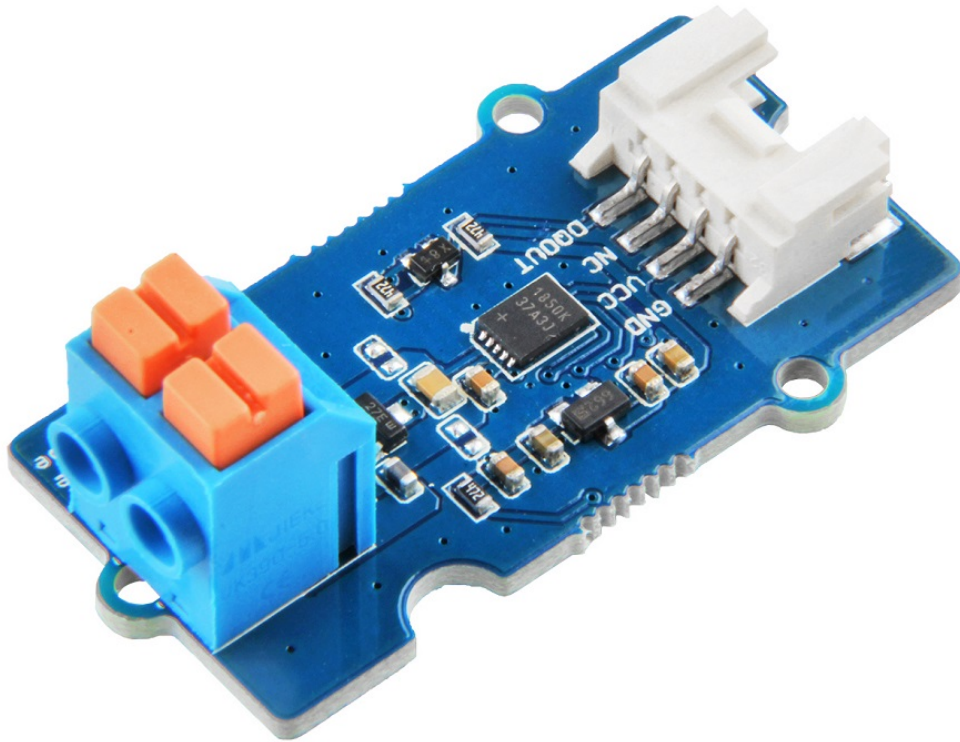# Grove - 1-Wire Thermocouple Amplifier(MAX31850K)

The Grove - 1-Wire Thermocouple Amplifier (MAX31850K) is a thermocouple-to-digital converters with 14-bit resolution and cold-junction compensation. This module is designed to be used in conjunction with a k-type thermocouple. The thermocouples have a much larger measurement range than thermistors. For example,

this k-type thermocouple
[https://www.seeedstudio.com/Thermocouple-Temperature-Sensor-
K-Type-1M-p-3132.html] on our website has a measurement range
of -50℃ to +600℃.

This module is based on the MAX31850K, which integrates
amplifier, ADC and 64-bit ROM. Thanks to the 64-bit ROM, each
device has a unique 64-bit serial code, which allows multiple units
to function on the same 1-Wire bus. Therefore, it is simple to use
one microcontroller (the master device) to monitor temperature
from many thermocouples distributed over a large area.

Again, this module can't work alone, it must work with a k-type
thermocouple, if you do not have one, you can consider
Thermocouple Temperature Sensor K Type-1M
[https://www.seeedstudio.com/Thermocouple-Temperature-Sensor-
K-Type-1M-p-3132.html] in our bazaar.

![Get One Now 🛒]

[https://www.seeedstudio.com/Grove-1-Wire-Thermocouple-
Amplifier-%28MAX31850K%29-p-3159.html]

Version

Product Version	Changes	Released Date
Grove - 1-Wire Thermocouple Amplifier (MAX31850K)	Initial	Aug 2018

Features

- Integrated Cold-Junction Compensation

- Wide Conversion Range: allow readings from -270℃ to +1768℃

- 14-Bit, 0.25℃ Resolution

- Will not work with any other kind of thermocouple except K type

- Detects Thermocouple Shorts to GND or VDD

- Detects Open Thermocouple

> ⚠ **Attention**
> Although this module can convert from -270℃ to +1768℃, the temperature measurment range also limited by the thermocouple you use.

Specification

| Item	Value
Operating Voltage	3.3V/5V
Temperature Resolution	14 bits
Temperature Accuracy	± 2℃
Operating Temperature Range	-40℃ to +125℃
Allow Readings Range	-270℃ to +1768℃
Storage Temperature Range	-65℃ to +150℃
Input Jack	DIP Female Blue-2Pin
Output Interface	1-Wire bus
Size	L: 40mm W: 20mm H: 18mm
Weight	4.8g
Package size	L: 140mm W: 90mm H: 20mm
Gross Weight	11g

Applications

- Medical

- Appliances

- Industrial

- HVAC(Heating, Ventilation and Air Conditioning)

Hardware Overview

Pin Map
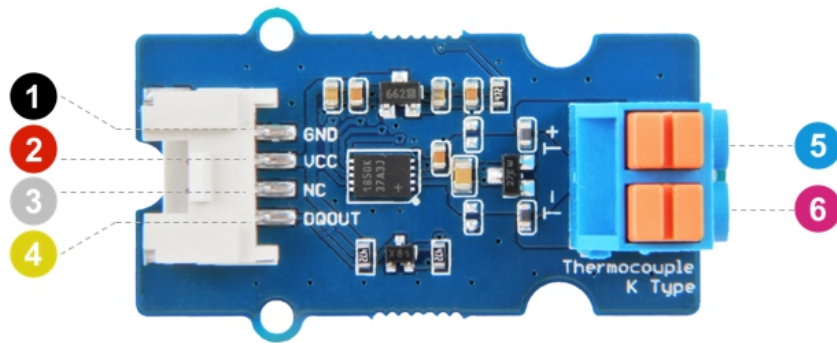
1. GND: connect this module to the system GND
2. VCC: you can use 5V or 3.3V for this module
3. NC: not connected
4. DQOUT: Data Input/Output. Open-drain 1-Wire interface pin.

5. T+: Thermocouple Input, positive pole.
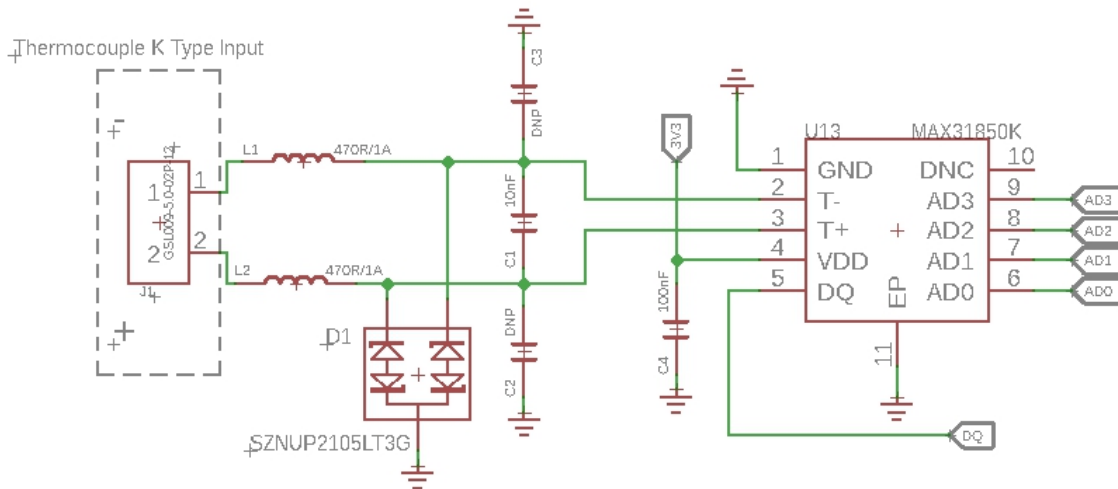6. T -: Thermocouple Input, negative pole
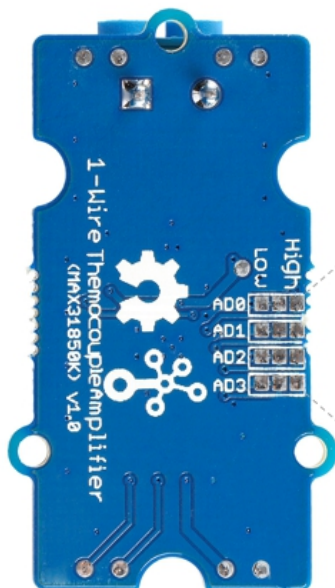
Four location address inputs simplify mapping of individual units to specific locations.

AD0(Least Significant Bit)

AD1

AD2

AD3(Most Significant Bit)
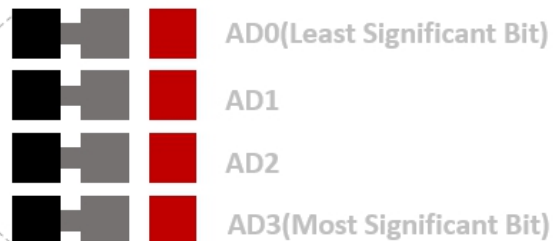
Default location address 0000

Schematic

Input Jack

Because of the small signal levels involved, we take a lot measures to filter the noise.

- **1--L1,L2** We use thermocouple up to 1 meter long. Such long wires can be regarded as antennas, which will receive spatial electric field interference and generate high frequency noise. So we use two inductances to filter the high frequency noise.

- **2--C1** It is strongly recommended by the chip manufacturer to add a 10nF ceramic surfacemount differential capacitor, placed across the T+ and T- pins, in order to filter noise on the thermocouple lines.

- **3--D1** We use the SZNUP2105LT3G DUAL BIDIRECTIONAL VOLTAGE SUPPRESSOR to protect this module from ESD(Electro-Static discharge).

### Bi-directional level shifter circuit
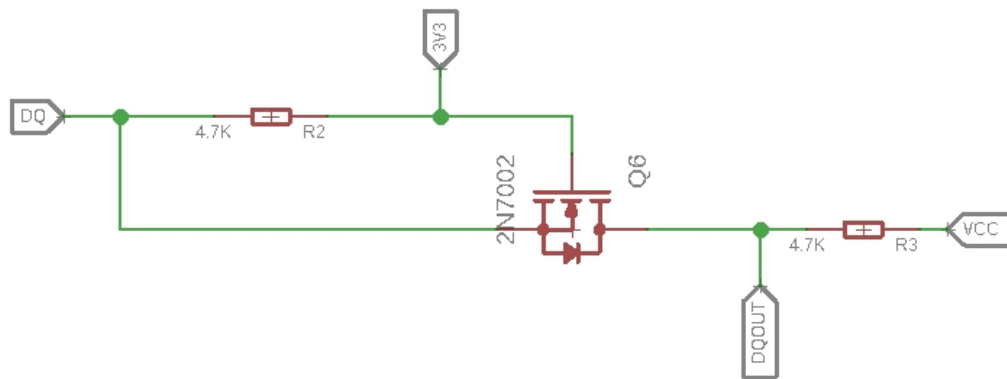
This is a typical Bi-directional level shifter circuit to connect two different voltage section. The left part, **DQ** pin of the MAX31850K use 3.3V, if the Arduino use 5V, this circuit will be needed. In the schematic above, **Q6** is N-Channel MOSFET 2N7002 [https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/2N7002A_datasheet.pdf], which act as a bidirectional switch. In order to better understand this part, you can refer to the AN10441 [https://files.seeedstudio.com/wiki/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/res/AN10441.pdf].

> ✏️ **Note**
>
> In this section we only show you part of the schematic, for the full document please refer to the Resources [https://wiki.seeedstudio.com/Grove-I2C_High_Accuracy_Temperature_Sensor-MCP9808/#resources]

Assembling Drawing
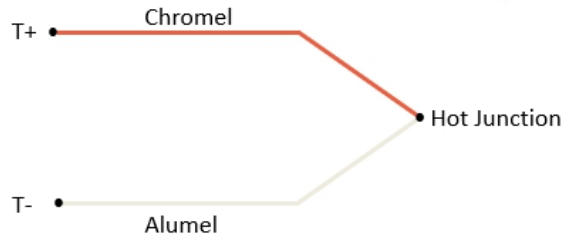
> **Tip**
>
> Please insert the **Red** wire of thermocouple into the **T+** port of the Grove - 1-Wire Themrocouple Amplifier (MAX31850K) , and the **White** wire into **T-** port. If you do not use the k-type thermocouple from our bazaar, the color may be different, please make sure the **+ -** with the seller.

Platforms Supported

Arduino	Raspberry Pi		

⚠️ **Caution**

The platforms mentioned above as supported is/are an indication of the module's software or theoritical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

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]

k-type thermocouple x 2	Grove - I2C Hub

Get One Now
[https://www.seeedstudio.com/Thermocouple-Temperature-Sensor-K-Type-1M-p-3132.html]

Get One Now
[https://www.seeedstudio.com/(I2C-Hub-p-851.html]

◀   ▶

> ✎ **Note**
> **1-** We've offered two software examples, **simple** and **multiple**, the materials requirement above is for the **multiple** example. If you want to test the **simple** example, then the Grove - I2C Hub is not necessary, and only one Grove - 1-Wire Thermocouple Amplifier will be enough.
>
> **2-** We use I2C Hub here not as a I^2C interface, but just as a normal one-to-two transfer interface.

- **Step 1.** Insert the red wire of k-type thermocouple into **T+**, insert the light white wire of k-type thermocouple into **T-**

- **Step 2.** Connect the Grove - I2C Hub to the **D3** port of the Base Shield.

- **Step 3.** Plug the Grove - 1-Wire Thermocouple Amplifier A and B into the Grove - I2C Hub.

- **Step 4.** Plug Grove - Base Shield into Seeeduino.

- **Step 5.** Connect Seeeduino to PC via a USB cable.
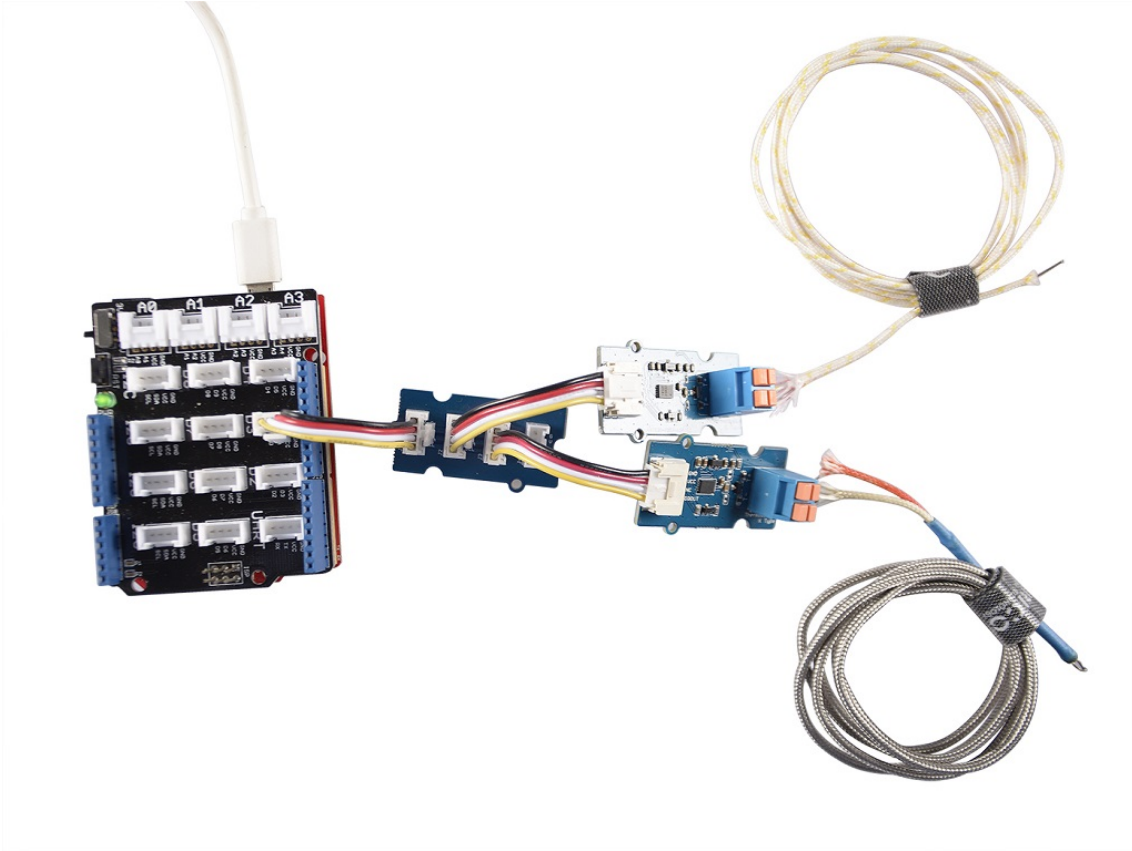
> ✏️ **Note**
>
> If we don't have Grove Base Shield, We also can directly connect this module to Seeeduino as below.

Seeeduino	Grove Cable	Grove - I2C Hub	Grove - 1-Wire Thermocouple Amplifier
GND	Black	G	GND
5V	Red	V	VCC
NC	White	SDA	NC
D3	Yellow	SCL	DQOUT

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

- **Step 1.** Download the Seeed_MAX31850K [https://github.com/Seeed-Studio/Seeed_MAX31850K] Library from Github.

- **Step 2.** Refer to How to install library [https://wiki.seeedstudio.com/How_to_install_Arduino_Library] to install library for Arduino.

- **Step 3.** Unzip the library folder, follow the path to find and open **Multiple.ino** ---> **xxxx\Arduino\libraries\Seeed_MAX31850K-master\examples\Multiple**. xxxx is the path you installed your Arduino.

This folder **\Arduino\libraries\Seeed_MAX31850K-master\examples** contains two examples： **Multiple.ino** and **Simple.ino**.

> Simple.ino--single mode(one host and one slave)
>
> Multiple.ino--multiple mode(one host and multiple slaves,base on rom addressing-one wire search algorithm)

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 <OneWire.h>
2   #include <DallasTemperature.h>
3
4   // Data wire is plugged into port 2 on the Arduino
5   #define ONE_WIRE_BUS 3
6   #define TEMP_RESOLUTION 9
7
8   #define MAX_NUM_OF_DEVICE  10
9
10  // Setup a oneWire instance to communicate with any One
11  OneWire oneWire(ONE_WIRE_BUS);
12
13  // Pass our oneWire reference to Dallas Temperature.
14  DallasTemperature sensors(&oneWire);
15
16  // arrays to hold device addresses
17  DeviceAddress Device_add[MAX_NUM_OF_DEVICE];
18  DeviceAddress insideThermometer, outsideThermometer;
19
20  void setup(void)
21  {
22    // start serial port
23    Serial.begin(115200);
24    Serial.println("Dallas Temperature IC Control Library
25
```

```
26      // Start up the library
27      sensors.begin();
28
29      // Locate devices on the bus
30      Serial.print("Locating devices...");
31      Serial.print("Found ");
32      Serial.print(sensors.getDeviceCount(), DEC);
33      Serial.println(" devices.");
34
35      Serial.print("Parasite power is: ");
36      if (sensors.isParasitePowerMode()) Serial.println("ON
37      else Serial.println("OFF");
38
39      for(int i=0;i<sensors.getDeviceCount();i++)
40      {
41        if(!sensors.getAddress(Device_add[i],i))
42        {
43          Serial.println("Find device error!!");
44        }
45        else
46        {
47          Serial.print("Device [");
48          Serial.print(i);
49          Serial.print("] addr =");
50          printAddress(Device_add[i]);
51        }
52        Serial.println(" ");
53      }
54
55  }
56
57  // function to print a device address
58  void printAddress(DeviceAddress deviceAddress)
59  {
60      for (uint8_t i = 0; i < 8; i++)
61      {
62        // zero pad the address if necessary
63        if (deviceAddress[i] < 16) Serial.print("0");
64        Serial.print(deviceAddress[i], HEX);
65      }
66  }
```

```
67
68   // function to print the temperature for a device
69   void printTemperature(DeviceAddress deviceAddress)
70   {
71     float tempC = sensors.getTempC(deviceAddress);
72     Serial.print("Temp C: ");
73     Serial.print(tempC);
74     Serial.print(" Temp F: ");
75     Serial.print(DallasTemperature::toFahrenheit(tempC));
76   }
77
78   // function to print a device's resolution
79   void printResolution(DeviceAddress deviceAddress)
80   {
81     Serial.print("Resolution: ");
82     Serial.print(sensors.getResolution(deviceAddress));
83     Serial.println();
84   }
85
86   // main function to print information about a device
87   void printData(DeviceAddress deviceAddress)
88   {
89     Serial.print("Device Address: ");
90     printAddress(deviceAddress);
91     Serial.print(" ");
92     printTemperature(deviceAddress);
93     Serial.println();
94   }
95
96   void loop(void)
97   {
98     // call sensors.requestTemperatures() to issue a glob
99     // request to all devices on the bus
100    Serial.print("Requesting temperatures...");
101    sensors.requestTemperatures();
102    Serial.println("DONE");
103
104    for(int i=0;i<sensors.getDeviceCount();i++)
105    {
106      printData(Device_add[i]);
107    }
```

```
108
109  }
```

- **Step 4.** Upload the demo. If you do not know how to upload the code, please check How to upload code [https://wiki.seeedstudio.com/Upload_Code/].

- **Step 5.** Open the **Serial Monitor** of Arduino IDE by click **Tool-> Serial Monitor**. Or tap the `Ctrl` + `Shift` + `M` key at the same time. Then set the baud rate **115200**.

✓ **Success**
If every thing goes well, you will get the result.

```
 1  DONE
 2  Device Address: 3B23211800ing temperatures... 77.00
 3  Requesting temperatures...DONE
 4  Device Address: 3B2321180000005C Temp C: 25.00 Temp F: 7
 5  Requesting temperatures...Dallas Temperature IC Control
 6  Locating devices...Found 2 devices.
 7  Parasite power is: OFF
 8  Device [0] addr =3B4C965D06D80C98
 9  Device [1] addr =3B2321180000005C
10  Requesting temperatures...DONE
11  Device Address: 3B4C965D06D80C98 Temp C: 26.25 Temp F: 7
12  Device Address: 3B2321180000005C Temp C: 25.25 Temp F: 7
13  Requesting temperatures...DONE
14  Device Address: 3B4C965D06D80C98 Temp C: 26.25 Temp F: 7
15  Device Address: 3B2321180000005C Temp C: 25.00 Temp F: 7
16  Requesting temperatures...DONE
17  Device Address: 3B4C965D06D80C98 Temp C: 26.25 Temp F: 7
18  Device Address: 3B2321180000005C Temp C: 25.25 Temp F: 7
19  Requesting temperatures...DONE
20  Device Address: 3B4C965D06D80C98 Temp C: 26.00 Temp F: 7
21  Device Address: 3B2321180000005C Temp C: 25.25 Temp F: 7
22  Requesting temperatures...DONE
23  Device Address: 3B4C965D06D80C98 Temp C: 26.00 Temp F: 7
```

```
24   Device Address: 3B2321180000005C Temp C: 25.25 Temp F: 7
```

Schematic Online Viewer

Resources

- **[Zip]** Grove - 1-Wire Thermocouple Amplifier eagle files
  [https://files.seeedstudio.com/wiki/Grove-

1_Wire_Themocouple_Amplifier-MAX31850K/res/Grove-1-
Wire_Themocouple_Amplifier-MAX31850K.zip]

- **[Zip]** Seeed_MAX31850K Library [https://github.com/Seeed-
  Studio/Seeed_MAX31850K/archive/master.zip]

- **[PDF]** Datasheet of MAX31850
  [https://files.seeedstudio.com/wiki/Grove-
  1_Wire_Themocouple_Amplifier-
  MAX31850K/res/Max31850.pdf]

Tech Support

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

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