest-Demo included by Adafruit Neopixel Library
 * Angepasst an Projekt = J.Reinert
 *
 * Benötigte Library: 'Adafruit Neopixel'
 *
 * Data In = nothing
 * Data Out = WS2812b (Pin3 am ATTiny-Chip)
 *
*************************************************************/

#define ATTINY                // aktivieren, wenn für ATTiny kompiliert werden soll
//#define Theater_marquee      // aktivieren, wenn Marquee-Effekt gewünscht ist

#include<Adafruit_NeoPixel.h>

#define LED_COUNT 29          // How many NeoPixels are attached to the Arduino? (48/29)

#ifdef ATTINY                 // Definitionen für ATTiny 85-20
                              // nicht genutzte Anschlüsse werden als Ausgang geschaltet

  #define LED_PIN  4          // WS2812-Datapin an ATTiny (Pin3 am Chip)
  #define RANDOM_INIT_PIN 3   // Analog-Input for init Random Generator (Pin2 am Chip)
  #define DUMMY2 2            // ungenutzt, wird Ausgang
  #define DUMMY1 1            // ungenutzt, wird Ausgang
  #define DUMMY0 0            // ungenutzt, wird Ausgang

#else                         // Definitionen für Arduino
```

```
  #define LED_PIN  13           // WS2812-Datapin an Arduino UNO/NANO
  #define RANDOM_INIT_PIN 0     // Analog-Input for init Random Generator
#endif

int waittime = 1000;
byte next_color = 1;
byte max_brightnes = 180;

// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_RGB + NEO_KHZ800);
// Argument 1 = Number of pixels in NeoPixel strip
// Argument 2 = Arduino pin number (most are valid)
// Argument 3 = Pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
//   NEO_RGBW    Pixels are wired for RGBW bitstream (NeoPixel RGBW products)


// setup() function -- runs once at startup --------------------------------

void setup() {
  #ifdef ATTINY
    pinMode(DUMMY0, OUTPUT);
    pinMode(DUMMY1, OUTPUT);
    pinMode(DUMMY2, OUTPUT);
  #endif
  strip.begin();                           // INITIALIZE NeoPixel strip object (REQUIRED)
  strip.show();                            // Turn OFF all pixels ASAP
  strip.setBrightness(max_brightnes);      // Set BRIGHTNESS to about 1/5 (max = 255)
  randomSeed(analogRead(RANDOM_INIT_PIN)); // Init Random Generator
}
```

```
// loop() function -- runs repeatedly as long as board is on ---------------

void loop() {
  // Fill along the length of the strip in various colors...
  waittime = random(5000, 10000);             // Wartezeit 5 bis 10 Sekunden
  next_color = random(0, 93);                 // Welche Farbe als nächstes?

  if (next_color>=0 && next_color<20) {
    colorWipe(strip.Color(255,   0,   0), 50); // Red
  }

  if (next_color>=20 && next_color<40) {
    colorWipe(strip.Color(  0, 255,   0), 50); // Green
  }

  if (next_color>=40 && next_color<60) {
    colorWipe(strip.Color(  0,   0, 255), 50); // Blue
  }

  if (next_color>=60 && next_color<80) {
    colorWipe(strip.Color(191,   0, 249), 50); // Lila
  }

  if (next_color>=80 && next_color<90) {
    colorWipe(strip.Color(255, 137,   0), 50); // Orange
  }

  // Special Effects without Waittime
  if (next_color>=90 && next_color<93) {
    waittime=1;

#ifdef Theater_marquee
    theaterChase(strip.Color(127, 127, 127), 120, random(30,50)); // White, half brightness
#else
    pulseBrightnes(5,random(6,15));
```

```
#endif

  }
  delay(waittime);
}


// Some functions of our own for creating animated effects -----------------

// Fill strip pixels one after another with a color. Strip is NOT cleared
// first; anything there will be covered pixel by pixel. Pass in color
// (as a single 'packed' 32-bit value, which you can get by calling
// strip.Color(red, green, blue) as shown in the loop() function above),
// and a delay time (in milliseconds) between pixels.

void colorWipe(uint32_t color, int wait) {
  for(int i=0; i<strip.numPixels(); i++) { // For each pixel in strip...
    strip.setPixelColor(i, color);         //  Set pixel's color (in RAM)
    strip.show();                          //  Update strip to match
    delay(wait);                           //  Pause for a moment
  }
}

#ifdef Theater_marquee
// Theater-marquee-style chasing lights. Pass in a color (32-bit value,
// a la strip.Color(r,g,b) as mentioned above), and a delay time (in ms)
// between frames.
void theaterChase(uint32_t color, int wait, int repeat) {
  for(int a=0; a<repeat; a++) {  // Repeat
    for(int b=0; b<3; b++) { //  'b' counts from 0 to 2...
      strip.clear();         //   Set all pixels in RAM to 0 (off)
      // 'c' counts up from 'b' to end of strip in steps of 3...
      for(int c=b; c<strip.numPixels(); c += 3) {
        strip.setPixelColor(c, color); // Set pixel 'c' to value 'color'
      }
```

```
      strip.show(); // Update strip with new contents
      delay(wait);  // Pause for a moment
    }
  }
}

#else

// Aktuelle Farbe mehrmals dimmen und wieder aufhellen (funktioniert nur mit Grundfarben richtig!)
void pulseBrightnes(int wait,int repeat){
 byte min_Brightnes=max_brightnes/5;
  for(int a=0; a<repeat; a++) {           // Repeat
    for(int b=max_brightnes; b>min_Brightnes; b--) {
      strip.setBrightness(b);
      strip.show();
      delay(wait);
    }
    for(int b=min_Brightnes; b<max_brightnes; b++) {
      strip.setBrightness(b);
      strip.show();
      delay(wait);
    }
  }
 strip.setBrightness(max_brightnes);
  strip.show();
}

#endif
```