Grove - LED Button



The Grove - LED Button is composed of Grove - Yellow Button, Grove - Blue LED Button and Grove - Red LED Button. This button is stable and reliable with a 100 000 times long life. With the build-in LED, you can apply it to many interesting projects, it is really useful to use the LED to show the status of the button. We use a highquality N-Channel MOSFET to control the LED to ensure the high swithching speed and a low consumption.All in all, you want some relly awesome button? Here you go ...

Get One Now 📜

[https://www.seeedstudio.com/Grove-Yellow-LED-Button-p-3101.html]



[https://www.seeedstudio.com/Grove-Blue-LED-Button-p-3104.html]



[https://www.seeedstudio.com/Grove-Red-LED-Button-p-3096.html]

Version

Product Version	Changes	Released Date
Grove-LED Button	Initial	Jun 2018

Version

Product Version	Changes	Released Date
Grove-LED Button	Initial	Jun 2018

Version

Product Version	Changes	Released Date
Grove-LED Button	Initial	Jun 2018

Features

- Long operating life
- Easy to use
- Grove Digital interface

Specification

ltem	Value
Working voltage	3.3V/5V
Operating Life without Load	100 000 times
LED rated current	50mA
Press Resistance ¹	<100mΩ
Release Resistance ²	>100MΩ
Size	L: 40mm W: 20mm H: 13mm
Weight	4.3g
Package size	L: 140mm W: 90mm H: 10mm
Gross Weight	11g



Tips

1,2- If you want to measure the resistance, please take the key cap off the board. Otherwise you will get the value of the equivalent resistance of the board instead of the true resistance of the key cap.

Hardware Overview

Pin Map



Schematic



SIG1 is the the LED control signal, the default value is low, so the N-Channel MOSFET is off, the LED is off too. When SIG1 becomes high, the N-Channel MOSFET trun on, and the LED light on.

SIG2 connect to the button pin. With a pull-up resistance, the default value of **SIG2** is high. When you press the button, the voltage is pulled low, the **SIG2** becomes to low.

Platforms Supported



Getting Started



Tips

In this part, we use the Grove - Red LED Button as an example. The following parts also apply to Yellow and Blue.

Play With Arduino

Hardware

Materials required



- Step 1. Grove- Red LED Button to port D3 of Grove-Base Shield.
- Step 2. Plug Grove Base Shield into Seeeduino.
- Step 3. 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- Red LED Button
5V	Red
GND	Black
SIG2	White
SIG1	Yellow

Software



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.** Open the Arduino IDE and create a new file, then copy the following code into the new file.

```
1
   #include "Arduino.h"
2
3
   #define LED_MODE
4
5
6
   const int ledPin = 3; // the number of the LED pin,
   const int buttonPin = 4; // the number of the pushbut<sup>.</sup>
7
8
   const boolean breathMode = true; // if or not the led l
9
10 // Variables will change:
11 int ledState = LOW; // the current state of the
12 int ledFadeValue = 0;
13 int ledFadeStep = 5;
14 int ledFadeInterval = 20; //milliseconds
15 int buttonState; // the current reading from
16 int lastButtonState = HIGH; // the previous reading from
17
18 unsigned long lastDebounceTime = 0; // the last time the
   unsigned long debounceDelay = 50; // the debounce time
19
   unsigned long lastLedFadeTime = 0;
20
21
22 void setup() {
23 pinMode(buttonPin, INPUT);
     pinMode(ledPin, OUTPUT);
24
25
     digitalWrite(ledPin, ledState);
26 }
27
28 void loop() {
     int reading = digitalRead(buttonPin);
29
30
```

```
31
32
     if (reading != lastButtonState) {
33
34
        lastDebounceTime = millis();
35
36
37
     if ((millis() - lastDebounceTime) > debounceDelay) {
38
39
40
41
42
       if (reading != buttonState) {
43
          buttonState = reading;
44
45
   #if LED MODE == 1
46
          if (buttonState == LOW) { //button is pressed
47
              ledState = !ledState;
48
             ledFadeValue = 0;
              lastLedFadeTime = millis();
49
50
51 #else
52
          if (buttonState == LOW) { //button is pressed
53
           ledState = HIGH;
54
            ledFadeValue = 0;
            lastLedFadeTime = millis();
55
56
         } else {
            ledState = LOW;
57
58
59 #endif
60
61
62
63
64
     if (breathMode && ledState != LOW) {
       if (millis() - lastLedFadeTime > ledFadeInterval) {
65
66
          lastLedFadeTime = millis();
          analogWrite(ledPin, ledFadeValue);
67
68
         ledFadeValue += ledFadeStep;
         if (ledFadeValue > 255){
69
70
            ledFadeValue = 255 - ledFadeStep;
71
            ledFadeStep = -ledFadeStep;
```

```
72
          } else if (ledFadeValue < 0) {</pre>
73
             ledFadeValue = 0;
74
            ledFadeStep = -ledFadeStep;
75
76
77
      } else {
        digitalWrite(ledPin, ledState);
78
79
80
81
      lastButtonState = reading;
82 }
```

6

Tip

In this demo, we choose mode 1 which is the toggle mode, you can change the line 4 **#define LED_MODE 1** into **#define LED_MODE 2** to use the follow mode.

- Step 2. 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 3.** Now, try to press you button, you will see the LED light on with a fade on/fade off effect.

It should be like:



Play With Raspberry Pi

Hardware

• Step 1. Things used in this project:



- ◀
- Step 2. Plug the Grove Base Hat into Raspberry.
- Step 3. Connect the red LED button to D5 port of the Base Hat.
- Step 4. Connect the Raspberry Pi to PC through USB cable.



Note

For step 3 you are able to connect the LED button to **any GPIO Port** but make sure you change the command with the corresponding port number.

Software

Attention

If you are using **Raspberry Pi with Raspberrypi OS >= Bullseye**, you have to use this command line **only with Python3**.

- Step 1. Follow Setting Software
 [https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_
 Pi/#installation] to configure the development environment.
- Step 2. Download the source file by cloning the grove.py library.



• Step 3. Excute below commands to run the code.

```
1 cd grove.py/grove
2 sudo python3 grove_ryb_led_button.py 5
```

Following is the grove_ryb_led_button.py code.

```
import time
1
2
   from grove.button import Button
3
   from grove.factory import Factory
4
5
   class GroveLedButton(object):
6
        def __init__(self, pin):
7
8
            self. led = Factory.getOneLed("GPIO-HIGH", pin)
9
10
            self. btn = Factory.getButton("GPIO-LOW", pin +
11
            self. on event = None
            self.__btn.on_event(self, GroveLedButton.__handle
12
13
14
       @property
        def on_event(self):
15
16
            return self.__on_event
17
18
       @on_event.setter
19
        def on_event(self, callback):
20
            if not callable(callback):
21
                return
22
            self.__on_event = callback
23
24
        def __handle_event(self, evt):
25
            if callable(self. on event):
26
```

```
27
                self.__on_event(evt['index'], evt['code'], e'
28
                return
29
30
            self. led.brightness = self. led.MAX BRIGHT
31
            event = evt['code']
32
            if event & Button.EV SINGLE CLICK:
33
                self. led.light(True)
                print("turn on LED")
34
35
            elif event & Button.EV_DOUBLE_CLICK:
                self. led.blink()
36
37
                print("blink
                               LED")
            elif event & Button.EV LONG PRESS:
38
39
                self.__led.light(False)
                print("turn off LED")
40
41
42
43
   Grove = GroveLedButton
44
45
   def main():
        from grove.helper import SlotHelper
46
        sh = SlotHelper(SlotHelper.GPI0)
47
        pin = sh.argv2pin()
48
49
50
        ledbtn = GroveLedButton(pin)
51
52
53
54
        # define a customized event handle your self
        def cust on event(index, event, tm):
55
56
            print("event with code {}, time {}".format(event
57
58
        ledbtn.on event = cust on event
59
        while True:
60
            time.sleep(1)
61
62
63
64 if __name__ == '__main__':
65
        main()
```

Success

If everything goes well, you will be able to see the LED turns on if you press it and turns off if you long press it. If you double click the LED button, the LED will blink.

```
pi@raspberrypi:~/grove.py/grove $ python3 grove_ryb_led_
1
   Hat Name = 'Grove Base Hat RPi'
2
3
   turn on LED
4
   turn on LED
5
   blink
           LED
6
   turn on LED
   turn off LED
7
8
   ^CTraceback (most recent call last):
9
     File "grove_ryb_led_button.py", line 101, in <module>
       main()
10
     File "grove_ryb_led_button.py", line 97, in main
11
       time.sleep(1)
12
13
```

You can quit this program by simply press Ctrl+C.

Schematic Online Viewer

Resources

[Zip] Grove-LED Button Eagle file
 [https://files.seeedstudio.com/wiki/Grove-Red_LED_Button/res/Grove-Red_LED_Button.zip]

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]