IKEA Star Hack

Kaum zu glauben, aber Weihnachten rückt immer näher. Der Weihnachtsurlaub steht kurz bevor und Sie möchten, natürlich in der Freizeit, ein bisschen dem Hobby **Maker** nachgehen. Was gibt es dafür Besseres, als einen kleinen IKEA-Hack umzusetzen. Heute wollen wir deswegen der IKEA Lampenschirm STRÅLA etwas modifizieren und zu einer anschaulichen Fensterdekoration umzufunktionieren?

Benötigte Hard- und Software

Die Hardware für diesen Versuchsaufbau ist relativ einfach, siehe Tabelle 1.

Pos	Anzahl	Bauteil	Link
1	1	Nano V3.0 CH340	https://www.az-delivery.de
2	1	Breadboard, am bestes als Kit	https://az-delivery.de
3	1	LED Ring 5V RGB WS2812B 12-Bit 37mm	https://az-delivery.de
4	1	Batterieclip 9-Volt I-Typ Clip Snap	https://az-delivery.de

Tabelle 1: Hardwareteile für Cyclone mit WS2812B-LEDring

Zusätzlich brauchen Sie noch eine 9V-Blockbaterie, damit später der Hack auch Strom bekommt. Für den Hack habe ich eigens ein kleines Gehäuse für den 3D-Drucker entworfen, man kann aber auch mit ein bisschen Pappe und Kleber ein eigenes Gehäuse basteln.

Zusätzlich müssen Sie sich noch bei Ikea den Lampenschirm STRÅLA organisieren, da es ja um den Hack eben dieses Lampenschirms geht.

Die benötigte Software für dieses Projekt ist überschaubar:

- Arduino IDE (https://www.arduino.cc/en/Main/Software), hier am besten die aktuelle Version herunterladen
- Die Bibliothek **Adafruit NeoPixel**, bei Erstellung dieses Beitrag in Version 1.10.0 verfügbar, mit allen Abhängigkeiten, die Sie über die Bibliotheksverwaltung, siehe https://www.az-delivery.de/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/arduino-ide-programmieren-fuer-einsteiger-teil-1 Abschnitt Bibliotheksverwaltung, installieren müssen.

Der Aufbau

Die Schaltung für den kleinen IKEA-Hack sollte kompakt, aber funktional sein, siehe Abbildung 1.

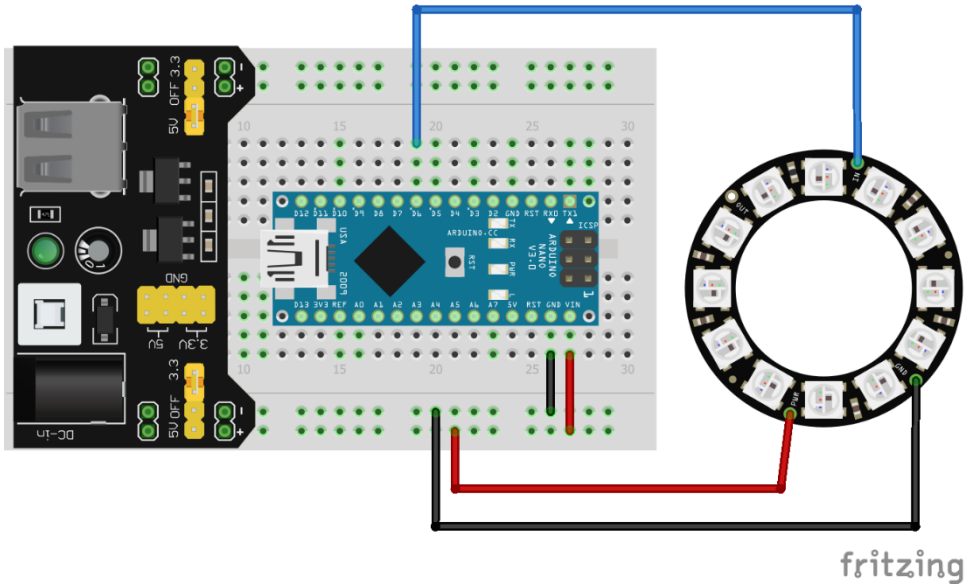

Abbildung 1: Aufbau der Schaltung

Bedenken Sie bitte, dass der LED-Ring eine 5V-Spannungsverorgung braucht. Die Masse des Nano V3.0 sollten sie mit dem LED-Ring koppeln.

Der Quellcode

Wie Sie es von uns gewohnt sind, ist der Quellcode wieder mit vielen Kommentaren versehen, siehe Code 1. Gleichzeitig ist dieser in der loop-Funktion so gestaltet, dass Sie weitere Lichtanimationen hinzufügen können.

```
//------------------------------------------------------
// Ikea Star Hack for Nano V3.0
// Autor:   Joern Weise
// License: GNU GPl 3.0
// Created: 07. Dec. 2021
// Update:
//------------------------------------------------------
//Include libraries
#include <Adafruit_NeoPixel.h>

//Defines
#define LED_PIN    6        //Pin on Arduino Nano D6
#define LED_COUNT 12        //Num of LEDs
#define TIME_CHANGE 120000 //Time to switch mode in ms 2 min = 120000ms
#define NUM_MODE 5          //Max Num of LED-Modes

//Init NeoPixel-strip
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);  //Init LED-Ring

//Some global vars
int iShowMode = -1;   //Change to valid mode to get a fix mode!!
int iCurrentMode = 0; //To switch through the modes
unsigned long ulLastTime = 0; //For time-checking

/*
  ======================================================================
  Function:     setup
```

```
  Returns:      void
  Description:  Needed setup routine for MicroController
  ======================================================================
*/
void setup() {
  Serial.begin(115200);
  strip.begin();           // INITIALIZE NeoPixel strip object (REQUIRED)
  strip.show();            // Turn OFF all pixels ASAP
  strip.setBrightness(50); // Set BRIGHTNESS to about 1/5 (max = 255)
  randomSeed(analogRead(A0)); //Make random more randome

}

/*
  ======================================================================
  Function:     loop
  Returns:      void
  Description:  Needed loop routine for MicroController
  ======================================================================
*/
void loop() {
  if(iShowMode == -1)
  {
    if(iCurrentMode >= NUM_MODE)
      iCurrentMode = 0;
      Serial.println("Current mode: " + String(iCurrentMode));
  }
  else
    iCurrentMode = iShowMode;

  //Simple state machine to go through the modes
  //More modes can be add, keep in mind to increase NUM_MODE
  switch(iCurrentMode)
  {
    case 0:
      Serial.println("Case: 0");
      randomColor();
      break;
    case 1:
      Serial.println("Case: 1");
      rainbow(10);
      break;
    case 2:
      Serial.println("Case: 2");
      theaterChaseRainbow(50);
      break;
    case 3:
      Serial.println("Case: 3");
      theaterChase(strip.Color(random(0,255), random(0,255), random(0,255)),250);
      break;
    case 4:
      Serial.println("Case: 4");
      shift(150);
      break;
    default:
      Serial.println("Case: default");
```

```
      randomColor();
      break;
    }
  }

  /*
    ================================================================
    Function:     randomColor
    Returns:      void
    Description:  Color a random LED with a random color
    ================================================================
  */
  void randomColor()
  {
    static unsigned long ulRandomLast = millis();
    ulLastTime = millis();
    Serial.println("ulLastTime: " + String(ulLastTime));
    strip.clear();
    while((millis() - ulLastTime) < TIME_CHANGE)
    {
      if(millis() - ulRandomLast > 250)
      {
        int iRed    = random(0,255);
        int iGreen  = random(0,255);
        int iBlue   = random(0,255);
        int iLED    = random(0,int(LED_COUNT) );
        strip.setPixelColor(iLED, strip.Color(iRed, iGreen, iBlue));         //  Set pixel's color (in
  RAM)
        strip.show();
        ulRandomLast = millis();
      }
    }
    Serial.println("Exit while-loop");
    iCurrentMode++;
  }

  /*
    ================================================================
    Function:     rainbow
    Returns:      void
    IN wait:      Time to wait for next LED change
    Description:  Color the ring like a rainbow
    ================================================================
  */
  void rainbow(int wait)
  {
    static unsigned long ulRainbowLast = millis();
    ulLastTime = millis();
    Serial.println("ulLastTime: " + String(ulLastTime));
    static int iFirstPixelHue = 0;
    strip.clear();
    while((millis() - ulLastTime) < TIME_CHANGE)
    {
      if(millis() - ulRainbowLast > wait)
      {
        iFirstPixelHue += 256;
```

```
      if(iFirstPixelHue >= 65536)
        iFirstPixelHue = 0;
      for(int i=0; i<strip.numPixels(); i++)
      {
        int iPixelHue = iFirstPixelHue + (i / strip.numPixels());
        strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(iPixelHue)));
      }
      ulRainbowLast = millis();
      strip.show();
    }
  }
  Serial.println("Exit while-loop");
  iCurrentMode++;
}

/*
  ====================================================================
  Function:     theaterChaseRainbow
  Returns:      void
  IN wait:      Time to wait for next LED change
  Description:  Circle rainbow-color around the LED-ring
  ====================================================================
*/
void theaterChaseRainbow(int wait)
{
  static unsigned long theaterChaseRainbow = millis();
  ulLastTime = millis();
  Serial.println("ulLastTime: " + String(ulLastTime));
  static int iFirstPixelHue = 0;
  strip.clear();
  static int iFirstPixel = 0;
  while((millis() - ulLastTime) < TIME_CHANGE)
  {
    if(millis() - theaterChaseRainbow > wait)
    {
      strip.clear();
      for(int iCount = iFirstPixel; iCount < strip.numPixels(); iCount += 3)
      {
        int      hue   = iFirstPixelHue + iCount * 65536L / strip.numPixels();
        uint32_t color = strip.gamma32(strip.ColorHSV(hue)); // hue -> RGB
        strip.setPixelColor(iCount, color); // Set pixel 'c' to value 'color'
      }
      theaterChaseRainbow = millis();
      strip.show();
      iFirstPixelHue += 65536 / 90;
      iFirstPixel++ ;
      if(iFirstPixel >= 3)
        iFirstPixel = 0;
    }
  }
  Serial.println("Exit while-loop");
  iCurrentMode++;
}

/*
  ====================================================================
```

```
  Function:     theaterChase
  Returns:      void
  IN wait:      Time to wait for next LED change
  IN color:     Random color for the LED-Ring
  Description:  Circle random-color around the LED-ring
  ================================================================
*/
void theaterChase(uint32_t color, int wait)
{
  static unsigned long theaterChase = millis();
  ulLastTime = millis();
  Serial.println("ulLastTime: " + String(ulLastTime));
  static int iFirstPixel;
  iFirstPixel = 0;
  strip.clear();
  while((millis() - ulLastTime) < TIME_CHANGE)
  {
    if(millis() - theaterChase > wait)
    {
      strip.clear();
      for(int iCount = iFirstPixel; iCount < strip.numPixels(); iCount += 3)
      {
        strip.setPixelColor(iCount, color); // Set pixel 'c' to value 'color'
      }
      theaterChase = millis();
      strip.show();
      iFirstPixel++ ;
      if(iFirstPixel >= 3)
        iFirstPixel = 0;
    }
  }
  Serial.println("Exit while-loop");
  iCurrentMode++;
}

/*
  ================================================================
  Function:     shift
  Returns:      void
  IN wait:      Time to wait for next LED change
  Description:  Color each LED-ring step by step to random color
  ================================================================
*/
void shift(int wait)
{
  static unsigned long ulShift = millis();
  ulLastTime = millis();
  Serial.println("ulLastTime: " + String(ulLastTime));
  static int iPixel;
  iPixel = 0;
  static int iShifter = 1;
  strip.clear();
  static uint32_t color = strip.Color(random(0,255), random(0,255), random(0,255));
  while((millis() - ulLastTime) < TIME_CHANGE)
  {
    if(millis() - ulShift > wait)
```

```
    {
      strip.setPixelColor(iPixel, color); // Set pixel 'c' to value 'color'
      ulShift = millis();
      strip.show();

      iPixel+= iShifter;
      if(iPixel >= (LED_COUNT - 1) )
      {
        color = strip.Color(random(0,255), random(0,255), random(0,255));
        iShifter = -1;
      }
      else if (iPixel <= 0)
      {
        iShifter = 1;
        color = strip.Color(random(0,255), random(0,255), random(0,255));
      }
    }
  }
  Serial.println("Exit while-loop");
  iCurrentMode++;
}
```

Code 1: Code für den Ikea Hack

Nutzen Sie für weitere Lichtanimationen z.B. die Funktion **randomColor()** die das Prinzip einer Animation einfach zeigt. Da ich persönlich kein Freund der Funktion delay() bin, nutze ich eine Subtraktion der aktuell laufenden Millisekunden zu dem zuletzt gespeicherten Wert. Wird eine Grenze überschritten, wird die Funktion weiter ausgeführt oder eine neue Lichtanimation angefangen.

Zusammenbau

Damit der Stern auch am Ende entsprechend leuchtet, habe ich extra noch ein kleines Gehäuse entworfen, siehe Abbildung 2. Das Gehäuse finden Sie auf meiner GitHub-Seite, damit Sie es selbst slicen können.

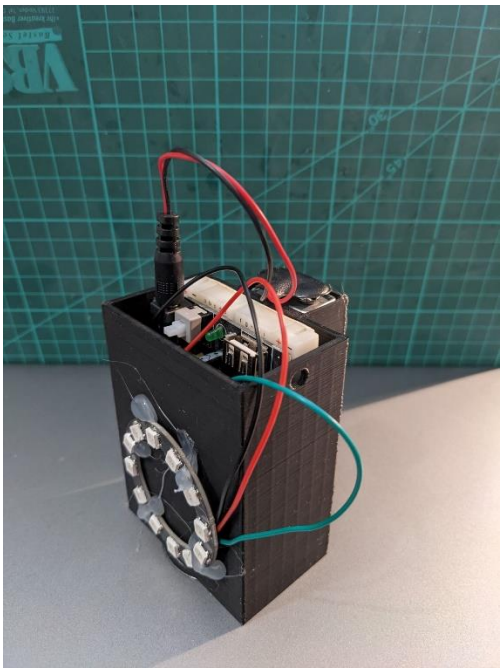

Abbildung 2: Gehäuse für die Technik

Da ich leider keine passenden Schrauben zum Befestigen vom LED-Ring hatte, tat es an dieser Stelle auch eine gute Portion Heißkleber. Im Anschluss noch das Gehäuse mit dem Halter des Sterns verbinden, zum Beispiel einfach mit Kabelbindern befestigen und fertig ist der leuchtende Stern, siehe Abbildung 3.

Abbildung 3: Der fertige Stern

Insgesamt dauert der Zusammenbau, ohne das Drucken des Gehauses, ungefähr 30 Minuten. Mit ein bisschen Basteltalent ist auch schnell ein eigenes Gehäuse aus Pappe für die Technik gebastelt.

Zusammenfassung

Ich wünsche Ihnen eine frohe Weihnacht und eine frohe Weihnachtszeit. Zwar hat uns Corona immer noch im Griff, das sollte aber den Maker-Trieb nicht bremsen. Wie schon beim Quellcode erwähnt, haben Sie die Möglichkeit eigene Effekte hinzuzufügen oder einfach alles so zu lassen. Denkbar sind auch LED-Schläuche, die um ein Gehäuse gewickelt werden. Alles ist möglich, lassen Sie Ihrer Kreativität einfach freien Lauf.

Weitere Projekte für Az-Delivery von mir, finden Sie unter https://github.com/M3taKn1ght/Blog-Repo.