Grove - Sound Sensor User Manual

Release date： 2015/9/23

Version： 1.0

Wiki:http://www.seeedstudio.com/wiki/index.php?title=Twig_-_Sound_Sensor

Bazaar: http://www.seeedstudio.com/depot/Grove-Sound-Sensor-p-752.html

Document Revision History

Revision	Date	Author	Description
1.0	Sep 23, 2015	Jiankai.li	Create file

Contents

Document Revision History ····· 2
1. Introduction ····· 2
2. Features ····· 3
3. Specifications ····· 4
4. Usage ····· 5
   4.1 Standalone ····· 5
   4.2 With Arduino ····· 5
   4.3 With TI LaunchPad ····· 7
   4.4 With Raspberry Pi ····· 9
   4.5 With Beaglebone Green ····· 10
5. Availability ····· 14
6. Resources ····· 15

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.

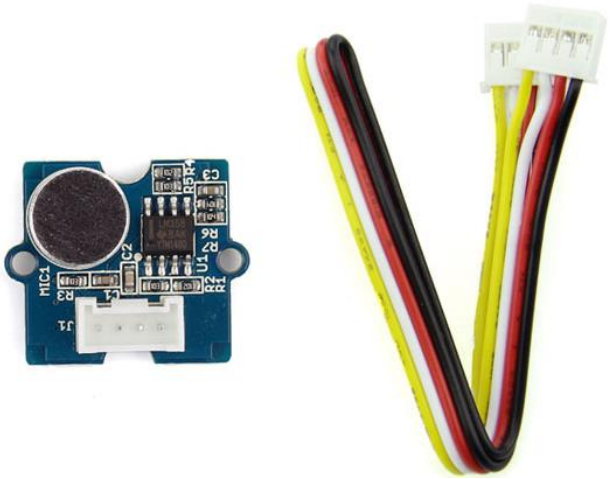
Under the supervision of Seeed Technology Inc., this manual has been compiled and published which covered the latest product description and specification. The content of this manual is subject to change without notice.

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

Grove - Sound Sensor can detect the sound strength of the environment. The main component of the module is a simple microphone, which is based on the LM358 amplifier and an electret microphone. This module's output is analog and can be easily sampled and tested by a Seeeduino.



2. Features

- Easy to use sound sensor module
- Provides analog output signal
- Easily integrates with Logic modules on the input side of Grove circuits
- Uses Standard 4-pin Grove Cables to connect to other Grove modules such as Grove Power Modules, Logic Gatesand Grove - Base Shield

3. Specifications

- Operating voltage range: 4-12V
- Operating current (Vcc=5V): 4-8mA
- Voltage Gain (VS=6V, f=1 kHz): 26dB
- Microphone sensitivity (1Khz): 52-48dB
- Microphone Impedance: 2.2KΩ
- Microphone Frequency: 16-20Khz
- Microphone S/N ratio: 54dB

4. Usage

4.1 Standalone

Follow these steps to build a sample Grove circuit using this module but without using any microcontroller board:

Connect the sound sensor module to the input side of your circuit (to the left of the power module).

On the output side of the circuit, you may use a range of User Interface modules (Grove - Red LED, Grove - LED String Light, Grove - Mini Fan, Grove - Buzzer, Grove - Recorder etc.)

Power up the circuit when complete.

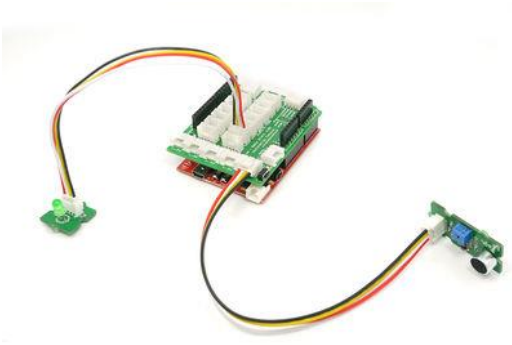
The sound sensor module can now be used to trigger an output. For example:

- When used in conjunction with a Grove - Red LED output module, observe that the LED turns ON when the sound sensor detects audio and turns off in silence. The same behavior can be seen when the sound sensor is used with the Grove - LED String Light module - when there is sound, the lights will turn ON, and when there is silence, the lights will turn OFF.
- Add a Grove - NOT module between the sound sensor and the power module to reverse the logic that triggers the LEDs on the Grove - Red LED or Grove - LED String Light modules. With the addition of the Logical NOT gate on the input side, you should see that the LEDs remain OFF when sound or music is detected, and turn ON when there is silence.
- You can use either the Grove - USB Power module or the Grove - DC Jack Power module for building this Grove circuit.

4.2 With Arduino

Follow these simple steps to build a Grove circuit using the light sensor:

When using the module in conjunction with an Arduino or a Seeeduino, use the Grove - Base Shield and connect the Grove - Sound Sensor module to the shield using a designated Grove Interface. Also attach an output module such as a Grove - LED or Grove - Red LED which will get triggered based on input received from the sound sensor (shown below).



Upload the following sample sketch to make the LED turn ON and OFF based on input from the sound sensor:

```
// Function: If the sound sensor senses a sound that is up to the threshold you set in the code, the LED is on for 200ms.
// Hardware: Grove - Sound Sensor, Grove - LED

/*macro definitions of the sound sensor and the LED*/
#define SOUND_SENSOR A0
#define LED 3      // the number of the LED pin

#define THRESHOLD_VALUE 400//The threshold to turn the led on 400.00*5/1024 = 1.95v
void setup()
{
    Serial.begin(9600);
    pins_init();
}

void loop()
{
     int sensorValue = analogRead(SOUND_SENSOR);//use A0 to read the electrical signal
     Serial.print("sensorValue ");
         Serial.println(sensorValue);
         if(sensorValue > THRESHOLD_VALUE)
     {
          turnOnLED();//if the value read from A0 is larger than 400,then light the LED
          delay(200);
     }
     turnOffLED();
}

void pins_init()
{
     pinMode(LED, OUTPUT);
```

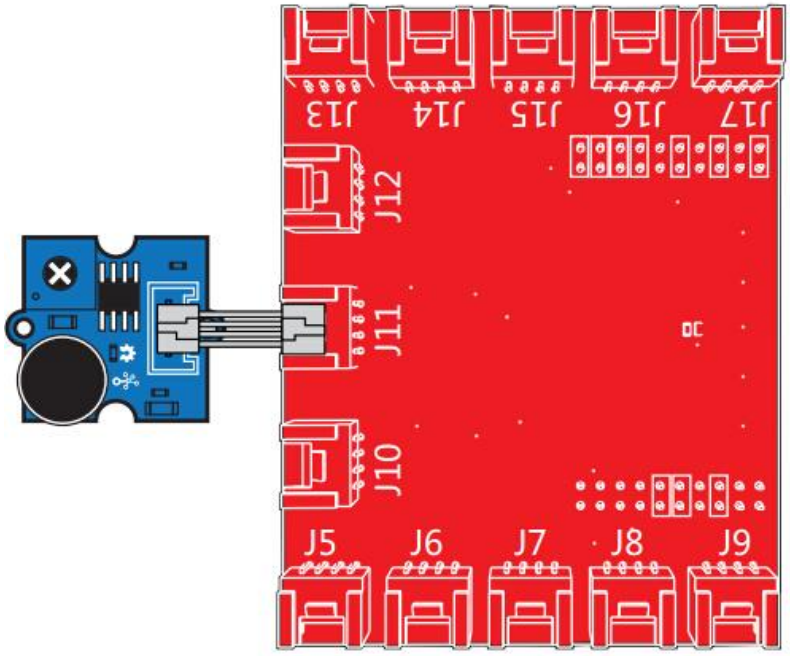
```
    pinMode(SOUND_SENSOR, INPUT);
}
void turnOnLED()
{
    digitalWrite(LED,HIGH);
}
void turnOffLED()
{
    digitalWrite(LED,LOW);
}
```

If the sound of the environment is greater than the threshold, then the LED will light up.

4.3 With TI LaunchPad

Turning on a LED by Sound (Sound Sensor)

This example program shows how to use the sound sensor (microphone. When the incoming sound amplitude exceeds a threshold, we will turn an LED on for 1 second.

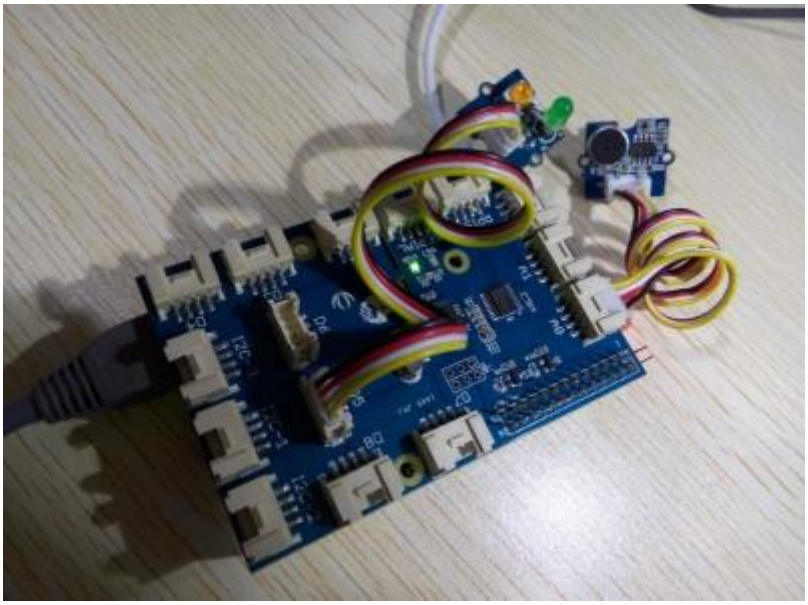



```
/*
 Sound Sensor
 A simple program demonstrate sound sensor senses a sound that is up to the threshold you set
 in the code, the LED is on for 1s.

 The circuit:
 * sig pin of the sound sensor to the analog pin 24 (J6 plug on Grove Base BoosterPack)
 * one side pin (either one) to ground
 * the other side pin to +VCC
```

```
 * LED anode (long leg) attached to RED_LED
 * LED cathode (short leg) attached to ground

 * Note:


 This example code is in the public domain.

 http://www.seeedstudio.com/depot/Grove-Sound-Sensor-p-752.html?cPath=25_128
 */

#define SOUND_SENSOR       24           /* sound sensor pin */
#define LED                RED_LED      /* LED pin */
#define THRESHOLD_VALUE    3000          /* Depending on your LaunchPad's ADC resolution you may
have to adjust the threshold */

#define ON                 HIGH                    /* led on */
#define OFF                LOW                     /* led off */
#define _handle_led(x)     digitalWrite(LED, x)    /* handle led */

/* Global Variables */
int sound_value = 0;

void setup() {

    /* Initialize led pin */
    pinMode(LED, OUTPUT);
    digitalWrite(LED, LOW);
}

void loop() {

    /* read the sound value */
    sound_value = analogRead(SOUND_SENSOR);
    /* if the value is larger than threshold, turn on led */
    if(sound_value > THRESHOLD_VALUE) {
        _handle_led(ON);
        delay(1000);
    }
    _handle_led(OFF);
}
```

4.4 With Raspberry Pi

You should connect to GrovePi+ with Grove - Sound Sensor and Grove - Green LED. Such as the following picture. When sound level greater than a certain value,the LED will turn on.



```
# GrovePi + Grove Sound Sensor + Grove LED

import time
import grovepi

# Connect the Grove Sound Sensor to analog port A0
# SIG,NC,VCC,GND
sound_sensor = 0

# Connect the Grove LED to digital port D5
# SIG,NC,VCC,GND
led = 5

grovepi.pinMode(sound_sensor,"INPUT")
grovepi.pinMode(led,"OUTPUT")

# The threshold to turn the led on 400.00 * 5 / 1024 = 1.95v
threshold_value = 400

while True:
    try:
        # Read the sound level
        sensor_value = grovepi.analogRead(sound_sensor)
```

```
        # If loud, illuminate LED, otherwise dim
        if sensor_value > threshold_value:
            grovepi.digitalWrite(led,1)
        else:
            grovepi.digitalWrite(led,0)

        print "sensor_value =", sensor_value
        time.sleep(.5)

    except IOError:
        print "Error"
```

Run the program

- Find the path to the file(According to your own path)

```
cd GrovePi/Software/Python/
```

- Run Program

```
sudo python grove_sound_sensor.py
```

4.5 With Beaglebone Green

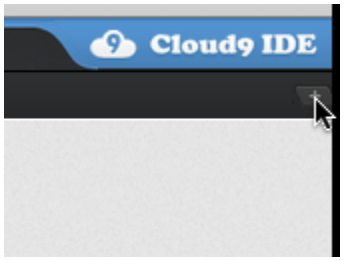
To begin editing programs that live on BBG, you can use the Cloud9 IDE.

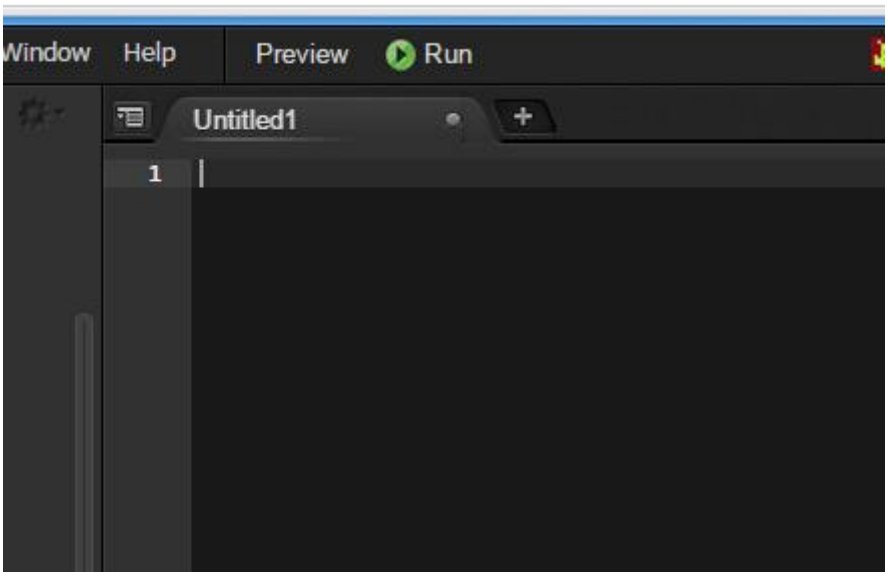
As a simple exercise to become familiar with Cloud9 IDE, creating a simple application to blink one of the 4 user programmable LEDs on the BeagleBone is a good start.

If this is your first time to use Cloud9 IDE, please follow this **link**.

Step1: Set the Grove - UART socket as a Grove - GPIO Socket, just follow this **link**.

Step2: Click the "+" in the top-right to create a new file.





Step3: Copy and paste the following code into the new tab

```
from Adafruit_I2C import Adafruit_I2C
import time

ADDR_ADC121 = 0x50

REG_ADDR_RESULT = 0x00
REG_ADDR_ALERT = 0x01
REG_ADDR_CONFIG = 0x02
REG_ADDR_LIMITL = 0x03
REG_ADDR_LIMITH = 0x04
REG_ADDR_HYST = 0x05
REG_ADDR_CONVL = 0x06
REG_ADDR_CONVH = 0x07

i2c = Adafruit_I2C(ADDR_ADC121)

class I2cAdc:
    def __init__(self):
        i2c.write8(REG_ADDR_CONFIG, 0x20)

    def read_adc(self):
        "Read ADC data 0-4095."
        data_list = i2c.readList(REG_ADDR_RESULT, 2)
        #print 'data list', data_list
        data = ((data_list[0] & 0x0f) << 8 | data_list[1]) & 0xfff
        return data
```

```
if __name__ == '__main__':
    # Connect the Grove - I2C ADC to I2C Grove port of Beaglebone Green.
    adc = I2cAdc()
    while True:
        print 'sensor value ', adc.read_adc()
        time.sleep(.2)
```

Step4: Save the file by clicking the disk icon with a name "grove_i2c_adc.py".

Step5: Create a new file Copy the following code into the new tab and save it with the .py extension.

```
import time
from logo import print_seeedstudio
import grove_i2c_adc
import Adafruit_BBIO.GPIO as GPIO

# Reference voltage of ADC is 3.0v
ADC_REF = 3.0

# Vcc of the grove interface is normally 3.3v
GROVE_VCC = 3.3



adc = grove_i2c_adc.I2cAdc()

def read_sound_sensor_values():
    "Read voltage values from Grove Sound Sensor"
    total_value = 0
    for index in range(5):
        sensor_value = adc.read_adc()
#        print "sensor_value = ", sensor_value
        total_value += sensor_value
        time.sleep(0.01)
#    print "total_value = ", total_value
    average_value = float(total_value / 5)

    voltage_value = average_value / 4095 * ADC_REF * 2
    return voltage_value

# Function: If the sound sensor senses a sound that is up to the threshold you set in the code, the
LED is on for 1s.
# Hardware: Grove - I2C ADC, Grove - Sound Sensor, Grove - LED(You can also replace Grove - LED with
Grove - Buzzer.)
```

```
# Note: Use P9_22(UART2_RXD) as GPIO.
# Connect the Grove LED to UART Grove port of Beaglebone Green.
# Connect the Grove - I2C ADC to I2C Grove port of Beaglebone Green, and then connect the Grove -
Sound Sensor to Grove - I2C ADC.
if __name__== '__main__':
    print_seeedstudio()

    while True:
        try:
            # Read voltage values from Grove Sound Sensor
            sensor_voltage_value = read_sound_sensor_values()

            print "sensor_voltage_value = ", sensor_voltage_value
#            time.sleep(2)

        except IOError:
            print "Error"
```

Step6: Connect Grove Sound Sensor to Grove I2C ADC which is connected to Grove I2C socket on BBG.

Step7: Run the code. You'll find that the terminal outputs Sound value every 2 seconds.

5. Availability

This Grove module is available as part of the following Grove Kit Series:

- Grove Mixer Pack V2
- Grove - Mixer Pack
- Grove - Toy Kit

Alternatively, it can be bought stand-alone here at the Seeed Studio Bazaar.

6. Resources

- Grove - Sound Sensor Schematic (pdf)
- Grove - Sound Sensor Source Files v1.3
- Grove - Sound Sensor Eagle files v0.9b
- LM386pdf
- how to upload code

Стандарт Электрон Связь

Мы молодая и активно развивающаяся компания в области поставок электронных компонентов. Мы поставляем электронные компоненты отечественного и импортного производства напрямую от производителей и с крупнейших складов мира.

Благодаря сотрудничеству с мировыми поставщиками мы осуществляем комплексные и плановые поставки широчайшего спектра электронных компонентов.

Собственная эффективная логистика и склад в обеспечивает надежную поставку продукции в точно указанные сроки по всей России.

Мы осуществляем техническую поддержку нашим клиентам и предпродажную проверку качества продукции. На все поставляемые продукты мы предоставляем гарантию .

Осуществляем поставки продукции под контролем ВП МО РФ на предприятия военно-промышленного комплекса России , а также работаем в рамках 275 ФЗ с открытием отдельных счетов в уполномоченном банке. Система менеджмента качества компании соответствует требованиям ГОСТ ISO 9001.

Минимальные сроки поставки, гибкие цены, неограниченный ассортимент и индивидуальный подход к клиентам являются основой для выстраивания долгосрочного и эффективного сотрудничества с предприятиями радиоэлектронной промышленности, предприятиями ВПК и научно-исследовательскими институтами России.

С нами вы становитесь еще успешнее!

Наши контакты:

Телефон: +7 812 627 14 35

Электронная почта: sales@st-electron.ru

Адрес: 198099, Санкт-Петербург,
Промышленная ул, дом № 19, литера Н,
помещение 100-Н Офис 331