



Grove – Button User Manual

Release date: 2015/9/22

Version: 1.0

Wiki: http://www.seeedstudio.com/wiki/Grove_-_Button

Bazaar: http://www.seeedstudio.com/depot/Grove-Button-p-766.html?cPath=85_50

Document Revision History

Revision	Date	Author	Description
1.0	Sep 22, 2015	Loovee	Create file

Contents

Document Revision History	2
1. Introduction	2
2. Features	3
3. Usage	4
3.1 Standalone	4
3.2 With Arduino	4
3.3 With Raspberry Pi	6
3.3.1 Run The Program	7
4. Availability	8
5. Resources	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

The Grove - Button is a momentary push button. It contains one independent "momentary on/off" button. "Momentary" means that the button rebounds on its own after it is released. The button outputs a HIGH signal when pressed, and LOW when released. The button signals the SIG Pin of the Grove Interface while NC is not used at all. There are two versions of this button available as shown in the pictures. The only difference between them is the direction of the Grove socket.



Grove - Button(P):



2. Features

- Easy to use momentary ON/OFF button
- Uses Standard 4-pin Grove Cables to connect to other Grove modules such as Grove Power Modules and Grove - Base Shield

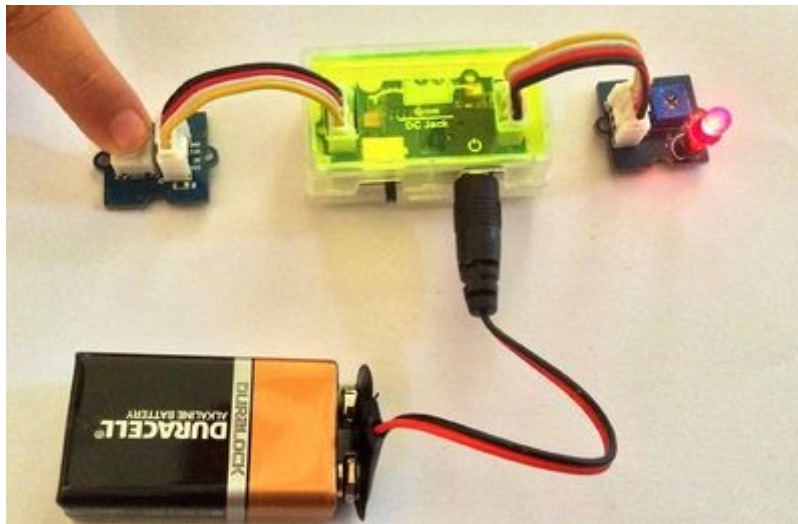
3. Usage

3.1 Standalone

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

- Connect the button 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 button 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 you press the button and turns OFF when you stop pressing it.



In terms of power modules, use either the [Grove - USB Power](#) module or the [Grove - DC Jack](#) Power module for the Grove circuit.

3.2 With Arduino

Follow these simple steps to build a Grove circuit using the momentary ON/OFF button:

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

- Upload the following sample sketch to make the LED turn ON and OFF based on input from Grove - Button:
- //Turns on and off a light emitting diode(LED) connected to digital pin 13, when pressing a pushbutton attached to pin 2.

```
/*
The circuit:
* LED attached from pin 13 to ground
* pushbutton attached to pin 2 from +5V
* 10K resistor attached to pin 2 from ground

* Note: on most Arduinos there is already an LED on the board
attached to pin 13.

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Button
*/

// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;     // the number of the LED pin

// variables will change:
int buttonState = 0;       // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:

```



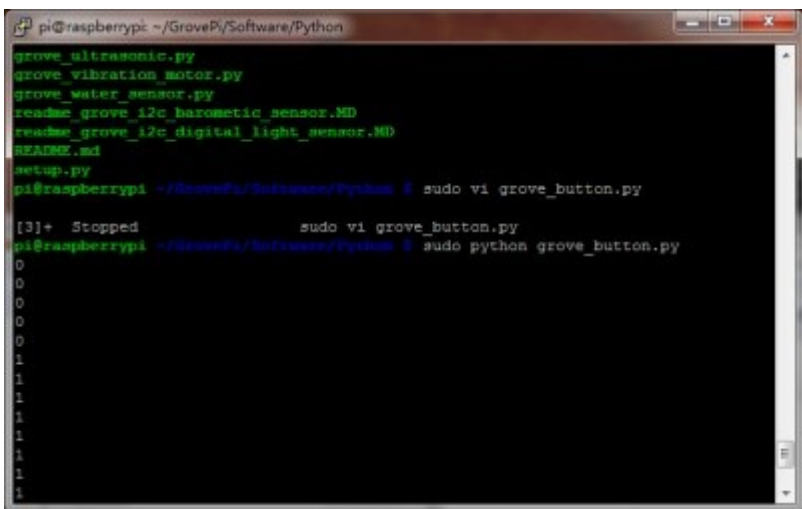
```

digitalWrite(ledPin, HIGH);
}
else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
}
}

```

3.3 With Raspberry Pi

This is a simple example of Raspberry Pi. Run Program and press button, it will print 1 on the terminal, else print 0. Such as the picture below.



http://www.seeedstudio.com/wiki/Grove_-_Button

```
import time
import grovepi

# Connect the Grove Button to digital port D3
# SIG, NC, VCC, GND
button = 3

grovepi.pinMode(button, "INPUT")

while True:
    try:
        print grovepi.digitalRead(button)
        time.sleep(.5)

    except IOError:
        print "Error"
```

3.3.1 Run The Program

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

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

- Run Program

```
sudo python grove_button.py
```

4. Availability

The standard Grove - Button module is available as part of the following [Grove Kit Series](#):

- [Grove - Mixer Pack V2](#)
- [Grove - Mixer Pack](#)
- [Grove - Starter Kit V1.1b](#)

Alternatively, it can be bought stand-alone [here](#) at the [Seeed Studio Bazaar](#). To buy the Panel Mount version of the module, go [here](#).

5. Resources

- [Grove-Button \(Eagle Files\)](#) for the version included in [Grove Mixer Pack V2](#)
- [how to upload code](#)