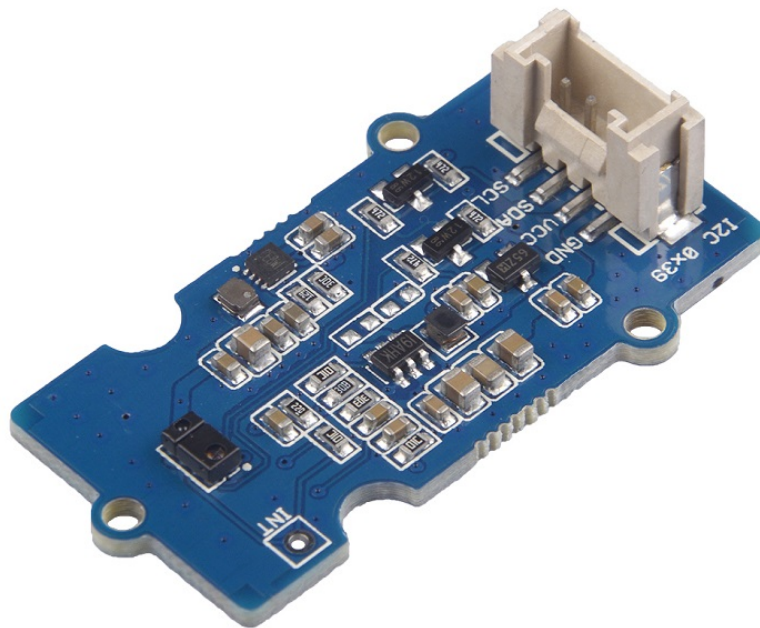


Grove - Light Gesture Color Proximity Sensor (TMG39931)



The Grove - Light & Gesture & Color & proximity sensor is TMG39931 based product, which features advanced Gesture detection, Proximity detection, Digital Ambient Light Sense(ALS), Color Sense(RGBC), and optical pattern generation/transmission for

broadcast. This four-in-one sensor allows you to collect the data from ambient environment and transfers over the I2C bus.



[<https://www.seeedstudio.com/Grove-Light-Gesture-Color-Proximity-Sensor-TMG39931.html>]

Feature

- Single device integrated optical solution
- Ambient light sensing
- UV and IR blocking filters
- Complex gesture sensing
- Ideal for operation behind dark glass - High sensitivity
- Proximity detection
- Barcode pattern generation and transmission
- Dual use of a single internal LED
- Trimmed to provide consistent reading

Specification

Item	Value
Supply Voltage	3.3V / 5V
Operating temperature	-30 ~ 85°C
Storage temperature	-40 ~ 85°C
Max. Lux [klx]	60
Interface	I2C
I2C address	0x39

Typical applications

- Gesture Detection
- Color Sense
- Ambient Light Sensing
- Cell Phone Touch Screen Disable
- Mechanical Switch Replacement
- Printed Bar Code Emulation

Working Principle

Gesture Detection

Gesture detection utilizes four directional photodiodes to sense reflected IR energy (sourced by the integrated LED) to convert physical motion information to a digital information.

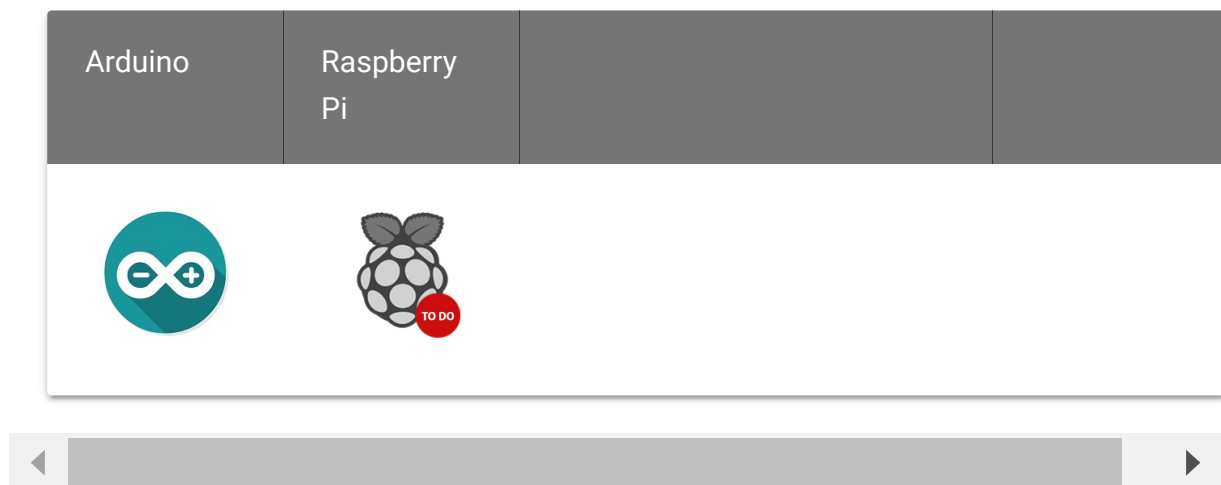
Proximity Detection

The proximity detection features provides object detection by photodiode detection of reflected IR energy(Sourced by the integrated LED).

Digital Ambient Light Sense & Color Sense

The Color and ALS detection feature provides red, green, blue and clear light intensity data. Each of the R, G, B, C channels have a UV and IR blocking filter and a dedicated data converter producing 16-bit data simultaneously. This architecture allows applications to accurately measure ambient light and sense color which enables devices to calculate illuminance and color temperature, control display backlight, and chromaticity.

Platforms Supported



Getting Started

Play With Arduino

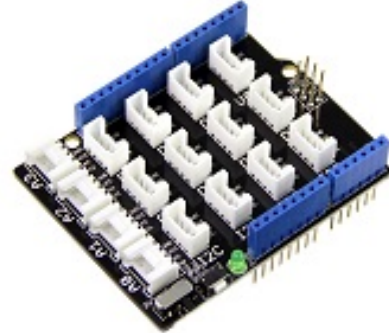
Hardware

Materials required

Seeeduino V4.2



Base Shield



Get ONE Now

[<https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html>]

Get ONE Now

[<https://www.seeedstudio.com/Base-Shield-V2-p-1378.html>]



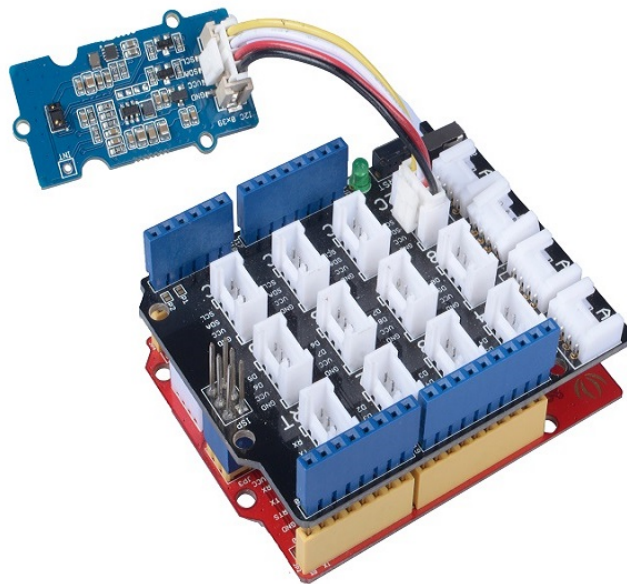
Note

1 Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click [here](https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html) [https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html] to buy

2 Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click [here](#)

[<https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html>] to buy.

- **Step 1.** Connect the Grove - Light&Gesture&Color&Proximity Sensor(TM39931) to port I²C of Grove-Base Shield.
- **Step 2.** Plug Grove - Base Shield into Seeeduino.
- **Step 3.** Connect Seeeduino to PC via a USB cable.



Software



Attention

If this is the first time you work with Arduino, we strongly recommend you to see [Getting Started with Arduino](#)

[https://wiki.seeedstudio.com/Getting_Started_with_Arduino/] before the start.

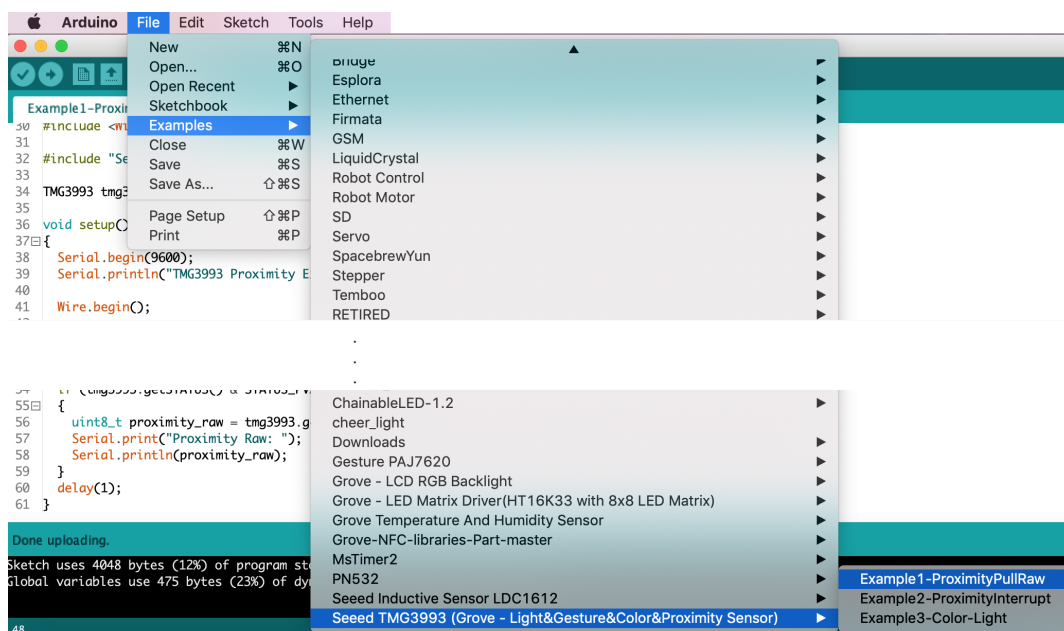


Tips

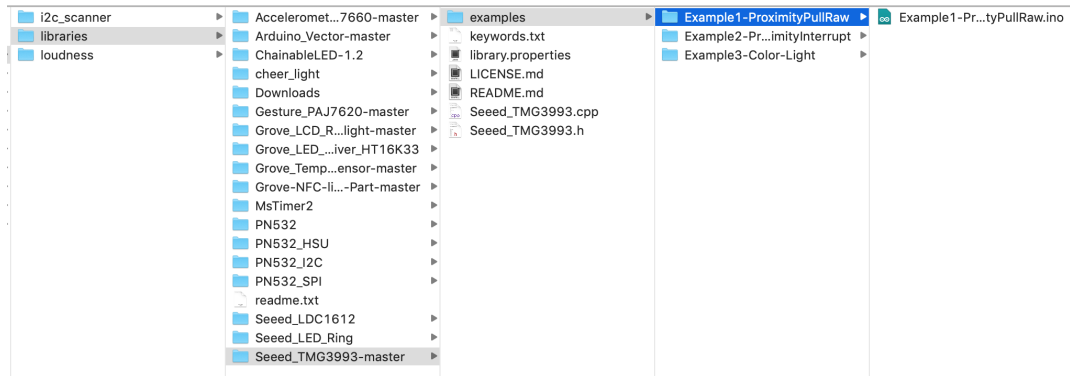
The gesture functionality is not supported yet, since this functionality needs algorithms provided by AMS. We will update this library as soon as we get the support from AMS.


Example 1

- **Step 1.** Download the [Seeed_TM3993](https://github.com/Seeed-Studio/Seeed_TM3993.git) [https://github.com/Seeed-Studio/Seeed_TM3993.git] Library from Github.
- **Step 2.** Refer to [How to install library](https://wiki.seeedstudio.com/How_to_install_Arduino_Library) [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open the example, you can open it in the following three ways:
 - a. Open it directly in the Arduino IDE via the path: **File** → **Examples** → **Seed TMG3993(Grove - Light&Gesture&Color&Proximity Sensor)** → **Example1-ProximityPullRaw**.



- a. Open it in your computer by click the **basic_demo.ino** which you can find in the folder **XXXX\Arduino\libraries\Seeed_TM3993-master\examples\Example1-ProximityPullRaw\Example1-ProximityPullRaw.ino**, **XXXX** is the location you installed the Arduino IDE.



- a. Or, you can just click the icon  in upper right corner of the code block to copy the following code into a new sketch in the Arduino IDE.

```

1  #include <Wire.h>
2
3  #include "Seeed_TM3993.h"
4
5  TMG3993 tmg3993;
6
7  void setup()
8  {
9      Serial.begin(9600);
10     Serial.println("TMG3993 Proximity Example");
11
12     Wire.begin();
13
14     if (tmg3993.initialize() == false)
15     {
16         Serial.println("Device not found. Check wiring.");
17         while (1);
18     }

```



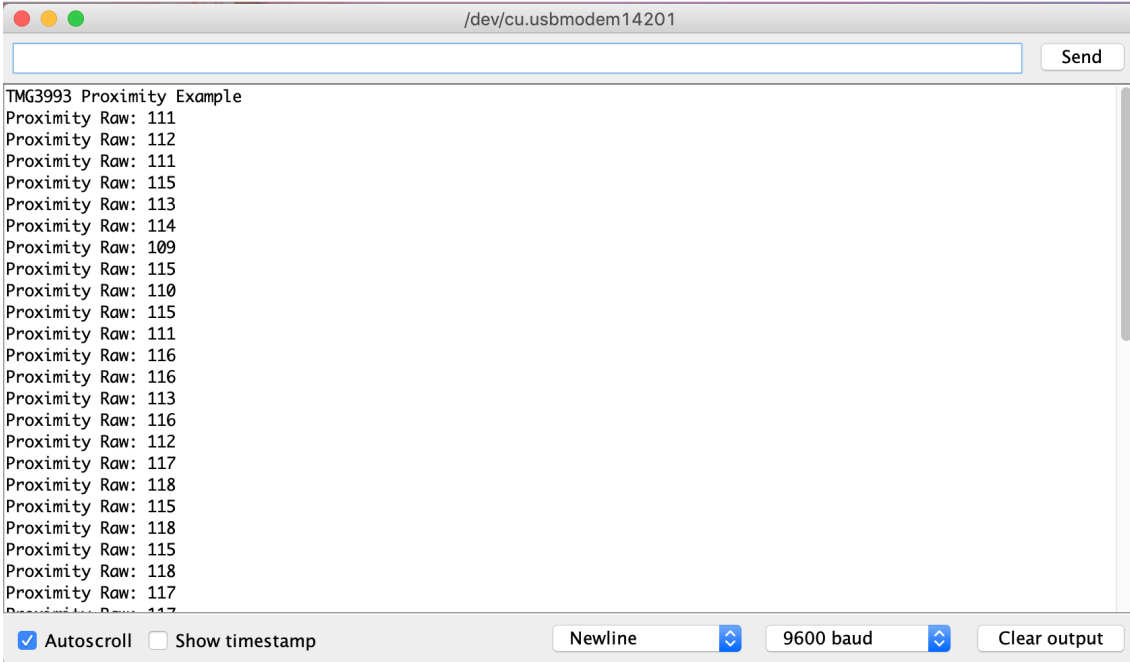
```
19  tmg3993.setupRecommendedConfigForProximity();
20  tmg3993.enableEngines(ENABLE_PON | ENABLE_PEN | ENABLE_
21  }
22
23  void loop()
24  {
25    if (tmg3993.getSTATUS() & STATUS_PVALID)
26    {
27      uint8_t proximity_raw = tmg3993.getProximityRaw();
28      Serial.print("Proximity Raw: ");
29      Serial.println(proximity_raw);
30    }
31    delay(1);
32  }
```

**Attention**

The library file may be updated. This code may not be applicable to the updated library file, so we recommend that you use the first two methods.

**Success**

If everything goes well, the raw (uint16_t) data of the proximity sensor should be able to read from Serial Monitor.



```
TMG3993 Proximity Example
Proximity Raw: 111
Proximity Raw: 112
Proximity Raw: 111
Proximity Raw: 115
Proximity Raw: 113
Proximity Raw: 114
Proximity Raw: 109
Proximity Raw: 115
Proximity Raw: 110
Proximity Raw: 115
Proximity Raw: 111
Proximity Raw: 116
Proximity Raw: 116
Proximity Raw: 113
Proximity Raw: 116
Proximity Raw: 112
Proximity Raw: 117
Proximity Raw: 118
Proximity Raw: 115
Proximity Raw: 118
Proximity Raw: 115
Proximity Raw: 118
Proximity Raw: 117
```

If you would like to run other examples, you may do similar processes.

Example 2

This example is also used to detect proximity, but in interrupt way.

Two thresholds need to be configured - low and high. Any proximity value which is inside this range will trigger nothing, but the value outside this range will trigger events. In detail, when the proximity sensor values are below the low threshold over 10 times, a proximity removed event will be emitted, when the proximity sensor values are higher than the high threshold over 10 times, a proximity detected event will be emitted. The 10 times here can be configured via `setInterruptPersistenceReg` (see `setupRecommendedConfigForProximity` for an example).

- **Step 1.** Download the [Seeed_TM3993](https://github.com/Seeed-Studio/Seeed_TM3993.git) [https://github.com/Seeed-Studio/Seeed_TM3993.git] Library from Github.

- **Step 2.** Refer to [How to install library](https://wiki.seeedstudio.com/How_to_install_Arduino_Library) [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open the example, the same way as in Example 1, but this time choose "Example2-ProximityInterrupt" instead of "Example1". Or copy the following code into Arduino IDE:

```
1  #include <Wire.h>
2
3  #include "Seeed_TMG3993.h"
4
5  TMG3993 tmg3993;
6  int last_interrupt_state = -1;
7
8  void setup()
9  {
10     Serial.begin(9600);
11     Serial.println("TMG3993 Proximity Example");
12
13     Wire.begin();
14
15     if (tmg3993.initialize() == false)
16     {
17         Serial.println("Device not found. Check wiring.");
18         while (1);
19     }
20     tmg3993.setupRecommendedConfigForProximity();
21     tmg3993.setProximityInterruptThreshold(25, 150); //Le.
22     tmg3993.enableEngines(ENABLE_PON | ENABLE_PEN | ENABLE_
23 }
24
25 void loop()
26 {
27     if (tmg3993.getSTATUS() & STATUS_PINT)
28     {
29         uint8_t proximity_raw = tmg3993.getProximityRaw();
30
```

```
31     if (proximity_raw >= 150 && last_interrupt_state != 1)
32         Serial.println("Proximity detected!!!");
33         Serial.print("Proximity Raw: ");
34         Serial.println(proximity_raw);
35         last_interrupt_state = 1;
36     } else if (proximity_raw <= 25 && last_interrupt_state != 0)
37         Serial.println("Proximity removed!!!");
38         Serial.print("Proximity Raw: ");
39         Serial.println(proximity_raw);
40         last_interrupt_state = 0;
41     }
42
43     // don't forget to clear the interrupt bits
44     tmg3993.clearProximityInterrupts();
45 }
46 delay(10);
47 }
```

**Attention**

The library file may be updated. This code may not be applicable to the updated library file, so we recommend that you use the first two methods.

You should see something like this in Serial Monitor after you upload the code to Seeeduino.

```

/dev/cu.usbmodem14201
TMG3
Proximity removed!!!
Proximity Raw: 16
TMG3993 Proximity Example
Proximity removed!!!
Proximity Raw: 15
Proximity detected!!!
Proximity Raw: 255
Proximity removed!!!
Proximity Raw: 19
Proximity detected!!!
Proximity Raw: 201
Proximity removed!!!
Proximity Raw: 13
Proximity detected!!!
Proximity Raw: 255
Proximity removed!!!
Proximity Raw: 14
Proximity detected!!!
Proximity Raw: 238
Proximity removed!!!
Proximity Raw: 13

```

Example 3

This example reads the R, G, B, C sensor values. C is defined as the non-color filtered clear channel in the chip's datasheet. This examples also calculates Lux value for the ambient light based on the equation in the Application Note of this sensor.

- **Step 1.** Download the [Seeed_TM3993](https://github.com/Seeed-Studio/Seeed_TM3993.git) [https://github.com/Seeed-Studio/Seeed_TM3993.git] Library from Github.
- **Step 2.** Refer to [How to install library](https://wiki.seeedstudio.com/How_to_install_Arduino_Library) [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.
- **Step 3.** Restart the Arduino IDE. Open the example, the same way as in Example 1, but this time choose "Example3-Color-Light" instead of "Example1". Or copy the following code into Arduino IDE:

```
1 #include <Wire.h>
```



```
2
3 #include "Seeed_TM3993.h"
4
5 TMG3993 tmg3993;
6
7 void setup()
8 {
9     Serial.begin(9600);
10    Serial.println("TMG3993 Color Example");
11
12    Wire.begin();
13
14    if (tmg3993.initialize() == false)
15    {
16        Serial.println("Device not found. Check wiring.");
17        while (1);
18    }
19    tmg3993.setADCIntegrationTime(0xdb); // the integration
20    tmg3993.enableEngines(ENABLE_PON | ENABLE_AEN | ENABLE_
21 }
22
23 void loop()
24 {
25     if (tmg3993.getSTATUS() & STATUS_AVALID)
26     {
27         uint16_t r,g,b,c;
28         int32_t lux, cct;
29         tmg3993.getRGBCRaw(&r, &g, &b, &c);
30         lux = tmg3993.getLux(r, g, b, c);
31         //the calculation of CCT is just from the `Applicati
32         //from the result of our test, it might have error.
33         cct = tmg3993.getCCT(r, g, b, c);
34
35         Serial.print("RGBC Data: ");
36         Serial.print(r);
37         Serial.print("\t");
38         Serial.print(g);
39         Serial.print("\t");
40         Serial.print(b);
41         Serial.print("\t");
42         Serial.println(c);
```

```
43
44     Serial.print("Lux: ");
45     Serial.print(lux);
46     Serial.print("\tCCT: ");
47     Serial.println(cct);
48     Serial.println("----");
49
50     // don't forget to clear the interrupt bits
51     tmg3993.clearALSInterrupts();
52 }
53 delay(100);
54 }
```



Attention

The library file may be updated. This code may not be applicable to the updated library file, so we recommend that you use the first two methods.

You should see something like this in Serial Monitor after you upload the code to Seeduino.

```
TMG3993 Color Example
RGBC Data: 13 15 13 38
Lux: 80 CCT: 4987
----
RGBC Data: 12 14 12 36
Lux: 74 CCT: 4987
----
RGBC Data: 12 13 12 34
Lux: 70 CCT: 4987
----
RGBC Data: 11 12 11 33
Lux: 70 CCT: 4987
----
RGBC Data: 11 13 11 33
Lux: 68 CCT: 4987
----
RGBC Data: 12 14 12 35
Lux: 74 CCT: 4987
----
RGBC Data: 12 14 12 35
Lux: 74 CCT: 4987
----
RGBC Data: 12 14 12 35
Lux: 74 CCT: 4987
```

Schematic Online Viewer



Resources

- **[Zip]** [Grove - Light&Gesture&Color&Proximity Sensor\(TM39931\) Eagle Files](https://files.seeedstudio.com/wiki/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931/res/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931.zip)
[https://files.seeedstudio.com/wiki/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931/res/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931.zip]

- **[Zip]** [Grove - Light&Gesture&Color&Proximity Sensor\(TMG39931\) Library](https://github.com/Seeed-Studio/Seeed_TMG3993/archive/master.zip) [https://github.com/Seeed-Studio/Seeed_TMG3993/archive/master.zip]
- **[PDF]** [Datasheet TMG39931](https://files.seeedstudio.com/wiki/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931/res/TMG3993.pdf) [https://files.seeedstudio.com/wiki/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931/res/TMG3993.pdf]
- **[PDF]** [PDF version WIKI](https://files.seeedstudio.com/wiki/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931/res/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931.pdf) [https://files.seeedstudio.com/wiki/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931/res/Grove-Light-Gesture-Color-Proximity_Sensor-TMG39931.pdf]

Tech Support

Please do not hesitate to submit the issue into our [forum](https://forum.seeedstudio.com/) [https://forum.seeedstudio.com/].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]