Grove - I2C Color Sensor

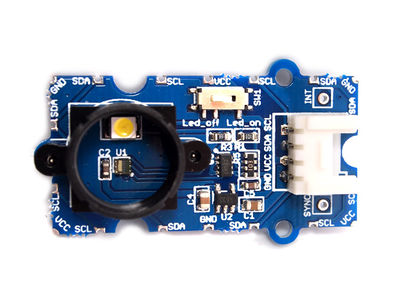
(Redirected from [Twig - I2C Color Sensor v0.9b](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b&redirect=no))

|  |
| --- |
| **Contents**   [[hide](javascript:toggleToc())]   * [1 Introduction](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Introduction) * [2 Features](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Features) * [3 Application Ideas](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Application_Ideas) * [4 Specification](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Specification)   + [4.1 Key Specification](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Key_Specification)   + [4.2 Electronic Characterstics](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Electronic_Characterstics) * [5 Usage](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Usage) * [6 Reference](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Reference) * [7 Version Tracker](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Version_Tracker) * [8 Resources](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Resources) * [9 Support](http://www.seeedstudio.com/wiki/index.php?title=Twig_-_I2C_Color_Sensor_v0.9b#Support) |

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.

Model:[SEN60256P](http://www.seeedstudio.com/depot/grove-i2c-color-sensor-p-854.html?cPath=144_148)

[](http://www.seeedstudio.com/wiki/File:I2C_Color_Sensor.jpg)

Features

* Grove compatible interface
* 16-Bit Digital Output with I 2C at 400 kHz
* SYNC Input Synchronizes Integration Cycle to Modulated Light Sources
* Operating Temperature Range -40℃ to 85℃
* Programmable Interrupt Function with User-Defined Upper and Lower Threshold Settings

Application Ideas

* Ambient light sensing
* Object color sensing
* Consumer Toys
* Industrial Process Control
* Tablets, Laptops, Monitors
* HDTVs

Specification

**Key Specification**

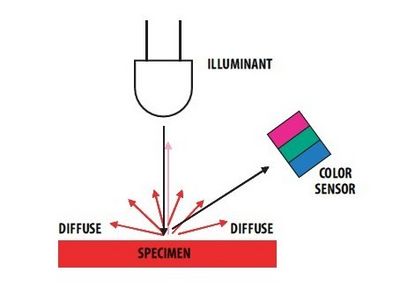
|  |  |
| --- | --- |
| **Items** | **Min** |
| PCB Size | 2.0cm\*4.0cm |
| Interface | 2.0mm pitch pin header |
| IO Structure | SCL,SDA,VCC,GND |
| ROHS | YES |

**Electronic Characterstics**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Items** | **Conditions** | **Min** | **Norm** | **Max** | **Unit** |
| VCC | - | 3.3 | 5.0 | 6.0 | Volts |
| Current Draw | VCC=5V,Led On,Sensor Power on | - | 16 | - | mA |
| VCC=5V,Led On,Sensor Power Down | - | 8 | - | mA |
| VCC=5V,Led Off,Sensor Power On | - | 10 | - | mA |
| VCC=5V,Led Off,Sensor Power Down | - | 700 | - | μA |
| Operating Free-air Temperature | Ambient | -40 | - | 85 | °C |

Usage

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.

[](http://www.seeedstudio.com/wiki/File:Reflcect.jpg)

The following sketch demonstrates a simple application of sensing the color of the paper.

* Connect the Color Sensor to Seeeduino through the [Grove-Base Shield](http://www.seeedstudio.com/wiki/Grove_-_Base_Shield) using the I²C interface.
* Connect the Seeeduino to PC via a USB cable.

(图片)

* Copy and paste code below to a new Arduino sketch.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <Wire.h>

#include <math.h>

#define COLOR\_SENSOR\_ADDR 0x39//the I2C address for the color sensor

#define REG\_CTL 0x80

#define REG\_TIMING 0x81

#define REG\_INT 0x82

#define REG\_INT\_SOURCE 0x83

#define REG\_ID 0x84

#define REG\_GAIN 0x87

#define REG\_LOW\_THRESH\_LOW\_BYTE 0x88

#define REG\_LOW\_THRESH\_HIGH\_BYTE 0x89

#define REG\_HIGH\_THRESH\_LOW\_BYTE 0x8A

#define REG\_HIGH\_THRESH\_HIGH\_BYTE 0x8B

#define REG\_BLOCK\_READ 0xCF

#define REG\_GREEN\_LOW 0xD0

#define REG\_GREEN\_HIGH 0xD1

#define REG\_RED\_LOW 0xD2

#define REG\_RED\_HIGH 0xD3

#define REG\_BLUE\_LOW 0xD4

#define REG\_BLUE\_HIGH 0xD5

#define REG\_CLEAR\_LOW 0xD6

#define REG\_CLEAR\_HIGH 0xD7

#define CTL\_DAT\_INIITIATE 0x03

#define CLR\_INT 0xE0

//Timing Register

#define SYNC\_EDGE 0x40

#define INTEG\_MODE\_FREE 0x00

#define INTEG\_MODE\_MANUAL 0x10

#define INTEG\_MODE\_SYN\_SINGLE 0x20

#define INTEG\_MODE\_SYN\_MULTI 0x30

#define INTEG\_PARAM\_PULSE\_COUNT1 0x00

#define INTEG\_PARAM\_PULSE\_COUNT2 0x01

#define INTEG\_PARAM\_PULSE\_COUNT4 0x02

#define INTEG\_PARAM\_PULSE\_COUNT8 0x03

//Interrupt Control Register

#define INTR\_STOP 40

#define INTR\_DISABLE 0x00

#define INTR\_LEVEL 0x10

#define INTR\_PERSIST\_EVERY 0x00

#define INTR\_PERSIST\_SINGLE 0x01

//Interrupt Souce Register

#define INT\_SOURCE\_GREEN 0x00

#define INT\_SOURCE\_RED 0x01

#define INT\_SOURCE\_BLUE 0x10

#define INT\_SOURCE\_CLEAR 0x03

//Gain Register

#define GAIN\_1 0x00

#define GAIN\_4 0x10

#define GAIN\_16 0x20

#define GANI\_64 0x30

#define PRESCALER\_1 0x00

#define PRESCALER\_2 0x01

#define PRESCALER\_4 0x02

#define PRESCALER\_8 0x03

#define PRESCALER\_16 0x04

#define PRESCALER\_32 0x05

#define PRESCALER\_64 0x06

int readingdata[20];

int i,green,red,blue,clr,ctl;

double X,Y,Z,x,y,z;

void setup()

{

Serial.begin(9600);

Wire.begin(); // join i2c bus (address optional for master)

}

void loop()

{

setTimingReg(INTEG\_MODE\_FREE);//Set trigger mode.Including free mode,manually mode,single synchronizition mode or so.

setInterruptSourceReg(INT\_SOURCE\_GREEN); //Set interrupt source

setInterruptControlReg(INTR\_LEVEL|INTR\_PERSIST\_EVERY);//Set interrupt mode

setGain(GAIN\_1|PRESCALER\_4);//Set gain value and prescaler value

setEnableADC();//Start ADC of the color sensor

while(1)

{

readRGB();

calculateCoordinate();

delay(1000);

clearInterrupt();

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void setTimingReg(int x)

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(REG\_TIMING);

Wire.write(x);

Wire.endTransmission();

delay(100);

}

void setInterruptSourceReg(int x)

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(REG\_INT\_SOURCE);

Wire.write(x);

Wire.endTransmission();

delay(100);

}

void setInterruptControlReg(int x)

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(REG\_INT);

Wire.write(x);

Wire.endTransmission();

delay(100);

}

void setGain(int x)

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(REG\_GAIN);

Wire.write(x);

Wire.endTransmission();

}

void setEnableADC()

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(REG\_CTL);

Wire.write(CTL\_DAT\_INIITIATE);

Wire.endTransmission();

delay(100);

}

void clearInterrupt()

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(CLR\_INT);

Wire.endTransmission();

}

void readRGB()

{

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.write(REG\_BLOCK\_READ);

Wire.endTransmission();

Wire.beginTransmission(COLOR\_SENSOR\_ADDR);

Wire.requestFrom(COLOR\_SENSOR\_ADDR,8);

delay(500);

if(8<= Wire.available()) // if two bytes were received

{

for(i=0;i<8;i++)

{

readingdata[i]=Wire.read();

//Serial.println(readingdata[i],BIN);

}

}

green=readingdata[1]\*256+readingdata[0];

red=readingdata[3]\*256+readingdata[2];

blue=readingdata[5]\*256+readingdata[4];

clr=readingdata[7]\*256+readingdata[6];

Serial.println("The RGB value and Clear channel value are");

Serial.println(red,DEC);

Serial.println(green,DEC);

Serial.println(blue,DEC);

Serial.println(clr,DEC);

}

void calculateCoordinate()

{

X=(-0.14282)\*red+(1.54924)\*green+(-0.95641)\*blue;

Y=(-0.32466)\*red+(1.57837)\*green+(-0.73191)\*blue;

Z=(-0.68202)\*red+(0.77073)\*green+(0.56332)\*blue;

x=X/(X+Y+Z);

y=Y/(X+Y+Z);

if((X>0)&&(Y>0)&&(Z>0))

{

Serial.println("The x,y value is");

Serial.print("(");

Serial.print(x,2);

Serial.print(" , ");

Serial.print(y,2);

Serial.println(")");

Serial.println("Please reference the figure(Chromaticity Diagram) in the wiki ");

Serial.println("so as to get the recommended color.");

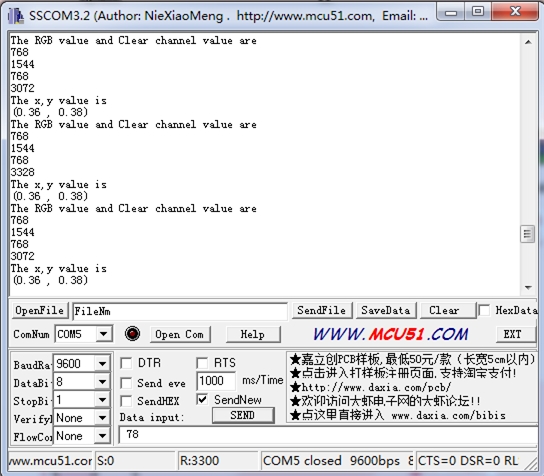
}

else

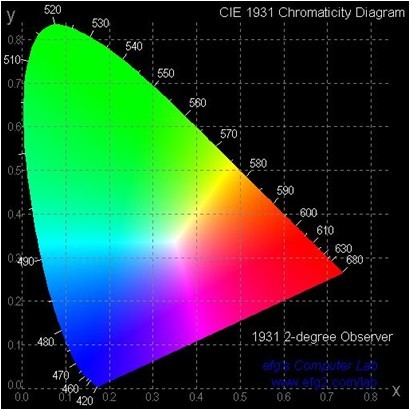
Serial.println("Error,the value overflow");

}

* Upload the code, Please click [here](http://www.seeedstudio.com/wiki/Upload_Code) if you do not know how to upload.
* Open the serial monitor, You will see the output result of Color Sensor as show below:

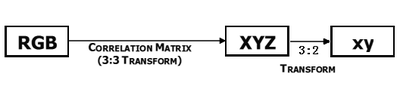
[](http://www.seeedstudio.com/wiki/File:I2C_Color_Sensor_Result.jpg)

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

[](http://www.seeedstudio.com/wiki/File:Chromaticity_Diagram.jpg)

Reference

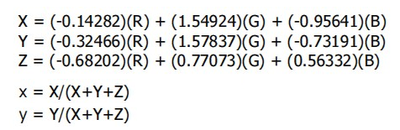
This module is based on the color sensor TCS3414CS. The TCS3414CS digital color sensor returns data from four channels: red(R), green(G), blue(B) and clear(C)(non-filtered). The response from the red, green and blue channels (RGB) can be used to determine a particular source’s chromaticity coordinates (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).

[](http://www.seeedstudio.com/wiki/File:Coordinates_transform.png)

[http://www.seeedstudio.com/wiki/skins/common/images/magnify-clip.png](http://www.seeedstudio.com/wiki/File:Coordinates_transform.png)

Chromaticity Calculation Process Overview

The equations to do the transformation:

[](http://www.seeedstudio.com/wiki/File:Equations.png)

[http://www.seeedstudio.com/wiki/skins/common/images/magnify-clip.png](http://www.seeedstudio.com/wiki/File:Equations.png)

Transformation Equations

Version Tracker

|  |  |  |
| --- | --- | --- |
| **Revision** | **Descriptions** | **Release** |
| v0.9b | Initial public release | date |

Resources

* [Grove-I2C Color Sensor Eagle File](http://www.seeedstudio.com/wiki/File:Twig_-_I2C_Color_sensor_v0.9b_eagle_files.zip)
* [TCS3414-A Datasheet](http://www.seeedstudio.com/wiki/File:TCS3404_TCS3414-A.pdf)