Grove - Mech Keycap

The Grove-Mech keycap is a mechanical switch with a build-in LED. The 255 full color RGB LED makes it simple and easy to show the statues of your switch. This keycap is very reliable with 20,000,000 times press operating life.

You will find that this is an interesting and stable module to make some really fun project or product. Actually, you even can make a

mechanical keyboard using several Grove-Mech keycaps.

Tips
20,000,000 cycles of operation shall be performed continuously at a rate of 300 cycles per minute without load.

Get One Now

[https://www.seeedstudio.com/-Grove-VOC-and-eCO2-Gas-Sensor-(SGP30)-p-3071.html]

Features

- Programmable LED
- Reliable mechanical structure
- Extremely long operating Life

Specification

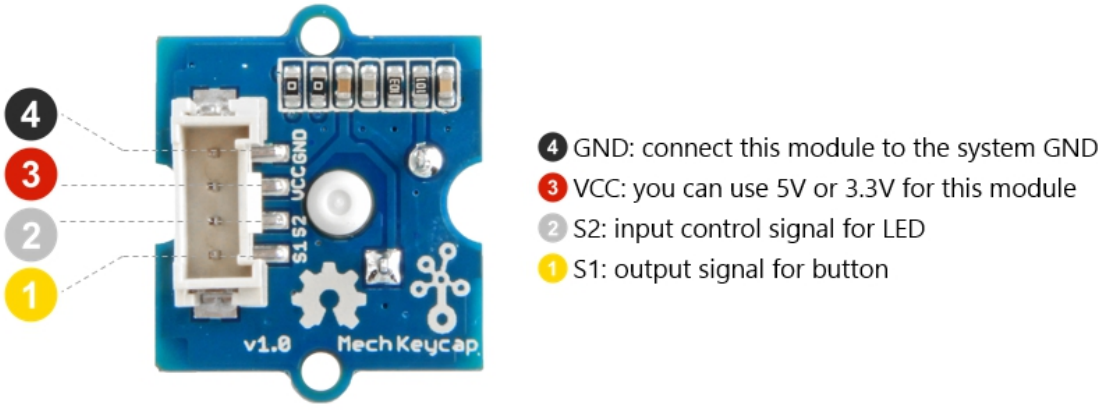
Item	Value
Working Voltage	3v-5v
Insulation Resistance	100MΩ Min.
Contract Resistance	200 mΩ Max.
Operating Life without Load	20,000,000

Applications

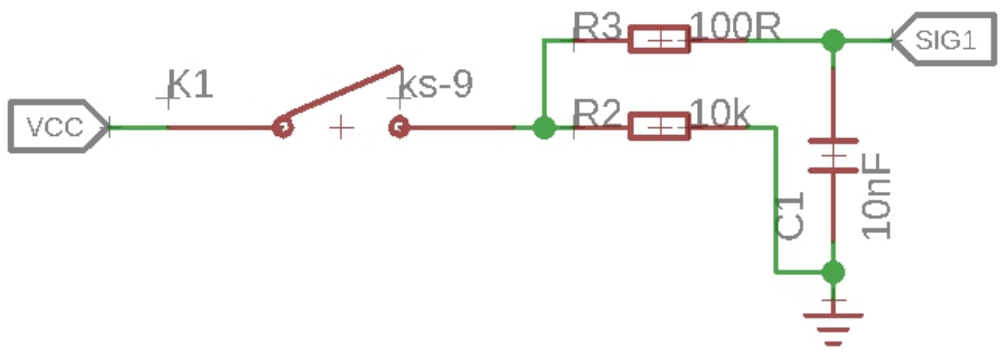
- automotive devices
- visual devices
- home electrical appliances
- information devices

Hardware

Pin Map



Schematic



The K1 is attached to the button, when the key is opened, the **SIG1** will be pulled-down by R2, then the output of **SIG1** should be low. Once the button is pushed, the K1 will be closed and the **SIG1** will connected to **VCC**, then the output of **SIG1** becomes high.

Note
In this section we only show you part of the schematic, for the full document please refer to the Resources [/#resources]

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

Note
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.

Play With Arduino

Hardware

Materials required

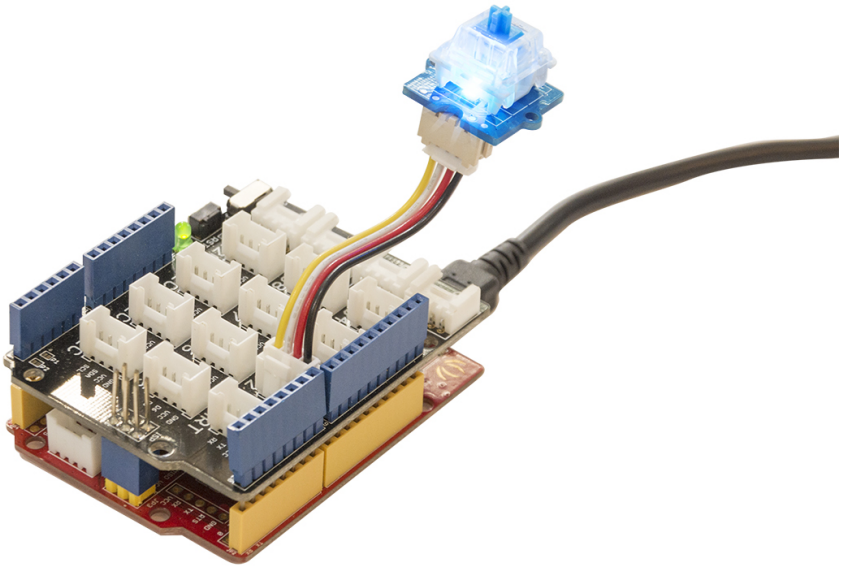
Seeeduino V4.2	Base Shield
	
Get One Now [https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html]	Get One Now [https://www.seeedstudio.com/Base Shield-V2-p-1378.html]

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] 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.** Grove-Mech keycap to port **D2** 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 Grove-Mech keycap to Seeeduino as below.

Seeeduino	Grove-Mech keycap
5V	Red
GND	Black
D3	White
D2	Yellow

Software

- **Step 1.** Download the Adafruit_NeoPixel-master [https://files.seeedstudio.com/wiki/Grove-Mech_Keycap/res/Adafruit_NeoPixel-master.zip] 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.** Open the Arduino IDE and create a new file, then copy the following code into the new file.

```
/**
 * This is an exmaple of the Grove - Mech Keycap.
 * Every press of the key will change the color the SK6
 *
 * Credit:
 * Adafruit_NeoPixel - https://github.com/adafruit/Adaf
 */

#include <Adafruit_NeoPixel.h>

#define BUTTON_PIN   2    // Digital IO pin connected t
                          // driven with a pull-up resi
                          // pull the pin to ground mom
                          // transition the button pres

#define PIXEL_PIN    3    // Digital IO pin connected t

#define PIXEL_COUNT 60

// Parameter 1 = number of pixels in strip,  neopixel s
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as need
//   NEO_RGB     Pixels are wired for RGB bitstream
//   NEO_GRB     Pixels are wired for GRB bitstream, co
//   NEO_KHZ400  400 KHz bitstream (e.g. FLORA pixels)
//   NEO_KHZ800  800 KHz bitstream (e.g. High Density L
```

```
27  Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT
28
29  bool oldState = LOW;
30  uint8_t color_pos = 0;
31  int i=0;
32  int longpress=2000;
33  long timecheck;
34
35  void setup() {
36    pinMode(BUTTON_PIN, INPUT_PULLUP);
37    strip.begin();
38    strip.clear();
39    strip.show(); // Initialize all pixels to 'off'
40    Serial.begin(9600);
41  }
42
43  void loop()
44  {
45
46    // Get current button state.
47    bool newState = digitalRead(BUTTON_PIN);
48
49    // Check if state changed from low to high (button pr
50    if (newState == HIGH && oldState == LOW) {
51        timecheck = millis();
52      // Short delay to debounce button.
53      delay(20);
54      // Check if button is still low after debounce.
55      newState = digitalRead(BUTTON_PIN);
56      if (newState == HIGH){
57        color_pos+=8;
58        strip.setPixelColor(0, Wheel(color_pos));
59        strip.show();
60      }
61    }
62
63   if( millis()-timecheck > 300)
64   {
65     if (digitalRead(BUTTON_PIN)==HIGH)
66     {
67   if(millis()-timecheck > longpress)
```

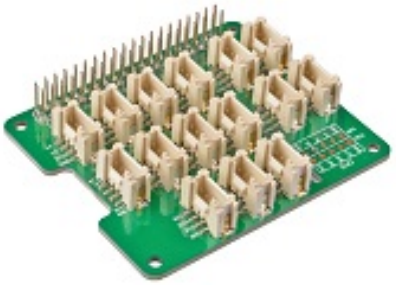
```
  {
   while(digitalRead(BUTTON_PIN) == HIGH)
   {
   strip.setPixelColor(0,Wheel(color_pos));
   strip.show();
   delay(300);

   strip.setPixelColor(0,0,0,0);
   strip.show();
   delay(300);
   bool newState = digitalRead(BUTTON_PIN);
   }
   strip.setPixelColor(0,0,0,0);
   strip.show();
    timecheck = millis();
  }
   }
    }

   // Set the last button state to the old state.
   oldState = newState;
}

// Input a value 0 to 255 to get a color value.
// The colours are a transition r - g - b - back to r.
uint32_t Wheel(byte WheelPos) {
  WheelPos = 255 - WheelPos;
  if(WheelPos < 85) {
    return strip.Color(255 - WheelPos * 3, 0, WheelPos
  }
  if(WheelPos < 170) {
    WheelPos -= 85;
    return strip.Color(0, WheelPos * 3, 255 - WheelPos
  }
  WheelPos -= 170;
  return strip.Color(WheelPos * 3, 255 - WheelPos * 3,
}
```

- **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.** Every time you press the Grove-Mech Keycap, you will see the LED color change. If you press and hold the button for about 2 sencond, you will see the breathing light effect.

Play With Raspberry Pi

Hardware

- **Step 1**. Things used in this project:

Raspberry pi	Grove Base Hat for RasPi
	
Get ONE Now [https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html]	Get ONE Now [https://www.seeedstudio.com/Grove Base-Hat-for-Raspberry-Pi-p-3186.html]

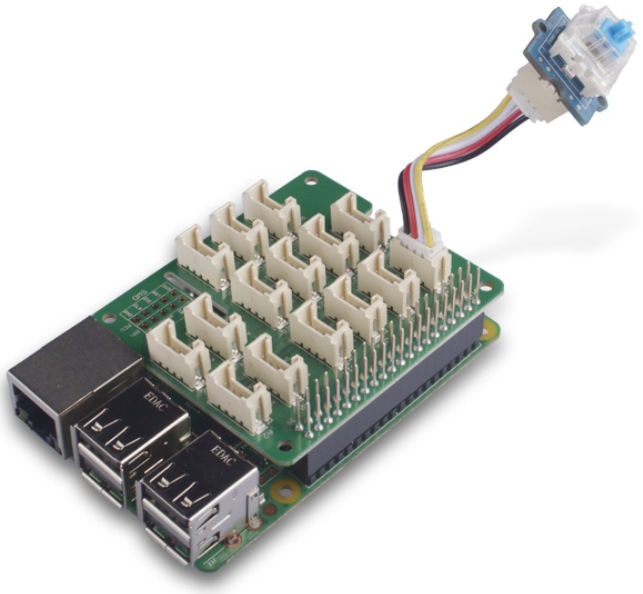
- **Step 2**. Plug the Grove Base Hat into Raspberry.

- **Step 3**. Connect the Grove - Mech Keycap to the PWM port(port 12) of the Base Hat.

> **Note**
> pin could be one of below values in the pin column for PWM function and connect the device to the corresponding slot.

Pin	Slot
18	D18
12	PWM

- **Step 4**. Connect the Raspberry Pi to PC through USB cable.



Software

- **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.

```
cd ~
git clone https://github.com/Seeed-Studio/grove.py
```

- **Step 3**. Excute below commands to run the code.

```
cd grove.py/grove
sudo python grove_mech_keycap.py 12
```

> **Caution**
> Unix has a "security model". As a normal users you can do stuff, but you should not be able to access other people's files on the same computer. And as a user you should not be able to cause the computer to stop working. Now "/dev/mem" allows you much, much more "mischief" than just changing a GPIO. So that's why /dev/mem must be protected against normal users. Thus in order to run this code, you should type **sudo python grove_mech_keycap.py** in the command line

Following is the grove_mech_keycap.py code.

```
import time
from grove.button import Button
from grove.factory import Factory

class GroveKeycap(object):
    def __init__(self, pin):
        # High = pressed
        self.__btn = Factory.getButton("GPIO-HIGH", pin)
        # single WS2812 LED
```

```
        self.__led = Factory.getOneLed("WS2812-PWM", pin
        self.__on_event = None
        self.__btn.on_event(self, GroveKeycap.__handle_e

    @property
    def on_event(self):
        return self.__on_event

    @on_event.setter
    def on_event(self, callback):
        if not callable(callback):
            return
        self.__on_event = callback

    def __handle_event(self, evt):
        # print("event index:{} event:{} pressed:{}".for
        if callable(self.__on_event):
            self.__on_event(evt['index'], evt['code'], e
            return

        self.__led.brightness = self.__led.MAX_BRIGHT
        event = evt['code']
        if event & Button.EV_SINGLE_CLICK:
            self.__led.light(True)
            print("turn on  LED")
        elif event & Button.EV_DOUBLE_CLICK:
            self.__led.blink()
            print("blink    LED")
        elif event & Button.EV_LONG_PRESS:
            self.__led.light(False)
            print("turn off LED")


Grove = GroveKeycap

def main():
    from grove.helper import SlotHelper
    sh = SlotHelper(SlotHelper.PWM)
    pin = sh.argv2pin()

    ledbtn = GroveKeycap(pin)
```

```
51
52      # remove ''' pairs below to begin your experiment
53      '''
54      # define a customized event handle your self
55      def cust_on_event(index, event, tm):
56          print("event with code {}, time {}".format(event
57
58      ledbtn.on_event = cust_on_event
59      '''
60      while True:
61          time.sleep(1)
62
63
64  if __name__ == '__main__':
65      main()
```

Success

If everything goes well, you will be able to see the following result. If you single click the keycap, you will see "turn on LED", if you double-click the keycap, you will see "blink LED". Long press the keycap will give "turn off LED".

```
1   pi@raspberrypi:~/grove.py/grove $ sudo python grove_mech
2   Hat Name = 'Grove Base Hat RPi'
3   turn on  LED
4   turn on  LED
5   blink    LED
6   turn on  LED
7   turn off LED
8   ^CTraceback (most recent call last):
9     File "grove_mech_keycap.py", line 98, in <module>
10      main()
11    File "grove_mech_keycap.py", line 94, in main
12      time.sleep(1)
13  KeyboardInterrupt
```

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

Schematic Online Viewer

Resources

- **[Zip]** Grove-Mech Keycap eagle file [https://files.seeedstudio.com/wiki/Grove-Mech_Keycap/res/Grove-Mech_Keycap_eagle.zip]

- **[Zip]** Adafruit_NeoPixel-master [https://files.seeedstudio.com/wiki/Grove-Mech_Keycap/res/Adafruit_NeoPixel-master.zip]
- **[PDF]** Product brief of the swith [https://files.seeedstudio.com/wiki/Grove-Mech_Keycap/res/DIP_Mech_Key.pdf]

Project

This is the introduction Video of this product, simple demos, you can have a try.

Tech Support

Please 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=newproducts]