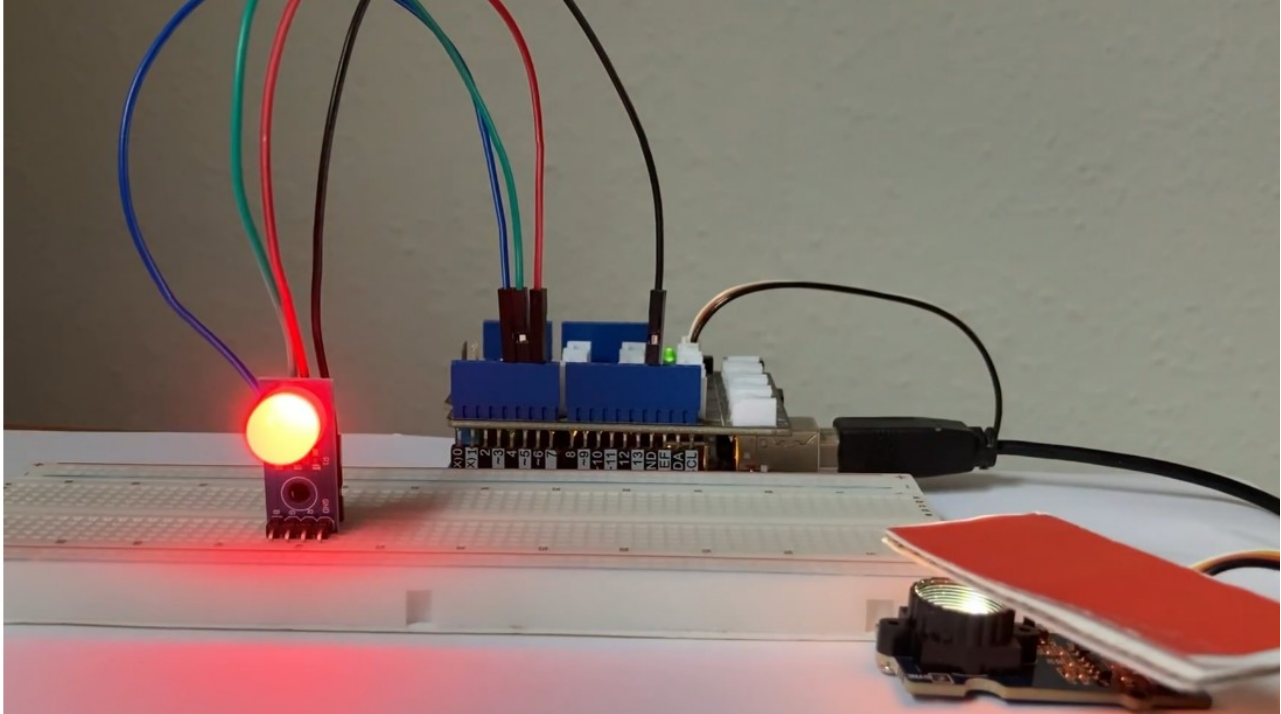


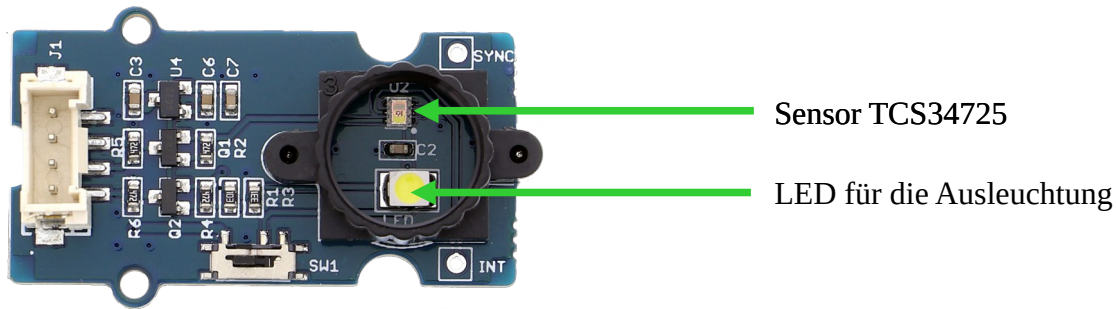
Ziel des Projekts

Wird eine Karte in den Farben rot, grün oder blau über den Farbsensor gehalten, leuchtet die entsprechende Farbe der RGB-LED.



Die Hardware

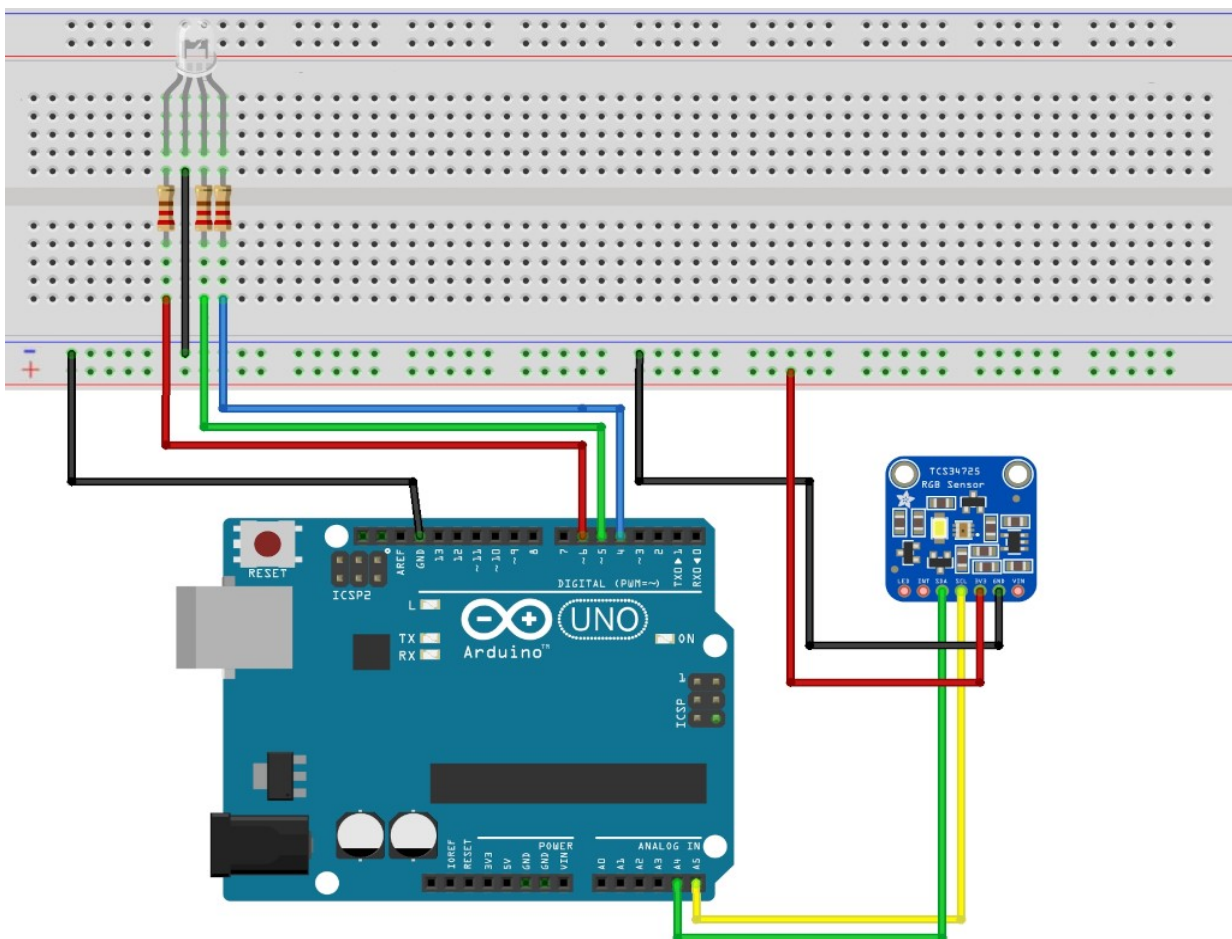
Der RGB-Sensor TCS34725 ist in verschiedenen Bauformen verfügbar. Diese Anleitung bezieht sich auf den Sensor mit dem Chip TCS34725. Der hier verwendete Sensor hat einen Grove-Stecker.



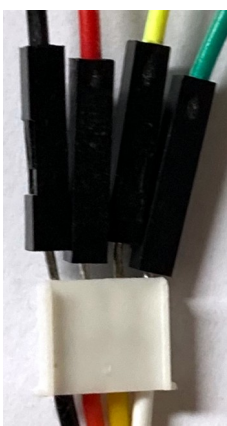
Benötigte Bauteile

- ➡ RGB-LED
- ➡ RGB-Sensor TCS34725
- ➡ Leitungsdrähte

Der Schaltplan



fritzing



Wenn das Grove-Shield nicht zur Verfügung steht, kann der Sensor auch direkt angeschlossen werden:

Farben am Stecker

→ rot - 5V



- ➔ schwarz - GND
- ➔ weiß - SDA
- ➔ gelb - SCL



Das Programm

Messung und Daten anzeigen

```
Ausgabe Serieller Monitor x
Nachricht (Enter um Nachricht für 'Arduino Uno' auf '/dev/ttyACM...
Neue Zeile
9600 Baud

Licht: 140
rot: 175 grün: 232 blau: 231
Farbtemperatur: 10773
Licht: 140
rot: 175 grün: 232 blau: 231
Farbtemperatur: 10773
Licht: 140
rot: 175 grün: 232 blau: 231
Farbtemperatur: 10773
```

Das dazugehörige Programm.

Die ermittelten Werte des Sensors kannst du beim eigentlichen Programm verwerten.

```
#include "Adafruit_TCS34725.h"

Adafruit_TCS34725 RGBSensor = Adafruit_TCS34725();

void setup()
{
  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial);
  delay(1000);

  if (RGBSensor.begin())
  {
    Serial.println("Sensor gefunden!");
  }

  // wenn der Sensor nicht gefunden wurde
  else
  {
    Serial.println("Sensor nicht gefunden");

    // Programm anhalten
    while (1);
  }

  // Empfindlichkeit einstellen
  // von 0x00 bis 0x03 (0, 2x, 16x, 60x)
  RGBSensor.setGain(0x03);
```



```
/*
  IntegrationTime -> Zeit für die Abtastung des Signals
  längere Zeiten erhöhen die Empfindlichkeit besonders bei schwachem Licht
  0xFF: < 2.4ms
  0xF6: < 24ms
  0xEB: < 50ms
  0xD5: < 101ms
  0xC0: < 154ms
  0x00: < 700ms
  Quelle: https://learn.adafruit.com/adafruit-color-sensors/library-reference
*/

RGBSensor.setIntegrationTime(0xFF);
}

void loop()
{
  // Variablen für die Farben
  int UngefilterteFarbe, rot, gruen, blau;

  // RGB-Sensor einschalten
  RGBSensor.setInterrupt(false);

  delay(60);

  // Daten erfassen
  RGBSensor.getRawData(&rot, &gruen, &blau, &UngefilterteFarbe);

  // Farbtemperatur gemessen in Kelvin
  int Farbtemperatur = RGBSensor.calculateColorTemperature(rot, gruen, blau);
  Serial.println("Farbtemperatur: " + String(Farbtemperatur));

  // Lichtverhältnis gemessen in LUX
  int Licht = RGBSensor.calculateLux(rot, gruen, blau);
  Serial.println("Licht: " + String(Licht));

  // RGB-Sensor ausschalten
  RGBSensor.setInterrupt(true);
  // Farbwerte berechnen
  float r, g, b;
  r = rot;
  r = r / UngefilterteFarbe;
  g = gruen;
  g = g / UngefilterteFarbe;
  b = blau;
  b = b / UngefilterteFarbe;

  r = r * 256;
  g = g * 256;
  b = b * 256;
}
```



```
// Farben anzeigen
Serial.print("rot:\t");
Serial.print((int)r);
Serial.print("\tgrün:\t");
Serial.print((int)g);
Serial.print("\tblau:\t");
Serial.println((int)b);

// delay-> Farbwerte für die Kalibrierung lesen
delay(500);
}
```

Steuerung der RGB-LED

```
#include "Adafruit_TCS34725.h"

Adafruit_TCS34725 RGBSensor = Adafruit_TCS34725();

int roteLED = 6;
int grüneLED = 5;
int blaueLED = 4;

void setup() {
  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial);
  delay(1000);

  pinMode(roteLED, OUTPUT);
  pinMode(grüneLED, OUTPUT);
  pinMode(blaueLED, OUTPUT);

  if (RGBSensor.begin())
  {
    Serial.println("Sensor gefunden!");
  }

  // wenn der Sensor nicht gefunden wurde
  else
  {
    Serial.println("Sensor nicht gefunden");

    // Programm anhalten
    while (1);
  }
}
```



```
// Empfindlichkeit einstellen
// von 0x00 bis 0x03 (0, 2x, 16x, 60x)
RGBSensor.setGain(0x03);

/*
  IntegrationTime -> Zeit für die Abtastung des Signals
  längere Zeiten erhöhen die Empfindlichkeit besonders bei schwachem Licht
  0xFF: < 2.4ms
  0xF6: < 24ms
  0xEB: < 50ms
  0xD5: < 101ms
  0xC0: < 154ms
  0x00: < 700ms
  Quelle: https://learn.adafruit.com/adafruit-color-sensors/library-reference
*/

RGBSensor.setIntegrationTime(0xFF);
}

void loop()
{
  // Variablen für die Farben
  int UngefilterteFarbe, rot, gruen, blau;

  // RGB-Sensor einschalten
  RGBSensor.setInterrupt(false);

  delay(60);

  // Daten erfassen
  RGBSensor.getRawData(&rot, &gruen, &blau, &UngefilterteFarbe);

  // Farbtemperatur gemessen in Kelvin
  int Farbtemperatur = RGBSensor.calculateColorTemperature(rot, gruen, blau);
  Serial.println("Farbtemperatur: " + String(Farbtemperatur));

  // Lichtverhältnis gemessen in LUX
  int Licht = RGBSensor.calculateLux(rot, gruen, blau);
  Serial.println("Licht: " + String(Licht));

  // RGB-Sensor ausschalten
  RGBSensor.setInterrupt(true);

  // Farbwerte berechnen
  float r, g, b;
  r = rot;
  r = r / UngefilterteFarbe;
  g = gruen;
  g = g / UngefilterteFarbe;
```



```
b = blau;
b = b / UngefilterteFarbe;

r = r * 256;
g = g * 256;
b = b * 256;

Serial.print("rot:\t");
Serial.print((int)r);
Serial.print("\tgrün:\t");
Serial.print((int)g);
Serial.print("\tblau:\t");
Serial.println((int)b);

// Grenzwerte -> RGB-LED bleibt aus
// mit Hilfe der Anzeige im Seriellen Monitor anpassen
if (r < 100 && g < 100 && b < 100)
{
    digitalWrite(roteLED, LOW);
    digitalWrite(grueneLED, LOW);
    digitalWrite(blaueLED, LOW);
}

// wird der Grenzwert überschritten leuchtet die RGB-LED
// in der entsprechenden Farbe
// mit Hilfe der Anzeige im Seriellen Monitor anpassen
if (r >= 100)
{
    digitalWrite(roteLED, HIGH);
}

if (g >= 100)
{
    digitalWrite(grueneLED, HIGH);
}

if (b >= 100)
{
    digitalWrite(blaueLED, HIGH);
}

// delay-> Farbwerte für die Kalibrierung lesen
delay(500);
}
```