

TCS3200 Farbsensor

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:

Schritt 1 – Anschließen des Sensors



Bild 1: TCS3200 Farbsensor

Schließen Sie den Farbsensor, wie im folgenden Bild 2, bzw. in folgender Tabelle 1, zu sehen, an die PINs des Arduinos an.

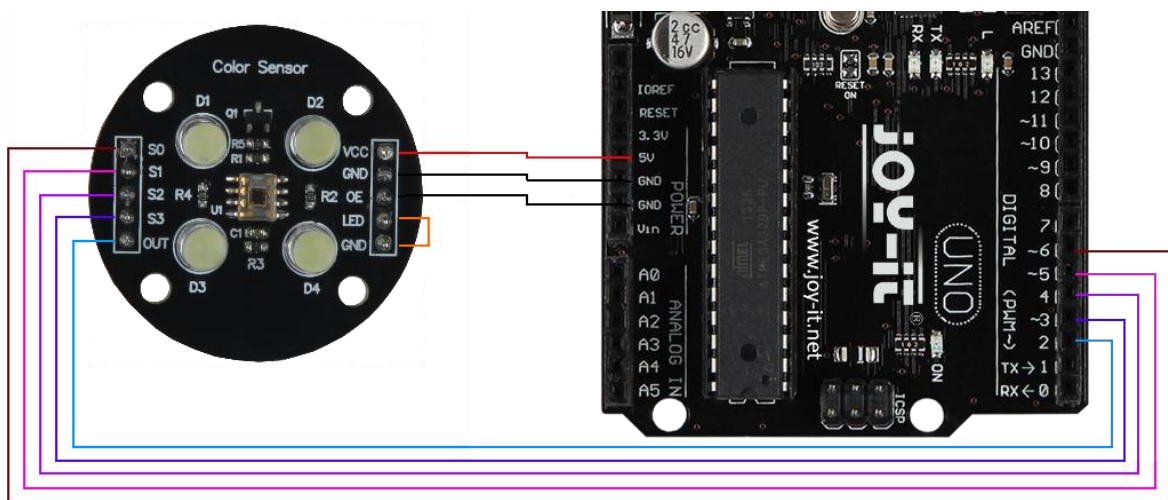


Bild 2: Verkabelung mit Arduino

TC3200 Farbsensor PIN	Arduino PIN
S0	6
S1	5
S2	4
S3	3
OUT	2
VCC	5V
GND	GND
OE	GND

Tabelle 1: PIN-Verbindung zwischen Farbsensor und Arduino

Die beiden nebeneinander liegenden PIN-Ausgänge **LED** und **GND** des Farbsensors müssen, entweder mit einem Jumper, oder einem Kabel, zusätzlich miteinander verbunden werden.

Schritt 2 – Installation der Bibliothek

Bevor Sie den entsprechen Quellcode auf Ihren Arduino übertragen, muss zunächst die **TimerOne** Bibliothek hinzugefügt werden.

Dazu klicken Sie bitte, wie in Bild 3 zu sehen, auf Sketch → Bibliothek einbinden → Bibliotheken verwalten.

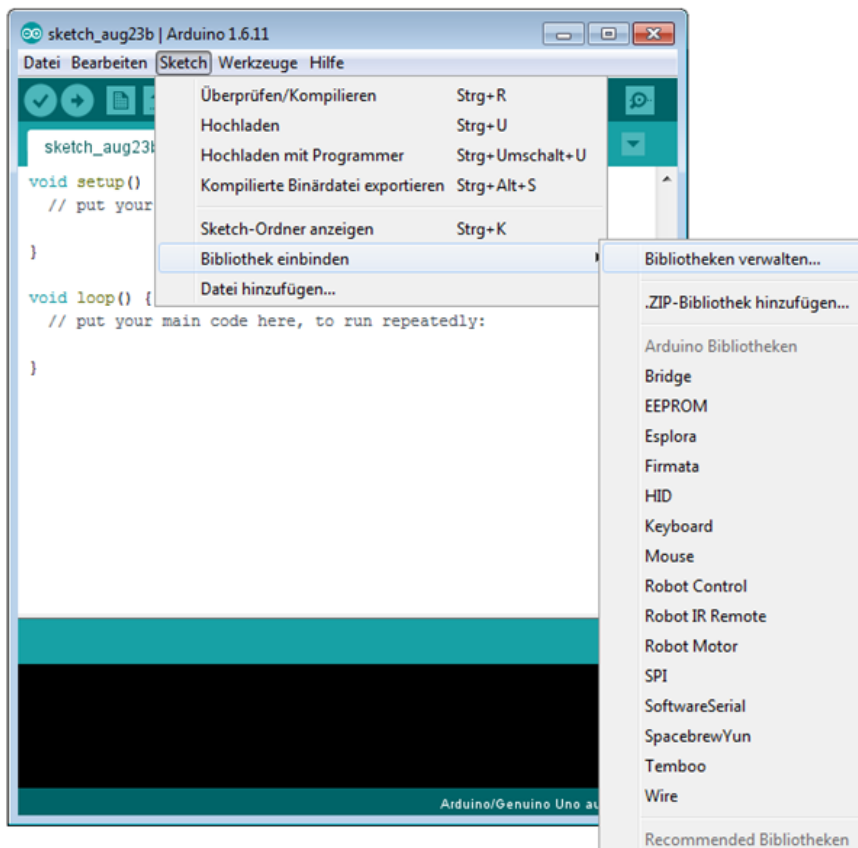


Bild 3: Bibliothekverwaltung im Arduino-Sketch

Im sich darauf öffnenden Bibliotheksverwalter suchen Sie, unter dem Suchbegriff **TimerOne**, die gleichnamige Bibliothek und installieren diese.

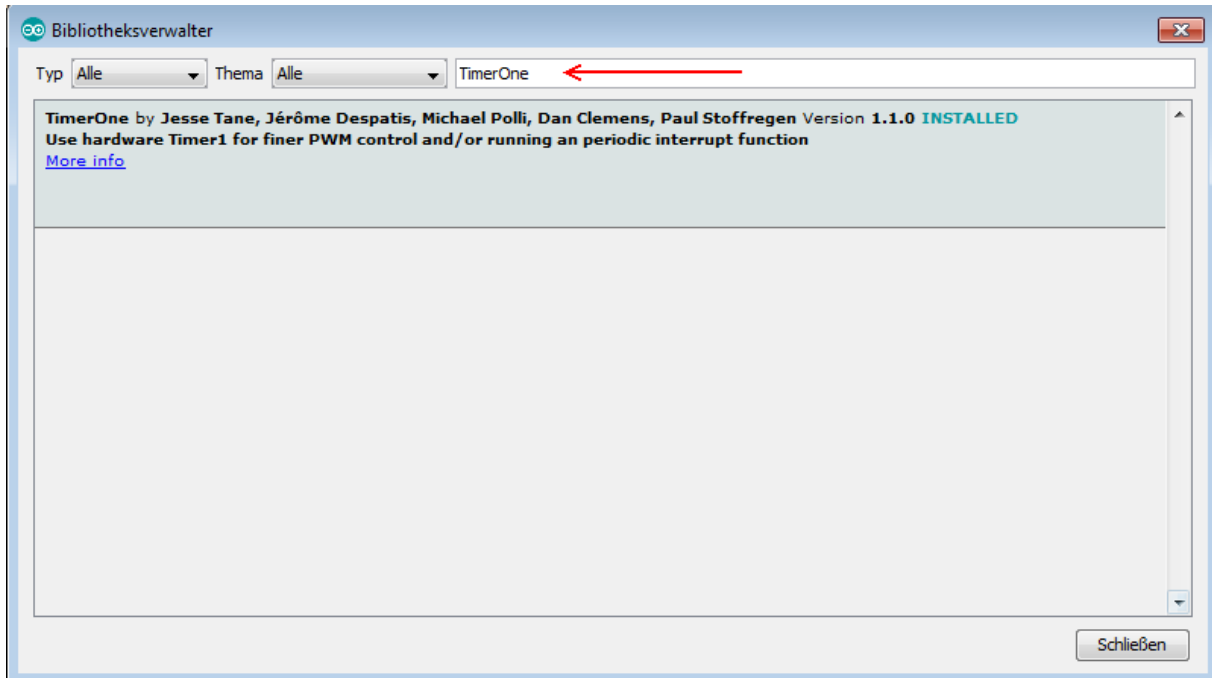


Bild 4: Installation der LiquidCrystal_I2C Bibliothek

Schritt 3 – Installation des Sensors

Im Nachfolgenden können Sie ein Codebeispiel zur Verwendung des Farbsensors entnehmen. Kopieren Sie bitte alle Teile vollständig auf Ihren Arduino.

```
#include <TimerOne.h>

#define S0  6
#define S1  5
#define S2  4
#define S3  3
#define OUT 2

int  g_count = 0;
int  g_array[3];
int  g_flag = 0;
float g_SF[3];

void TSC_Init()
{
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);

  digitalWrite(S0, LOW);
  digitalWrite(S1, HIGH);
}

void TSC_FilterColor(int Level01, int Level02)
{
  if(Level01 != 0)
    Level01 = HIGH;

  if(Level02 != 0)
    Level02 = HIGH;

  digitalWrite(S2, Level01);
  digitalWrite(S3, Level02);
}

void TSC_Count()
{
  g_count ++;
}
```

Code 1: Teil 1 des Arduino Quellcodes

```

void TSC_Callback()
{
  switch(g_flag)
  {
    case 0:
      Serial.println("->WB Start");
      TSC_WB(LOW, LOW);      //Filter ohne Rot
      break;
    case 1:
      Serial.print("->Frequency R=");
      Serial.println(g_count);
      g_array[0] = g_count;
      TSC_WB(HIGH, HIGH);   //Filter ohne Grün
      break;
    case 2:
      Serial.print("->Frequency G=");
      Serial.println(g_count);
      g_array[1] = g_count;
      TSC_WB(LOW, HIGH);   //Filter ohne Blau
      break;

    case 3:
      Serial.print("->Frequency B=");
      Serial.println(g_count);
      Serial.println("->WB End");
      g_array[2] = g_count;
      TSC_WB(HIGH, LOW);   //Kein Filter
      break;
    default:
      g_count = 0;
      break;
  }
}

void TSC_WB(int Level0, int Level1) //Weißabgleich
{
  g_count = 0;
  g_flag ++;
  TSC_FilterColor(Level0, Level1);
  Timer1.setPeriod(1000000);
}

```

Code 2: Teil 2 des Arduino Quellcodes

```
void setup()
{
  TSC_Init();
  Serial.begin(9600);
  Timer1.initialize();
  Timer1.attachInterrupt(TSC_Callback);
  attachInterrupt(0, TSC_Count, RISING);

  delay(4000);

  for(int i=0; i<3; i++)
    Serial.println(g_array[i]);

  g_SF[0] = 255.0/ g_array[0]; //R-Wert
  g_SF[1] = 255.0/ g_array[1]; //G-Wert
  g_SF[2] = 255.0/ g_array[2]; //B-Wert

  Serial.println(g_SF[0]);
  Serial.println(g_SF[1]);
  Serial.println(g_SF[2]);
}
void loop()
{
  g_flag = 0;
  for(int i=0; i<3; i++)
    Serial.println(int(g_array[i] * g_SF[i]));
  delay(4000);
}
```

Code 3: Teil 3 des Arduino Quellcodes

Schritt 3 – Verwendung des Sensors

Nach dem Übertragen und dem Start des Quellcodes, richten Sie den Sensor auf einen weißen Untergrund, um diesen zu kalibrieren.

Danach ist der Sensor bereit zur Farberkennung.

Öffnen Sie hierzu bitte in Ihrem Arduino-Sketch den Seriellen Monitor.

Die richtige Bestimmung der RGB-Werte können Sie dem folgenden Beispielbild entnehmen.



```
COM7 (Arduino/Genuino Uno)
->Frequency R=1292
->Frequency G=1045
->Frequency B=1240
->WB End
45
42
42
->WB Start
->Frequency R=1476
->Frequency G=1034
->Frequency B=1245
->WB End
52 ← Rot
41 ← Grün
42 ← Blau
```

The screenshot shows the Serial Monitor window for COM7 (Arduino/Genuino Uno). The output text is as follows:
->Frequency R=1292
->Frequency G=1045
->Frequency B=1240
->WB End
45
42
42
->WB Start
->Frequency R=1476
->Frequency G=1034
->Frequency B=1245
->WB End
52
41
42
Hand-drawn arrows point from the text 'Rot' to the value '52', from 'Grün' to '41', and from 'Blau' to '42'. The 'Autoscroll' checkbox is checked, and the settings at the bottom are 'Kein Zeilenende' and '9600 Baud'.

Bild 5: RGB Bestimmung im Seriellen Monitor