TCS34725 I2C Color Sensor For Arduino SKU: SEN0212

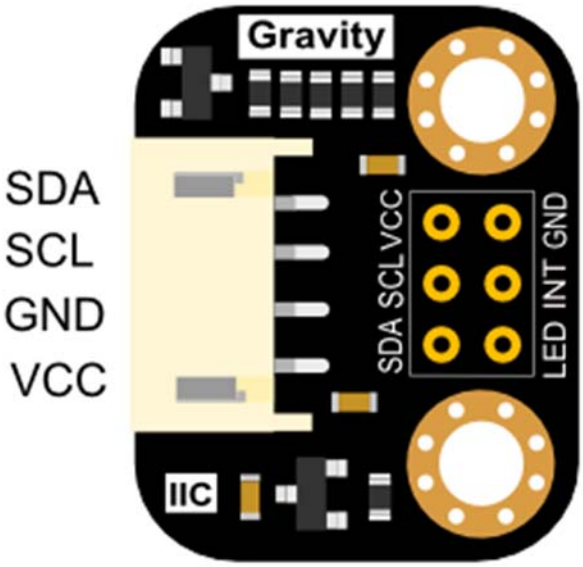
Introduction

The TCS34725 has a high sensitivity, wide dynamic range, and includes an IR blocking filter making it an ideal color sensing solution for use under varied lighting conditions. The sensor also includes four ultra-bright LEDs to allow the sensor to work without external light resources.
The module works via your Arduino's I2C bus and includes PH2.0-4P and XH2.54 (breadboard) interfaces to meet a range of user scenarios.

Specification

- Operating Voltage: 3.3~5V
- Operating Current: 65 uA
- Detection Range: 3-10 mm
- The Clock Frequency: 0-400 KHZ
- Interface: IIC interface
- IIC Address: 0x29
- Temperature Range: - 30 ℃ ~ + 70 ℃
- Feet inches: 18.5 * 23 mm/ 0.73 * 0.9 inches
- Weight: 12 g

Board Overview



Num	Label	Description
1	SDA	I2C-SDA
2	SCL	I2C-SCL
3	VCC	3.3~5V
4	GND	GND
5	LED	Active-High/Vacant On
6	INT	Active Low

Note:

1. I2C address: 0x29
2. XH2.54 interface (BreadBoard Compatible) need soldering.
3. The paper of the blocking filter could be teared out.

Tutorial

In this tutorial, we'll detect the specimen RGB value, and simulate it with RGB LEDs

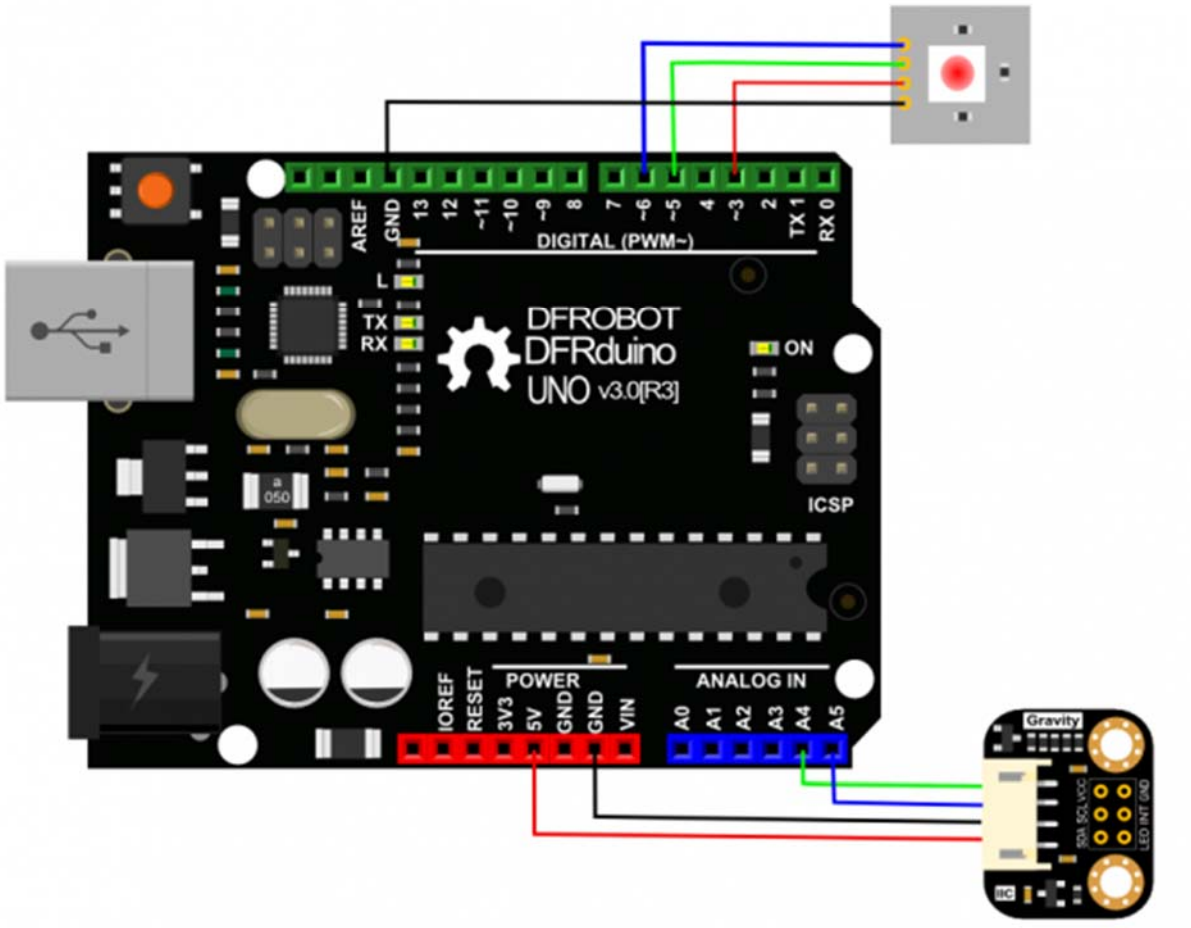
Requirements

- **Hardware**
  DFRduino UNO (or similar) x 1
  RGB LED Module
  M-M/F-M/F-F Jumper wires

- **Software**
  Arduino IDE (Version requirements: V1.6.x), Click to Download Arduino IDE from Arduino®
  https://www.arduino.cc/en/Main/Software%7C

Note: The sensor should be placed above the specimen, 3 ~ 10 mm

Connection Diagram



TCS34725 I2C Color Sensor For Arduino

Sample Code

Install the Arduino Library Download here How to install Libraries in Arduino IDE

https://github.com/DFRobot/DFRobot_TCS34725/raw/master/DFRobot_TCS34725.rar

https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0%7C

```
 1 /*!
 2  * @file colorview.ino
 3  * @brief DFRobot's Color Sensor
 4  * @n [Get the module here]
 5  * @n This example read current R,G,B,C value by the IIC bus
 6  * @n [Connection and Diagram](http://wiki.dfrobot.com.cn/index.php?
title=(SKU:SEN0212)Color_Sensor-TCS34725_%E9%A2%9C%E8%89%B2%E4%BC%A0%E6
%84%9F%E5%99%A8)
 7  *
 8  * @copyright [DFRobot](http://www.dfrobot.com), 2016
 9  * @copyright GNU Lesser General Public License
10  *
11  * @author [carl](carl.xu@dfrobot.com)
12  * @version  V1.0
13  * @date  2016-07-12
14  */
15 #include <Wire.h>
16 #include "DFRobot_TCS34725.h"
17
18 // Pick analog outputs, for the UNO these three work well
19 // use ~560  ohm resistor between Red & Blue, ~1K for green (its bri
ghter)
20 #define redpin 3
21 #define greenpin 5
22 #define bluepin 6
23 // for a common anode LED, connect the common pin to +5V
24 // for common cathode, connect the common to ground
25
```

```
// set to false if using a common cathode LED
#define commonAnode true

// our RGB -> eye-recognized gamma color
byte gammatable[256];

DFRobot_TCS34725 tcs = DFRobot_TCS34725(TCS34725_INTEGRATIONTIME_50MS, TCS34725_GAIN_4X);
void setup() {
  Serial.begin(115200);
  Serial.println("Color View Test!");

  if (tcs.begin()) {
    Serial.println("Found sensor");
  } else {
    Serial.println("No TCS34725 found ... check your connections");
    while (1); // halt!
  }
  // use these three pins to drive an LED
  pinMode(redpin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  pinMode(bluepin, OUTPUT);

  // thanks PhilB for this gamma table!
  // it helps convert RGB colors to what humans see
  for (int i=0; i<256; i++) {
    float x = i;
    x /= 255;
    x = pow(x, 2.5);
    x *= 255;

    if (commonAnode) {
      gammatable[i] = 255 - x;
    } else {
      gammatable[i] = x;
```

```
    }
    //Serial.println(gammatable[i]);
  }
}

void loop() {
  uint16_t clear, red, green, blue;
  tcs.getRGBC(&red, &green, &blue, &clear);
  tcs.lock();  // turn off LED
  Serial.print("C:\t"); Serial.print(clear);
  Serial.print("\tR:\t"); Serial.print(red);
  Serial.print("\tG:\t"); Serial.print(green);
  Serial.print("\tB:\t"); Serial.print(blue);
  Serial.println("\t");

  // Figure out some basic hex code for visualization
  uint32_t sum = clear;
  float r, g, b;
  r = red; r /= sum;
  g = green; g /= sum;
  b = blue; b /= sum;
  r *= 256; g *= 256; b *= 256;
  Serial.print("\t");
  Serial.print((int)r, HEX); Serial.print((int)g, HEX); Serial.print((int)b, HEX);
  Serial.println();

  //Serial.print((int)r ); Serial.print(" "); Serial.print((int)g);Serial.print(" ");  Serial.println((int)b );
  //Set the color lamp
  analogWrite(redpin, gammatable[(int)r]);
  analogWrite(greenpin, gammatable[(int)g]);
  analogWrite(bluepin, gammatable[(int)b]);
}
```

Expected Results