



Grove - I2C Color Sensor User Manual

Release date: 2015/9/22

Version: 1.0

Wiki:[http://www.seeedstudio.com/wiki/index.php?title=Twig - I2C Color Sensor v0.9b](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b)

Bazaar:<http://www.seeedstudio.com/depot/Grove-I2C-Color-Sensor-p-854.html>

Document Revision History

Revision	Date	Author	Description
1.0	Sep 22, 2015	jiankai.li	Create file

Contents

1. Introduction	2
2. Specification	3
3. Demonstration	4
4. With Arduino/Seeeduino	5
4.1 Suggest Reading for Starter	5
4.2 Hardware Installation	5
4.3 Color Sensor Library	5
4.4 Color Sensor Examples/Applications	6
5. Resources	8
6. Other Reference	9

Disclaimer

For physical injuries and possessions loss caused by those reasons which are not related to product quality, such as operating without following manual guide, natural disasters or force majeure, we take no responsibility for that.

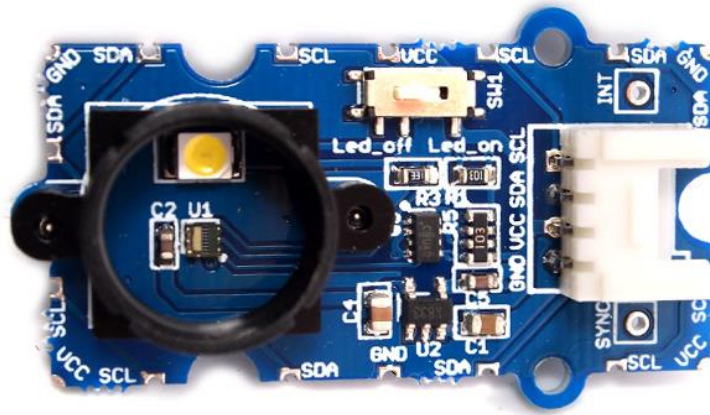
Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

Copyright

The design of this product (including software) and its accessories is under tutelage of laws. Any action to violate relevant right of our product will be penalized through law. Please consciously observe relevant local laws in the use of this product.

1. Introduction

This module is based on the color sensor TCS3414CS with digital output I2C. Based on the 8*2 array of filtered photodiodes and 16-bit analog-to-digital converters, you can measure the color chromaticity of ambient light or the color of objects. Of the 16 photodiodes, 4 have red filters, 4 have green filters, 4 have blue filters and 4 have no filter(clear). With the synchronization input pin, an external pulsed light source can provide precise synchronous conversion control.

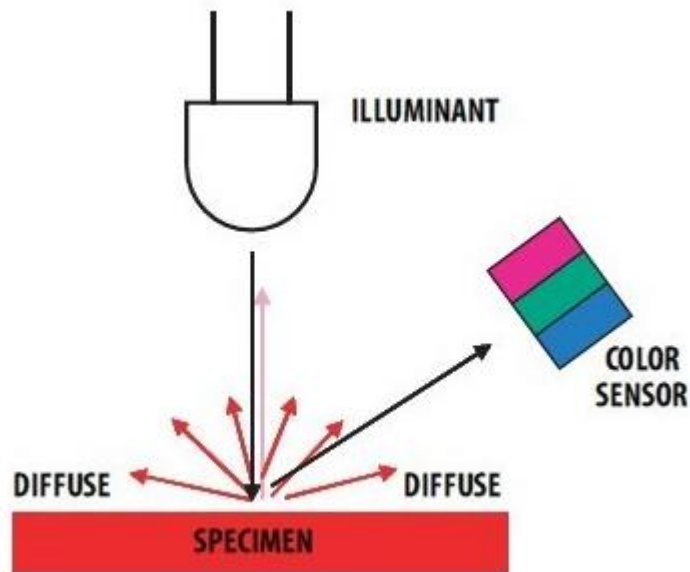


2. Specification

- PCB Size : 2.0cm*4.0cm
- Interface : 2.0mm pitch pin header
- IO Structure : SCL,SDA,VCC,GND
- ROHS : YES
- VCC :3.3 ~6.0 V
- Operating Free-air Temperature :-40 ~ 85 °C

3. Demonstration

This module can be used to detect the color of light source or the color of objects. When used to detect the color of the light source, the led switch should be turned off, and the light source should shine the sensor directly. When used to detect the color of things, the led should be on and you should put the object on the top of the enclosure closely. The theory of sensing the color of objects is Reflective Sensing Theory. Like the picture below.



4. With Arduino/Seeeduino

4.1 Suggest Reading for Starter

- [Download Arduino and install Arduino driver](#)
- [Getting Started with Seeeduino/Arduino](#)

4.2 Hardware Installation

Grove products have a eco system and all have a same connector which can plug onto the Base Shield. Connect this module to the I2C port of Base Shield, however, you can also connect Grove - I2C Color Sensor to Arduino without Base Shield by jumper wires.

Arduino UNO	Grove - I2C Color Sensor
5V	VCC
GND	GND
SDA	SDA
SCL	SCL

4.3 Color Sensor Library

We have created a library to help you start playing quickly with the Seeeduino/Arduino, in this section we'll show you how to set up the library.

Setup

- [Download the library code as a zip file from the Grove I2C Color Sensor github page.](#)
- Unzip the downloaded file into your `.../arduino/libraries`.
- Rename the unzipped folder "Color_Sensor"
- Start the Arduino IDE (or restart if it is open).

Description of functions

These are the most important/useful function in the library, we invite you to look at the `.h` and `.cpp` files yourself to see all the functions available.

1. Read RGB data through the library function

- `readRGB(int *red, int *green, int *blue)`
 - `*red`: The variable address to save R.
 - `*green`: The variable address to save G.
 - `*blue`: The variable address to save B.


```

void loop()
{
  int red, green, blue;
  GroveColorSensor colorSensor;
  colorSensor.ledStatus = 1;          // When turn on the color sensor LED, ledStatus = 1; When
turn off the color sensor LED, ledStatus = 0.
  while(1)
  {
    colorSensor.readRGB(&red, &green, &blue); //Read RGB values to variables.
    delay(300);
    Serial.print("The RGB value are: RGB( ");
    Serial.print(red, DEC);
    Serial.print(", ");
    Serial.print(green, DEC);
    Serial.print(", ");
    Serial.print(blue, DEC);
    Serial.println(")");
    colorSensor.clearInterrupt();
  }
}

```

4.4 Color Sensor Examples/Applications

These example are going to show you how to use features of Grove - I2C Color Sensor. And via [Chainable RGB LED Grove](#) to display the detected color.

Notice: If you haven't download [Grove-Chainable RGB LED library](#) to your Arduino IDE before, please download and set up the library first.

- Open File->Examples->Color_Sensor->example->ColorSensorWithRGB-LED sketch for a complete example, or copy and paste code below to a new Arduino sketch.

Description: This example can measure the color chromaticity of ambient light or the color of objects, and via Chainable RGB LED Grove displaying the detected color.

You also can use other display module to display the detected color by Grove - I2C Color Sensor.

```

#include <Wire.h>
#include <GroveColorSensor.h>
#include <ChainableLED.h>

#define CLK_PIN 7
#define DATA_PIN 8
#define NUM_LEDS 1          //The number of Chainable RGB LED

```

```
ChainableLED leds(CLK_PIN, DATA_PIN, NUM_LEDS);

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

void loop()
{
  int red, green, blue;
  GroveColorSensor colorSensor;
  colorSensor.ledStatus = 1;          // When turn on the color sensor LED, ledStatus = 1; When
turn off the color sensor LED, ledStatus = 0.
  while(1)
  {
    colorSensor.readRGB(&red, &green, &blue); //Read RGB values to variables.
    delay(300);
    Serial.print("The RGB value are: RGB( ");
    Serial.print(red, DEC);
    Serial.print(", ");
    Serial.print(green, DEC);
    Serial.print(", ");
    Serial.print(blue, DEC);
    Serial.println(")");
    colorSensor.clearInterrupt();
    for(int i = 0; i<NUM_LEDS; i++)
    {
      leds.setColorRGB(i, red, green, blue);
    }
  }
}
```

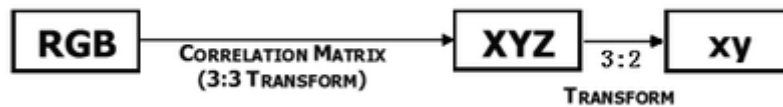
- Upload the code to the development board.
- Then Grove_-_Chainable_RGB_LED would display the color which is detected.

5. Resources

- [Library Grove - I2C Color Sensor](#)
- [Grove-I2C Color Sensor Eagle File](#)
- [TCS3414-A Datasheet](#)

6. Other Reference

This module is based on the color sensor TCS3414CS. The TCS3414CS digital color sensor returns data from four channels: red(R), green(G), blue(B) and clear(C)(non-filtered). The response from the red, green and blue channels (RGB) can be used to determine a particular source's chromaticity coordinates (x, y). These standards are set by the Commission Internationale de l'Eclairage (CIE). The CIE is the main international organization concerned with color and color measurement. In order to acquire the color of a given object using TCS3414CS, we must first map the sensor response (RGB) to the CIE tristimulus values (XYZ). It is then necessary to calculate the chromaticity coordinates (x, y).



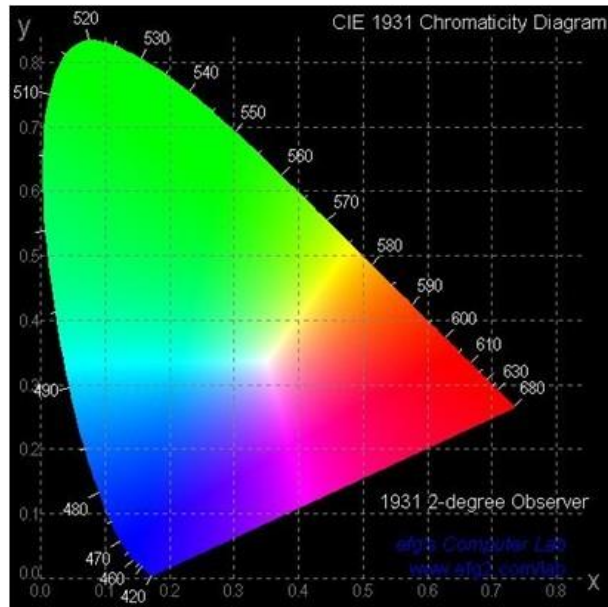
Chromaticity Calculation Process Overview

The equations to do the transformation:

$$\begin{aligned}
 X &= (-0.14282)(R) + (1.54924)(G) + (-0.95641)(B) \\
 Y &= (-0.32466)(R) + (1.57837)(G) + (-0.73191)(B) \\
 Z &= (-0.68202)(R) + (0.77073)(G) + (0.56332)(B) \\
 x &= X/(X+Y+Z) \\
 y &= Y/(X+Y+Z)
 \end{aligned}$$

Transformation Equations

- When we get coordinates (x, y), please reference the below figure so as to get the recommended color.



X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Optical Sensor Development Tools](#) category:

Click to view products by [Seeed Studio](#) manufacturer:

Other Similar products are found below :

[MT9V034C12STCH-GEVB](#) [MT9V115EBKSTCH-GEVB](#) [416015300-3](#) [ISL29102IROZ-EVALZ](#) [MT9M021IA3XTMH-GEVB](#)
[AR1335CSSC11SMKAH3-GEVB](#) [MAXCAMOV10640#](#) [MT9M031I12STMH-GEVB](#) [TSL2581CS-DB](#) [TMD3700-DB](#) [NANOUSB2.2](#)
[ASX340AT3C00XPEDH3-GEVB](#) [AR0144ATSM20XUEAH3-GEVB](#) [AR0144CSSC00SUKAH3-GEVB](#) [AR0522SRSC09SURAH3-GEVB](#)
[AR0522SRSM09SURAH3-GEVB](#) [AR0521SR2C09SURAH3-GEVB](#) [MARS1-MAX9295A-GEVK](#) [MARS1-MAX9296B-GEVB](#)
[ISL29112IROZ-EVALZ](#) [AR0233AT2C17XUEAH3-GEVB](#) [AR0431CSSC14SMRAH3-GEVB](#) [MARS-DEMO3-MIPI-GEVB](#) [TCS3430-DB](#)
[AR0234CSSC00SUKAH3-GEVB](#) [AR0130CSSM00SPCAH-GEVB](#) [TSL2521-DB](#) [TSL2520-DB](#) [EVALZ-ADPD2212](#) [TMD2772EVM](#)
[TMG3993EVM](#) [MIKROE-2103](#) [TSL2672EVM](#) [1384](#) [MT9M114EBLSTCZDH-GEVB](#) [SEN0043](#) [SEN0162](#) [TMD2771EVM](#) [TMD3782EVM](#)
[TSL4531EVM](#) [1918](#) [AS7225 DEMO KIT](#) [SEN0097](#) [SEN0212](#) [SEN0228](#) [AR0134CSSC00SUEAH3-GEVB](#) [AP0100AT2L00XUGAH3-GEVB](#)
[AR0144CSSM20SUKAH3-GEVB](#) [725-28915](#) [EVAL-ADPD1081Z-PPG](#)