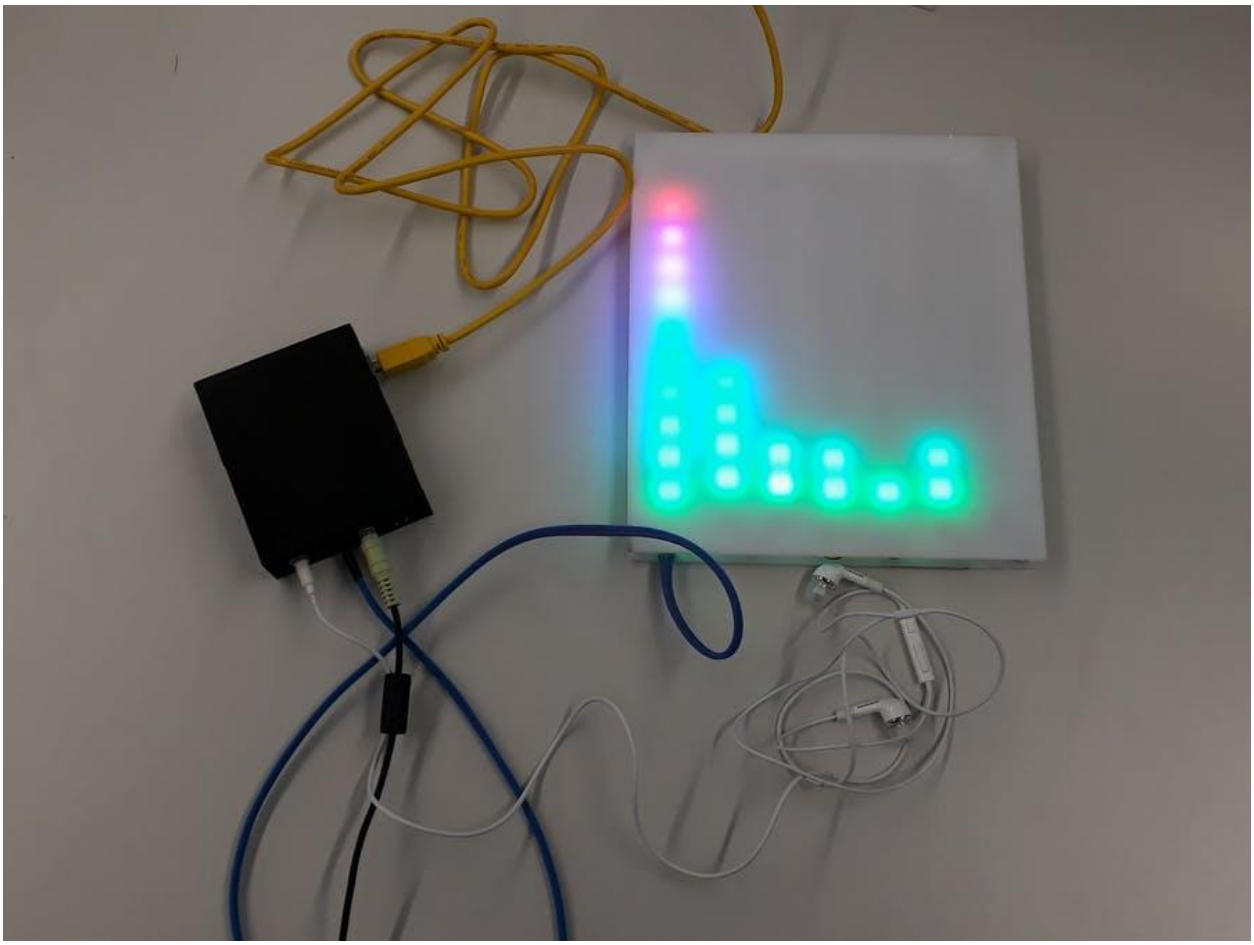
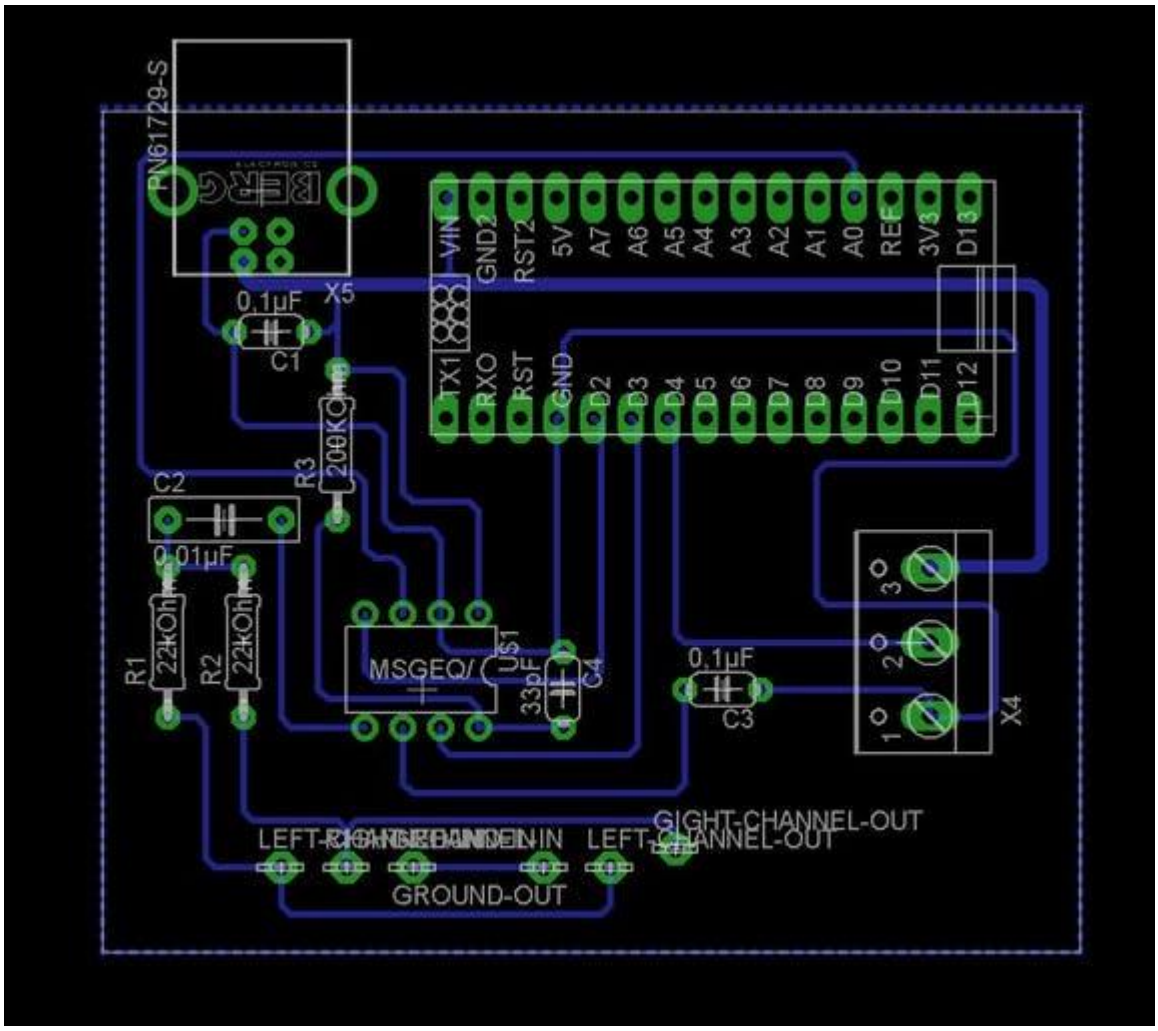


Spectrum Analyzer with RGB LEDs

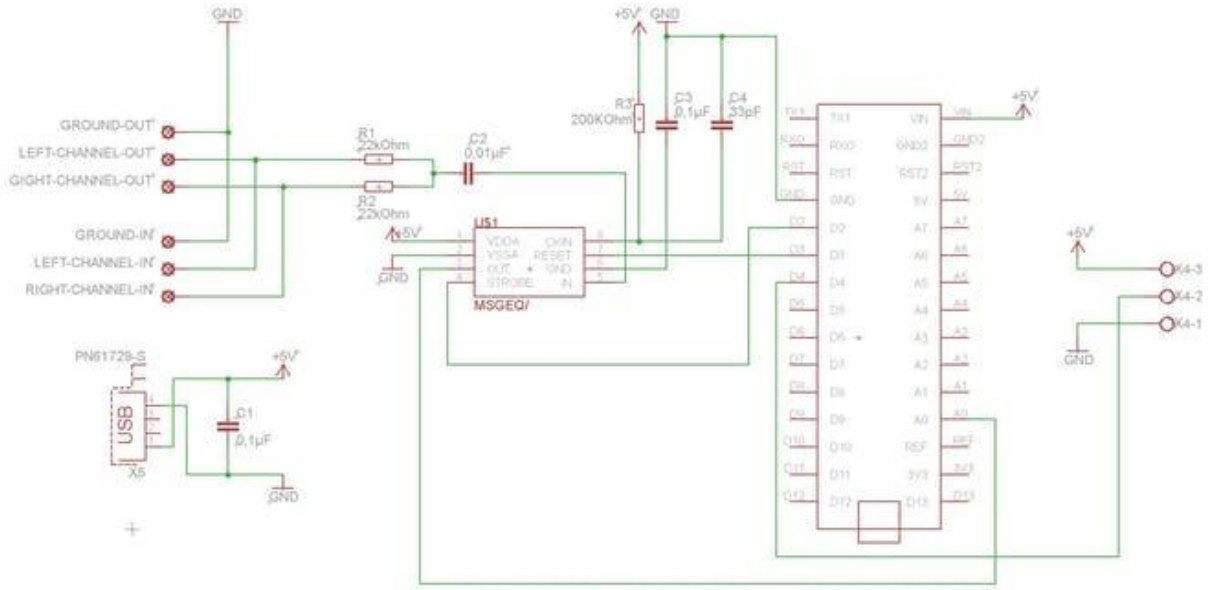
A short and simpel project for visiualize Audio signals with LEDs.

About this project

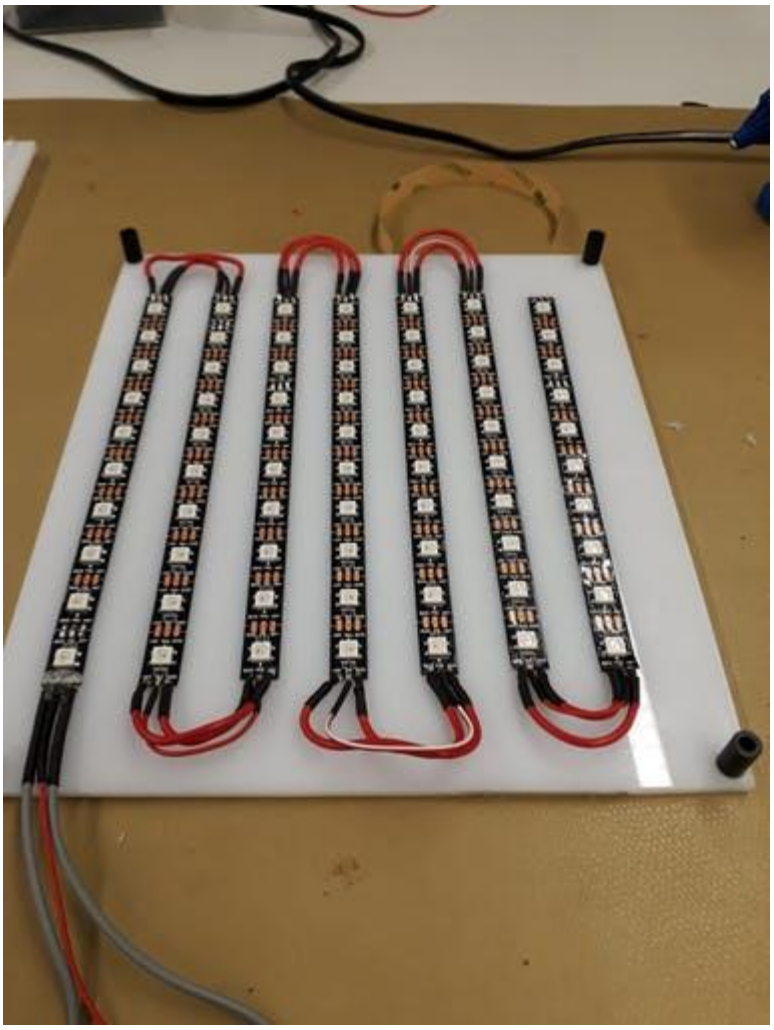
The project is to visualize an audio signal with a RGB Led Stripe. The used MSGE7 have 7 bandpasses, thats why I made 7 lines with the LED stripes.



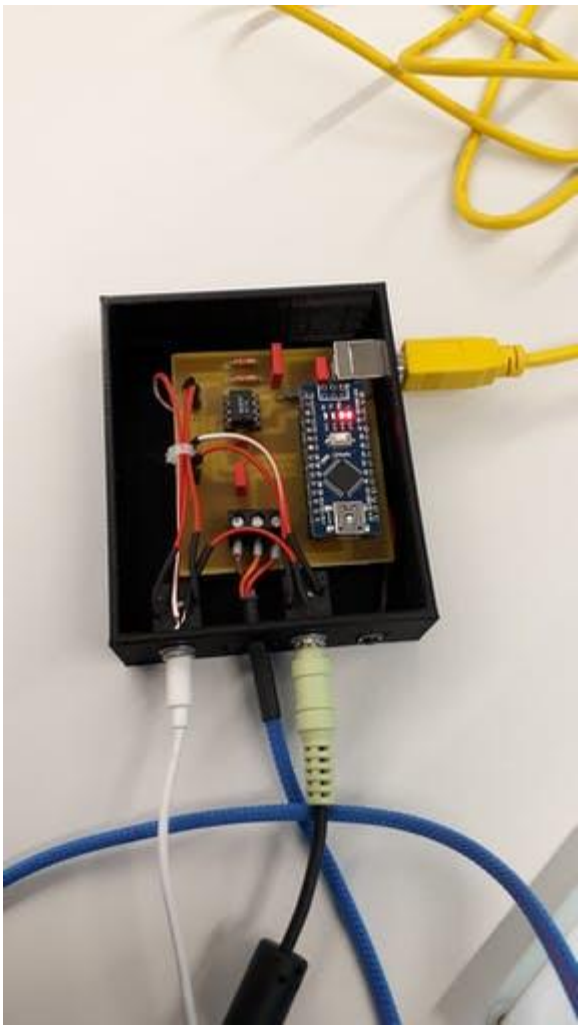
For the Power supply I used a normal USB Type B socket. On the Bottom are the connectors for the AUX wire and X4 is for the LED-Stripe.



MSGEQ7, Arduino Nano and the jumper are fixed on one circut board.



I just soldered the single stripes behid each other and glued them on transperend PE.



The circuit board is in a 3D printed case with Aux sockets for input and output signal.

Code

```
//Arduino Spectrum Analyzer with RGB-LED Stripe and MSGEQ7 by HeinzKetchup
// declarations for the Neopixel by Adafruit
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif
#define PIN 4 // Pin for the RGB Stripe
#define NUMPIXELS 70  // Number of Pixels
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);
int strobePin = 2;  // Strobe Pin on the MSGEQ7
int resetPin = 3; // Reset Pin on the MSGEQ7
int outPin = A0;  // Output Pin on the MSGEQ7
int level[7]; // An array to hold the values from the 7 frequency bands
int l;
uint32_t aus =  pixels.Color(0,0,0);
uint32_t gr =  pixels.Color(0,200,0);
uint32_t grb =  pixels.Color(0,160,40);
uint32_t grbl =  pixels.Color(0,120,80);
uint32_t gbl =  pixels.Color(0,80,120);
uint32_t bl =  pixels.Color(0,40,160);
uint32_t blr =  pixels.Color(0,0,200);
uint32_t blro =  pixels.Color(40,0,160);
uint32_t bro =  pixels.Color(80,0,120);
uint32_t ro =  pixels.Color(120,0,80);
uint32_t rog =  pixels.Color(160,0,0);

void setup()
{
  Serial.begin(9600);
  pinMode(strobePin, OUTPUT); // Define our pin modes
  pinMode(resetPin,  OUTPUT);
  pinMode(outPin, INPUT);
  pinMode(3,OUTPUT);
  digitalWrite(resetPin, LOW);  // Create an initial state for our pins
  digitalWrite(strobePin, LOW);
  delay(1);
  digitalWrite(resetPin, HIGH); // Reset the MSGEQ7 as per the datasheet
timing diagram
  delay(1);
  digitalWrite(resetPin, LOW);
  digitalWrite(strobePin, HIGH);
  delay(1);
  pixels.begin(); // enables Adafruit Neopixels
  pixels.show();  // reset Pixels
  for (int i = 0; i < 70; i++)
  {
    int ii = i-5; //snake effect at the start
    pixels.setPixelColor(i, gr);
    pixels.setPixelColor(ii, aus);
    pixels.show();
    delay(20);
  }
}
```

```
void LEDaus1(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
   for (int i=a; i <=k; i++)
   {
    pixels.setPixelColor(i, aus);
    pixels.show();
   }
}
void LEDaus2(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
   for (int i=a; i >=k; i--)
   {
    pixels.setPixelColor(i, aus);
    pixels.show();
   }
}
void LED0(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k,
uint32_t gr)
{
 pixels.setPixelColor(a, gr);
 for (int i=b; i <= k; i++) pixels.setPixelColor(i, aus);
}
void LED1(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 for (int i=k; i <= c; i++) pixels.setPixelColor(i, aus);
}
void LED2(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 for (int i=d; i <= k; i++) pixels.setPixelColor(i, aus);
}
void LED3(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 pixels.setPixelColor(d, gbl);
 for (int i=k; i <= e; i++) pixels.setPixelColor(i, aus);
}
void LED4(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 pixels.setPixelColor(d, gbl);
 pixels.setPixelColor(e, bl);
 for (int i=f; i <= k; i++) pixels.setPixelColor(i, aus);
}
void LED5(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
```

```
 pixels.setPixelColor(d, gbl);
 pixels.setPixelColor(e, bl);
 pixels.setPixelColor(f, blr);
 for (int i=k; i <= g; i++) pixels.setPixelColor(i, aus);
}
void LED6(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 pixels.setPixelColor(d, gbl);
 pixels.setPixelColor(e, bl);
 pixels.setPixelColor(f, blr);
 pixels.setPixelColor(g, blro);
 for (int i=h; i <= k; i++) pixels.setPixelColor(i, aus);
}
void LED7(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 pixels.setPixelColor(d, gbl);
 pixels.setPixelColor(e, bl);
 pixels.setPixelColor(f, blr);
 pixels.setPixelColor(g, blro);
 pixels.setPixelColor(h, bro);
 for (int i=k; i <= j; i++) pixels.setPixelColor(i, aus);
}
void LED8(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 pixels.setPixelColor(d, gbl);
 pixels.setPixelColor(e, bl);
 pixels.setPixelColor(f, blr);
 pixels.setPixelColor(g, blro);
 pixels.setPixelColor(h, bro);
 pixels.setPixelColor(j, ro);
 pixels.setPixelColor(k, aus);
}
void LED9(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 pixels.setPixelColor(a, gr);
 pixels.setPixelColor(b, grb);
 pixels.setPixelColor(c, grbl);
 pixels.setPixelColor(d, gbl);
 pixels.setPixelColor(e, bl);
 pixels.setPixelColor(f, blr);
 pixels.setPixelColor(g, blro);
 pixels.setPixelColor(h, bro);
 pixels.setPixelColor(j, ro);
 pixels.setPixelColor(k, rog);
}
void abfolge(int a,int b,int c,int d,int e,int f,int g,int h,int j,int k)
{
 switch (l)
```

```
 {
  case 93 ... 104:/*<---------------------------------------first LED area---
-------------------------------------------------*/
  LED0(a,b,c,d,e,f,g,h,j,k,gr);
  break;
  case 105 ... 139:/*<--------------------------------------second LED area-
-------------------------------------------------*/
  LED1(a,b,c,d,e,f,g,h,j,k);
  break;
  case 140 ... 164:/*<--------------------------------------third LED area--
-------------------------------------------------*/
  LED2(a,b,c,d,e,f,g,h,j,k);
  break;
  case 165 ... 199:/*<--------------------------------------fourth LED area-
-------------------------------------------------*/
  LED3(a,b,c,d,e,f,g,h,j,k);
  break;
  case 200 ... 234:/*<--------------------------------------fith LED area---
-------------------------------------------------*/
  LED4(a,b,c,d,e,f,g,h,j,k);
  break;
  case 235 ... 269:/*<--------------------------------------sixth LED area--
-------------------------------------------------*/
  LED5(a,b,c,d,e,f,g,h,j,k);
  break;
  case 270 ... 304:/*<--------------------------------------seventh LED
area-------------------------------------------------*/
  LED6(a,b,c,d,e,f,g,h,j,k);
  break;
  case 305 ... 339:/*<--------------------------------------eighth LED area-
-------------------------------------------------*/
  LED7(a,b,c,d,e,f,g,h,j,k);
  break;
  case 340 ... 374:/*<--------------------------------------nineth LED area-
-------------------------------------------------*/
  LED8(a,b,c,d,e,f,g,h,j,k);
  break;
  case 375 ... 1000:/*<-------------------------------------tenth LED area-
-------------------------------------------------*/
  LED9(a,b,c,d,e,f,g,h,j,k);
  break;
 }
}
void loop()
{
 // Cycle through each frequency band by pulsing the strobe.
 for (int i = 0; i < 7; i++)
 {
  digitalWrite       (strobePin, LOW);
  delayMicroseconds  (100);                    // Delay necessary due to
timing diagram
  level[i] =         analogRead (outPin);
  digitalWrite       (strobePin, HIGH);
  delayMicroseconds  (100);                    // Delay necessary due to
timing diagram
 }
```

```
 /*-------------------------------------------------------Band1(For the
first LED stripe)-----------------------------------------------------------
---*/
 // allocation for the Numbers of LEDs
 l = level [0];
 abfolge(0,1,2,3,4,5,6,7,8,9);
 if (l < 92) LEDaus1(0,1,2,3,4,5,6,7,8,9);
 /*-------------------------------------------------------Band2(For the
second LED stripe)----------------------------------------------------------
----*/
 // allocation for the Numbers of LEDs
 l = level [1];
 abfolge(19,18,17,16,15,14,13,12,11,10);
 if (l < 92) LEDaus2(19,18,17,16,15,14,13,12,11,10);
 /*-------------------------------------------------------Band3(For the
third LED stripe)-----------------------------------------------------------
---*/
 // allocation for the Numbers of LEDs
 l = level [2];
 abfolge(20,21,22,23,24,25,26,27,28,29);
 if (l < 92) LEDaus1(20,21,22,23,24,25,26,27,28,29);
 /*-------------------------------------------------------Band4(For the
fourth LED stripe)----------------------------------------------------------
----*/
 // allocation for the Numbers of LEDs
 l = level [3];
 abfolge(39,38,37,36,35,34,33,32,31,30);
 if (l < 92) LEDaus2(39,38,37,36,35,34,33,32,31,30);
 /*-------------------------------------------------------Band5(For the
fifth LED stripe)-----------------------------------------------------------
---*/
 // allocation for the Numbers of LEDs
 l = level [4];
 abfolge(40,41,42,43,44,45,46,47,48,49);
 if (l < 92) LEDaus1(40,41,42,43,44,45,46,47,48,49);
 /*-------------------------------------------------------Band6(For the
sixth LED stripe)-----------------------------------------------------------
---*/
 // allocation for the Numbers of LEDs
 l = level [5];
 abfolge(59,58,57,56,55,54,53,52,51,50);
 if (l < 92) LEDaus2(59,58,57,56,55,54,53,52,51,50);
 /*-------------------------------------------------------Band7(For the
seventh LED stripe)----------------------------------------------------------
-----*/
 // allocation for the Numbers of LEDs
 l = level [6];
 abfolge(60,61,62,63,64,65,66,67,68,69);
 if (l < 92) LEDaus1(60,61,62,63,64,65,66,67,68,69);
}
```

Schematics